The Application of Statistical Classification to Predict Sovereign Default

A thesis submitted in fulfillment of the requirements for the degree of

Master of Science

in

Mathematical Statistics

in the

Department of Statistics

of

Rhodes University

by

Vele Rendani

December 2022

Abstract

When considering sovereign loans, it is imperative for a financial institution to have a good understanding of the sovereign they are transacting with. Defaults can occur if proper evaluation steps are not considered. To aid in the prediction of potential sovereign defaults, financial institutions, together with grading companies, quantify the risk associated with issuing a loan to a sovereign by developing sovereign default early warning systems (EWS). Various classification models are considered in this study to develop sovereign default EWS. These models are the binary logit, probit, Bayesian additive regression trees, and artificial neural networks. This study investigates the predictive performance of the various classification techniques. Sovereign information is not readily available, so missing data techniques are considered in order to counter the data availability issue. Sovereign defaults are rare, which results in an imbalance in the distribution of the binary dependent variable. To assess data sets with such characteristics, metrics for imbalanced data are considered for model performance comparison. From the findings, the Bayesian additive regression technique generated better results than the other techniques when considering a basic data analysis. Moreover when cross-validation was considered, the neural network technique performed best. In addition, regional models had better results than the global model when considering model predictive capability. The significance of this study is to develop sovereign default prediction models using various classification techniques focused on enhancing previous literature and analysis through the application of Bayesian additive regression trees.

Keywords: Artificial Neural Networks, Bayesian Additive Regression Trees, Brier Score, Classification, Case-Wise Deletion, Default, Imbalanced Data, K-fold Cross-Validation, Logit, Multiple Imputation via Chained Equation, Missing Data, Markov Chain Monte Carlo, Neuron, Probit, Regularization Prior, Sovereign Debt.

Contents

List of Figures **ix**

List of Tables **xv**

List of Abbreviations **xxv**

Acknowledgements **xxvii**

1 Introduction **1**

2 Sovereign Debt and Classification **4**

2.1 An Introduction to Sovereign Debt and Default 4

2.2 Sovereign Debt Measures 5

2.3 Probability of Default (PD) 5

2.4 Loss Given Default (LGD) 6

2.5 Exposure at Default (EAD) 7

2.6 Gross Domestic Product (GDP) 7

2.7 Sovereign Bonds 9

2.8 Previous Literature 9

3 Classification **12**

3.1 Imbalanced Data 13

3.2 Evaluation Metrics for Imbalanced Classification 13

3.2.1 Threshold Metrics for Imbalanced Classification 14

3.2.2 Ranking Metrics 15

3.2.3 Probability Metrics for Imbalanced Classification 17

3.3 k-Fold Cross Validation 18

3.4 Root Mean Square Error . . . 19
3.5 Missing Data . . . 20
3.5.1 Case Wise Deletion (CWD) . . . 20
3.5.2 Mean Imputation . . . 20
3.5.3 Multiple Imputation via Chained Equation (MICE) . . . 21

4 Statistical Classifiers **23**
4.1 The Logit Classifier . . . 23
4.1.1 Logit Transformation . . . 24
4.1.2 Logistic Regression . . . 26
4.2 The Probit Classifier . . . 27
4.2.1 Probit Model Application . . . 28
4.3 The Probability of Default . . . 29
4.4 Bayesian Additive Regression Trees Classifier . . . 31
4.4.1 BART . . . 31
4.4.2 Sum-Of-Trees Model . . . 33
4.4.3 Regularization Prior . . . 34
4.4.4 Prior Independence and Symmetry . . . 34
4.4.5 The T_j Prior . . . 35
4.4.6 The $\mu_{ij}|T_j$ Prior for $P(\mu_{ij}|T_j)$ . . . 35
4.4.7 The σ Prior for $P(\sigma)$ . . . 36
4.4.8 Selection of m . . . 36
4.4.9 Posterior Information Extraction . . . 36
4.4.9.1 Bayesian Back-Fitting MCMC Algorithm . . . 36
4.4.9.2 Posterior Inference Statistics . . . 38
4.4.10 BART Probit for Classification . . . 40
4.4.11 BART Variable Importance . . . 42
4.5 Artificial Neural Networks Classifier . . . 42
4.6 Neuron . . . 43
4.6.1 Artificial Neuron . . . 44

5 An Analysis of Sovereign Default **48**
5.1 The Data and a Preliminary Analysis . . . 48

5.2 The Case-Wise Deletion (CWD) Data Set . . . 52
5.2.1 Default of Sovereigns Per Region . . . 53
5.2.2 Box plot Visualization of the Independent Variables . . . 54
5.3 Binary Logit Classification Models . . . 59
5.3.1 Case-Wise Deletion Logit Models . . . 60
5.3.2 Comparison of Global and Regional Binary Logistic Model Results for Case-Wise Deletion . . . 64
5.3.3 Mean Imputation Logit Models . . . 65
5.3.4 Comparison of Global and Regional Binary Logistic Model Results for Mean Imputation . . . 70
5.3.5 MICE Logit Models . . . 71
5.3.6 Comparison of Global and Regional Binary Logistic Model Results for MICE . 76
5.4 Probit Classifier Models . . . 77
5.4.1 Case Wise Deletion Probit Models . . . 78
5.4.2 Comparison of Global and Regional Binary Probit Model Results for Case-Wise Deletion . . . 82
5.4.3 Mean Imputation Probit Models . . . 83
5.4.4 Comparison of Global and Regional Binary Probit Model Results for Mean Imputation . . . 88
5.4.5 MICE Probit Models . . . 89
5.4.6 Comparison of Global and Regional Binary Probit Model Results for MICE . . 94
5.5 Bayesian Additive Regression Trees Classifier . . . 95
5.5.1 Comparison of Global and Regional BART Model Results . . . 95
5.6 Artificial Neural Network Classifier . . . 98
5.6.1 Comparison of Global and Regional ANN Model Results . . . 98
5.7 Comparison of Classifiers across Data Sets . . . 101
5.7.1 The Global Region . . . 101
5.7.2 South and East Asia . . . 102
5.7.3 Latin America . . . 104
5.7.4 The Middle East and Africa . . . 105
5.7.5 Western Europe . . . 106
5.8 Performance: Regional and Global Performance . . . 107
5.9 Additional Artificial Neural Networks Analysis . . . 111

6 Conclusion **116**

6.1 Logit Classifier . . . 116
6.2 Probit Classifier . . . 117
6.3 BART Classifier . . . 118
6.4 Neural Network Classifier . . . 118
6.5 Cross-Validation vs Non-Cross-Validation . . . 118
6.6 Variable Importance . . . 119
6.7 Data Imputation . . . 119
6.8 Limitations of Research . . . 119
6.9 Future Research . . . 120

References **121**

Appendix A: A Linear Regression Analysis of the Sovereignty Data **132**

A.1 Regression Analysis for the Data . . . 133
A.1.1 Global . . . 133
A.1.2 South and East Asia . . . 135
A.1.3 Latin America . . . 136
A.1.4 The Middle East and Africa . . . 138
A.1.5 Western Europe . . . 139
A.1.6 Conclusion . . . 140

Appendix B: Cross-Validation Classification **141**

B.1 5-fold Cross Validation: Logistic Regression . . . 141
B.1.1 5-Fold Cross-Validation Case-Wise Deletion Logit Models . . . 142
B.1.2 5-Fold Cross-Validation Mean Imputation Logit Models . . . 147
B.1.3 5-Fold Cross-Validation MICE Logit Models . . . 152
B.1.4 Performance comparison . . . 157
B.2 5-fold Cross Validation: Probit Model . . . 158
B.2.1 5-Fold Cross-Validation Case-Wise Deletion Probit Models . . . 159
B.2.2 5-Fold Cross-Validation Mean Imputation Probit Models . . . 164
B.2.3 5-Fold Cross-Validation MICE Probit Models . . . 169
B.2.4 Performance Comparison . . . 174
B.3 5-fold Cross Validation: BART . . . 175

B.3.1 5-Fold Cross-Validation Case-Wise Deletion BART Models 176
B.3.1.1 Global 176
B.3.1.2 South and East Asia 179
B.3.1.3 Latin America 182
B.3.1.4 The Middle East and Africa 184
B.3.1.5 Western Europe 187
B.3.2 5-Fold Cross-Validation Mean Imputation BART Models 188
B.3.2.1 Global 188
B.3.2.2 South and East Asia 191
B.3.2.3 Latin America 194
B.3.2.4 The Middle East and Africa 196
B.3.2.5 Western Europe 199
B.3.3 5-Fold Cross-Validation MICE BART Models 202
B.3.3.1 Global 202
B.3.3.2 South and East Asia 205
B.3.3.3 Latin America 208
B.3.3.4 The Middle East and Africa 210
B.3.3.5 Western Europe 213
B.3.4 Model Comparison 213
B.4 5-fold Cross Validation: Neural Network 215
B.4.1 5-Fold Cross-Validation Case Wise Deletion ANN Models 216
B.4.2 5-Fold Cross-Validation Mean Imputation ANN Models 218
B.4.3 5-Fold Cross-Validation MICE ANN Models 221
B.4.4 ANN Best Model Outputs Per Neuron 223
B.5 5-Fold Cross-Validation Model Comparison of All Techniques by Data Set 225
B.5.1 Global Data 225
B.5.2 South and East Asia 226
B.5.3 Latin America 228
B.5.4 The Middle East and Africa 229
B.5.5 Western Europe 231

Appendix C: BART Analysis **233**

C.1 Case Wise Deletion BART Models . . . 233

C.1.1 Global Data . . . 233

C.1.2 South and East Asia Case Wise Deletion . . . 236

C.1.3 Latin America Case Wise Deletion . . . 238

C.1.4 The Middle East and Africa Case Wise Deletion . . . 241

C.2 Mean Imputation BART Models . . . 243

C.2.1 Global Data . . . 243

C.2.2 South and East Asia . . . 246

C.2.3 Latin America . . . 248

C.2.4 The Middle East and Africa . . . 251

C.2.5 Western Europe . . . 253

C.3 MICE BART Models . . . 256

C.3.1 Global Data . . . 256

C.3.2 South and East Asia . . . 258

C.3.3 Latin America . . . 261

C.3.4 The Middle East and Africa . . . 263

C.3.5 Western Europe . . . 266

Appendix D: Neural Networks Analysis **269**

D.1 Case-wise Deletion ANN Models . . . 269

D.2 Mean Imputation ANN Models . . . 271

D.3 MICE ANN Models . . . 273

Appendix E: R-Code. **276**

E.1 Summary Statistics . . . 276

E.2 MICE . . . 279

E.3 Logistic Classifier . . . 284

E.4 Probit Classifier . . . 305

E.5 Bayesian Additive Regression Trees Classifier . . . 327

E.6 Artificial Neural Network Classifier . . . 363

E.7 Regression Analysis . . . 388

E.8 5-Fold Binary Logit Classifier . . . 396

E.9 5-Fold Probit Classifier 423
E.10 5-Fold Bayesian Additive Regression Trees Classifier 451
E.11 5-Fold Artificial Neural Network Classifier 486
E.12 Additional ANN analysis 506

List of Figures

2.1 South Africa's GDP from 2010 to 2020 (Trading Economics, 2021). 8

3.1 Receiver operator curve (ROC) plot (MachineLearningMastery, 2021). 16
3.2 A precision-recall curve (MachineLearningMastery, 2021). 17

4.1 The logit transformation. 26
4.2 Inclusion proportion graph of the variables included in this study. 42
4.3 A biological neuron.(Abraham, 2005) 44
4.4 Model of an artificial neuron.(Bennell et al., 2006) 45
4.5 Sigmoid function.(Dongare et al., 2012) 45
4.6 Neural network architecture.(Yadav et al., 2015) 46

5.1 Graphical representation of countries involved in Dawood et al. (2017) study. 48
5.2 Missing data plot of all regions. 52
5.3 Box plot of the total debt, trade openness, export growth, and current account for all observations. 56
5.4 Box plot of the FDI, real GDP growth, REER and government expenditure for all observations. 56
5.5 Box plot of the national savings for all observations. 57
5.6 Box plot of the global interest with the full sample observation and trimmed sample observation. Trimming was a result of cutting out the outliers to have a better view of the boxplot distribution. 57
5.7 Box plot of the inflation rate with the full sample and a trimmed sample. Trimming was a result of cutting out the outliers to have a better view of the boxplot distribution. 58
5.8 Box plot of the domestic credit and bank assets for all observations. As seen there is an extreme outlier, it is represented by Lebanon as a result of the IMF credit accounting to over 1500% of quota Dawood et al. (2017). 58

5.9 5-fold accuracy to number of neurons plot for all regions. 113
5.10 5-fold sensitivity to number of neurons plot for all regions. 114
5.11 5-fold specificity to number of neurons plot for all regions. 114
5.12 5-fold ROC AUC to number of neurons plot for all regions. 115

1 Diagnostics plots of the fitted GLM for the global MICE data. 134
2 Diagnostics plots of the fitted GLM for the South and East Asia MICE data 135
3 Diagnostics plots of the fitted GLM for the Latin America MICE data 137
4 Diagnostics plots of the fitted GLM for the Middle East and Africa MICE data 138
5 Diagnostics plots of the fitted GLM for the Western Europe MICE data 140

6 Area under the curve graph applied to case-wise deletion global data. 176
7 Plots and box plots of the model output results for training and testing case-wise deletion global data set. 176
8 RMSE box plot and line plot for case-wise deletion global data. 177
9 5-fold cross validation BART variable importance for the global case wise deletion data set. 178
10 Area under the curve graph applied to case-wise deletion South and East Asia data set. 179
11 Plots and box plots of the model output results for training and testing case-wise deletion South and East Asia data set. 179
12 RMSE box plot and line plot for case-wise deletion South and East Asia data. 180
13 5-fold cross validation BART variable importance for the South and East Asia case wise deletion data set. 181
14 Area under the curve graph applied to case-wise deletion Latin America data set. . . . 182
15 Plots and box plots of the model output results for training and testing case-wise deletion Latin America data set. 182
16 RMSE box plot and line plot for case-wise deletion Latin America data. 183
17 5-fold cross validation BART variable importance for the Latin America case wise deletion data set. 184
18 Area under the curve graph applied to case-wise deletion the Middle East and Africa data set. 184
19 Plots and box plots of the model output results for training and testing case-wise deletion Middle East and Africa data set. 185
20 RMSE box plot and line plot for case-wise deletion Middle East and Africa data. . . . 186

21 5-fold cross validation BART variable importance for the Middle East and Africa case wise deletion data set. . . . 187
22 Area under the curve graph applied for the mean imputed global data set. . . . 188
23 Plots and box plots of the model output results for training and testing mean imputed global data set. . . . 188
24 RMSE box plot and line plot for mean imputed global data. . . . 189
25 5-fold cross validation BART variable importance for the global mean by region data set. 190
26 Area under the curve graph applied for the mean imputed South and East Asia data set. 191
27 Plots and box plots of the model output results for training and testing mean imputed Middle East and Asia data set. . . . 191
28 RMSE box plot and line plot for mean imputed South and East Asia data. . . . 192
29 5-fold cross validation BART variable importance for the South and East Asia mean by region data set. . . . 193
30 Area under the curve graph for the mean imputed Latin America data set. . . . 194
31 Plots and box plots of the model output results for training and testing mean imputed Latin America data set. . . . 194
32 RMSE box plot and line plot for mean imputed Latin America data. . . . 195
33 5-fold cross validation BART variable importance for the Latin America mean by region data set. . . . 196
34 Area under the curve graph applied for the mean imputed Middle East and Africa data set. . . . 196
35 Plots and box plots of the model output results for training and testing mean imputed Middle East and Africa data set. . . . 197
36 RMSE box plot and line plot for mean imputed Middle East and Africa data. . . . 198
37 5-fold cross validation BART variable importance for the Middle East and Africa mean by region data set. . . . 199
38 Area under the curve graph applied for the mean imputed Wesern Europe data set. . . . 199
39 Plots and box plots of the model output results for training and testing mean imputed Western European data set. . . . 200
40 RMSE box plot and line plot for mean imputed Western European data. . . . 200
41 5-fold cross validation BART variable importance for the Western Europe mean by region data set. . . . 201
42 Area under the curve graph applied to the MICE global data set. . . . 202

43 Plots and box plots of the model output results for training and testing MICE global data set. . . . 202
44 RMSE box plot and line plot for case-wise deletion global data. . . . 203
45 5-fold cross validation BART variable importance for the global MICE data set. . . . 204
46 Area under the curve graph applied to the MICE South and East Asia data set. . . . 205
47 Plots and box plots of the model output results for training and testing MICE South and East Asia data set. . . . 205
48 RMSE box plot and line plot for case-wise deletion global data. . . . 206
49 5-fold cross validation BART variable importance for the South and East Asia MICE data set. . . . 207
50 Area under the curve graph applied to the MICE Latin America data set. . . . 208
51 Plots and box plots of the model output results for training and testing MICE Latin America data set. . . . 208
52 RMSE box plot and line plot for case-wise deletion global data. . . . 209
53 5-fold cross validation BART variable importance for the Latin America MICE data set. 210
54 Area under the curve graph applied to the MICE Middle East and Africa data set. . . . 210
55 Plots and box plots of the model output results for training and testing MICE Middle East and Africa data set. . . . 211
56 RMSE box plot and line plot for case-wise deletion global data. . . . 211
57 5-fold cross validation BART variable importance for the Middle East and Africa MICE data set. . . . 212

58 ROC curve applied to the case-wise deletion global data. . . . 233
59 Plot and box plot of the model output results for training and testing case-wise deletion global data. . . . 234
60 RMSE box plot and line plot for case-wise deletion global data. . . . 234
61 BART variable importance for the global case wise deletion data set. . . . 235
62 ROC curve applied to the case-wise deletion South and East Asia data. . . . 236
63 Plot and box plot of the model output results for training and testing case-wise deletion South and East Asia data. . . . 236
64 RMSE box plot and line plot for case-wise deletion South and East Asia data. . . . 237
65 BART variable importance for the South and East Asia case wise deletion data set. . . . 238
66 ROC curve applied to the case-wise deletion Latin America data. . . . 238
67 Plot and box plot of the model output results for training and testing case-wise deletion Latin America data. . . . 239

68 RMSE box plot and line plot for case-wise deletion Latin America data. 239
69 BART variable importance for the Latin America case wise deletion data set. 240
70 ROC curve applied to the case-wise deletion Middle East and Africa data. 241
71 Plot and box plot of the model output results for training and testing case-wise deletion The Middle East and Africa data. 241
72 RMSE box plot and line plot for case-wise deletion The Middle East and Africa data. . 242
73 BART variable importance for the Middle East and Africa case wise deletion data set. . 243
74 ROC curve applied to the mean imputated global data. 243
75 Plot and box plot of the model output results for training and testing mean imputation global data. 244
76 RMSE box plot and line plot for mean imputation global data. 244
77 BART variable importance for the global mean by region data set. 245
78 ROC curve applied to the mean imputated South and East Asia data. 246
79 Plot and box plot of the model output results for training and testing mean imputation South and East Asia data. 246
80 RMSE box plot and line plot for mean imputation South and East Asia data. 247
81 BART variable importance for the South and East Asia mean by region data set. 248
82 ROC curve applied to the mean imputated Latin America data. 248
83 Plot and box plot of the model output results for training and testing mean imputation Latin America data. 249
84 RMSE box plot and line plot for mean imputation Latin America data. 249
85 BART variable importance for the Latin America mean by region data set. 250
86 ROC curve applied to the mean imputated Middle East and Africa data. 251
87 Plot and box plot of the model output results for training and testing mean imputation The Middle East and Africa data. 251
88 RMSE box plot and line plot for mean imputation the Middle East and Africa data. . . 252
89 BART variable importance for the Middle East and Africa mean by region data set. . . 253
90 ROC curve applied to the mean imputated Western Europe data. 253
91 Plot and box plot of the model output results for training and testing mean imputation Western Europe data. 254
92 RMSE box plot and line plot for mean imputation Western Europe data. 254
93 BART variable importance for the Western European mean by region data set. 255
94 ROC curve applied to the MICE global data. 256

95 Plot and box plot of the model output results for training and testing MICE global data. 256
96 RMSE box plot and line plot for MICE global data. 257
97 BART variable importance for the global MICE data set. 258
98 ROC curve applied to the MICE South and East Asia data. 258
99 Plot and box plot of the model output results for training and testing MICE South and East Asia data. 259
100 RMSE box plot and line plot for MICE South and East Asia data. 259
101 BART variable importance for the South and East Asia MICE data set. 260
102 ROC curve applied to the MICE Latin America data. 261
103 Plot and box plot of the model output results for training and testing MICE Latin America data. 261
104 RMSE box plot and line plot for MICE Latin America data. 262
105 BART variable importance for the Latin America MICE data set. 263
106 ROC curve applied to the MICE the Middle East and Africa data. 263
107 Plot and box plot of the model output results for training and testing MICE The Middle East and Africa data. 264
108 RMSE box plot and line plot for MICE the Middle East and Africa data. 264
109 BART variable importance for the Middle East and Africa MICE data set. 265
110 ROC curve the applied to MICE Western Europe data. 266
111 Plot and box plot of the model output results for training and testing MICE Western Europe data. 266
112 RMSE box plot and line plot for MICE Western Europe data. 267
113 BART variable importance for the Western European MICE data set. 268

List of Tables

3.1 The confusion matrix. 15

3.2 The 5-fold cross validation process. 19

5.1 Variables considered in Dawood et al. (2017) study in order to predict sovereign debt. . 50

5.2 Crises episode for Latin American sovereigns based on Dawood et al. (2017). 53

5.3 Crises episode for South and East Asia sovereigns based on the criteria in Dawood et al. (2017). 53

5.4 Crises episode for Middle East and African sovereigns based on the criteria in Dawood et al. (2017). 54

5.5 Crises episode for Western European sovereigns based on the criteria in Dawood et al. (2017). 54

5.6 Table showing the number of observations (n), average and variance of the default and non-default periods for all sovereigns along with the independent, two-sample t-test and F-test. 55

5.7 Binary logit model summary when applied to the global case-wise deletion data set. . . 60

5.8 Binary logit model summary when applied to the South and East Asia case-wise deletion data set. 61

5.9 Binary logit model summary when applied to the Latin America case-wise deletion data set. 62

5.10 Binary logit model summary when applied to the Middle East and Africa case-wise deletion data set. 63

5.11 Binary logit models of sovereign debt from the year 1980-2005, case-wise deletion data set. Pseudo McFadden's R^2 and percentage of out of sample crisis predicted by each model, global and regional are reported. The Western Europe model is omitted as there are no observations for this region in the case-wise deletion data. 64

5.12 Binary logit model summary when applied to the global Mean Imputation data set. . . 65

5.13 Binary logit model summary when applied to the South and East Asia mean imputation data set. 66
5.14 Binary logit model summary when applied to the Latin America mean imputation data set. 67
5.15 Binary logit model summary when applied to the Middle East and Africa mean imputation data set. 68
5.16 Binary logit model summary when applied to the Western Europe mean imputation data set. 69
5.17 Binary logit models of sovereign debt from the year 1980-2005, where the missing values are filled using the average of the regions of which the country resides. Pseudo McFadden's R^2 and percentage of in and out of sample crisis predicted by each model ,global and regional are reported. 70
5.18 Binary logit model summary when applied to the global MICE data set. 71
5.19 Binary logit model summary when applied to the South and East Asia MICE data set. . 72
5.20 Binary logit model summary when applied to the Latin America MICE data set. 73
5.21 Binary logit model summary when applied to the Middle East and Africa MICE data set. 74
5.22 Binary logit model summary when applied to the Western Europe MICE data set. . . . 75
5.23 Binary logit models of sovereign debt from the year 1980-2005, where the missing values are filled using the MICE technique. The Pseudo McFadden's $R^2$and percentage of out of sample crisis predicted by each model, global and regional are reported. . . . 76
5.24 Probit model summary when applied to the global case-wise deletion data set. 78
5.25 Probit model summary when applied to the South and East Asia case-wise deletion data set. 79
5.26 Probit model summary when applied to the Latin America case-wise deletion data set. 80
5.27 Probit model summary when applied to the Middle East and Africa case-wise deletion data set. 81
5.28 Binary probit models of sovereign debt from the year 1980-2005, where case-wise deletion technique was used to compile the data. Pseudo McFadden's R^2 and percentage of out-of-sample crisis predicted by each model, global and regional are reported. . 82
5.29 Probit model summary when applied to the global mean imputation data set. 83
5.30 Probit model summary when applied to the South and East Asia mean imputation data set. 84
5.31 Probit model summary when applied to the Latin America mean imputation data set. . 85
5.32 Probit model summary when applied to the Middle East and Africa mean imputation data set. 86

5.33 Probit model summary when applied to the Western European mean imputation data set. 87

5.34 Binary probit models of sovereign debt from the year 1980-2005, where mean imputation technique was used to compile the data. Pseudo McFadden's R^2 and percentage of out-of-sample crises predicted by each model, global and regional are reported. . . . 88

5.35 Probit model summary when applied to the global MICE data set. 89

5.36 Probit model summary when applied to the South and East Asia MICE data set. 90

5.37 Probit model summary when applied to the Latin America MICE data set. 91

5.38 Probit model summary when applied to the Middle East and Africa MICE data set. . . 92

5.39 Probit model summary when applied to the Western European MICE data set. 93

5.40 Binary probit models of sovereign debt from the year 1980-2005, where the missing values are filled using MICE. Pseudo McFadden's $R^2$and as well as metrics used for prediction performance are considered for the global and regional are reported. 94

5.41 BART regional model comparison for the case-wise deletion data sets. 95

5.42 BART regional model comparison for the mean imputations data sets. 96

5.43 BART regional model comparison for the MICE data sets. 97

5.44 Artificial neural network classifier summary for the of the case-wise deletion data sets. 98

5.45 Artificial neural network summary for the of the mean imputation data sets. 99

5.46 Artificial neural network summary for the of the MICE data sets. 100

5.47 Classifier accuracy summary for the case-wise deletion for the global region. 101

5.48 Classifier accuracy summary for the mean imputation for the global region. 101

5.49 Classifier accuracy summary for the Mice for the global region. 102

5.50 Classifier accuracy summary for the case-wise deletion for the South and East Asia region. 102

5.51 Classifier accuracy summary for the mean imputation for the South and East Asia region 103

5.52 Classifier accuracy summary for the MICE for the South and East Asia region. 103

5.53 Classifier accuracy summary for the case-wise deletion for the Latin America region. . 104

5.54 Classifier accuracy summary for the mean for the Latin America region. 104

5.55 Classifier accuracy summary for the MICE for the Latin America region. 105

5.56 Classifier accuracy summary for the case-wise deletion for the Middle East and Africa region. 105

5.57 Classifier accuracy summary for the mean imputation for the Middle East and Africa region 106

5.58 Classifier accuracy summary for the MICE for the Middle East and Africa region. . . . 106

5.59 Classifier accuracy summary for the mean imputation for the Western Europe region. . 107

5.60 Classifier accuracy summary for the MICE for the Western Europe region. 107

5.61 Summary of the best classifier by data set and region. An "X" is assigned to the classifier with the best overall performance in that particular instance. 108

5.62 Summary of the best 5-fold cross validation classifier by data set and region. An "X" is assigned to the classifier with the best overall performance in that particular instance. 108

5.63 Tabular representation of global to regional model outcomes. In cases where a regional model performs better than the global model, only that model is assigned an "X" and if it happens that no regional model outperforms the global model, the "X" will be assigned to the global model. The results considered are those from the test set which ranges from 2006 to 2012. All conclusion are based on the ROC AUC. 109

5.64 Tabular representation of global to regional model outcomes. In cases where a regional model performs better than the global model, only that model is assigned an "X" and if it happens that no regional model outperforms the global model, the "X" will be assigned to the global model. The results considered are those from the test set which ranges from 2006 to 2012. All conclusion are based on the ROC AUC. 109

5.65 Variable importance in terms of model development for the BART classifier per data set. The most important variable is labelled one (1) and the least relevant fourteen (14). The Western Europe importance is not considered therefore coloured red, due to lack of data. 110

5.66 Table representing variable importance in terms of model development for the cross validation data for the BART classifier per data set. The most important variable is labelled one (1) and the least relevant fourteen (14). The Western Europe importance is not considered therefore coloured red, due to lack of data. 111

5.67 A summary comparison of the variable reduced ANN classifiers with 2-4 nodes and comparing those classifiers to the best classifier by data set and region. Case-wise deletion was considered here. 112

5.68 A summary comparison of the variable reduced ANN classifiers with 2-4 nodes and comparing those classifiers to the best classifier by data set and region. Mean imputation was considered here. 112

5.69 A summary comparison of the variable reduced ANN classifiers with 2-4 nodes and comparing those classifiers to the best classifier by data set and region. MICE was considered here. 113

6.1 Official rank for the variable importance in default model development. 119

2 Variable mean and variance for crisis and non crisis cases along with the p-value of both the t-test and the F-test at a 95%, where $\alpha = 0.05$, for the global MICE data set. . 133

3 Variable mean and variance for crisis and non crisis cases along with the p-value of both the t-test and the F-test at a 95%, where $\alpha = 0.05$, for the South and East Asia MICE data set. 135

4 Variable mean and variance for crisis and non crisis cases along with the p-value of both the t-test and the F-test at a 95%, where $\alpha = 0.05$, for the Latin America MICE data set. 136

5 Variable mean and variance for crisis and non crisis cases along with the p-value of both the t-test and the F-test at a 95%, where $\alpha = 0.05$, for the Middle East and Africa MICE data set. 138

6 Variable mean and variance for crisis and non crisis cases along with the p-value of both the t-test and the F-test at a 95%, where $\alpha = 0.05$, for the Western Europe MICE data set. 139

7 5-fold cross-validation binary logit model summary when applied to the global case-wise deletion data set. 142

8 5-fold cross-validation binary logit model summary when applied to the South and East Asia case-wise deletion data set. 143

9 5-fold cross-validation binary logit model summary when applied to the Latin America case-wise deletion data set. 144

10 5-fold cross-validation binary logit model summary when applied to the Middle East and Africa case-wise deletion data set. 145

11 5-fold cross-validation binary logit model summary when applied to the Western Europe case-wise deletion data set. 146

12 5-fold cross-validation binary logit model summary when applied to the global mean imputation data set. 147

13 5-fold cross-validation binary logit model summary when applied to the South and East Asia mean imputation data set. 148

14 5-fold cross-validation binary logit model summary when applied to the Latin America mean imputation data set. 149

15 5-fold cross-validation binary logit model summary when applied to the Middle East and Africa mean imputation data set. 150

16 5-fold cross-validation binary logit model summary when applied to the Western Europe mean imputation data set. 151

17 5-fold cross-validation binary logit model summary when applied to the global MICE data set. . . . 152
18 5-fold cross-validation binary logit model summary when applied to the South and East Asia MICE data set. . . . 153
19 5-fold cross-validation binary logit model summary when applied to the Latin America MICE data set. . . . 154
20 5-fold cross-validation binary logit model summary when applied to the Middle East and Africa MICE data set. . . . 155
21 5-fold cross-validation binary logit model summary when applied to the Western Europe MICE data set. . . . 156
22 Case-wise deletion 5-fold cross-validation logit model summary. . . . 157
23 mean imputation 5-fold cross-validation logit model summary. . . . 157
24 MICE 5-fold cross-validation logit model summary. . . . 158
25 5-fold cross-validation binary probit model summary when applied to the global case-wise deletion data set. . . . 159
26 5-fold cross-validation binary probit model summary when applied to the South and East Asia case-wise deletion data set. . . . 160
27 5-fold cross-validation binary probit model summary when applied to the Latin America case-wise deletion data set. . . . 161
28 5-fold cross-validation binary probit model summary when applied to the Middle East and Africa case-wise deletion data set. . . . 162
29 5-fold cross-validation binary probit model summary when applied to the Western Europe case-wise deletion data set. . . . 163
30 5-fold cross-validation binary probit model summary when applied to the global mean imputation data set. . . . 164
31 5-fold cross-validation binary probit model summary when applied to the South and East Asia mean imputation data set. . . . 165
32 5-fold cross-validation binary probit model summary when applied to the Latin America mean imputation data set. . . . 166
33 5-fold cross-validation binary probit model summary when applied to the Middle East and Africa mean imputation data set. . . . 167
34 5-fold cross-validation binary probit model summary when applied to the Western Europe mean imputation data set. . . . 168
35 5-fold cross-validation binary probit model summary when applied to the global MICE data set. . . . 169

36 5-fold cross-validation binary probit model summary when applied to the South and East Asia MICE data set. . . . 170
37 5-fold cross-validation binary probit model summary when applied to the Latin America MICE data set. . . . 171
38 5-fold cross-validation binary probit model summary when applied to the Middle East and Africa MICE data set. . . . 172
39 5-fold cross-validation binary probit model summary when applied to the Western Europe MICE data set. . . . 173
40 Case-wise deletion 5-fold cross-validation probit model summary. . . . 174
41 mean imputation 5-fold cross-validation probit model summary. . . . 174
42 MICE 5-fold cross-validation probit model summary. . . . 175
43 5-fold cross-validation BART model summary when applied to the global case-wise deletion data set. . . . 177
44 5-fold cross-validation BART model summary when applied to the South and East Asia case-wise deletion data set. . . . 180
45 5-fold cross-validation BART model summary when applied to the Latin America case-wise deletion data set. . . . 183
46 5-fold cross-validation BART model summary when applied to the Middle East and Africa case-wise deletion data set. . . . 186
47 5-fold cross-validation BART model summary when applied to the Western Europe case-wise deletion data set. . . . 187
48 5-fold cross-validation BART model summary when applied to the global mean imputation data set. . . . 189
49 5-fold cross-validation BART model summary when applied to the South and East Asia mean imputation data set. . . . 192
50 5-fold cross-validation BART model summary when applied to the Latin America mean imputation data set. . . . 195
51 5-fold cross-validation BART model summary when applied to the Middle East and Africa mean imputation data set. . . . 198
52 5-fold cross-validation BART model summary when applied to the Western Europe mean imputation data set. . . . 201
53 5-fold cross-validation BART model summary when applied to the global MICE data set. . . . 203
54 5-fold cross-validation BART model summary when applied to the South and East Asia MICE data set. . . . 206

55 5-fold cross-validation BART model summary when applied to the Latin America MICE data set. . . . 209
56 5-fold cross-validation BART model summary when applied to the Middle East and Africa MICE data set. . . . 212
57 5-fold cross-validation BART model summary when applied to the Western Europe MICE data set. . . . 213
58 BART regional model comparison for the case-wise deletion data. . . . 214
59 BART regional model comparison for the case-wise deletion data. . . . 214
60 BART regional model comparison for the case-wise deletion data. . . . 215
61 5-fold cross-validation ANN model summary when applied to the global case wise deletion data set. . . . 216
62 5-fold cross-validation ANN model summary when applied to the South and East Asia case wise deletion data set. . . . 216
63 5-fold cross-validation ANN model summary when applied to the Latin America case wise deletion data set. . . . 217
64 5-fold cross-validation ANN model summary when applied to the Middle East and Africa data set. . . . 217
65 5-fold cross-validation ANN model summary when applied to the Western Europe case wise deletion data set. . . . 218
66 5-fold cross-validation ANN model summary when applied to the global mean imputation data set. . . . 218
67 5-fold cross-validation ANN model summary when applied to the South and East Asia mean imputation data set. . . . 219
68 5-fold cross-validation ANN model summary when applied to the Latin America mean imputation data set. . . . 219
69 5-fold cross-validation ANN model summary when applied to the Middle East and Africa mean imputation data set. . . . 220
70 5-fold cross-validation ANN model summary when applied to the Western Europe mean imputation data set. . . . 220
71 5-fold cross-validation ANN model summary when applied to the global MICE data set. 221
72 5-fold cross-validation ANN model summary when applied to the South and East Asia MICE data set. . . . 221
73 5-fold cross-validation ANN model summary when applied to the Latin America MICE data set. . . . 222

74 5-fold cross-validation ANN model summary when applied to the Middle East and Africa MICE data set. . . . 222
75 5-fold cross-validation ANN model summary when applied to the Western Europe mean imputation data set. . . . 223
76 ANN regional model comparison for the case-wise deletion data. . . . 223
77 ANN regional model comparison for the mean imputation data. . . . 224
78 ANN regional model comparison for the MICE data. . . . 224
79 Model comparison summary for the case-wise deletion global date set. . . . 225
80 Model comparison summary for the mean imputation global date set. . . . 225
81 Model comparison summary for the MICE global date set. . . . 226
82 Model comparison summary for the case-wise deletion South and East Asia date set. . . 226
83 Model comparison summary for the mean imputation South and East Asia data set . . 227
84 Model comparison summary for the MICE South and East Asia date set. . . . 227
85 Model comparison summary for the case-wise deletion Latin America date set. . . . 228
86 Model comparison summary for the mean imputation Latin America. . . . 228
87 Model comparison summary for the MICE Latin America date set. . . . 229
88 Model comparison summary for the case-wise deletion the Middle East and Africa date set. . . . 229
89 Model comparison summary for the mean imputation the Middle East and Africa data set 230
90 Summary of model comparisons for the MICE Middle East and Africa date set. . . . 230
91 Model comparison summary for the case-wise deletion Western Europe date set. . . . 231
92 Model comparison summary for the mean imputation Western Europe data set. . . . 231
93 Model comparison summary for the MICE Western Europe date set. . . . 232

94 BART model summary of the global case-wise deletion data set. . . . 235
95 BART model summary of the South and East Asia case-wise deletion data set. . . . 237
96 BART model summary of the Latin America case-wise deletion data set. . . . 240
97 BART model summary of the Middle East and Africa case-wise deletion data set. . . . 242
98 BART model summary of the global mean imputation data set. . . . 245
99 BART model summary of the South and East Asia mean imputation data set. . . . 247
100 BART model summary of the Latin America mean imputation data set. . . . 250
101 BART model summary of the Middle East and Africa mean imputation data set. . . . 252

102 BART model summary of the Western Europe mean imputation data set. 255
103 BART model summary of the global MICE data set. 257
104 BART model summary of the South and East Asia MICE data set. 260
105 BART model summary of the Latin America MICE data set. 262
106 BART model summary of the Middle East and Africa MICE data set. 265
107 BART model summary of the Western Europe MICE data set. 267

108 ANN model summary applied to the global case-wise deletion data set. 269
109 ANN model summary applied to the South and East Asia case-wise deletion data set. . 269
110 ANN model summary applied to the Latin America case-wise deletion data set. 270
111 ANN model summary applied to the Middle East and Africa case-wise deletion data set. 270
112 ANN model summary applied to the global mean imputation data set. 271
113 ANN model summary applied to the South and East Asia mean imputation data set. . . 271
114 ANN model summary applied to the Latin America mean imputation data set. 272
115 ANN model summary applied to the Middle East and Africa mean imputation data set. 272
116 ANN model summary applied to the Western Europe mean imputation data set. 272
117 ANN model summary applied to the global MICE data set. 273
118 ANN model summary applied to the South and East Asia MICE data set. 273
119 ANN model summary applied to the Latin America MICE data set. 274
120 ANN model summary applied to the Middle East and Africa MICE data set. 274
121 ANN model summary applied to the Western Europe MICE data set. 274

List of Abbreviations

ANN: Artificial Neural Networks.

AUC: Area Under Curve.

BART: Bayesian Additive Regression Trees.

BACT: Bayesian Additive Classification Trees.

BCBS: Basel Committee of Banking Supervision.

BPNN: Back Propagation Neural Network.

CAR: Capital Adequacy Ratio.

CWD: Case-Wise Deletion.

EAD: Exposure at Default.

EWS: Early Warning Systems.

FDI: Foreign Direct Investment.

GDP: Gross Domestic Product.

FDIC: Federal Deposit Insurance Corporation.

FN: False Negative.

IID: Independent and Identical Distribution.

IMF: International Monetary Fund.

LGD: Loss Given Default.

Nan: Not a Number.

MAR: Missing At Random.

MCMC: Markov Chain Monte Carlo.

MICE: Multivariate Imputation via Chained Equations.

MH: Metropolis-Hastings.

MLE: Maximum Likelihood Estimate.

MNP: Multinomial Probit Model.

MSM: Method of Simulated Moments.

OLS: Ordinary Least Squares.

PD: Probability of Default.

PR: Precision Recall.

QQ: Quantile-Quantile.

REER: Real Effective Exchange Rate.

RMSE: Root Mean Square Error.

ROC: Receiver Operating Characteristic.

RRMSE: Relative Root Mean Square Error.

RWA: Risk-Weighted Assets.

TP: True Positive.

USA: United States of America.

WLS: Weighted Least Squares.

Acknowledgments

I would like to thank the Tiso Foundation group for giving me the opportunity to chase my academic dreams through their masters scholarship program.

My deep and sincere gratitude to my parents, Joel and Peggy Vele, for their continuous and unparalleled love, help, and support. I am indebted to my parents for giving me the opportunities and experiences that have made me who I am. Also, not forgetting my sisters for always being there for me.

A great thank you to my family and friends for the great words of wisdom and guidance. To my girlfriend, thank you for being with me every step of the way, your continuous help and motivation helped me.

Thank you to my supervisor, Jeremy Baxter, for your aid and support in the commencement and conclusion of this study. I also want to thank Tebogo Mabele and Roelof Sheppard from the Standard Bank model development team for their support and guidance throughout the study, and last but not least, thank God for staying faithful.

Chapter 1

Introduction

In 1974, the central bank governors of Argentina, Australia, Belgium, Brazil, Canada, China, European Union, France, Germany, Hong Kong SAR, India, Indonesia, Italy, Japan, Korea, Luxembourg, Mexico, the Netherlands, Russia, Saudi Arabia, Singapore, South Africa, Spain, Sweden, Switzerland, Turkey, the United Kingdom, and the United States established the Basel Committee of Banking Supervision (BCBS) (Goodhart, 2011). The objective of the BCBS was to moderate the authorities in charge of banking processes. For example, bank loan models were required to go through four (4) stages of approval before they could be utilized. These stages were tier one (1), two (2), and three (3) validation, and the final tier was approval from the banking regulation committee. This process was developed because clients' final risk grade should be determined by more than just a model but rather by a collaboration of the model development division and credit risk analysis. Goodhart (2011) indicates that the primary goal of the BCBS was to enhance and improve the quality of banking supervision worldwide. The BCBS is responsible for framing guidelines and standards in different areas of banking; for example, the amount of funding that can be issued to a client based on their creditworthiness or determining how much capital can be placed aside to avoid losing depositors' funds through loans and contracts.

This study investigates how to identify high risk clients when issuing loans within the BCBS framework. The creditworthiness of various sovereigns was determined based on qualitative and quantitative data obtained from the sovereign's data, either through financial analysis or questionnaires (Peter, 2002). This data was subsequently used to develop various statistical models to determine and identify factors that contribute to the likelihood of a sovereign failing to settle its debt; see, for example, Schimmelpfennig et al. (2003). Mitigation of sovereign default is imperative for a financial institution. In the event that a sovereign fails to settle its debts or defaults, financial institutions are required to utilize their designated risk-weighed assets (RWA) to account for the lost funds. RWA are financial institutions' assets or off-balance-sheet exposure, weighed relative to the risk associated with a client (Pereira et al., 2018). These calculations are utilized to determine the capital requirement, or capital adequacy

ratio (CAR), for financial institutions (Goodhart, 2011). The motivation behind such mitigation is that when money is being issued to clients that require credit, there is a certain risk that is associated with issuing out funds. Financial institutions are required to have RWA, where the funds cannot be affected by any financial activity. RWA's are used to offset lost funds incurred when creditors default. This is a benefit to investors since they do not lose their investments and can have access to any of their funds at any given time.

This study focuses on the classification of sovereign debt, as introduced in Chapters 1, 2 and 3. The word "sovereign" is an adjective derived from the French word "Soverain," which describes a title given to the highest leader in various categories (Blackburn, 2005). In the context of this thesis, the word sovereign refers to the governments of different countries. Sovereign debt is money that has been issued, typically as bonds, to a sovereign by a financial institution as a loan (Schimmelpfennig et al., 2003). This thesis aims to develop models that predict sovereign default through the use of different statistical techniques, as discussed in Chapter 4. The models considered in this thesis are fitted to data from the Dawood et al. (2017) study, which is a collection of thirty-eight (38) advanced and emerging sovereigns around the world with a time frame from 1980 to 2012. Bayesian additive regression trees (BART), artificial neural networks (ANN), binary logistic models, and probit models are applied to this data in Chapter 4. The data was collected from three online sources: The World Group (2021), Knoema (2021), and The Global Economy (2021). The mitigation of sovereign default is imperative because when a financial institution issues debt to a sovereign, it is investors' money that financial institutions use.

This thesis aims to investigate if machine learning techniques can aid in the development of a sovereign default early warning system. The objectives of this study are:

1. Study previous literature on sovereign debt, machine learning algorithms, missing data techniques, and metrics for evaluating imbalanced binary data;

2. Develop a sovereign debt crisis early warning systems (EWS) models that use historical data to conclude the state of the sovereign's financial stance;

3. Determine whether regional EWS's are better default predictors than a general global model,

4. Share best practices to aid in the development of sovereign debt crisis EWS.

Chapter 2 focuses on sovereign literature, Chapter 3 looks at methods that aid in the model development along with model prediction measuring metrics, and Chapter 4 focuses on the classification models used in the paper. Chapter 5 contains a comparative analysis of the various classifiers applied to these data sets, including those applied to data sets constructed using two different data imputation techniques. Chapter 6 provides a conclusion of the analysis done. As this study is focused on Dawood et al. (2017) findings, it is important to note that the effects of imbalanced data were not considered but

are introduced in this study, as is imputing the data and using receiver operating characteristic (ROC) area under curve (AUC) as a performance indicator. BART has also not been applied as an EWS, so this is an introduction to using BART to detect sovereign defaults.

Chapter 2

Sovereign Debt and Classification

2.1 An Introduction to Sovereign Debt and Default

Sovereign debt is typically one of two forms: either internal or external debt. Internal debt refers to funds issued to a borrower within the country (Goodhart, 2011). External debt refers to funds issued to a borrower in a foreign country (Eaton and Fernandez, 1995). Eaton and Fernandez (1995) argue that a method used to classify sovereign debt is the duration until the debt repayment is due. Bai and Zhang (2012) indicate that there are two types of debt classification: short-term and long-term debt. The duration of payment for short-term debt is at most a year. Typically, long-term debt has a duration of up to ten (10) years.

Acquiring government bonds and bills and issuing assets is how most sovereign debt is formed (Miu and Ozdemir, 2006). In order to borrow directly from international organizations such as the World Bank, the International Monetary Fund (IMF), or other international financial institutions, sovereigns with bad credit ratings must obtain permission from their respective governments and reserve banks (Calomiris, 2000). Sovereigns may find it hard to repay their debt due to negative changes in currency rates and an unduly optimistic appraisal of the return from debt-financed projects. Since banks cannot confiscate the government's assets, their sole option is to renegotiate the loan's terms. Governments weigh the dangers of taking on sovereign debt since countries that default on their obligations may have trouble acquiring loans in the future. Even though there is always a risk of default with sovereign debt, lending money to a national government in the sovereign's currency is a risk-free investment since the borrowing sovereign can repay the debt through raising taxes, cutting spending, or simply printing more money within certain limits (Goodhart, 2011). Governments can eliminate the obligation to pay interest through funding their projects. This strategy lowers government interest expenses and may result in hyperinflation (Fuertes and Kalotychou, 2006). As a result, sovereigns continue to rely on the assistance of foreign governments to fund their projects.

To mitigate sovereign default, various studies have been conducted to develop sovereign default early

warning systems (EWS); Schimmelpfennig et al. (2003) applied the logit model, a binary recursive tree strategy, and a combination of the two to develop models to predict defaults. The model was based on data from forty-seven (47) countries. Dawood et al. (2017) then added to Schimmelpfennig et al. (2003)'s study by focusing on the difference between creating regional models and a global model and including the application of the multinomial logit classifier and dynamic signal extraction.

2.2 Sovereign Debt Measures

Sovereign debt crises in the 1990's initiated the need to understand the effects and causes of sovereign debt crisis. This led to the development of statistical and economic models that detect sovereign default occurrences (Sechrest, 1993). Fioramanti (2008) showed that economic theory has developed three (3) generations of risk models that explain sovereign debt-related financial crises. The first and second generational models focused on currency crises, while the third generational model looked at a wider variety of crises and better-explained episodes in the 1990's.

Fioramanti (2008) argues that the first generational models generated poor economic policies. These models' primary aim was to fix exchange rates and to produce a continuous loss in foreign exchange reserves. Once reserves pass a critical level, the designated authorities are then compelled to withdraw from the exchange rate peg.

The fundamentals of the second generational model were to develop models that look at the existence of multiple exchange rate equilibria and look at self-fulfilling speculative attacks, such as investors' fear of a particular crisis, for example, the Corona-virus, protests, and other financially threatening crises (Buchheit and Gulati, 2021). In the presence of a sound economy, a sovereign may consider the high cost of maintaining a fixed exchange rate once the currency is at risk of a speculative attack. The third generational model was developed in response to the Asian crisis (Hunter et al., 2012). Its focus was more on the private imbalance instead of the public since, public finances in given countries were quite sound, while those of the corporate and banking sectors displayed excesses.

2.3 Probability of Default (PD)

The probability of default (PD), or default probability, is the likelihood that a client will not make a scheduled repayment over a specific period, usually one year (Yeh and Lien, 2009). PD is applied in a wide variety of financial risk management or credit analysis scenarios. It is influenced by the client's characteristics, for example, credit score, age, job status, and it involves the economic environment (Goodhart, 2011).

In cases where a client possesses a higher PD score, creditors issue small loans at high-interest rates to compensate for the associated risk. Whereas clients with a lower PD score, are in a better position

to obtain much larger loans at a lower interest rate. To further evaluate risk, financial metrics such as cash flows relative to debt, revenues or operating margin trends, and the use of leverage as standard risk evaluators are considered (Miu and Ozdemir, 2006). Another factor that is sometimes considered is a client's ability to execute a business plan and their willingness to pay.

The credit rating of sovereigns indicates the PD. Historical data and statistical techniques can be used to estimate PD's (Goodhart, 2011). In several risk management models, PD is coupled with loss given default (LDG), discussed in Section 2.4, and exposure at default (EAD), discussed in Section 2.5, to predict potential losses suffered by lenders. In general, the bigger the PD, the higher the interest rate charged by the bank to the borrower. When investors purchase and sell fixed-income assets on the open market, the same rationale applies. Sovereigns or companies with plenty of cash and a low probability of default will be able to be issued loans at cheaper interest rates. When compared to riskier debt, investors will price these bonds higher on the open market. In other words, the yield on safer bonds will be lower. Investors in the bond market will respond to the increased risk by trading bonds at lower prices and higher yields, which leads to a company's financial health deteriorating over time since bond prices move in the opposite direction to yields (Eaton and Fernandez, 1995). However, since high-yield bonds have the most significant risk of default, they pay a high yield or interest rate. Goodhart (2011) notes that government bonds, such as U.S. Treasury notes, are on the other end of the scale, with the lowest yields and the lowest chance of default, since for example, governments can always print more money to pay back debt to avoid defaulting.

2.4 Loss Given Default (LGD)

Loss Given Default (LGD) refers to the amount that a credit issuer will lose if the borrower defaults, depicted as a percentage of the total exposure at the time of default (Goodhart, 2011). For financial institutions, the total LGD calculations occur after assessing all outstanding loans using cumulative losses and exposure (Miu and Ozdemir, 2006). In addition, banks and financial institutions determine credit losses by analysing actual defaults. The quantifying of losses can be complicated and needs analysis of multiple variables. Financial analysts consider these variables when assessing all loans issued by a financial institution to determine the LGD. The analyst assesses the sovereign or company's financial statements to account for credit losses, including determining both allowances for credit losses and doubtful accounts (Miu and Ozdemir, 2006).

LGD is a crucial component of Basel Models (Miu and Ozdemir, 2006). The calculation for expected loss is given by a loan's LGD multiplied by both its PD and the financial institution's exposure at default (EAD). The key distinction between LGD and EAD is that LGD considers default recovery. An example would be considering the case of a borrower who takes out a R400,000 loan to buy a Mercedes. After a few years of making instalment payments on the loan, the borrower runs into

financial hardship and defaults when the loan balance, or exposure at default, reaches R300,000. The Mercedes is repossessed by the bank, which is able to sell it for R240,000. The bank has a net loss of R60,000 from (R300,000 – R240,000), with an LGD of 20% (R300,000 – R240,000)/R300,000.

2.5 Exposure at Default (EAD)

Exposure at default (EAD) is the sum of money a financial institution is exposed to when a debtee's loan defaults (Jacobs et al., 2010). Financial institutions use the internal rating-based (IRB) approach to calculate their risks (Goodhart, 2011). Financial institutions use internal risk management default models to predict respective EAD systems. EAD is also known as credit exposure, outside the banking industry. When a sovereign defaults on a loan, EAD is the estimated amount of loss a bank will incur. Banks frequently generate an EAD value for each loan and use this information to assess their total default risk. The EAD is a changing statistic that shows how quickly a debtor reimburses a loan.

At default, there are two methods for determining exposure (Witzany, 2011). The first method, known as foundation internal ratings-based (F-IRB), is used by regulators. Banking institutions utilize the second method, known as advanced internal ratings-based (A-IRB), which is more versatile since it allows financial institutions to have different ratings per portfolio or product, instead of using one general rating when considering all product offerings.

Banks are required to reveal their risk exposure. Data and internal analysis, such as sovereign characteristics and cash flow, are used to arrive at this figure (Jacobs et al., 2010). The credit risk capital of banking firms is calculated using EAD and loss-given default (LGD) and probability of default (PD).

2.6 Gross Domestic Product (GDP)

The sum of all monetary or market values of all finished items and services produced within a sovereign's boundaries over a given time period is known as the gross domestic product (GDP) (Goodhart, 2011). It serves as a comprehensive indicator of a sovereign's economic health since it is a broad measure of total domestic production. Gross national product (GNP), its precursor, was created in the 1930s to aid America's recovery from the Great Depression (O'Neill, 2014). This was driven by the government's lack of sufficient statistics on the condition of the economy at the time.

Abramovitz (1986) undertook a study to determine if policy actions were effective. Simon Kuznets, a Russian-American economist, compiled the first set of standard GNP accounts (Abramovitz, 1986). Abramovitz (1986) underlying concept was direct: they wanted to simplify economic production data into a single figure that would rise in good times and fall in bad. GDP is usually calculated on a yearly basis, but it is also calculated on a quarterly basis on occasion. The US government, for example, produces annualized GDP estimates for each fiscal quarter and the calendar year. Since the data in

this analysis of GDP is given in actual numbers, it is adjusted for inflation (Miu and Ozdemir, 2006). The GDP calculation takes into account all private and public consumption, investments, government outlays, paid-in building expenses, additions to private inventories, and the international balance of trade (Trading Economics, 2021). The external trade balance is the most important of all the factors that are used to calculate a sovereign's GDP. A sovereign's GDP increases when the total value of products and services sold to foreign sovereign exceeds the net value of foreign goods and services acquired by domestic consumers (Goodhart, 2011). When this happens, a sovereign is said to have a trade surplus. A trade deficit develops when the amount that domestic consumers spend on foreign items exceeds the entire amount that domestic producers are able to sell to overseas consumers (O'Neill, 2014). The GDP of a country tends to decline under this situation.

GDP can be computed in two ways (O'Neill, 2014): nominally and realistically, with the latter accounting for inflation. Real GDP is a better approach to depicting long-term national economic performance because it uses a stable currency. Assume that in 2010, a sovereign's nominal GDP is R50 billion. The nominal GDP of this country will increase to R75 billion by 2020. Over the same time span, prices increased by 100%. If you only consider the nominal GDP in this case, the economy appears to be thriving well. However, real GDP (rands) would only be R37.5 billion, indicating that real economic performance actually declined over this period at time. South Africa's GDP from 2010 to 2020 is illustrated in Figure 2.1 (Trading Economics, 2021) and is of great concern.

Figure 2.1: South Africa's GDP from 2010 to 2020 (Trading Economics, 2021).

2.7 Sovereign Bonds

A sovereign bond is referred to as a debt security issued by a government to generate wealth for government programs, debt repayment, interest payments on current debt, and other government expenditures (Buchheit and Gulati, 2002). Sovereign bonds can either be issued in the local currency or in a foreign currency. Governments use sovereign bonds to generate money in addition to tax revenue. A sovereign bond works similarly to a standard bond in that the buyer receives periodic interest installments from the issuer, in this case, the sovereign, and is repaid the face amount when it matures (Choi et al., 2012). The interest rate paid, or yield, is determined by the sovereign's risk profile, just like any other bond. If a country is deemed hazardous, the yield on a sovereign bond will be higher (Goodhart, 2011). To estimate the chance of a country defaulting on its sovereign debt, creditors look at the sovereign's economic profile, exchange rate, and political risks. Sovereign credit ratings are provided by rating organizations such as Standard & Poor's, Moody's, and Fitch Ratings to aid debt issuers in understanding the risks of issuing credit in a particular country (Luo et al., 2017). These organizations also give credit ratings to businesses and their corporate bonds.

2.8 Previous Literature

Currency crises related to the breakdown of a fixed exchange rate regime are becoming less common as more countries move towards flexible exchange rates (Pereira et al., 2018). In contrast, sovereign debt servicing problems and, in certain circumstances, outright defaults have become more regular in recent years (Buchheit and Gulati, 2002). However, the macroeconomic misalignment that contributes to debt crises are still poorly understood. The majority of the work has focused on attempting to foresee currency crises, with varying degrees of success, as well as the associated banking crises in some of these periods.

Although there is significant empirical literature on "twin" currency and banking crises, there has been little recent research on predicting sovereign debt crises (Pereira et al., 2018). In recent years sovereign debt servicing problems have taken a variety of forms, ranging from outright defaults on domestic and external debt to rollover/liquidity crises, in which a solvent but the illiquid country was on the verge of defaulting on its debt due to investors' unwillingness to roll over short-term debts approaching maturity (Reusens and Croux, 2017). For a broad sample of nations loosely characterized as having market access and different definitions of a debt crisis, Schimmelpfennig et al. (2003) apply a range of methodologies to examine the impact of macroeconomic fundamentals on the probability of sovereign default and a debt crisis. The crisis criteria in work done by Schimmelpfennig et al. (2003) are unique in that they cover not just situations of outright default or forcible restructuring, but also cases where a near-default was avoided because of large-scale public financing provided by the IMF (Calomiris, 2000).

The paper by Schimmelpfennig et al. (2003) was based on the following questions: What set of economic factors, is most likely to lead to a debt-servicing crisis? What role do such disparities play in the onset of a crisis versus the resolution of a crisis? Can we identify critical junctures beyond which the chance of default skyrockets? Can we create a debt crisis early warning system (EWS) model that can assist in anticipating vulnerability to a crisis early on?

For the 1970–2002 period, Schimmelpfennig et al. (2003) employed a panel data set for 47 market access countries. They estimated and employed debt crisis logit models, a binary recursive tree strategy, and a combination of the two. The classification and regression tree analysis was used in one EWS, while the logit model estimation was used in another. Schimmelpfennig et al. (2003) discovered that theory-based economic variables could forecast crises. Perhaps most importantly, they found that early warnings provided a solid estimate of the likelihood of entering a debt crisis. The paper outlines macro variables that affect investors' confidence and the sovereign's ability to service debt. This includes external debt ratios that measure solvency and debt sustainability, measures of illiquidity or refinancing risk, measures of external imbalance and debt-servicing pressures, other macro variables that affect investors' confidence and the sovereign's ability to service debt, macroeconomic (primarily monetary) instability, and some political-economic factors that lead to policy unpredictability. Schimmelpfennig et al. (2003) study formed the base to determine which variables should be considered in this thesis.

Creating early warning systems for sovereign credit has been a challenge. The lack of proper modeling and evaluation has proven that financial institutions may crash without proper early warning system models. An example would be the financial crisis that occurred in 2008, in which instant home loans were issued to almost anyone that applied, which resulted in a large number of defaults, which led the federal deposit insurance corporation (FDIC) to close four-hundred and sixty five (465) failed banks in the United States of America (USA) (Altman and Sabato, 2013). In that case, no proper models were put in place to assess different clients properly.

Dawood et al. (2017) evaluated the predictive capability of Western Europe's econometric models across Western Europe and emerging regions. This study developed models based in the region of the sovereign instead of using one, globally derived model. The purpose of the study was to outline the distinct nature of sovereigns' economies, their vulnerability to shocks, and the effectiveness of their institutions' policy responses. Dawood et al. (2017) imply that the causes of sovereign debt differ as assessed by region. This shows that the factors that influence sovereign defaults vary. The data set used in this thesis is similar to the data set as that used in the (Dawood et al., 2017) study with two missing variables as discussed in Chapter 5. The models used to determine the plausibility of Dawood et al. (2017)'s study are binary logit, multinomial logit, and the dynamic signal extraction approach. In terms of model performance, Dawood et al. (2017)'s results suggests that the regional models perform better than a general model.

Dawood et al. (2017) examined the efficacy of multiple econometric techniques, including binary logit, multinomial logit, and dynamic signal extraction, that have recently been used in Western Europe to

create more effective EWS for sovereign debt crises in industrialized and emerging countries. These models reveal that it is critical to incorporate factors that account for the likelihood of spillover from the banking sector and the foreign exchange market in order to design an effective EWS for sovereign debt crises. Dawood et al. (2017) concluded that the binary logit model outperformed other classifiers in producing fewer false alarms and correctly forecasting the duration of a crisis. The binary logit model employing an innovative dynamic-recursive forecasting technique adequately anticipated the majority of out-of-sample crisis onset periods, while producing half as many false signals as the dynamic signal extraction technique. In this thesis four techniques were considered when developing sovereign default EWS models, namely the binary logit, probit, BART and ANN.

When considering the results without cross-validation, the BART model had the best performance in terms of its prediction capability and it generated the best predictive performance metrics scores, followed by the logit, probit and ANN models. The European model was not considered, since the data set does not have sufficient defaults to build an adequate model.

When considering the 5-fold cross-validation analysis, the ANN technique generated models that performed extremely well, generating the best scores in nine (9) of the twelve (12) models. The BART model came second, followed by the logit and probit tied for third place. The readings from the European model were not considered, since the data set does not have sufficient defaults to build an adequate model.

Chapter 3

Classification

Classification refers to a method used to predict qualitative responses based on a training set containing observations (Kotsiantis et al., 2007). An example of classification would be looking at an input vector X with a response Y. If the independent variable is not quantifiable and is in the form of categories, it is then referred to as a qualitative response variable (McFadden, 1984). Classification is also referred to as a method for pattern recognition in machine learning, which is typically an instance of supervised learning (Breiman, 2001). In addition, classification is a statistical model developed to classify observations into groups or classes (Sun et al., 2009). An example would be looking at a data set used to determine whether Ferrari clients will keep or sell their cars after ten (10) years. The model's outcome would be either "keep" or "sell", but the prediction model would be built based on historical variables that influence the outcome. This classification process is attributed to a machine learning model's ability to learn the relationship between input variables and the response variable.

Supervised learning refers to building a model with historical data of both the input variables and response (Sathya et al., 2013). This method allows the model to learn what inputs influence different responses. An example would be determining whether a Ferrari client will "keep" or "sell" their cars after ten (10) years, where the data used to construct the classifiers relies on the response variable "keep" or "sell". Unsupervised learning methods build a model without using the response variables in the model learning or estimation stage (Sathya et al., 2013). These models look for patterns, similarities, or outliers within a given data set and uses this information to classify observations. This thesis focuses on supervised learning, since the classifiers that will be considered will be constructed using the response, as a dependent variable, and the independent variables to fit a model.

When developing a classification model, the data set is typically split into a training and a testing set (Dawood et al., 2017). Let the complete set be denoted by D. Denote the training set by D_{train} and the test set by D_{test}. D_{train} is used when developing or fitting the model. This data set is usually larger than the testing set, since the data set is used in estimating the parameters of the model. This allows for the adequate fitting of the model while also outlining the best relationship between the dependent

and independent or feature variables (Kotsiantis et al., 2007). The testing set, D_{test}, is used when determining the predictive power of the model. The input variable of the test set is used in the model to observe the responses. These responses are then compared with the test responses using different measures to determine model accuracy and predictive power, see section 3.2

3.1 Imbalanced Data

In classification, imbalanced data refers to the scenario where classes are not equally represented (He and Garcia, 2009). For example, one might have one-hundred (100) instances of a 2-class or binary classification problem. A total of 80 cases are classified as class-1, while the remaining 20 are classified as class-2. This is an unbalanced data set with an 80:20 (or 4:1) ratio of class-1 to class-2 occurrences. This is referred to as imbalanced data because there is an un even distribution of the binary classes (Sun et al., 2009). Both binary and multi-class classification problems can have a class imbalance problem (He and Garcia, 2009). The majority of classification techniques can be applied to either. The subsequent discussions will assume a binary categorisation problem as classification of sovereign default as binary: default or non-default.

3.2 Evaluation Metrics for Imbalanced Classification

It is imperative to determine the metric used when evaluating a model. This is because if the wrong metric is chosen, a flawed model will be concluded as the best fit (Sun et al., 2009). The best fit model is a model that can predict or forecast more accurately given that it also produces a good score for the metric used to assess its predictability (Altman and Sabato, 2013). Selecting the best model is crucial, since it helps predict future outcomes or occurrences of the variable of interest. Determining the best metric is a challenge particularly for imbalanced classification (Zou et al., 2016). This selection process is complicated, since most metrics that are generally used assume that the classifications are balanced. Not all classes and prediction errors are equal in the case of imbalanced classification.

To determine the best metric, we first look at what an evaluation metric is. When training a model on a data set, a test set is used. The model will predict relevant results based on the test set inputs. The evaluating metrics compare the expected output of the test set, and the actual observations (Japkowicz, 2013). Model selection, as well as data preparation methods, are both guided by evaluation metrics. Tests are done on different models, and the results of the various tests are quantified with metrics (Sun et al., 2009). Evaluation measures are crucial when assessing a classifiers performance and hence guides the process of selecting a classifier.

Standard metrics for evaluating predictive classification models, such as classification accuracy and classification error, are frequently employed. Standard metrics function effectively for balanced data

(Zou et al., 2016). All measurements make assumptions about the problem or what is important in the problem. As a result, the analyst or project stakeholders must choose an assessment metric that best expresses what is believed to be significant about the model or forecasts, making selecting model evaluation metrics difficult. Sun et al. (2009) states that evaluation metrics are divided into three groups, namely

Threshold Metrics,

Ranking Metrics, and

Probability Metrics.

3.2.1 Threshold Metrics for Imbalanced Classification

Threshold metrics are metrics that quantify the classification prediction errors. Their purpose is to summarise the fraction, ratio, or rate of a models performance when the predicted class outcome does not equal the actual outcome in a test set (Zou et al., 2016). The most generally used threshold metric is classification accuracy:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}.$$

The complement of classification accuracy is classification error that is:

$$\text{Error} = \frac{\text{Incorrect Predictions}}{\text{Total Prediction}}.$$

Despite its widespread use, classification accuracy is virtually always ineffective for imbalanced categorization. The reason for this is because a no-skill model that merely predicts the majority class can achieve a high accuracy or low error (Japkowicz, 2013). When dealing with imbalanced classification, the class with most observations is typically referred to as the negative outcome, for example "No default" or "Negative test". The majority class is typically referred to as the negative class and the minority class as the positive class,

The confusion matrix is a quick and effective summary of performance in binary classification. The confusion matrix provides more details which is not just the predictive performance of a model as it indicates which classifications are correctly or incorrectly assigned, as well as the errors involved in the predictions (Zou et al., 2016). In a confusion matrix, all cells have a specified and well-understood name, summarized as follows (Zou et al., 2016):

Table 3.1: The confusion matrix.

		Model prediction	
		Positive	Negative
Observed Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

A true positive (TP) is an observation which the model correctly predicts as a positive. A true negative (TN) is an observation that the model correctly predicts as negative. A false positive (FP) is an observation that the model incorrectly predicts as positive. A false negative (FN) is an observation that the model incorrectly predicts as negative.The sensitivity-specificity and precision-recall metrics are two measures that may be effective for unbalanced classification because they focus on one class (Boyd et al., 2013). The true positive rate is referred to as sensitivity, and it quantifies the positive class predictions (Boyd et al., 2013). The complement of sensitivity, or the true negative rate, is specificity, which sums the negative class predictions (Boyd et al., 2013). The fraction of positive class examples that belong to the positive class is summarized by precision. Recall highlights how effectively the positive class was predicted. Their respective formulas are given as:

$$\begin{aligned} \text{Sensitivity} &= TP/(TP+FN), \\ \text{Specificity} &= TN/(TN+FP), \\ \text{Precision} &= TP/(TP+FP), \\ \text{Recall} &= TP/(TP+FN). \end{aligned}$$

When dealing with binary classification, the sensitivity and recall values are equal (Drews and Greenland, 1990).

3.2.2 Ranking Metrics

Ranking metrics focus on evaluating classifiers on how effective they are at separating classes (Cortes and Mohri, 2003). A classifier must predict a score or a probability of class membership in order to use these metrics. Different thresholds can be used to measure the effectiveness of the classifiers (as discussed in section 3.2.1). Models that retain a high score across a variety of thresholds have better class separation and are hence ranked higher (Narkhede, 2018). Standard ranking metrics include the receiver operating characteristic (ROC) curve or ROC analysis (Cortes and Mohri, 2003). ROC describes a method used to analyze binary classifiers by looking at the way they discriminate between classes. A ROC curve is a diagnostic graphic that illustrates a model's behaviour by computing the false positive and true positive rates for a series of predictions made by the model at various thresholds

(Hand, 2009). Recall or sensitivity is the true positive rate, and the rate of false positives are computed as follows: True Positive Rate $= TP/(TP+FN) =$ Sensitivity $=$ Recall and False Positive Rate $= FP/(FP+TN)$.

Each threshold is represented by a point on the plot, which is joined to form a curve, see Figure 3.1. A diagonal line from the bottom left to the top right represents a classifier with no skill, that is a classifier predicting the majority class under all thresholds (Narkhede, 2018).

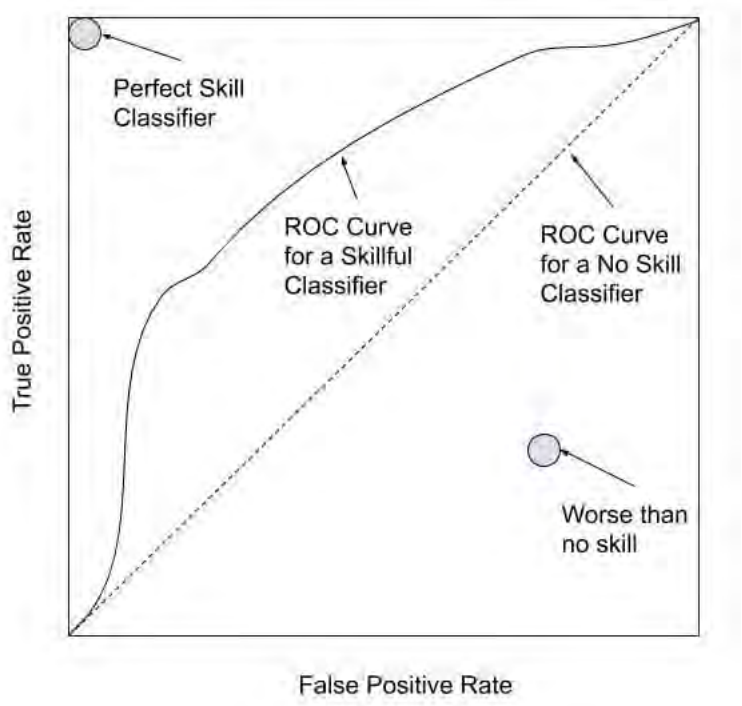

Figure 3.1: Receiver operator curve (ROC) plot (MachineLearningMastery, 2021).

According to Narkhede (2018) any points below this line indicate a lack of expertise. An ideal model would be a point in the plot's upper left corner.

Another measure of interest related to the ROC curve is the ROC area under the curve (AUC). The ROC AUC is calculated by finding the area under the ROC curve. This measure provides a single numerical summary of a plot and can be used to compare models. The higher the score, the more perfect the classifier. The ROC area under the graph is represented by: ROC AUC $= \int$ True Positve Rate d(False Positive Rate).

While generally helpful, the ROC Curve and ROC AUC can be overly optimistic in cases of significant class imbalance, especially when the number of samples in the minority class is minimal (Boyd et al., 2013). An alternative method to the ROC curve is the precision-recall curve. A precision-recall curve is similar to the ROC curve but concentrates on the classifier's performance on the minority class (Boyd et al., 2013). Different thresholds are applied to a set of predictions by a model, and the precision and recall are computed. The points create a curve. Classifiers that perform better across a range of thresholds are given a higher ranking.

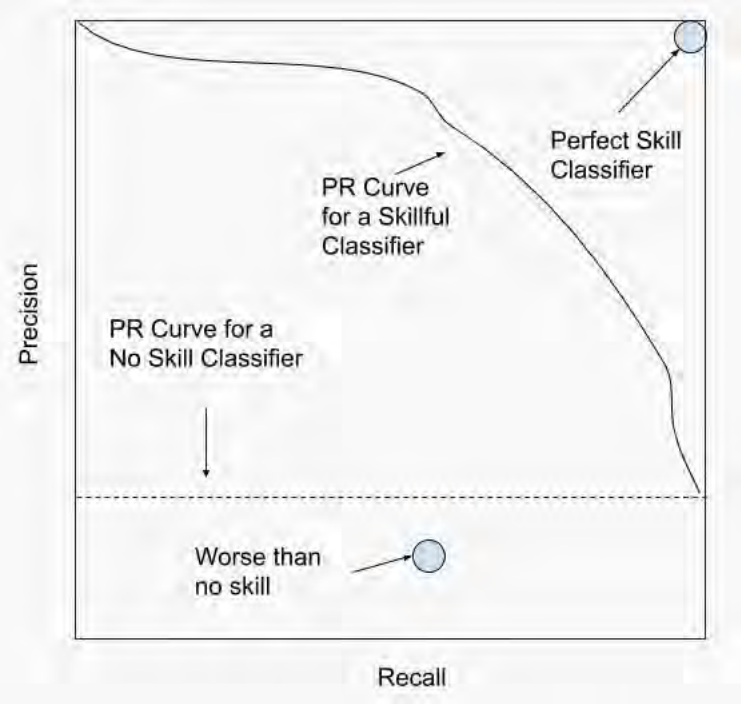

Figure 3.2: A precision-recall curve (MachineLearningMastery, 2021).

According to Boyd et al. (2013), the horizontal line on Figure 3.2 represents a no-skill classifier. The classifiers has a precision proportionate to the number of positive examples in the data set. This will be 0.5 for a balanced data set. A point in the top right represents a perfect classifier. The area under this curve is the ROC AUC. This value can be used to compare classifiers. The precision-recall AUC is more beneficial for imbalanced classification problems in this scenario because of the focus on the minority class (Boyd et al., 2013). The value for the precision-recall area under the graph is given as: PR AUC $= \int$ Precision d(Recall).

3.2.3 Probability Metrics for Imbalanced Classification

Probabilistic metrics are beneficial for when there is less concern about the correct as opposed to incorrect class predictions and more concern on the model's uncertainty in forecasts and penalizing erroneous but extremely confident predictions (Japkowicz, 2013). The probabilities must be calibrated before a model can be evaluated based on the anticipated probability. Specific classifiers are trained using a probabilistic framework, such as maximum likelihood estimation (Cao et al., 2014). A good example is a logistic regression. Since many nonlinear classifiers are not trained in a probabilistic framework, their probabilities must be calibrated against a data set before being evaluated using a probabilistic measure. Support vector machines and k-nearest neighbors are two examples of machine learning techniques that need calibration (Japkowicz, 2013). Log loss for binary classification, or negative log-likelihood which is also known as cross-entropy, is a commonly used metric for evaluating

projected probabilities (Vovk, 2015); For a binary classification where the expected values are y and the predicted values are $\hat{y}$, log loss is given by;

$$\text{Log Loss} = -\frac{1}{N}\sum y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i).$$

The Brier score is another prominent measure for probabilities. The Brier score has the advantage of concentrating on the positive class, which is the minority class in imbalanced classification (Rufibach, 2010). This distinguishes it from log loss, which considers the entire probability distribution; The mean squared error between the expected probability for the positive class and the projected probabilities are used to generate the Brier score; The average of the squared discrepancies between the values is the mean error;

$$\text{Brier score} = \frac{1}{n}\sum (f_i - o_i),$$

where f_i is the forecast probability and o_i is the outcome. The Brier score of a perfect classifier is zero (0).

3.3 k-Fold Cross Validation

For k-fold cross validation, the data are randomly partitioned into k disjoint approximately equal-sized sets (Refaeilzadeh et al., 2009). The validation set is one fold, the remaining $k-1$ folds are utilized to train or estimate the model (Xia et al., 2021). The algorithm undergoes k repetitions, with each fold serving as the validation set. Denote the k folds as $D_1, D_2, ..., D_k$. Denote the binary classifier generated from D/D_j as $M_j, j = 1,2,...,k$, where D/D_j represents the data set D with D_j's observations removed. The validation set D_j is used to evaluate the classifier M_j on each iteration. The accuracy in the jth fold is denoted by Acc_{υ}^{j} ,that is $Acc_{\upsilon}^{j} = \frac{1}{n_j}\sum_{i=1}^{n_j} I(y_i = \hat{y}_i, X_i \varepsilon D_j)$ with the associated error rate given by $Err_{\upsilon}^{j} = \frac{1}{n_j}\sum_{i=1}^{n_j} I(y_i \neq \hat{y}_i, X_i \varepsilon D_j)$ calculated for fold D_j. The total accuracy of a model utilizing the k-fold cross validation method is calculated by averaging the accuracy rates obtained from the k iterations, using:

$$Acc_{\upsilon} = \frac{1}{k}\sum_{j=1}^{k} Acc_{\upsilon}^{j} = \frac{1}{k}\sum_{j=1}^{k}\left[\frac{1}{n_j}\sum_{i=1}^{n_j} I(y_i = \hat{y}_i)\right].$$

Where $X_i \varepsilon D_j$ and $n_j = |D_j|$ is the number of data points in D_j. The total error rate of the model is calculated by averaging the error rates from the k iterations:

$$Err_{\upsilon} = \frac{1}{k}\sum_{j=1}^{k} Err_{\upsilon}^{j} = \frac{1}{k}\sum_{j=1}^{k}\left[\frac{1}{n_j}\sum_{i=1}^{n_j} I(y_i \neq \hat{y}_i)\right].$$

Where $X_i \varepsilon D_j$ and $n_j = \left|D_j\right|$.

Consider an example of 5-fold cross-validation see Table 3.2. The data set is divided into five folds, labeled $D_1, D_2, ..., D_5$.The training data, D_{train}, in the first iteration consists of all folds except D_1, i.e. $D_{train} = D \setminus D_1$ (Anguita et al., 2012). The validation set D_1 is used to calculate the accuracy, Acc_υ^1, and error rate, Err_υ^1. In the second iteration, the training data includes all folds except D_2, thus $D_{train} = D \setminus D2$. The validation set D_2 is used to calculate the accuracy, Acc_υ^2, and error rate, Err_υ^2 (Fushiki, 2011). The method is repeated for all five (5) folds. The overall accuracy and error rate are calculated as:

$$Acc_\upsilon = \tfrac{1}{5}\sum_{j=1}^{5} Acc_\upsilon^j \quad \text{and} \quad Err_\upsilon = \tfrac{1}{5}\sum_{j=1}^{5} Err_\upsilon^j \ .$$

Typically the value of k is set between 5 and 10 (Anguita et al., 2012). Cross validation with $k = n$ is known as leave-one-out cross-validation (Refaeilzadeh et al., 2009), as depicted in the table below.

Iteration	Training set	Validation set	Accuracy	Error rate
1	$D_{train} = D \setminus D_1$	D_1	Acc_υ^1	Err_υ^1
2	$D_{train} = D \setminus D_2$	D_2	Acc_υ^2	Err_υ^2
3	$D_{train} = D \setminus D_3$	D_3	Acc_υ^3	Err_υ^3
4	$D_{train} = D \setminus D_4$	D_4	Acc_υ^4	Err_υ^4
5	$D_{train} = D \setminus D_5$	D_5	Acc_υ^5	Err_υ^5

Table 3.2: The 5-fold cross validation process.

3.4 Root Mean Square Error

Root mean square error (RMSE) is a statistical metric used to measure model performance, calculated using the testing set independent variable and model outcomes. RMSE is known for its ability to represent errors adequately. Wang and Lu (2018) states that RMSE varies with the variability of the error magnitudes and sum of errors. The basic assumption when dealing with RMSE is that the errors are unbiased and are normally distributed. Therefore, RMSE assists in providing a somewhat accurate error distribution (Willmott and Matsuura, 2005). RMSE is simply the square root of the sum of the squared differences between the observed and predicted value denoted by n:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

where $\hat{y}$ represents the models predictions and y represents the actual observations. The lower the RMSE the better the fit (Wang and Lu, 2018).

3.5 Missing Data

When creating models, missing data occurs in practically all domains of research (Afghari et al., 2019). In statistical modeling, missing data may result in biased and inefficient model parameters (Graham, 2009). Bhaskaran and Smeeth (2014) describes three frameworks utilized in classifying of missing data:

1. Missing completely at random (MCAR): The likelihood of missing data in a variable is unrelated to itself or the other variables in the study. An example, consider investigating the weight of rats to determine life expectancy. The weight scales battery might go flat. In this case, some of the weight variable observations will not have readings due to such circumstances.

2. Missing at random (MAR): The likelihood of missing data in a variable is related to the study's other variables. For example an investigation of drinking addiction may be MAR if male members are less likely than female participants to complete a survey concerning drinking addiction.

3. Missing not at random (MNAR): A variable's chance of missing data is proportional to its own values. These could occur when, for example, the majority of the people who are absent from work are the most sick.

The MAR assumption can be made more reasonable by gathering more explanatory factors and including them in the study. It is not possible to discriminate between MAR and MNAR based on observable data alone. In this thesis, three missing data techniques were employed; case-wise deletion (CWD), multiple imputation through chain equation (MICE), and filling in the missing data with the mean of the variable.

3.5.1 Case Wise Deletion (CWD)

The data for any observation with one or more missing values is discarded using the case-wise deletion method (Afghari et al., 2019). If any record for variable x_i is missing, the observation is erased. As a result, model development occurs with fewer observations than the original data.

3.5.2 Mean Imputation

In this method, all missing values are simply replaced with the mean of the variable as shown by Donders et al. (2006). Extreme outlier observations are not included in the calculations of the mean imputation value ,i.e standard score of >3.5. In fact, imputing the mean maintains the mean of the observed data. As a result, when data points are missing totally at random, the mean estimate retains its objectivity and this demonstrates positive model development (Donders et al., 2006). The sample

size can be maintained to reach the full sample size by replacing the mean which further presents a benefit. Mean imputation will not prejudice the parameter estimate if the focus is on estimating averages. This is rarely the case in research studies, and the data is missing completely at random.

3.5.3 Multiple Imputation via Chained Equation (MICE)

Presented here is the multiple imputation via chained equation (MICE) algorithm. Consider a set of variables, $x_1, ..., x_n$, with some values. All missing values are initially filled at random. An initial variable with a minimum of one missing observation, say x_1, is regressed on the other variables, $x_2, ..., x_n$. Observations with an observed x_1 are barred from participating in the estimation (Casella and George, 1992). Proper imputation replaces missing values in $\mathbf{x}_1$ with simulated draws from the derived predictive distribution of x_1. All other variables, $x_1, x_3, ..., x_n$, are regressed on the missing values of a variable , say x_2. Missing observation prediction is limited to individual observations with known x_2 and utilize on imputed x_1 values. Missing values in x_2 are substituted once more using draws from x_2's posterior predictive distribution. The method is repeated for each missing value variable in turn: one round is a cycle. This method, similar to the one proposed in Casella and George (1992), is usually repeated for ten cycles to produce a single imputed data set to stabilize the results. Van Buuren et al. (1999) initially recommended twenty (20) cycles, but Slade and Naylor (2020) state that ten or even five cycles may be sufficient. Royston et al. (2011) conducted simple tests to see if the sampling distribution of the imputed data converged. They showed that more than ten (10) cycles are required for convergence only if the variables with missing values to be imputed are significantly linked, that is a correlation coefficient above 0.6. Such high correlations are uncommon in most of the applications that Slade and Naylor (2020) encountered. The technique is repeated M times in a row, resulting in M imputed data sets.

According to standard texts on M, small quantities of imputed data sets (M = 3 to 5) are sufficient (Buuren and Groothuis-Oudshoorn, 2010). In recent years, public opinion has turned in favor of higher M values. For example, Slade and Naylor (2020) recommend that M be at least equal to the fraction of incomplete cases in the data set. The rule will imply $M = 20$ if 80% of cases have complete data on all relevant variables. Larger values of M, in the range of 100 to 1000, are required to control Monte Carlo error in studies comparing approaches (Slade and Naylor, 2020). MICE can handle various variable types, since each variable is imputed using its own imputation model (for example, continuous, binary, unordered categorical, or ordered categorical). MICE is implemented as follows in this thesis:

1. A simple imputation is initially performed: A mean imputation is performed for the missing values in the data set. This initial imputation can be referred to as a "place holder".

2. Set the "place holder" mean imputation for one variable back to missing.

3. The observed variable from the previous step is regressed on other variables in the imputed model, which may have all the other variables in the data set. The variable is the dependent variable in a regression model; all the other variables are the independent variables.

4. The missing observations of the variable are then replaced with imputations from the regression model. If a variable is used as an independent variable in the regression model , both the observed and imputed values are considered.

5. Repeat steps 2-4 for each missing variable. All variables are considered in each all iterations. After the initial iteration, all missing values are replaced with imputed values from the regressions that consider the relationships observed in the observed data.

6. Repeat steps 2-4 for a set number of cycles. In this thesis five (5) iterations were conducted, with the imputations updated each time.

Chapter 4

Statistical Classifiers

A detailed linear regression analysis was conducted to get an understanding of what data was available in the various regions, to assess if there were linear relationships between the independent variables and the target (or dependent variable). Since the target is binary, these models did not meet the inference requirements of linear regression. The regression analysis was conducted and can be found in Appendix A. This chapter discusses the binary logit classifier (section 4.1), probit classifier (section 4.2), Bayesian additive regression trees (section 4.4), and neural networks (section 4.5) and their application to predict sovereign default.

4.1 The Logit Classifier

Financing strategies have a history of requiring liquidation prediction methods and the implementation of an accounting-based credit default model. Sun and Benekohal (2007) suggest that having a subjective credit quality assessment is preferable. A range of credit models are used to determine the creditworthiness of a sovereign (Wiginton, 1980; Trinkle and Baldwin, 2007; Manasse and Roubini, 2009; Trustorff et al., 2011). These studies made use of the following models: discriminant analysis, binary models such as the logit, neural networks, and support vector machines.

There are a variety of ways to predict the likelihood of bankruptcy or credit default (Goodhart, 2011). The logit-based approach appears to be widely employed as a method among researchers and practitioners (Aldrich and Nelson, 1984). When identifying binary models like the logit model, assumptions that contribute to the functional form of regressors and the distribution family of the error terms are made. To improve the accuracy of accounting-based credit models, Andrews and Monahan (1992), Dakovic et al. (2010), and Altman and Sabato (2013) implemented nonlinear transformations of the regressors that allow for heteroskedastic variances. Porath (2006) and Lennox (1999) argued that identifying the distribution family of the error terms has little to no impact on the accuracy of financial models. When it comes to these components of functional model identification, there has not been

enough research conducted on the importance of fixed versus variable parameters in accounting-based credit default models (Lennox, 1999). Kukuk and Rönnberg (2013) show that in an accounting-based credit default model, one could think of at least three reasons why parameters vary across a population:

1. Financial statements frequently reflect sector-specific characteristics (Sutcliffe, 2003). As a result, financial features may affect credit risk differently depending on sectors, sovereigns, accounting rules, or regulatory frameworks. This may result in a mixed relationship between accounting ratios and default risk.

2. Evidence-based models illustrate that credit models may be improved using non-financial data (Grunert et al., 2005). This information is difficult to obtain. Such an omission may result in heterogeneity.

3. Company-specific accounting policies within a cluster of otherwise homogeneous enterprises may impede financial figure comparability and promote heterogeneity (Malmendier, 2018).

By allowing some or all parameters to be stochastic, the hybrid logit model explicitly accounts for parameter heterogeneity (Train, 2009). This model creation methodology has been considered from an economic standpoint as the most promising discrete choice model (Hensher and Greene, 2003). Despite this, credit default modeling rarely uses the hybrid logit approach (Kukuk and Rönnberg, 2013). Jones and Hensher (2004) and Hensher et al. (2007) applied empirical methods in ordinal and multi-nomial situations, focusing on group membership proportions rather than individual probabilities. Many research studies do not use widely accepted validation criteria, such as Kukuk and Rönnberg (2013)'s discriminating power and interest calibrations in credit default models.

The binary response probability is commonly used as the default probability for bankruptcy prediction (Savona and Vezzoli, 2015). However, a large variety of explanatory variables are used. Just like in discriminant analysis, the logistic technique weighs the independent variables and assigns a score y_i in the form of a probability of default to each sovereign in the sample. Let y_i denote the response of the sovereign i's reaction to the outcome of the explanatory variables.

4.1.1 Logit Transformation

The first step is to define a model for the data's systematic structure (Sun and Benekohal, 2007). Let the probabilities π_i be defined in terms of a vector of observed covariates $\mathbf{x}_i$. Let π_i represent a linear function of the covariates. That is

$$\pi_i = \mathbf{x}_i'\beta = \beta'\mathbf{x}_i, \tag{4.1}$$

where β is a regression coefficient vector. Equation 4.1 is referred to as the linear probability model (Cakmakyapan et al., 2013). Ordinary least squares β(OLS) are used in the estimation of this model

through observing individual data (Hensher et al., 2007). An issue related to equation 4.1 is that the probability π_i on the left-hand side must range between zero (0) and one (1), whereas the linear predictor $\mathbf{x}_i'\beta$ on the right-hand side can take any real value. Thus there is no assurance that the predicted values will be within the correct range unless the coefficients are subjected to complex restrictions. Lennox (1999) suggests that a straightforward solution to this issue is to transform the probability to remove the range restrictions, and then model the transformation as a linear function of the covariates. This is done in the two following steps:

The first step involves re-evaluating the probability π_i by considering the odds:

$$\text{odds}_i = \frac{\pi_i}{1-\pi_i}. \tag{4.2}$$

The odds are described as favourable to unfavourable cases cases, or the ratio of the probability to its complement (Cakmakyapan et al., 2013). The chances are one-to-one or even if the probability of an occurrence is half. Another example would be that the odds are referred to as one-to-two if the probability of occurrence is $1/3$. The odds are referred to as "long" if the probability is extremely small.

The second step involves taking logarithms and calculating the logit or log odds:

$$\eta_i = \text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right). \tag{4.3}$$

This transformation effectively eliminates the floor constraint (Hensher and Greene, 2003), since as the probability approaches zero (0), the odds approach zero (0) and logit approaches negative infinity $(-\infty)$. However, as the probability approaches one (1), the odds and logit both approach positive infinity $(+\infty)$ (Lennox, 1999). Therefore the logit transforms probabilities between the intervals (0, 1) to the complete real line. In the case where the odds are even, the logit is zero (0) if the probability is 1/2. Probabilities below half are represented by negative logits, whereas those over half are represented by positive logits (Cakmakyapan et al., 2013). An illustration of the logit transformation is shown in Figure 4.1. The binomial mean $\mu_i = n_i\pi_i$ can be used to define logits as the log of the ratios of expected successes, μ_i , to expected failures, $n_i - \pi_i$, (Berry et al., 2010). The logit transformation is one-to-one. Logits can be inversely transformed into probabilities using the inverse transformation, which is sometimes called the anti-logit (Sun and Benekohal, 2007).

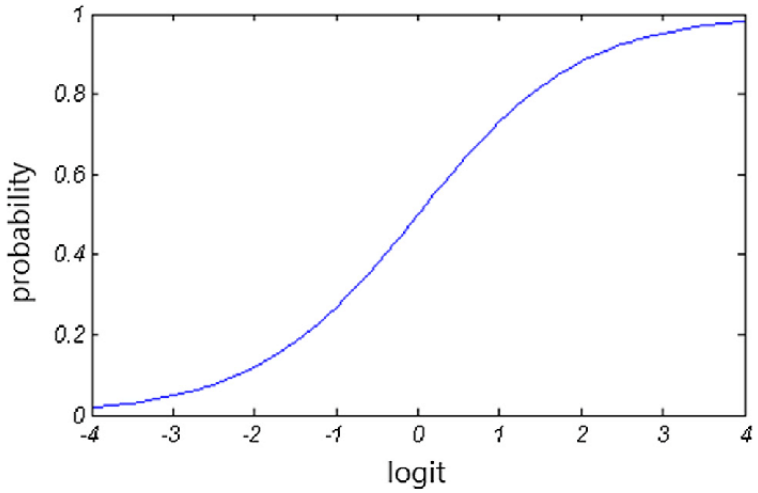

Figure 4.1: The logit transformation.

When solving for π_i in equation 4.3 the result gives

$$\pi_i = \text{logit}^{-1}(\eta_i) = \frac{exp(\eta_i)}{1 - exp(\eta_i)}. \tag{4.4}$$

Since the logit of the probability π_i is assumed to be linear, a logistic regression model can now be constructed, where η_i represents the log of the odds.

4.1.2 Logistic Regression

Assume that there are k independent observations $y_1, ..., y_k$ and the i^{th} observation to be an actualization of some random variable y_i. y_i is assumed to follow a binomial distribution

$$y_i \sim B(n_i, \pi_i), \tag{4.5}$$

where π_i is the probability and n_i being the binomial denominator. In the case where $n_i = 1$ for all i. This is how the model's stochastic structure is defined. Furthermore, lets assume that the logit of the underlying probability π_i follows a linear relationship with the predictors.

$$\text{logit}(\pi_i) = \mathbf{x}_i'\beta, \tag{4.6}$$

where β is a vector of regression coefficients and $\mathbf{x}_i$ represents a vector of covariates. This is the systematic structure of the model's definition. For the generalized linear model with binomial response and link logit, see equation 4.5 and 4.6. It should be noted that considering the distribution of the response y_i rather than the distribution of the implied error term $y_i - \mu_i$ is more logical (Berry et al., 2010), where μ_i represents the binomial mean. β coefficients can be read similarly to linear coefficients, with

the exception that the left-hand side of equation 4.6 is a logit rather than a mean (Wiginton, 1980). In other words, if we hold all other predictors constant while making a unit change to the j^{th} predictor, we get the logit probability change represented by β_j (Cakmakyapan et al., 2013). Exponentiating equation 4.6 results in the odds for the i^{th} unit, that is

$$\frac{\pi_i}{1-\pi_i} = exp(\mathbf{x}_i'\beta).$$

This formula allows calculations of the odds using a multiplicative model (Aldrich and Nelson, 1984). As an example, while all other variables remain constant, changing one unit of the j-the predictor would result in an increase of $exp(\beta_j)$. As an illustration, consider the linear predictor $\mathbf{x_i'}\beta$ where x_j is increased by one resulting in $\mathbf{x}_i'\beta + \beta_j$. After considering the exponential, the results are $exp(\mathbf{x}_i'\beta)$ time $exp(\beta_j)$. The exponentiated coefficient $exp(\beta_j)$ denotes an odd ratio.

Equation 4.6's solution for the probability π_i yields the model

$$\pi_i = \frac{exp(\mathbf{x}_i'\beta)}{1-exp(\mathbf{x}_i'\beta)}.$$

There is no straightforward method to quantify the effect of increasing a predictor by one unit while keeping the other variables constant since the right-hand side of the this equation is a non-linear function of the predictors (Cakmakyapan et al., 2013). Taking derivatives with respect to x_j can yield an approximation to the solution, but this only makes sense for continuous predictors (Berry et al., 2010). Applying the quotient rule yields:

$$\frac{\mathrm{d}\pi_i}{\mathrm{d}x_{ij}} = \beta_j\pi_i(1-\pi_i).$$

The influence of the j-th predictor on the probability π_i is dependent on the coefficient β_j as well as the value of the probability (Jones and Hensher, 2004). In some circumstances, analysts evaluate this product by comparing its π_i value to the sample mean for example the percentage of cases in the sample that have the desired feature (Aldrich and Nelson, 1984). The result is close to how the covariates affects the response when it is near the mean.

4.2 The Probit Classifier

The multinomial probit (MNP) model was initially utilized in the econometric literature by Hausman and Wise (1978) and Sheffi et al. (1982). The probit model is favored because it eliminates the multinomial logit model's irrelevant alternative properties, such as not being able fit a better classifier than the probit in a random effects model with medium sized data set (Aldrich and Nelson, 1984). Recently researchers have only started using the MNP, due to computational difficulties in computing the max-

imum likelihood estimate (MLE) of the MNP (Harville, 1977). Stern (1992) proposed an alternate method known as the method of simulated moments (MSM) approach, which avoids using a direct evaluation of the likelihood function in favour of using simulated scores. Both the MLE and MSM techniques employ a method for determining choice probabilities. In the case of an MNP model entails $(p-1)$-dimensional integrals of a multivariate normal distribution above a cone, where p is the number of selection choices (Patil et al., 2017).

Trevino and Thomas (2000) show that when modeling sovereign credit ratings, the ordered probit technique is a sufficient economic tool. The ordered probit model is demonstrated to be more robust when compared to other econometric methods (Hausman and Wise, 1978). This is because the estimation takes into account the dependent variable's discrete, ordinal nature. The dependent variable is not continuously distributed. The absence of continuity implies that pursuing models that represent a qualitative dependent variable as continuous can lead to serious inference errors, as it applies to the use of ordinary least squares (Hung et al., 2013).

4.2.1 Probit Model Application

The probit model is a common financial institution default prediction model used by rating agencies like Moody's, S&P, and Fitch, among others (Hausman and Wise, 1978). An example of how the model has been used is demonstrated by Blackburn (2005). Modeling for client default is based on a non-linear function $f(\mathbf{x}_i)$, that is

$$P(Y = 1|\mathbf{x}_i) = f(\mathbf{x}_i).$$

The relationship assumed for the probit model is usually denoted as

$$P(Y = 1|\mathbf{x}_i) = \Phi(\beta'\mathbf{x}_i)$$

where Φ represents the distribution function of the standard normal distribution, except when the probability being predicted is very small or very large, the results of a probit model should be similar to those of a logistic regression model (Aldrich and Nelson, 1984). Feng et al. (2008), Hung et al. (2013), and Reusens and Croux (2017) discuss the application of ordered probit models to credit rating. For the binary case there are two rating categories, namely default, $y_i = 1$, and non-default, $y_i = 0$. The result of a latent variable, z_i, results in the default capability of case i, that is

$$z_i = \beta'\mathbf{x}_i + \varepsilon_i$$

where ε_i represents a random variable with a standard normal distribution and z_i being the results of the latent variable. The logit and probit models are both econometric approaches for analyzing

binary dependent variables (Aldrich and Nelson, 1984). The latent-variable approach assumes an unobservable or, latent, variable z which is related to the borrower's characteristics in a linear fashion, that is

$$z_i = \beta' \mathbf{x}_i + \varepsilon_i.$$

β_i represent the parameters that capture the impact of a change in the characteristics on a default, $\mathbf{x}_i$ represents the characteristics of the borrower, and ε_i is the residual variable that contains the variation not captured by the characteristics $\mathbf{x}_i$. The variable z_i represents the metrically scaled dependent variable and triggers the value of the binary default variable y_i (Hung et al., 2013).

$$P(y_i = 1) = P(\varepsilon_i > -\beta' \mathbf{x}) = F\left(-\beta' \mathbf{x}\right)$$

Where $F(.)$ denotes the unknown distribution function. The density function above assumes that the distribution function has a symmetric density around zero (0) (Hensher and Greene, 2003). The selection of the distribution function $F(.)$ is based on the residual distribution assumptions, that is the assumption about ε_i. Stern (1992) shows that, if a normal distribution is assumed, the probit model is given by

$$F\left(\beta' \mathbf{x}\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\beta' \mathbf{x}} \exp\left(\frac{-t^2}{2}\right) dt. \tag{4.7}$$

In the case that residuals are assumed to follow a logistic distribution, the logit model is given by

$$F\left(\beta' \mathbf{x}\right) = \frac{\exp\left(\beta' \mathbf{x}\right)}{1 + \exp\left(\beta' \mathbf{x}\right)}. \tag{4.8}$$

4.3 The Probability of Default

The estimation of probability of default (PD) can be achieved via the logit and probit models. However default probability cannot be viewed as a knowledge of PD in the case of single borrowers (Hung et al., 2013). The reported default frequencies, on the other hand, can be viewed as a PD for a large number of borrowers. Fit an ordinary least square regression, $\pi_i = \beta' \mathbf{x}_i + \varepsilon_i$ where i denotes the group formed by considering several observations. π_i denotes the default frequency observed in group i, and $\mathbf{x}_i$ is the characteristics observed for group i (Bennell et al., 2006). However, these models are inadequate. Consider the result $E(y_i|\mathbf{x}_i) = \beta' \mathbf{x}_i$. This expected value is not limited to the values zero and one and thus cannot be interpreted as a probability. Hence we can not assume that this linear function can predict a probability. The linear expression $\beta' \mathbf{x}$ is transformed by a non-linear link function F in the second step, yielding

$$\pi_i = F\left(\beta' \mathbf{x}_i\right) \tag{4.9}$$

Appropriate link functions transform the values of $\beta' \mathbf{x_i}$ to a scalar in the interval $[0,1]$. The link function chosen in equation 4.9 determines the type of model. A logistic link function yields the logit model, as given in equation 4.8. Choosing the normal distribution as the link function results in a probit model, equation 4.7. When estimating π_i using OLS, the resulting coefficients will be heteroscedastic, since $Var(\varepsilon_i) = Var(\pi_i) = P(\mathbf{x}_i).(1 - P(\mathbf{x}_i))$. One way to attain homoscedasticity would be to compute the weighted least squares (WLS) estimate of β in π_i (Cakmakyapan et al., 2013). Although possible, this is not a common practice since if the data is homoscedastic the standard errors are the same as the standard errors estimated by OLS (Lennox, 1999) . To observe default frequencies, the data has to be grouped before estimation (Dawood et al., 2017). Grouping considers practical problems like defining the size and number of the groups and treating different covariates within the single groups. A more efficient way of estimating the logit and probit models which do not require grouping is the maximum-likelihood (ML) method. Considering a binary dependent variable, the likelihood function is given by

$$L(\beta) = \prod_i P(\beta' \mathbf{x}_i)^{y_i} [1 - P(\beta' \mathbf{x}_i)]^{1-y_i} \tag{4.10}$$

In the probit model $P(.)$ is the normal density function; for the logit model $P(.)$ is the logistic density function (Hung et al., 2013). Model estimation of equation 4.10 is relatively computationally simple and not complex (Dawood et al., 2017). The ML-approach lends itself to a broad set of tests to evaluate the model and each variable. In most cases, the use of the link function is not theoretically driven (Sun and Benekohal, 2007). The differences in the results of both classes of models are often negligible (Stern, 1992). This is because both distribution functions have a similar form except for the tails, which are heavier for the logit model. The computation of the logit estimator is relatively straightforward, see for example Radloff (2019). In general, the coefficient of the logit model can be easily computed and interpreted. Transform the logit model, in equation 4.8, as follows;

$$\frac{\pi_i}{1 - \pi_i} = \exp\left(\beta' \mathbf{x}_i\right) \tag{4.11}$$

The left-hand side of equation 4.11 are the odds, i.e., the relation between default and non-default probability. Holding all other variables constant, a variation of variable x_k by one unit has an impact of $exp(\beta_k)$ on the odds, when β_k represents the coefficient of the variable x_k (Lennox, 1999).

4.4 Bayesian Additive Regression Trees Classifier

4.4.1 BART

Chipman et al. (2010) defined Bayesian additive regression trees (BART) as a learning technique that attempts to find the relationship between an unknown function $f(\mathbf{x})$ that forecasts an output y and an n-dimensional vector of inputs $\mathbf{x}' = (x_1, ..., x_n)$ when

$$y = f(\mathbf{x}) + \varepsilon, \text{where } \varepsilon \sim N(0, \sigma^2).$$

$f(\mathbf{x})$ represents the additive regression inputs and ε the residual. Consider approximating $f(\mathbf{x}) = E(Y|\mathbf{x})$, the mean of y given $\mathbf{x}$, with a sum of i regression trees: $f(\mathbf{x}) \approx h(\mathbf{x}) = \sum_{i=1}^{j} g_i(\mathbf{x})$ where $g_i(\mathbf{x})$ denotes a regression tree. In this context $f(\mathbf{x})$ can be approximated to

$$y \approx h(\mathbf{x}) + \varepsilon, \quad \text{where } \varepsilon \sim N(0, \sigma^2) \ . \tag{4.12}$$

A sum-of-trees model is an additive model with multivariate components (Chipman et al., 2010). In this context, a generalized additive model has the form,

$$E(y|\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) = \alpha + f_1(\mathbf{x}) + f_2(\mathbf{x}) + ... + f_n(\mathbf{x})$$

Unidentified smooth and non-parametric functions are referred to as $f_j(x)$'s (Chipman et al., 2010). Multivariate components, as opposed to additive models based on the sum of low-dimensional smoothers, can more easily integrate interaction effects (Cappa and Cantet, 2008). Compared to a single tree model, the sum-of-trees model can more easily incorporate additive effects. Ensemble methods are models that combine a number of tree models (Altman and Krzywinski, 2017). These methods include boosting (Natekin and Knoll, 2013), bagging (Lemmens and Croux, 2006) and random forest (Biau and Scornet, 2016), which make use of unique techniques to fit a linear combination of trees. Boosting fits a single tree pattern, using each tree to fit data variation not detailed by the previous tree's sequences (Friedman et al., 2000). Boosting is based on the idea that by combining the best feasible future model with the prior models, the overall prediction error is minimized (Lemmens and Croux, 2006). If a little modification in a case's prediction generates no change in inaccuracy, the case's subsequent goal outcomes are zero (0) (Chipman et al., 2010). Bagging and random forests use randomization to create a large number of separate trees, which are then averaged to reduce prediction variance (Altman and Krzywinski, 2017). Bayesian model averaging is an approach that results in a linear combination of trees applied to the posterior coming from a Bayesian single-tree model (Wasserman, 2000; Montgomery and Nyhan, 2010; Pratola, 2016). The expected forecast from individual trees is weighted using posterior probabilities in this model averaging (Wasserman, 2000).

Chipman et al. (2010) developed BART as a Bayesian technique that uses the sum-of-trees to express or approximate $f(\mathbf{x}) = E(y|\mathbf{x})$. BART improves the sum-of-trees model, equation 4.12, by providing a prior that regularizes the fit by minimizing individual tree impacts. Where the regression trees $g_i(\mathbf{x})$'s, become a dimensionally adaptive random basis of "weak learners" when they are applied (Chipman et al., 2010).

Strong learners are models that perform as well as possible on a classification or regression problem, whereas weak learners are simple models that perform only marginally better than chance (Cao et al., 2020). Chipman et al. (1998) showed that BART produces a sum-of-trees, each of which explains a little and varied piece of $f(\mathbf{x})$ by weakening the $g_i(\mathbf{x})$ effects. BART is not similar to posterior averaging of single tree fits of the whole function of $f(\mathbf{x})$. Instead, when fitting the sum-of-trees model, BART uses a modified version of Bayesian back-fitting MCMC (Hastie et al., 2000), which iteratively creates and fits consecutive residuals. As a result BART classification is similar to the gradient boosting method (Duan et al., 2020). BART is unique in how it weakens the individual trees by using a prior and iterative fitting instead of using Bayesian back fitting on a set number of trees (Abu-Nimeh et al., 2008a). BART can be thought of as a Bayesian non-parametric approach that fits a parameter rich model through utilizing a strong influential prior distribution. The back fitting algorithm, which is effectively an MCMC sample from the prompted posterior mean estimate of $f(\mathbf{x}) = E(y|\mathbf{x})$ for any input value $\mathbf{x}$, is used to draw inference from BART models. Point-wise uncertainty intervals for $f(\mathbf{x})$ are easy to gather from the drawn sample's corresponding quantiles (Chipman et al., 2010). Point and interval estimates are thus brought together for functionals of $f(\mathbf{x})$, such as partial dependence functions that show the $\mathbf{x}$ components' marginal effects (Kapelner and Bleich, 2013). By keeping watch of the relative frequencies with which $\mathbf{x}$ components appear in the sum-of-trees model iterations, BART can be utilized to identify the importance of components when explaining y's variation. This variable selection method is model-free because it does not concern an encompassing parametric model's usual claim (Chipman et al., 2010).

Zhang and Härdle (2010) used a nonlinear classification method known as Bayesian Additive Classification Trees (BACT), an extension of BART. The similarities between BART and BACT are that they are both Bayesian non-parametric additive models defined by a prior and a likelihood where the additive components are trees that are fitted by an iterative MCMC algorithm. In BACT, each tree learns unique parts of the function relating the dependent variable to the input variables. The sum of trees offers a flexible and robust model. Zhang and Härdle (2010) apply the BACT method to tell whether a firm was insolvent and various benchmark cases and application to credit risk modeling. BACT can generate flexible class boundaries and is a serious contender to the logit model, CART, SVM, random forest, and gradient boosting. Abu-Nimeh et al. (2008a) used BART to create an enhanced spam detection system to improve email privacy. BART is not used in its original form in this thesis, but as modified to apply the algorithm to classification problems, termed the CBART method. CBART performs promisingly for spam classification (Chipman et al., 2010).

Hill et al. (2020) analyzed BART's performance and they concluded that BART is flexible in fitting the mean structure of regression because of its non-parametric sum-of-trees model. They state that Chipman et al. (2010) developed an improvement by accommodating a wider variety of data structures and that improvements in the prior specifications resulted in more generalization to support high dimensions and better approximate smoother functions.

According to Chipman et al. (2010) the Bayesian additive regression tree (BART) model is composed of two parts: the sum-of-trees model and a regularization prior to the model's parameters.

4.4.2 Sum-Of-Trees Model

T is defined as a binary tree with a set of interior node decision rules and terminal nodes. Note this is a single tree. $M = \{\mu_1, \mu_2, ..., \mu_j\}$ represents a set of parameter values associated with each of T's j terminal nodes. Binary splits based on decision rules result in predictor spaces of the form $\{\mathbf{x} \in A\}$ against $\{\mathbf{x} \not\in A\}$ in the case of A being a subset of the range of $\mathbf{x}$ (Chipman et al., 2010). They are of the form $\{x_i \leq c\}$ against $\{x_i > c\}$ for continuous x_i and are based on the single components of $\mathbf{x} = (x_1, ..., x_n)$. There is a single terminal node of T for individual $\mathbf{x}$'s determined by the sequence of decision rules, from the highest to the lowest, resulting in the allocation to μ_i the value correlating to a terminal node. When looking at T and M, $g(\mathbf{x} : T, M)$ represents the function that allocates a $\mu_i \in M$ to $\mathbf{x}$. Thus,

$$y = g(\mathbf{x} : T, M) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2) \tag{4.13}$$

represents a single tree model; a classification and regression tree (CART) model (Chipman et al., 2010). In equation 4.13, the conditional mean of y given $\mathbf{x}$ $E(y|\mathbf{x})$ is equal to the terminal node parameter μ_i assigned by $g(\mathbf{x} : T, M)$. In this the notation, the sum-of tree model can be expressed as

$$y = \sum_{j=1}^{m} g(\mathbf{x} : T_j, M_j) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2) \tag{4.14}$$

Each binary regression tree T_j and its associated terminal node parameters , $\mu_{ij} \in M_j$, is assigned by the function $g(x : T_j, M_j)$ to x. The mean of y given x, $E(y|x)$, is equal to the total of terminal nodes $\mu_{ij} \in M_j$ allocated to $\mathbf{x}$ via the $g(\mathbf{x} : T_j, M_j)$ function. In the case where $m > 1$, individual μ_{ij} are part of $E(y|\mathbf{x})$, rather than the single tree model, equation 4.13. Each μ_{ij} represents an effect of when $g(\mathbf{x} : T_j, M_j)$ is dependent on a single component of $\mathbf{x}$. Thus represents an interaction effect when $g(\mathbf{x} : T_j, M_j)$ is dependent on an excess of one component of $\mathbf{x}$ (Chipman et al., 2010). The sum-of-tree model can incorporate both main effects and interaction effects. Equation 4.14 is based on trees of different sizes and there may be variations in the interaction effect's order. It is important to note that in the case where all terminal node assignments depend on one component of $\mathbf{x}$, the sum-

of-trees model goes through reduction and results in a simple additive function the total of a step function of the individual components of **x**. With many trees, a sum-of-trees model gains increased representation flexibility, enabling BART to have exceptional predictive power (Nalenz and Villani, 2018). The flexibility is a result of rapidly increasing the number of parameters. When considering a fixed number of trees m, each sum-tree-model is dependent on $(T_1,M_1),...,(T_m,M_m)$ and σ, which includes all the terminal node parameters and also the tree structures along with the decision rules. Every single tree leads to substantial redundancy across the tree components (Kapelner and Bleich, 2013). $\{g(\mathbf{x};T_1,M_1),...,g(\mathbf{x};T_n,M_n)\}$ can be regarded as an "over-complete basis" in the sense that multiple individual choices of $(T_1,M_1),...,(T_m,M_m)$ results in an identical function $\sum_{j=1}^{m} g(\mathbf{x}:T_j,M_j)$.

4.4.3 Regularization Prior

Chipman et al. (2010) imposed a prior across all the parameters of the sum-of-tree model. Priors that effectively regularize the fit by holding the unique trees' effects from being unduly influential were advocated by Nalenz and Villani (2018). In the absence of regularizing influence, many tree components would potentially overwhelm the rich structure of the sum of trees model equation 4.14, resulting in a limitation of the advantages of the additive representation concerning both computation and function approximation.

In practice automatic default specifications are recommended since they appear highly effective (Chipman et al., 2010). An initial step is to limit the interpretable hyper-parameters which govern the priors on T_j,M_j and σ. Chipman et al. (2010) showed that the recommended priors are found by making use of the observed variation in y to limit reasonable hyper-parameter values in the case where external subjective information is not available. Another method considers a range of plausible hyper-parameter values and then uses cross-validation to select from these values (Kapelner and Bleich, 2013).

4.4.4 Prior Independence and Symmetry

The regularization prior is simplified by restricting attention to priors where trees are independent of each other, thus

$$\begin{aligned} P((T_1,M_1),...,(T_m,M_m),\sigma) &= \prod_j P(T_j,M_j)p(\sigma) \\ &= \prod_j P(M_j|T_j)P(T_j)p(\sigma) \end{aligned}$$

and $P(M_j|T_j) = \prod_i P(\mu_{ij}|T_j)$, where $\mu_{ij} \in M_j$. Under such a priors, tree components (T_j,M_j) are independent of each other and σ, and the terminal nodes of individual parameters are independent. The simplicity of the prior specification scenario to the specification of form for just $P(T_j)$, $P(\mu_{ij}|T_j)$ and $P(\sigma)$ is a result of independence (Abu-Nimeh et al., 2008a). This is a specification that is further simplified by the use of identical forms for all $P(T_j)$ and $P(\mu_{ij}|T_j)$ (Kapelner and Bleich, 2013). The

identical prior forms as proposed by Chipman et al. (2010) for Bayesian CART are known as CGM98. These priors have exceptional computational benefits and their forms are maintained by a few interpretable hyper-parameters. Using data to yield an effective default specification for regularization of the sum-of-trees model can allow calibration. Kapelner and Bleich (2013) noted that due to the choice of hyper-parameter values for BART, BART is different from Bayesian CART.

4.4.5 The T_j Prior

The prescribed form of $P(T_j)$ is broken down into three specifications:

1. The probability that a node depth $d = (0,1,2,...)$ is non-terminal, denoted as,

$$\alpha(1+d)^{-\beta}, \quad \text{where } \alpha \in (0,1), \beta \in [0,\infty),$$

2. The distribution of splitting variable assignments at each interior node.

3. The splitting rule applied to individual interior nodes, with a variable splitting distribution condition.

For aspects 2 and 3, the simple defaults are described in CGM89. They are the uniform prior to present variables for aspect 2 and uniform prior to the discrete set of present splitting values in aspect 3. Even though it does not follow Bayesian logic, the last choice has the reconsideration of invariance under monotone transformations of the splitting variables (Chipman et al., 2010). In the case of a single tree model, that is $m = 1$, to model a complicated structure a tree with $f(\mathbf{x})$ terminal nodes is needed. For the sum-of-trees model, a regularization prior that keeps individual tree components small (Kapelner and Bleich, 2013) is needed.

4.4.6 The $\mu_{ij}|T_j$ Prior for $P(\mu_{ij}|T_j)$

In order to margin out μ_{ij}, Chipman et al. (2010) makes use of the conjugate normal distribution $N(\mu_\mu, \sigma_\mu^2)$ which has high computational benefits. When guiding the hyper-parameters μ_μ and σ_μ, $E(y|\mathbf{x})$ represents the total of m μ_{ij}'s below the sum-of-trees model. The induced prior on $E(y|\mathbf{x})$ is $N(m\mu_\mu, m\sigma_\mu^2)$ since the μ_{ij}'s are apriori independent and identical distributed. There is a high probability that $E(y|\mathbf{x})$ lies within y_{min} and y_{max}, which are the minimum and maximum of y in the data set. Thus we select μ_μ and σ_μ such that $m\mu_\mu - k\sqrt{m}\sigma_\mu = y_{min}$ and $m\mu_\mu + k\sqrt{m}\sigma_\mu = y_{max}$ for a chosen k.

Kapelner and Bleich (2013) show that the nature of the method is to select μ_μ and σ_μ such that $N(m\mu_\mu, m\sigma_\mu^2)$ allocates probabilities to the interval (y_{min}, y_{max}). Parts of the observed data, namely

y_{min} and y_{max}, are used to ensure the implicit prior $E(y|\mathbf{x})$ is correctly determined. As a result Chipman et al. (2010) assigned a high probability to the entire range of $E(y|\mathbf{x})$ values while keeping away from over-dispersion and over-concentration. Chipman et al. (2010) show that when these conditions are met, BART is very robust to changes in the exact specifications.

4.4.7 The σ Prior for $P(\sigma)$

The inverse chi-squared distribution $\sigma^2 \sim \nu\lambda/\chi^2_\nu$, is used as the conjugate prior. A data-informed prior approach is crucial to guide the specification of the hyper-parameters ν and λ (Kapelner and Bleich, 2013). This ensures that considerable probability is assigned to the region of credible values of σ. As seen in Chipman et al. (2010), prior degrees of freedom ν and λ are calibrated with the objective of making use of "rough data-based overestimate" $\hat{\sigma}$ of σ. Kapelner and Bleich (2013) concluded that, there are two typical selections for $\hat{\sigma}$, namely the "inexperienced" specification, in which the standard deviation of y is taken from $\hat{\sigma}$, and the linear model specification, where $\hat{\sigma}$ is taken as the residual standard deviation from a least squared linear regression of y on the initial $\mathbf{x}'s$. Applications of $P(\sigma)$ can be seen in Chipman et al. (2010).

4.4.8 Selection of m

The main difference between BART and boosting techniques is that when considering a fixed number of trees m, BART utilizes the iterative back-fitting algorithm to go over m trees (Chipman et al., 2010). Chipman et al. (2010) shows that when using BART to estimate $f(\mathbf{x})$ or predict y, it is best to assume m is an unknown parameter, which results in assigning a prior to m through the Bayes implementation of BART. One method that may yield the best fit for selecting m is cross-validation from a range of suitable options. Chipman et al. (2010) concluded that the predictive performance of BART models improves as m is increased from 1. A significant increase in predictability is observed until a certain point, then it levels off and then gradually starts to degrade for large values of m. It is crucial not to select an m that is too small. Chipman et al. (2010) show BART's predictive performance on multiple examples with a default $m = 200$ which is considered the BART default value.

4.4.9 Posterior Information Extraction

4.4.9.1 Bayesian Back-Fitting MCMC Algorithm

The Bayesian model specification generates a posterior distribution based on the observed data (Chipman et al., 2010). This relates to each of the unknowns that make up a sum-of-trees model. The complete calculation is impossible due to the large size of the parameter space. A back-fitting MCMC

approach can be used to sample from this posterior. Chipman et al. (2010)'s Bayesian setup generates a posterior distribution based on the observed data y as follows,

$$P(T_1,M_1),...,P(T_m,M_m,\sigma|y)$$

At the basic level, the approach is based on the Gibbs sampler (Smith and Roberts, 1993). Let $T_{(j)}$ denote the collection of all trees in the sum that are not T_j, and M denote the set of all trees in the sum that are not $T_{(j)}$. $T_{(j)}$ will be a collection of $m-1$ trees, and $M_{(j)}$ denotes the terminal node parameters. In this case, the Gibbs sampler involves m consecutive draws of (T_j,M_j) conditional on $(T_{(j)},M_{(j)})$: $(T_j,M)|T_{(j)},M_{(j)},...,M_m,\sigma,y$, where $j=1,..,m$. From the entire conditional, a draw of: $\sigma|T_1,...,T_m,M_1,...,M_m,y$.

For additive and extended additive models with fixed parameters, Hastie et al. (2000) considers similar use of the Gibbs sampler for posterior sampling. The result of the study indicated that it was a stochastic generalization of the back-fitting approach for such models. The algorithm is referred to as back-fitting MCMC. The selection of σ is merely a draw from an inverse gamma distribution and hence may be easily generated using standard methods. Complications arise when applying m draws of (T_j,M_j). This can be accomplished by utilizing the following reductions. The conditional distribution $P(Tj,Mj|T(j),M(j),\sigma,y)$ is exclusively dependent on $(T(j),M(j),y)$ through $R_j \equiv y-\sum_{k\neq j} g(\mathbf{x};T_k,M_k)$, resulting in a n-vector of partial residuals from a fit that does not include the j-th tree. As a result the m draws from (T_j,M_j) given $(T(j),M(j),\sigma,y)$ are identical to m draws from $(T_j,M_j)|R_j,\sigma,$ where$j=1,...,m$.

The latter formula is mathematically identical to the $R_j = g(\mathbf{x};T_j,M_j)+\varepsilon$, which is the posterior of a single tree model, where R_j is the data y. This is due to the use of the conjugate prior for M_j, namely

$$P(T_j|R_j,\sigma) \propto p(T_j)\int P(R_j|M_j,T_j,\sigma)P(M_j|T_j,\sigma)dM_j \tag{4.15}$$

This can be derived in closed form, based on a standard constant (Abu-Nimeh et al., 2008a). As a result, we may complete each drawing in two steps:

$$\begin{aligned} &\text{Step 1:} \quad T_j|R_j,\sigma, \\ &\text{Step 2:} \quad M_j|T_j,R_j,\sigma \end{aligned}$$

The Metropolis-Hastings (MH) algorithm can be used to create T_j's draws (Chipman et al., 2010), using one of four movements. This method offers a new tree based on the present tree. The moves with their respective proposal probabilities include growing a terminal node (0.25), pruning a pair of terminal nodes (0.25), updating a non-terminal rule (0.40), and swapping a rule between parent and child (0.10). Green (1995) states that the growing and pruning steps modify the number of terminal nodes. Green (1995) avoids the difficulties of reversible leaps across continuous spaces of various

dimensions by integrating out M_j in equation 4.15. The M_j selections are a collection of individual draws from a normal distribution of the terminal node μ_{ij}. The calculation of the successive residual $R_{(j+1)}$, which is essential for the following selection of T_j, is made possible by the M_j draw. Chipman et al. (2010) show that a sophisticated reversible jump is not required. Instead start the chain with m basic single-node trees, then repeats iterations until the chain converges satisfactorily (Abu-Nimeh et al., 2008a). In each iteration, each tree may change one or two decision criteria, or increase or reduce the number of terminal nodes by one. Each μ will change, either by vanishing or by forming a new μ, resulting in a change of σ. As the process iterates, it is not uncommon for a tree to become enormous and then collapse back down to a single node.

The sum-of-trees model permits "fit" to be freely moved from one tree to another with its many unexplained factors (Kapelner and Bleich, 2013). We might consider the method as being similar to sculpting a complex sculpture by adding and subtracting small dabs of clay because each operation only produces modest incremental changes to the fit (Abu-Nimeh et al., 2008a). Chipman et al. (2010) back-fitting MCMC technique blends better than the CGM98's single tree model MCMC approach. In cases where only the single tree models are evaluated, the MCMC algorithm is more likely to gravitate quickly toward a single large tree and then become stuck in its local neighborhood. As a result, Chipman et al. (2010) prescribe running BART on one lengthy chain rather than several beginnings.

4.4.9.2 Posterior Inference Statistics

Back-fitting is ergodic, resulting in a series of draws $(T_1,M_1),...,(T_m,M_m)$, all of which are converging in distribution to the posterior $P((T_1,M_1),...,(T_m,M_m),\sigma|y)$ (Abu-Nimeh et al., 2008a). The sum-of-trees functions induced the sequence,

$$f^*(.) = \sum_{j=1}^{n} g(\cdot;T_j^*,M_j^*)$$

converges to $P(f|y)$, the posterior distribution on the "true" $f(.)$, for the sequence of draws $(T_1^*,M_1^*),...,(T_m^*,M_m^*)$. After a suitable burn-in period, the series of f^* drawings, say, $f_1^*,...,f_K^*$, can be treated as an approximation, dependent on a sample of size K from $P(f|y)$ by running the algorithm long enough. The number of iterations required for reliable inferences will vary depending on the application. Chipman et al. (2010) demonstrates that the necessary iterations are relatively low. The average of the after burn-in sample $f_1^*,...,f_K^*$, is a natural choice for estimating $f(\mathbf{x})$ or predicting Y at a specific $\mathbf{x}$, in-sample or out-of-sample. Thus $\frac{1}{K}\sum_{K=1}^{K} f_k^*(\mathbf{x})$ is an approximation of the posterior mean $E(f(\mathbf{x})|y)$. Another approach would be to use the median of $f_1^*(\mathbf{x}),...,f_K^*(\mathbf{x})$, which approximates the posterior median of $f(\mathbf{x})$. The variation of $f_1(\mathbf{x}),...,f_K(\mathbf{x})$ can be used to estimate posterior uncertainty about $f(\mathbf{x})$. An illustration would be the interval between the upper and lower $\alpha/2$ quantile for $f_1^*(\mathbf{x}),...,f_K^*(\mathbf{x})$ which is a natural and convenient $100(1-\alpha)\%$ percent posterior interval for $f(\mathbf{x})$.

Other functionals of $f(\mathbf{x})$ can be estimated using $f_1^*(\mathbf{x}), ..., f_K^*(\mathbf{x})$. Friedman et al. (2000) indicates that the partial dependence function describes the marginal effect of one or more predictors on the response. The partial dependency function is defined as $f(\mathbf{x}) = f(x_s, \mathbf{x}_c)$ where $\mathbf{x}$ has been partitioned into the interest predictors, x_s, and the complement, $\mathbf{x}_c = \mathbf{x}/x_s$ for,

$$f(x_s) = \frac{1}{n}\sum_{i=1}^{n} f(x_s, x_{ic})$$

where x_{ic} is the data's ith observation of x_c (Chipman et al., 2010) and (x_s, x_{ic}) is not usually one of the observed data items. Simply computing

$$f_k^*(x_s) = \frac{1}{n}\sum_{i} f_k^*(x_s, x_{ic})$$

yields a draw from the induced BART posterior $P(f(x_s)|y)$ for any value of x_s. The upper and lower $\alpha/2$ quantiles offer endpoints of $100(1-\alpha)\%$ percent posterior intervals for $f(x_s)$, and the average of $f_1^*(x_s), ..., f_K^*(x_S)$ yields an estimate of $f(x_s)$.

Chipman et al. (2010) demonstrates that BART may be used to select variables by identifying those margins that appear most frequently in the fitted sum-of-trees models. When m is large, this technique is less effective since the redundancy provided by so many trees tends to mix in many irrelevant predictors with the important ones. BART tends to prioritize relevant predictors for its fit as m decreases and redundancy decreases. When m is small, the predictors "fight" for the best fit. This model-free approach of variable selection is done through observing what happens to the $\mathbf{x}$ component usage frequencies in a series of MCMC samples, $f_1^*, ..., f_K^*$, as the number of trees m is reduced. Let z_{ik} be the fraction of all splitting rules that employ the ith component of $\mathbf{x}$ for each simulated sum-of-trees model f_k^*.

$$v_i \equiv \frac{1}{K}\sum_{k=1}^{K} z_{ik} \tag{4.16}$$

For the i^{th} observation of x, equation 4.16 is the estimated use per splitting rule Kapelner and Bleich (2013). As m decreases, sum-of-trees models tend to favor the inclusion of x components that improve y prediction while excluding x components that are unrelated to y. A bottleneck appears to form as m decreases, causing the $\mathbf{x}$ components to compete for a spot in the sum-of-trees model. Chipman et al. (2010) concluded that the $\mathbf{x}$ components with the largest v_i's will provide the most significant information for forecasting y. Different techniques of assessing component usage might be valuable, such as weighting variables by the number of data points existing in the node, emphasizing the relevance of early node splits, see example in Chipman et al. (2010).

4.4.10 BART Probit for Classification

Abu-Nimeh et al. (2008a) states that the first obstacle faced when using BART is that it requires a continuous output variable rather than a binary one. For the sovereignty binary classification, from the training set we let $y = 1$ denote sovereign default and $y = 0$ denote no default. Define a latent variable z in relation to y in the spirit of equation 4.14 (Abu-Nimeh et al., 2008b),

$$z = f(\mathbf{x}) + \varepsilon, \quad \varepsilon \sim N(0,1).$$

Then let

$$y = \begin{cases} 1 & \text{if } z > 0; \\ 0 & \text{if } z \leq 0. \end{cases} \tag{4.17}$$

$f(\mathbf{x})$ is the sum-of-trees model, set σ to 1 in this case due to the binary nature of y (Abu-Nimeh et al., 2008b). The probit link function is applied to the default probability π and the function $f(\mathbf{x})$,

$$\pi = P(y = 1|\mathbf{x}) = P(z > 0|\mathbf{x}) = \Phi(f(\mathbf{x})), \tag{4.18}$$

where $\Phi(.)$ represents the cumulative density function of a $N(0,1)$ random variable. After selecting the prior distribution of m, the MCMC algorithm given for posterior computation, can utilize BART to learn $f(\mathbf{x})$ from data (Abu-Nimeh et al., 2008a) where if the fitted $f(\mathbf{x}) > 0.5$, estimate $y = 1$, otherwise, estimate $y = 0$. Abu-Nimeh et al. (2008b) also shows that equation 4.18 can also be used to estimate π. Let T signify a binary tree with a set of interior node decision rules and a set of terminal nodes, and let $m = \mu_1; \mu_2; ...; \mu_b$ denote a set of parameter values associated with each of T's b terminal nodes. The model's ith component, $g_i(\mathbf{x})$, is denoted by $g_i(\mathbf{x}; T_i; M_i)$. Let $T_{(j)}$ denote the set of all trees in the sum that are not T_j, and $M_{(j)}$ be the associated terminal node parameters. In the training data, let y represent the observed default state of sovereigns. The algorithm will use the posterior distribution to produce the outputs

$$P((T_1, M_1), ..., (T_m, M_m), z|y). \tag{4.19}$$

Instead of picking from $P((T_1, M_1), ..., (T_m, M_m), z|\sigma)$ m successive draws of tree component $(T_j; M_j)$ which depend on $(T_{(j)}; M_{(j)}; z)$ are typical draws from the new posterior given in equation 4.19 resulting in:

$$\begin{array}{c}(T_1;M_1)|T_{(1)};M_{(1)};y;z\\(T_2;M_2)|T_{(2)};M_{(2)};y;z\\\vdots\\(T_m;M_m)|T_{(m)};M_{(m)};y;z\end{array} \tag{4.20}$$

After this, z is drawn from the complete conditional:

$$z(T_1;M_1);...;(T_m;M_m),y. \tag{4.21}$$

Since the σ_i algorithm is set to 1, there is no need to draw σ (Abu-Nimeh et al., 2008a). The first stage is roughly the same as in the original algorithm. This is due to the fact that in equation 4.20, y provides no further information when z is given because y can be totally identified by z through equation 4.17. As a result, we may delete the redundant y in equation 4.20 and draw from equation 4.20 using the original g_i technique by replacing y with z and setting $\sigma = 1$. Because z is latent, we must take an extra step to extract z values in equation 4.21 (Abu-Nimeh et al., 2008b). $z_j|(T_1,M_1),...,(T_m,M_m),y$ is distributed as $N(\sum_{i=1}^{m} g_i(\mathbf{x};T_i,M_i),1)$ truncated at the left by 0 if $y_j = 1$, and distributed as $N(\sum_{i=1}^{m} gi(\mathbf{x};Ti,Mi),1)$ truncated at the right by 0 if $y_j = 0$. As a result, drawing from equation 4.21 is as simple as taking values from the normal distribution band and truncating them by 0 from the right or left depending on the value of y (Abu-Nimeh et al., 2008a; Hill et al., 2020).

In an MCMC run, simply select the best $f^*(\mathbf{x})$ from the sequence of visited models based on posterior probabilities, Bayes factor, or other criteria, and save it for future prediction. Because BART automatically screens input variables, the selected $f(\mathbf{x})$ may have a smaller number of input variables than π (Van Gestel et al., 2006). This makes it easier to predict upcoming sovereign defaults because considerably less information is needed to be taken from the quantitative and qualitative sovereign data.

A better technique to anticipate sovereign default, is to utilize the posterior mean of $f(\mathbf{x})$, which may be approximated by averaging the $f^*(\mathbf{x})$ over the many draws from equation 4.19, and to further evaluate the uncertainty of our prediction by looking at the variation throughout the draws (Kapelner and Bleich, 2013). However, this necessitates the storage of several models in a physical location for future usage. Using the best B fitted models for prediction, which account for 95% of the posterior probabilities across the space of sum-of-tree models, is a more realistic approach. When forecasting whether a new default scenario is or is not, B is usually a value less than twenty (20), and a far smaller number of input variables than π is likely to be employed (Abu-Nimeh et al., 2008a).

4.4.11 BART Variable Importance

Following the construction of a BART model, it is normal to raise the question: what variables are significant? The "inclusion proportions" of each *m* tree over the post-burn-in iterations of the Monte Carlo simulation are examined to determine variable importance (Kapelner and Bleich, 2013). The inclusion proportion for each given predictor is the percentage of times that a variable is picked as a splitting rule among all the splitting rules in the sum-of-trees model's posterior draws.

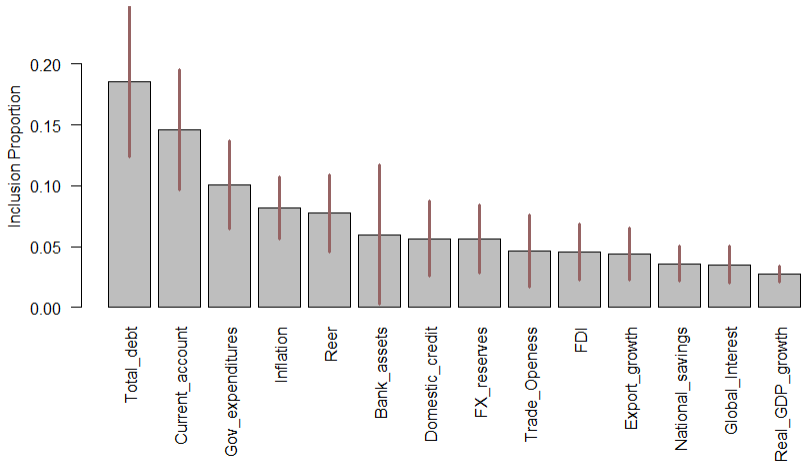

Figure 4.2: Inclusion proportion graph of the variables included in this study.

4.5 Artificial Neural Networks Classifier

The human brain has been of scientific interest for more than a thousand years. With the evolution of science and technology, it is only natural that science uses the same thinking process to compute results. The initial step toward developing artificial neural networks (ANN) was commenced by McCulloch and Pitts (1943), a neuro-physiologist and a young mathematician who researched how neurons might work. They built the first neural network with an electrical circuit. Hebb (1949) established that neuron pathways are strengthened each time they are utilized. As computers evolved, it became easier to model the significant aspects of theories concerning human thoughts. Nathaniel (1951), an IBM researcher, successfully simulated a neural network. During this period computing began to bloom, but primarily with a focus on computing. McCarthy et al. (1956) furthered artificial intelligence (AI) and neural network (NN) research, by focusing on the lower level of neural processing part of the brain. McCarthy et al. (1956) and Ulam (1958) enhanced simple neuron function through telegraph relays or vacuum tubes. Rosenblatt (1961), a neuro-biologist from Cornell, initiated his work on perceptrons. His interest was in the operation of the eye of a fly. The majority of the processing action that distinguishes a fly from a flee occurs in the fly's eye. The perceptron resulted from his research.

A single layer perceptron was analyzed to help classify a continuous-valued set of inputs into one of two classes. The perceptron computes a weighted sum of inputs from the training set, subtracts the threshold, and passes one of two possible values out as a result (Trinkle and Baldwin, 2007). Unfortunately the perceptron was limited, as shown by Minsky and Papert (1969). Widrow and Hoff (1960) developed models called ADALINE and MADALINE, named for their use of multiple adaptive linear elements. MADALINE was the first neural network to be applied to a real-world scenario (Widrow and Lehr, 1990). It is an adaptive filter for phone lines that removes echoes. This NN is still in use in the commercial world (Wang, 2003). People exaggerated the potential of neural networks due to their earlier accomplishments, in part owing to the limitations of the electronics available at the time (Dongare et al., 2012).

In 1982, several events ignited a new interest in NN. Hopfield (1982) presented his paper to the National Academy of Sciences. He showed how networks would work and their potential. The US-Japan Joint Conference on Cooperative and Competitive Neural Networks was hosted in Kyoto, Japan. At this conference Japan announced its Fifth Generation effort (Yadav et al., 2015). The 1985 meeting of the American Institute of Physics, Neural Networks for Computing was held with the purpose of furthering NN research in the USA. The initial International Conference on Neural Networks, held by the Institute of Electrical and Electronics Engineers (IEEE) in 1987, gathered almost 1800 people. The Small and Medium Business Innovation Research Program of the United States Department of Defense identified 16 subjects in which neural networks were thoroughly examined, with an additional 13 themes indicating the potential use of neural networks (Widrow and Lehr, 1990).

4.6 Neuron

Artificial neural network processes are based on the functions of the biological neuron. When observed under a microscope, neurons are formed with similar parts regardless of their different sizes and shapes (Dongare et al., 2012). A biological neuron has four regions these regions are dendrites, soma, axons and synapses, see Figure 4.3, (Wang, 2003).

Dendrites are root-like structures responsible for receiving signals from adjacent neurons and sending those signals to the soma (Dongare et al., 2012). The signal is an electric impulse that is chemically conveyed through a synaptic gap. The chemical process buffers the incoming signal by scaling the received signal's frequency, which is similar to the contribution of weights in a ANN. The soma is described as the core of the cell and is where the cell nucleus is found to maintain and grow the structure of the neuron (Abraham, 2005). The soma links the incoming signals, similar to the ANN combining method. When the signal information is adequate, the processed signal is carried by an axon. A axon is a single fiber connected to other neurons across a synaptic gap known as a synapse (Abraham, 2005). A synapse refers to the site of transmission of electric nerve impulses between two

nerve neurons (Dongare et al., 2012).

Wang (2003) shows that a cell either transmits a signal or does not at any given time. Thus signal transmission is seen as binary. This gives rise to the use of binary and sigmoid functions as activation functions in ANN (Dongare et al., 2012). Activation functions are essential since they form good relationships between the value of the function at a given point and the value of the derivative at that same point. This aids in reducing the computational burden when training an ANN (Abraham, 2005). The transmission rate varies and is observed as a signal of either high or low magnitude. This relates to observing discrete time steps, and adding all activities at a specific point in time in a ANN.

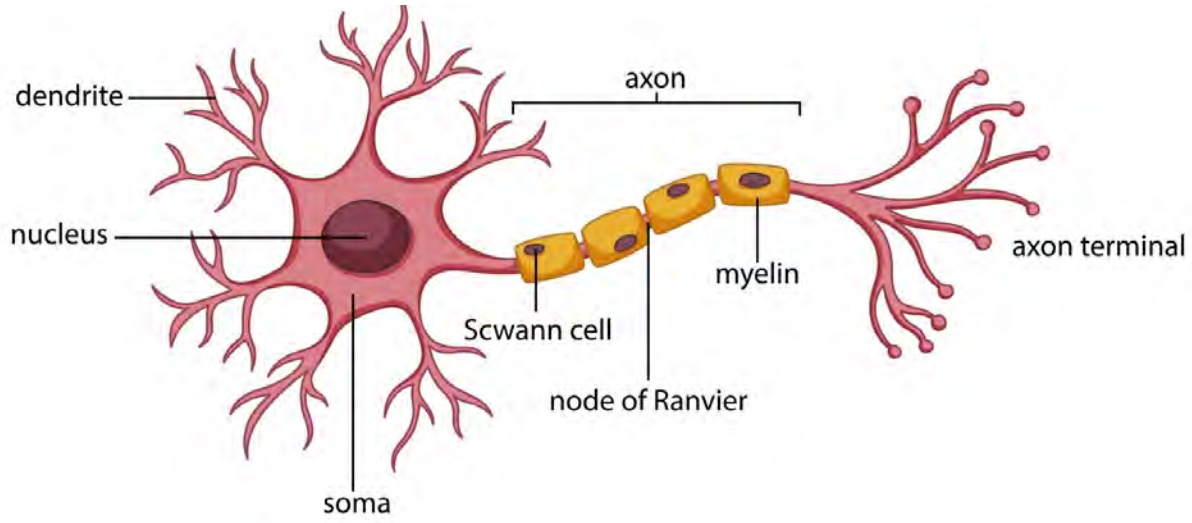

Figure 4.3: A biological neuron.(Abraham, 2005)

4.6.1 Artificial Neuron

Artificial neurons are the processing units that make up an artificial neural network (Yegnanarayana, 2009). An artificial neuron replicates the structure and behavior of a biological neuron. A biological neuron has one output, known as a synapse, which travels through an axon. A biological neuron has one set of inputs referred to as dendrites. The function of the neuron determines the activation of the neuron (Hopfield, 1982).

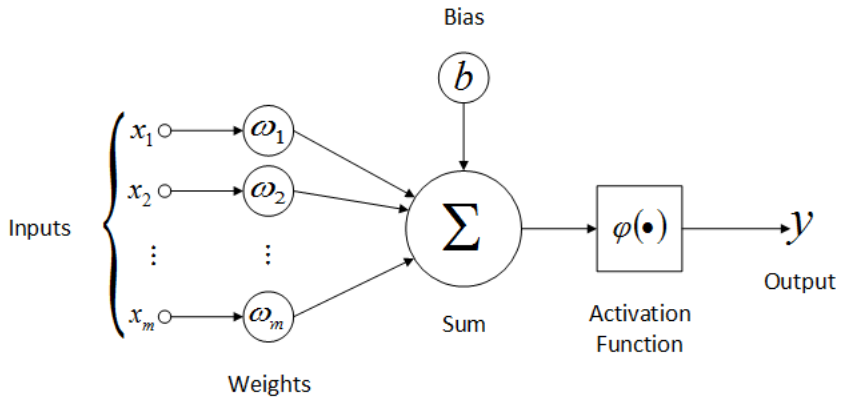

Figure 4.4: Model of an artificial neuron.(Bennell et al., 2006)

Consider Figure 4.4. A mathematical model of a biological neuron which has $x_1, ..., x_m$ independent inputs to the neuron and a bias which is included in the neuron along with the inputs. Typically the bias value is set to one (1) (Yegnanarayana, 2009). $\omega_1, ..., \omega_m$ denote the various weights. Weight represents the connection to the signal. The signal's strength is calculated as the product of weight and the input. A neuron gets several inputs from various sources and produces just one output. Activation can be accomplished via a variety of mathematical functions. The sigmoid function see Figure 4.5, is given by

$$F(x) = \frac{1}{1 + e^{-\Sigma_{i=1}^{n} x_i \omega_i}}$$

The sigmoid function is one of the commonly used activation functions. The sum $\sum_{i=1}^{m} x_i \omega_i$ equals the weighted sum of the inputs multiplied by the weights applied to each input. The sigmoid function, see Figure 4.5 , is a continuous and differentiable approximation of a step function.

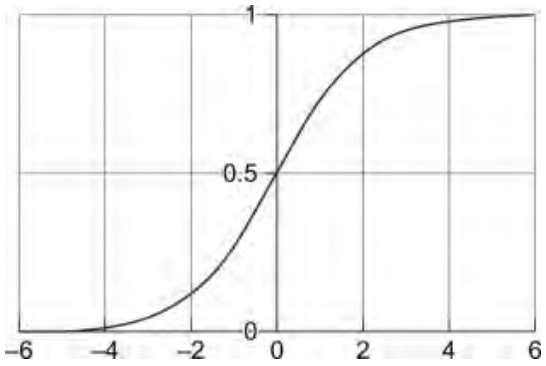

Figure 4.5: Sigmoid function.(Dongare et al., 2012)

The step, linear, ramp, and hyperbolic tangent functions are examples of other activation functions. The hyperbolic tangent (tanh) function curve is not similar to the sigmoid since its limit ranges from -1 to

+1, rather than from 0 to 1 (Hopfield, 1982). Step or ramp functions are utilized as activation functions. The perceptron is the most straightforward ANN architecture, consisting of one neuron with two inputs and one output. Multi-layer perceptions (MLP) are employed for more complex applications, and they consist of one input layer, an output layer, and one to many hidden layers. ANN architecture, see Figure 4.6 , comprises of:

1. An input layer: This layer is in charge of receiving input variables.
2. The hidden layer(s): The set of neurons located between the input and output layers which can consist of one or more layers.
3. The output layer: This is typically one neuron which has an output that ranges between zero (0) and (1). Multiple outputs can be present.

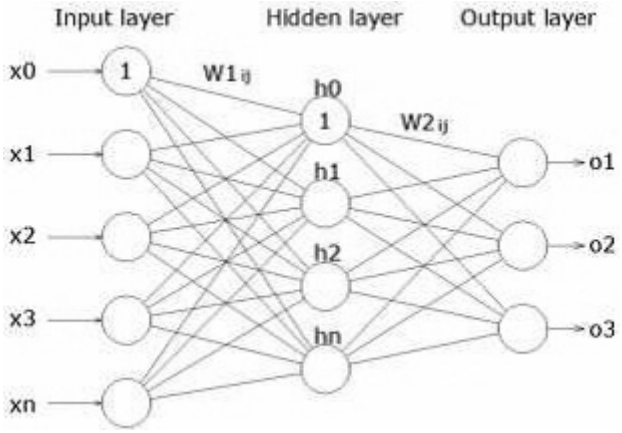

Figure 4.6: Neural network architecture.(Yadav et al., 2015)

Inter-unit connection strengths, known as weights, are used to store processing capabilities. The weight value determines the strength of the input. The value of the weight can be positive, negative, or zero. The signal is decreased or suppressed when the weight is negative. There is no connection between the two neurons if the weight is zero. To get the desired output, the weights are modified. There are algorithms for adjusting ANN weights to obtain the desired output. The process of changing weights is referred to as learning or training (Abraham, 2005).

The back propagation algorithm is frequently used as an approach for neural network training (Yegnanarayana, 2009). The difference between desired and actual output is propagated back to the layers, and the weights are adjusted. A supervised learning approach and feed-forward architecture are used in a back propagation neural network (BPNN) (Abraham, 2005). The outputs of hidden layers are propagated to the output layer (feed forward), where the output is calculated. For the provided input, this output is compared to the desired output. Based on this difference, the error is transmitted back from the output to the hidden layer and from hidden layer to the input layer. An epoch is a cycle that

progresses from input to output and from output to input (Dongare et al., 2012). A neural network is presented as a set of known input data and is then asked to produce a known output (Yadav et al., 2015). This is referred to as network training. The network runs through a series of these epochs until the error, that is the difference between actual and expected output, is within a particular tolerance. The network is now believed to have been trained. The weights between all of the neurons in all of the layers are adapted throughout the training phase. The weights obtained from a trained network are used to calculate the network's response to unknown data. More information on backpropagation can be found in Hecht-Nielsen (1992).

Chapter 5

An Analysis of Sovereign Default

5.1 The Data and a Preliminary Analysis

Dawood et al. (2017) used a data sample consisting of thirty-eight (38) advanced and emerging economies from 1980 to 2012. Because sovereign financial crises last for long periods of time (Schimmelpfennig et al., 2003), the article employs an annual frequency of data. A training data set was extracted from data entries between the years 1980 and 2005, and a testing data set was extracted from data entries between 2005 and 2012, referred to as in-sample and out-of-sample data sets, respectively, to create an EWS model and estimate the associated parameters. According to Dawood et al. (2017), this was difficult due to the small number of past sovereign debt crises in Western European countries. Dawood et al. (2017) selected data entries from five primary geographic regions: the Middle East and Africa, South and East Asia, Latin America, and Western Europe. Figure 5.1 is a graphical representation of the countries from which the data was collected.

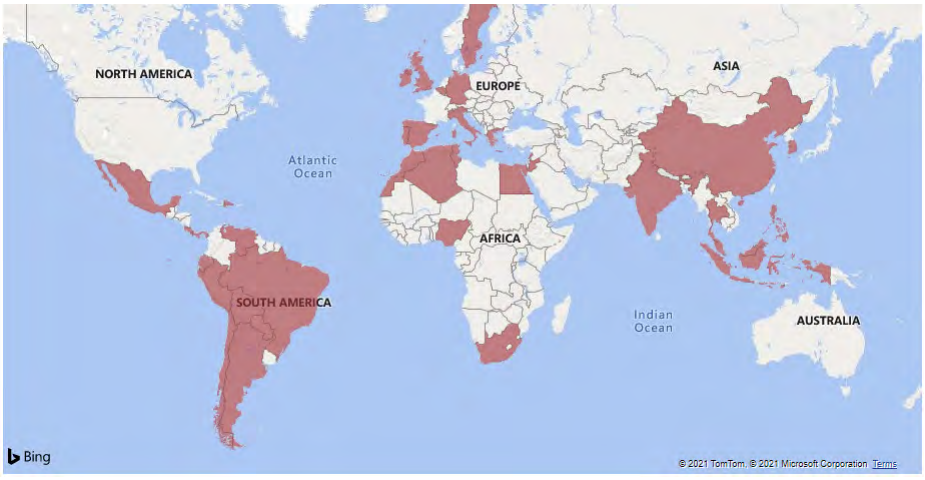

Figure 5.1: Graphical representation of countries involved in Dawood et al. (2017) study.

Schimmelpfennig et al. (2003) defined a country as being in a financial crisis if it is rated as being in

default by Standard & Poor's in order to indicate both actual and potential defaults on sovereign debt. Thus, a country is in default if it is either failing to meet its external financial obligations or has been granted a loan in excess of one hundred percent (100%) of its quota by the IMF as a means of resolving its financial problems. This definition was also applied in Fioramanti (2008), Manasse and Roubini (2009), Savona and Vezzoli (2015) and Zarantonello et al. (2013). According to Ciarlone and Trebeschi (2005), a country is in crisis if the unpaid interest or principal payments exceed five percent (5) of the sovereign's outstanding external debt, or if the sovereign is restructuring or rescheduling. These are additional criteria that should be taken into consideration, as done in this thesis. The following crisis definition was used in this thesis. The dependent variable, *y*, is set to one (1) if the emerging economy faces any of the following four occurrences:

1. Interest and principal arrears account for more than five percent (5%) of total outstanding debt,
2. Obtaining an IMF loan that exceeds the sovereign quota by more than a hundred percent (100%),
3. If a sovereign engages in debt restructuring and rescheduling schemes such as buybacks and reductions, and the total amount of IMF credit secured exceeds two hundred percent (200%) of the quota,
4. A debt rescheduling plan that involves more than twenty percent (20%) of the outstanding debt.

Due to the lack of published specifics on the arrears and amounts involved in restructuring and rescheduling plans, a different criteria was used for Western European countries. As a result, in addition to the two events involving IMF loans, the dependent variable for Western European sovereigns is set to one (1) if the outstanding government debt is greater than one hundred and fifty percent (150%) of nominal GDP. Showing different variables from the study and the sources that consider those variables for sovereign EWS model development.

Table 5.1: Variables considered in Dawood et al. (2017) study in order to predict sovereign debt.

Groups	Indicators	Measurement	Literature
Debt exposure	Total debt	Gross external debt as % of GDP	Ciarlone and Trebeschi (2005)
			Fuertes and Kalotychou (2007)
			Laušev et al. (2011)
	IMF credit	IMF credit Loans from IMF as % of GDP	Fuertes and Kalotychou (2006)
	Global interest	Global lending interest rate	Ciarlone and Trebeschi (2005)
External sector	Foreign Exch reserves	As % of GDP	Zarantonello et al. (2013)
	Trade openness	Ratio of exports plus imports to GDP	Fuertes and Kalotychou (2007)
			Fuertes and Kalotychou (2006)
			Zarantonello et al. (2013)
	Export growth	Annual exports growth rate	Ciarlone and Trebeschi (2005)
	Current account	Current account balance as % of GDP	Fioramanti (2008)
			Manasse and Roubini (2009)
	FDI	Net FDI inflows as % of GDP	Peter (2002)
Domestic macroeconomic conditions	Real GDP growth	Annual growth of real GDP	Fioramanti (2008),Savona and Vezzoli (2015)
	REER overall	Deviation of REER from 5-year rolling mean	Manasse and Roubini (2009)
			Pescatori and Sy (2007)
	Inflation	Rate of change in CPI	Schimmelpfennig et al. (2003)
			Manasse and Roubini (2009)
	M2/reserves	ratio of M2 to foreign exchange reserves	Lestano and Kuper (2003)
	National Savings	Ratio to GDP	Lestano and Kuper (2003),
			Zarantonello et al. (2013)
	Gov expenditure	As % of GDP	Laušev et al. (2011),
			Zarantonello et al. (2013)
Banking sector	Domestic credit	Ratio of domestic credit to GDP	Fuertes and Kalotychou (2006)
			Fuertes and Kalotychou (2007)
	Bank assets	Ratio of bank assets to GDP	Lestano and Kuper (2003)
	Gov bank loans	Net bank claims on central gov.	Manasse and Roubini (2009)

Dawood et al. (2017) suggest various indicator variables that can be used to provide warning signals of a future crisis. These variables were grouped into four main categories, see Table 5.1 showing the different variables considered along with the literature that supports their use.

1. Debt exposure illustrates a country's exposure to sovereign debt. External debt or IMF credit that exceeds a country's GDP raises the likelihood of unsustainable debt. The burden of servicing external debt is estimated using the GDP-weighted average of bank lending interest rates in seven main Western European nations.

2. External sectors are factors that raise the possibility of sovereign default owing to currency depreciation. A good current account balance, improved export earnings, and a net inflow of foreign direct investment (FDI) all help the sovereign lower its financial commitments for foreign debt

acquisition. A low degree of openness can impair trade surpluses and enhance a country's willingness to repudiate its debt, whereas an increase in openness exposes the sovereign to external shocks.

3. Domestic macroeconomic conditions are factors that are projected to deteriorate prior to a debt crisis. Reduced GDP growth, decreased national savings, increases in inflation, and the money and quasi-money (M2) ratio, a measure of the money supply that includes cash, checking deposits, and easily convertible near money, all indicate a sovereign may fail to meet its financial obligations. Foreign exchange reserves erode foreign competitiveness and indicate the magnitude of implicit government liabilities that are not backed by assets. This increase creates a confidence crisis, since lenders believe the government is seeking to inflate away the value of its external debt. Overvaluation of the actual exchange rate is believed to be necessary to account for the effect of an imminent currency crisis on the country's capacity to satisfy external obligations. Increased government spending can raise the chance of a debt crisis, which is why governments typically implement austerity measures during times of crisis. Thus, increased public spending is connected with peaceful times with a low probability of a debt crisis,

4. Banking sector conditions include increasing bank assets and a greater domestic credit ratio, which reflect the banking industry's expansion positively, the latter, however, increases the banking sector's exposure to macroeconomic shocks. Additionally, net bank claims on the central government, i.e., loans minus deposits (Dawood et al., 2017).

Data from this thesis was based on Dawood et al. (2017) data sets. The data was collected from three primary databases: The World Group (2021), Knoema (2021), and The Global Economy (2021). When collecting data from different data sources, there are variables where the data is either insufficient to build sovereign detection models or is unavailable. The missing variables based on these are IMF credit, M2 reserves, and government bank loans.

Based on Dawood et al. (2017), data was collected from three sources, namely The World Group (2021), The Global Economy (2021), and Knoema (2021). While collecting and compiling the data, it was noted that some entries and a few variables were not available. The percentage of data missing for total debt was 47.02%, global interest was 24.67%, REER was 15.98%, national savings was 11.43%, export growth was 3.84%, government expenditure was 1.72%, bank assets was 1.72%, domestic credit was 1.62%, trade openness was 0.91%, real GDP growth was 0.91%, FDI was 0.81%, current account was 0.4%, and inflation was 0.3% No data was available for M2/reserves, IMF credit, and government bank loans. As a result, these variables were not included in this study. The biggest issue with building machine learning models used in this study is that they can not be built with missing observations (Marlin, 2008). To resolve the missing data issue, three (3) techniques are utilized. A case-wise deletion data set was derived. In this data set, an observation of a sovereign was removed if

any entry of the sovereign variables were missing. In addition, if a variable has more than forty five percent (45%) missing data, the variable was not considered and removed from the data set. A mean imputed data set is generated by replacing missing values with the average of the variable. Extreme outlier values were not considered when generating the average value. A multiple imputation via chain equation data set was generated using an imputation technique that generates missing value substitution by means of regression, see Section 3.5. How this study aims to improve (Dawood et al., 2017)'s study is that it introduces the use of imbalanced data metrics, missing data techniques and also applies both the BART and ANN classifiers to the data. As a result of a the highly imbalanced data, it is important to note that majority of determining the best model will be conlcuded by looking at the ROC AUC score.

5.2 The Case-Wise Deletion (CWD) Data Set

The proportion of missing data for each variable is summarized in Figure 5.2. The variable is removed from the data set if more than forty-five percent (45%) of the observations are missing.

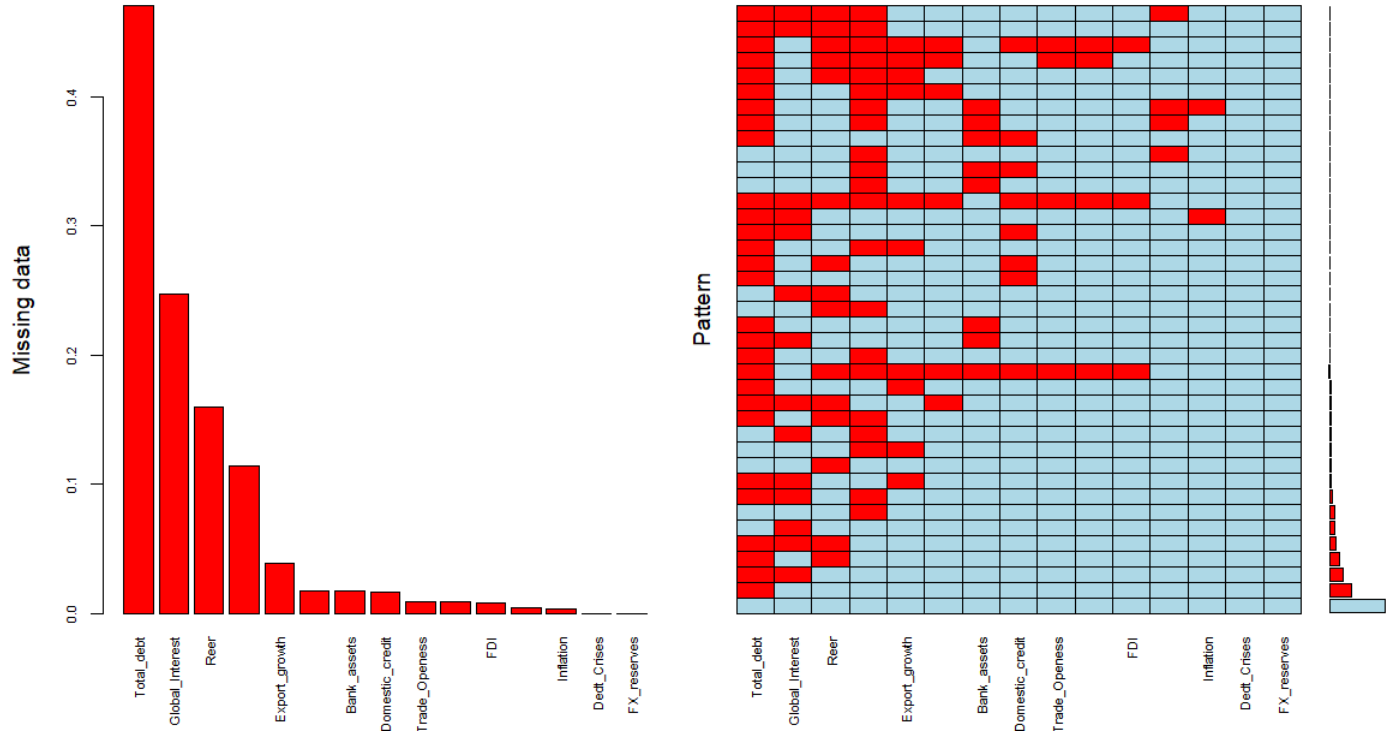

Figure 5.2: Missing data plot of all regions.

The variable total-debt has more than forty-five percent (45%) missing data. It will be entirely removed for the case-wise deletion data set (Allison, 2001).

5.2.1 Default of Sovereigns Per Region

The following are tables showing instances of when sovereigns defaulted along with what criteria brought about the anomaly.

Table 5.2: Crises episode for Latin American sovereigns based on Dawood et al. (2017).

Country	Crisis episode	Comment
Latin America		
Argentina	1983–1992	Arrears exceeded 20% of total debt
	2001–2005	Credit from IMF exceeded 500% of quota
Brazil	1983–1994	Massive rescheduling and restructuring schemes
	1999	IMF credit exceeded 200% of country quota
	2002–2003	IMF credit reached more than 600%
Mexico	1982–1990	40% of the debt was rescheduled or forgiven
	1995–1996	IMF loans increased to more than 600%
Chile	1983–1990	Rescheduling of 20% of debt
Paraguay	1986–1990	Defaults on 20% of debt
Dominican Rep.	1983–1999	IMF loans increased to about 250% of quota
	2003–2005	Rescheduling 20% of outstanding debt
	2010–2011	IMF credit reached 400% of country quota
Ecuador	1983–1995	Arrears reached 40% of total debt
	1999–2000	Rescheduling 30% of the debt
Venezuela	1989–1996	Rescheduling over 50% of the debt
Bolivia	1980–1985	Arrears increased to 20% of outstanding debt
	1986–1994	IMF credit reached 200% of quota
Peru	1980–1997	Arrears increased to 50% of outstanding debt
Panama	1983–1997	Arrears reached about 60% of outstanding debt
Costa Rica	1981–1991	IMF credit reached over 200% of quota

Table 5.3: Crises episode for South and East Asia sovereigns based on the criteria in Dawood et al. (2017).

Country	Crisis episode	Comment
South and East Asia		
Indonesia	1998–2003	IMF credit of about 400% of quota
Philippines	1981–1990	10% of debt was restructured
China	-	No significant external debt problems
India	-	No significant external debt problems
Malaysia	-	No significant external debt problems
Thailand	1981–1982	IMF loans increased to 280% of quota
	1997–1999	IMF loans reached 400% of quota
South Korea	1980-1982	IMF loans reached 400% of quota
	1997-1998	IMF credit accounted to over 1500% of quota
Lebanon	1985-1991	Arrears reached 12% of total debt
Singapore	-	No significant external debt problems

Table 5.4: Crises episode for Middle East and African sovereigns based on the criteria in Dawood et al. (2017).

Country	Crisis episode	Comment
Middle East and Africa		
Egypt	1980-1991	Arrears increase to more than 20% of debt
South Africa	1985-1989	Failure to meet about 50% of debt obligations
Morocco	1981-1989	Loans from IMF reached about 400% of quota
Tunisia	1986-1991	IMF credit reached 150% of country quota
Jordan	1989-1994	Default on more than 10% of the debt
Algeria	1990-1996	Rescheduling over 10% of debt principal
Nigeria	1988-1999	Defaulting on 60% of outstanding debt
Central Africa	1981-2006	Arrears increased to more than 30% of debt

Table 5.5: Crises episode for Western European sovereigns based on the criteria in Dawood et al. (2017).

Country	Crisis episode	Comment
Advanced Western Europe		
Greece	2010-	Debt reached 160% of GDP, IMF credit is over 1700% of quota
Portugal	1986	IMF credit amounted to 150% of quota
	2011-	IMF credit of 1700% of quota
Spain	-	No significant external debt problems
Ireland	2011-	IMF credit reached 1300% of quota
Italy	-	No significant external debt problems
Belgium	1992-1994	Debt increased to over 140% of GDP
Sweden	-	No significant external debt problems
Germany	-	No significant external debt problems
UK	-	No significant external debt problems

5.2.2 Box plot Visualization of the Independent Variables

A visualization of the data was conducted to determine how the chosen variables behave during default times compared to non-default times; this was done to observe whether there is a difference in characteristics of the variable between the two times. Summary statistics of each variable given the default and non-default periods are shown in Table 5.6. Univariate t-tests were used to assess if the means in these two periods were significantly different.

Table 5.6: Table showing the number of observations (n), average and variance of the default and non-default periods for all sovereigns along with the independent, two-sample t-test and F-test.

	Non-Crisis			Crisis			P-value		Observed	
Indicator	n	Mean	Var	n	Mean	Var	t-test	F-test	t-test	F-test
	Non Crisis			Crisis						
Debt exposure										
Total debt	461	60.2455	1 027.7843	62	78.6332	1 989.9166	0.0024	0.0002	3.1385	1.9361
Global interest	566	14.6212	101.9315	178	60.8498	112 735.2318	0.0679	<0.0001	1.8367	1106
External sector										
Foreign Exch reserves	719	26.3441	2 673.1509	270	4.6692	64.2109	<0.0001	<0.0001	-10.902	0.0239
Export growth	711	6.9760	84.1802	240	5.8506	114.9163	0.1462	0.0024	-1.4649	1.3638
Current account	715	-1.1696	41.9030	270	-2.6127	41.9607	0.0019	0.997	-3.1366	1.0011
Trade openness	714	74.7172	4 110.1180	266	56.3843	853.5021	<0.0001	<0.0001	-6.1242	0.2073
FDI	714	2.9014	19.1362	267	1.1807	3.0238	<0.0001	<0.0001	-8.8251	0.1578
Macroeconomic condition										
Real GDP growth	714	4.1483	14.5492	266	2.7150	41.1982	0.0007	<0.0001	-3.4093	2.8325
Inflation	716	7.7005	85.3442	270	161.0723	825 252.8097	0.0059	<0.0001	2.7741	9656.2
REER overall	641	114.3698	2 652.8640	189	110.3512	2505.3046	0.336	0.643	-0.9635	0.9443
Gov expenditures	710	15.05299	24.3036	262	12.0888	26.2612	<0.0001	0.4375	8.0937	1.0797
National savings	637	24.5964	93.5080	239	20.4815	87.1098	<0.0001	0.5227	-5.7483	0.9301
Banking sector										
Domestic credit	712	57.26424	1 059.6329	261	29.7244	642.7984	<0.0001	<0.0001	-13.857	0.6059
Bank assets	706	63.8180	1 208.7562	266	31.5247	660.4140	<0.0001	<0.0001	-15.774	0.5458

There are significant differences in the means of all variables, except REER and export growth, between the default and non-default periods. Total debt, trade openness, export growth, FDI and REER are all right skewed, see Figure 5.3, 5.4, and 5.5. GDP and current account are symmetric, but with many extreme observations, see Figure 5.4 and 5.3.

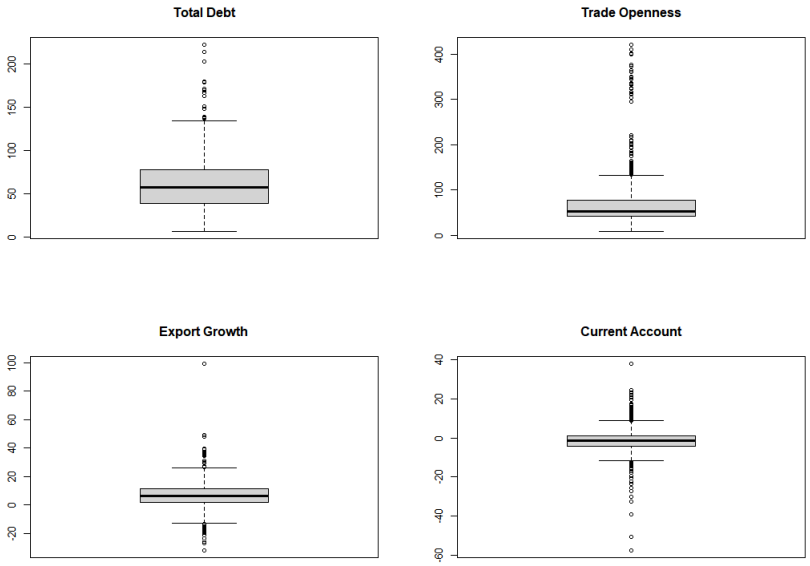

Figure 5.3: Box plot of the total debt, trade openness, export growth, and current account for all observations.

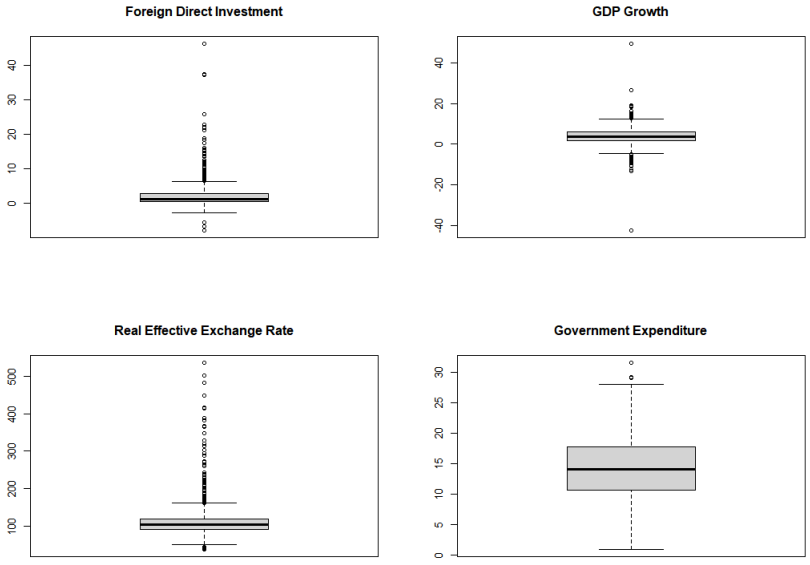

Figure 5.4: Box plot of the FDI, real GDP growth, REER and government expenditure for all observations.

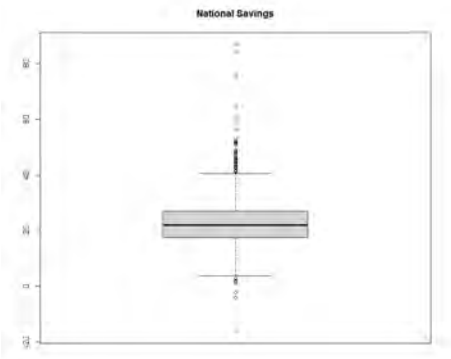

Figure 5.5: Box plot of the national savings for all observations.

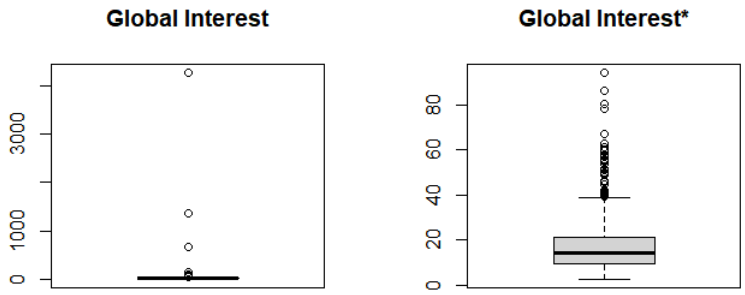

Figure 5.6: Box plot of the global interest with the full sample observation and trimmed sample observation. Trimming was a result of cutting out the outliers to have a better view of the boxplot distribution.

There are very extreme outliers in the global interest variable, see Figure 5.6. China brought about this anomaly in 2003. This resulted from the calls by United States lawmakers for legislation aimed at forcing the administration to take a stricter policy stand on China for keeping its currency relatively weak. This policy encouraged US trade protectionists to press Beijing for a rapid yuan appreciation in order to reduce China's deficit (Wei and Zhichao, 2009).

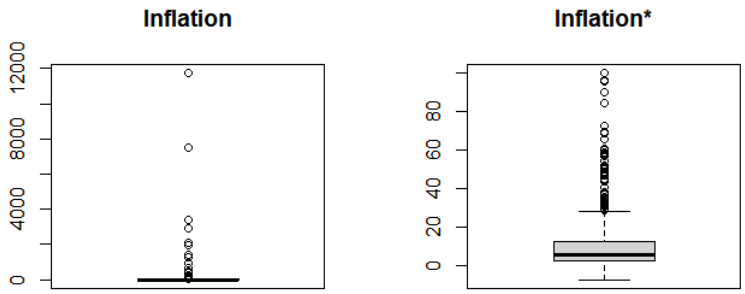

Figure 5.7: Box plot of the inflation rate with the full sample and a trimmed sample. Trimming was a result of cutting out the outliers to have a better view of the boxplot distribution.

There are extremely large values in the inflation data. The most notable observation comes from Bolivia in 1985, when the country printed money to cover its government's budget deficit (Pastor, 1991). The second highest extreme value is an observation from Peru in 1990, which resulted from bad policies implemented by General Alvarados, a realist policymaker who emphasized protectionism and questioned foreign aid (Deininger and Byerlee, 2011).

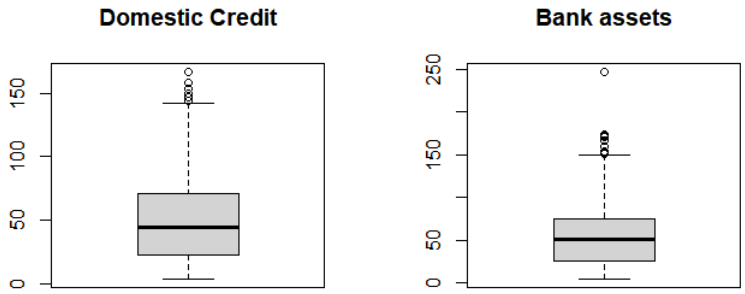

Figure 5.8: Box plot of the domestic credit and bank assets for all observations. As seen there is an extreme outlier, it is represented by Lebanon as a result of the IMF credit accounting to over 1500% of quota Dawood et al. (2017).

To formally test the significance of the detection model and associated variables' performance as signaling indicators of the sovereign debt crisis, four techniques are applied to the data sets: binary logit (Section 5.3), probit (Section 5.4), artificial neural networks (Section 5.5), and Bayesian additive regression trees (Section 5.6). These models are compared and contrasted in Sections 5.7 and 5.8.

5.3 Binary Logit Classification Models

Based on the partitioned training and testing sets, a binary logit classifier was built and tested. Post-crisis bias is avoided by looking at defaults based on the sovereign's individual entries (Dawood et al., 2017). The binary dependent variable y_i is assigned the value of one (1) for all crisis periods and zero (0) during non-default times.

As per the following sections, five models are developed considering the following missing data techniques; case-wise deletion, MICE, and mean imputation, which include all countries in the sample: the South and East Asia model, Latin America model, Middle East and Africa model, and the Western European model. For each model, the beta coefficients of the logged odds ratio, McFadden's Pseudo R^2, in and out of sample correctness, the deviance of the model, and metrics used in the evaluation of model performance for unbalanced data are reported. The dependent variable in these models is the debt crisis indicator variable. The time frame that is used to build the models is from 1980 to 2005, and the period that is used for the out-of-sample testing is from 2006 to 2012. The performance of the models listed below are bucketed into performance groups, this is applied to better elaborate the outcome of the predictions. The following are the different buckets for the ROC AUC: 0% to 50%, suggest no discrimination and, therefore poor, 51% to 69% represents slighlty accaptable which is considered moderately good, 70% to 80% is acceptable and therefore noted as good, 81% to 90% is excellent and anything greater than 90% is outstanding (Mandrekar, 2010). The main metric used to assess model prediction accuracy will be the ROC AUC, therefore conclusions will be based on the ROC AUC results. In the delevelopment classifiers, multicolinearity has been ignored. In the development of these classifiers multicolinearity has been ignored. As a result, interpretation of the variables in each model should be treated with caution.

5.3.1 Case-Wise Deletion Logit Models

Table 5.7: Binary logit model summary when applied to the global case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	−0.5961	2.6921	0.8247		Insignificant
Global interest	−0.0214	0.0355	0.5468	4.1892	Insignificant
Foreign Exch reserves	−0.0507	0.0332	0.1272	3.3223	Insignificant
Trade openness	0.0070	0.0077	0.3637	1.7043	Insignificant
Export growth	0.0379	0.0351	0.2796	0.4134	Insignificant
Current account	−0.0111	0.0893	0.9007	0.7972	Insignificant
FDI	−0.3474	0.1679	0.0385	3.7308	Significant
Real GDP growth	0.1592	0.1005	0.1132	1.0789	insignificant
Inflation	0.2083	0.0634	0.0010	4.5453	Significant
REER overall	0.0227	0.0128	0.0779	0.4439	Insignificant
Gov expenditures	−0.1928	0.0934	0.0390	3.4807	Significant
National savings	−0.1410	0.0673	0.0363	0.9959	Significant
Domestic credit	−0.0029	0.0214	0.8886	0.0239	Insignificant
Bank assets	0.0158	0.0255	0.5342	1.0244	Insignificant
Pseudo R^2	0.8373				
Out of sample correctness	0.9256				
In sample correctness	0.8611				
Deviance	80.5245				
Sensitivity	0.9739				
Specificity	0.5				
Precision	0.9256				
Recall	0.9739				
PR AUC	0.5295				
ROC AUC	0.7370				
Brier Score	0.0902				

From analysing the global data set considering case-wise deletion; the significant variables show to be FDI, inflation, government expenditure, and national savings. Based on the considered metric, the model is good at predicting sovereign default since it generated a ROC AUC value of 73.70%, see Table 5.7.

Table 5.8: Binary logit model summary when applied to the South and East Asia case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$5.622e^{-1}$	$3.659e^{6}$	1		Insignificant
Global interest	-1.856	$4.935e^{4}$	1	0.4308	Insignificant
Foreign Exch reserves	$-2.771e^{-1}$	$8.118e^{3}$	1	1.8503	Insignificant
Trade openness	$-2.486e^{-1}$	$5.205e^{3}$	1	0.2914	Insignificant
Export growth	-1.184	$6.163e^{4}$	1	1.3732	Insignificant
Current account	-3.338	$2.120e^{4}$	1	0.5108	Insignificant
FDI	$-6.747e^{-1}$	$2.561e^{5}$	1	2.6564	Insignificant
Real GDP growth	-2.950	$4.072e^{4}$	1	0.2585	Insignificant
Inflation	1.043	$5.590e^{4}$	1	0.6681	Insignificant
REER overall	$-1.505e^{-1}$	$1.269e^{4}$	1	2.0451	Insignificant
Gov expenditures	-8.986	$7.980e^{4}$	1	1.3282	Insignificant
National savings	2.646	$4.564e^{4}$	1	2.3608	Insignificant
Domestic credit	-1.278	$2.049e^{4}$	1	0.8211	Insignificant
Bank assets	1.680	$2.550e^{4}$	1	1.9252	Insignificant
Pseudo R^2	1				
Out of sample correctness	0.8571				
In sample correctness	0.8312				
Deviance	$8.7386e^{-10}$				
Sensitivity	0.857				
Specificity	0.875				
Precision	0.9333				
Recall	0.875				
PR AUC	0.4824				
ROC AUC	0.8661				
Brier Score	0.1304				

From analysing the South and East Asia data set considering case-wise deletion; there are no significant variables that could aid in the detection of sovereign debt. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 86.61%, see Table 5.8. The high standard error is a result of the deletion of entries which leads to the sample not closely representing the population (Afghari et al., 2019).

Table 5.9: Binary logit model summary when applied to the Latin America case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	−39.4790	26.0678	0.1299		Insignificant
Global interest	0.2169	0.2124	0.3072	3.3762	Insignificant
Foreign Exch reserves	−0.4102	0.3792	0.2794	2.2849	Insignificant
Trade openness	0.0606	0.0603	0.3152	0.2804	Insignificant
Export growth	0.2949	0.2035	0.1474	1.8792	Insignificant
Current account	−0.3886	0.4099	0.3431	2.3165	Insignificant
FDI	0.8711	0.8569	0.3093	1.6951	Insignificant
Real GDP growth	−0.4314	0.3755	0.2506	1.6962	Insignificant
Inflation	0.3579	0.2069	0.0837	3.6200	Insignificant
REER overall	0.2111	0.1292	0.1024	0.0513	Insignificant
Gov expenditures	−0.7465	0.6165	0.2260	0.8974	Insignificant
National savings	0.7153	0.5401	0.1853	1.2279	Insignificant
Domestic credit	0.7579	0.4791	0.1137	1.1146	Insignificant
Bank assets	−0.9607	0.5936	0.1056	1.2908	Insignificant
Pseudo R^2	0.9070				
Out of sample correctness	0.9076				
In sample correctness	0.7778				
Deviance	19.0254				
Sensitivity	0.9259				
Specificity	0.8182				
Precision	0.9615				
Recall	0.9259				
PR AUC	0.4845				
ROC AUC	0.8721				
Brier Score	0.0923				

From analysing the Latin America data set considering case-wise deletion; there are no significant variables that could aid in the detection of sovereign debt. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 87.21%, see Table 5.9.

Table 5.10: Binary logit model summary when applied to the Middle East and Africa case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$-2.983e^{2}$	$4.301e^{5}$	1		Insignificant
Global interest	5.630	$7.756e^{3}$	1	0.8094	Insignificant
Foreign Exch reserves	-5.115	$2.402e^{4}$	1	0.9027	Insignificant
Trade openness	$3.580e^{-1}$	$5.364e^{3}$	1	0.8494	Insignificant
Export growth	-1.411	$3.530e^{3}$	1	0.8645	Insignificant
Current account	$1.084e^{1}$	$7.868e^{3}$	1	0.9304	Insignificant
FDI	$2.573e^{1}$	$1.082e^{5}$	1	0.8005	Insignificant
Real GDP growth	3.233	$5.489e^{3}$	1	0.8244	Insignificant
Inflation	3.160	$4.896e^{3}$	1	0.7808	Insignificant
REER overall	$3.098e^{-1}$	$3.579e^{2}$	1	0.9386	Insignificant
Gov expenditures	7.440	$1.790e^{4}$	1	0.9129	Insignificant
National savings	$-6.641e^{-1}$	$5.281e^{3}$	1	1.0040	Insignificant
Domestic credit	$-8.645e^{-1}$	2.598^{3}	1	0.9108	Insignificant
Bank assets	$3.146e^{-1}$	$3.398e^{3}$	1	0.8719	Insignificant
Pseudo R^2	1				
Out of sample correctness	0.96551				
In sample correctness	0.8889				
Deviance	$5.7176e^{-10}$				
Sensitivity	0.875				
Specificity	0.9047				
Precision	0.9130				
Recall	0.875				
PR AUC	0.4922				
ROC AUC	0.8899				
Brier Score	0.1111				

From analysing the Middle East and Africa data set considering case-wise deletion; there are no significant variables that could aid in the detection of sovereign debt. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 88.99%, see Table 5.10. The high standard error is a result of the deletion of entries which leads to the sample not closely representing the population (Afghari et al., 2019).

For the Western European case-wise deletion data set, there was no default observation in the test and training data set. As a result, no model can be built.

5.3.2 Comparison of Global and Regional Binary Logistic Model Results for Case-Wise Deletion

Table 5.11: Binary logit models of sovereign debt from the year 1980-2005, case-wise deletion data set. Pseudo McFadden's R^2 and percentage of out of sample crisis predicted by each model, global and regional are reported. The Western Europe model is omitted as there are no observations for this region in the case-wise deletion data.

	Global	South and East Asia	Latin America	Middle East and Africa	Western Europe
Pseudo R^2	0.8373	1	0.9070	1	-
Out of sample correctness	0.9256	0.8571	0.7924	1	-
In sample correctness	0.8611	0.8312	0.7778	0.7586	-
Deviance	80.5245	$8.7386e^{-10}$	19.0254	$5.7176e^{-10}$	-
Sensitivity	0.9739	0.857	0.9259	0.875	-
Specificity	0.5	0.875	0.8182	0.9047	-
Precision	0.9256	0.9333	0.9615	0.9130	-
Recall	0.9739	0.875	0.9259	0.875	-
PR AUC	0.5295	0.4824	0.4845	0.4922	-
ROC AUC	0.7370	0.8661	0.8721	0.8899	-
Brier Score	0.0902	0.1304	0.0923	0.1111	-
sample size (n)	569	160	180	87	-

The results for Table 5.11 are as follows: The Western Europe model will not be considered since not enough observations are available to develop an adequate model. When considering the out-of-sample correctness, only one regional model, the Middle East and Africa model scored better than the global model, implying that the global model correctly classed more observations from the testing sample than the regional models. For the in-sample correctness, Table 5.11 shows that the global model, produced a higher score than the regional models, suggesting that the regional models struggled to predict more than 85% of the outcomes considered in the data used to develop the model. The global model produced the highest deviance, suggesting that the regional models had a better goodness-of-fit. In terms of sensitivity and recall, the global model generated the best sensitivity score, which suggests that the model had the highest true positive prediction. When considering specificity, the regional models produced the highest score, suggesting that they generated the best true negative prediction rate. When looking at precision, the global model, with the exception of the Latin America model, produced the highest score, which suggests that the global model could correctly classify a positive that is not a negative. Considering the PR AUC, the global model produced the highest score, implying that it did well in terms of classifying. The ROC AUC suggests that the regional models produced a higher score than the global model, which implies that they had the best performance measurements for classification at different thresholds. In terms of the Brier score, the global model generated the lowest score, which means that it had the lowest mean squared error between the predicted probability and observed value. When looking at the ROC AUC score on Table 5.11, the regional models performed better,

making using regional models the best classification method when case-wise deletion is considered for the logit classification method.

5.3.3 Mean Imputation Logit Models

Table 5.12: Binary logit model summary when applied to the global Mean Imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	0.8651	0.6961	0.2878		Insignificant
Total debt	0.0251	0.0046	$5.55e^{-7}$	5.0063	Significant
Global interest	0.0175	0.0075	0.0199	2.3276	Significant
Foreign Exch reserves	−0.0954	0.0158	$3.91e^{-9}$	5.8878	Significant
Trade openness	0.0076	0.0029	0.0094	2.5936	Significant
Export growth	−0.0007	0.0093	0.9398	0.0755	Insignificant
Current account	0.0299	0.0172	0.0907	1.6915	Insignificant
FDI	−0.1599	0.0531	0.0031	2.9536	Significant
Real GDP growth	−0.0037	0.0200	0.8514	0.1872	Insignificant
Inflation	0.0365	0.0073	$7.63e^{-7}$	4.9446	Significant
REER overall	−0.0048	0.0024	0.0472	1.9839	Significant
Gov expenditures	−0.0963	0.0252	0.0001	3.7690	Significant
National savings	−0.0416	0.0132	0.0018	3.1202	Significant
Domestic credit	−0.0149	0.0050	0.0334	2.1265	Significant
Bank assets	−0.0058	0.0068	0.3905	0.8587	Insignificant
Pseudo R^2	0.3995				
Out of sample correctness	0.9360				
In sample correctness	0.8279				
Deviance	695.8678				
Sensitivity	0.9881				
Specificity	0.36				
Precision	0.9398				
Recall	0.9881				
PR curve	0.5374				
ROC AUC	0.6741				
Brier Score	0.0683				

From analysing the global data set considering mean imputation; the significant variables show to be total debt, global interest, trade openness, FDI, inflation, REER overall, government expenditure, national savings, and domestic credit. Based on the considered metric, the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 67.41%, see Table 5.12.

Table 5.13: Binary logit model summary when applied to the South and East Asia mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	4.7961	3.2227	0.1367		Insignificant
Total debt	0.0438	0.0164	0.0074	2.6783	Significant
Global interest	0.0473	0.0515	0.3576	0.9199	Insignificant
Foreign Exch reserves	−0.0891	0.0396	0.0248	2.2443	Significant
Trade openness	0.0073	0.0073	0.3155	1.0037	Insignificant
Export growth	−0.0564	0.0215	0.0089	2.6137	Significant
Current account	0.0329	0.0422	0.4352	0.7802	Insignificant
FDI	−0.6625	0.2234	0.0030	2.9647	Significant
Real GDP growth	0.0065	0.0440	0.8816	0.1489	Insignificant
Inflation	0.0711	0.0288	0.0137	2.4633	Significant
REER overall	−0.0555	0.0250	0.0264	2.2200	Significant
Gov expenditures	−0.3214	0.1084	0.0030	2.9643	Significant
National savings	−0.0539	0.0476	0.2579	1.1312	Insignificant
Domestic credit	0.0458	0.0200	0.0224	2.2821	Significant
Bank assets	−0.0055	0.0138	0.6889	0.4003	Insignificant
Pseudo R^2	0.4529				
Out of sample correctness	0.9662				
In sample correctness	0.9153				
Deviance	120.2475				
Sensitivity	0.95891				
Specificity	1				
Precision	1				
Recall	0.95891				
PR AUC	0.4834				
ROC AUC	0.9794				
Brier Score	0.0337				

From analysing the South and East Asia data set considering mean imputation; the significant variables show to be total debt, foreign exchange reserves, export growth, FDI, inflation, REER overall, government expenditure, and domestic credit. Based on the considered metric, the model is outstanding at predicting sovereign default since it generated a ROC AUC value of 97.94%, see Table 5.13.

Table 5.14: Binary logit model summary when applied to the Latin America mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	3.0918	1.4366	0.0313		Significant
Total debt	−0.0001	0.0101	0.9911	0.0110	Insignificant
Global interest	0.0189	0.0097	0.0516	1.9465	Insignificant
Foreign Exch reserves	−0.0873	0.0238	0.0002	3.6612	Significant
Trade openness	0.0046	0.0068	0.4972	0.6788	Insignificant
Export growth	0.0125	0.0148	0.3981	0.8450	Insignificant
Current account	0.0101	0.0420	0.8097	0.2408	Insignificant
FDI	−0.2283	0.0797	0.0041	2.8632	Significant
Real GDP growth	0.0241	0.0391	0.5385	0.6150	Insignificant
Inflation	0.0300	0.0080	0.0002	3.7112	Significant
REER overall	−0.0223	0.0068	0.0010	3.2781	Significant
Gov expenditures	−0.0483	0.0446	0.2990	1.0385	Insignificant
National savings	−0.0749	0.0333	0.0244	2.2507	Significant
Domestic credit	−0.0488	0.0199	0.0143	2.4491	Significant
Bank assets	0.0554	0.0251	0.0275	2.2037	Insignificant
Pseudo R^2	0.3269				
Out of sample correctness	0.9642				
In sample correctness	0.7724				
Deviance	290.8733				
Sensitivity	0.9647				
Specificity	0.8181				
Precision	0.9761				
Recall	0.9647				
PR AUC	0.4938				
ROC AUC	0.8914				
Brier Score	0.0520				

From analysing the Latin America data set considering mean imputation; the significant variables show to be foreign exchange reserves, FDI, inflation, REER overall, national savings, and domestic credit. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 89.14%, see Table 5.14.

Table 5.15: Binary logit model summary when applied to the Middle East and Africa mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	7.1731	4.6532	0.1231		Insignificant
Total debt	−0.0320	0.0396	0.4195	0.7392	Insignificant
Global interest	0.0953	0.0993	0.3369	1.1030	Insignificant
Foreign Exch reserves	−1.0874	0.2833	0.0001	3.6769	Significant
Trade openness	0.0184	0.0247	0.4569	0.9768	Insignificant
Export growth	0.0561	0.0352	0.1111	1.4541	Insignificant
Current account	0.1692	0.1138	0.1370	1.5894	Insignificant
FDI	−1.1114	0.4391	0.0113	2.4794	Significant
Real GDP growth	0.0119	0.0721	0.8685	0.2321	Insignificant
Inflation	0.1601	0.0574	0.0053	2.7851	Significant
REER overall	−0.0128	0.0058	0.0267	2.2148	Significant
Gov expenditures	−0.0471	0.1686	0.7797	0.2797	Insignificant
National savings	−0.0255	0.0507	0.6148	0.5031	Insignificant
Domestic credit	−0.0762	0.0230	0.0009	3.3114	Significant
Bank assets	0.0365	0.0359	0.3097	1.0157	Insignificant
Pseudo R^2	0.6673				
Out of sample correctness	0.9120				
In sample correctness	0.8524				
Deviance	82.2766				
Sensitivity	0.8235				
Specificity	1				
Precision	1				
Recall	0.8235				
PR AUC	0.4387				
ROC AUC	0.9118				
Brier Score	0.1475				

From analysing the Middle East and Africa data set considering mean imputation; the significant variables show to be foreign exchange reserves, FDI, inflation, REER overall, and domestic credit. Based on the considered metric, the model is outstanding at predicting sovereign default since it generated a ROC AUC value of 91.18%, see Table 5.15.

Table 5.16: Binary logit model summary when applied to the Western Europe mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$-3.611e^{2}$	$5.724e^{5}$	0.999		Insignificant
Total debt	2.603	$1.247e^{3}$	0.998	0.0021	Insignificant
Global interest	5.903	$6.572e^{3}$	0.999	0.0009	Insignificant
Foreign Exch reserves	-1.952	$2.390e^{3}$	0.999	0.0008	Insignificant
Trade openness	$-0.8257e^{-1}$	$2.329e^{3}$	1	0.0004	Insignificant
Export growth	$-0.9582e^{-1}$	$1.048e^{4}$	1	0.0001	Insignificant
Current account	$15.09e^{1}$	$9.977e^{3}$	0.999	0.0015	Insignificant
FDI	-6.829	$2.654e^{4}$	1	0.0003	Insignificant
Real GDP growth	7.065	$2.371e^{4}$	1	0.0003	Insignificant
Inflation	$-8.041e^{-1}$	$7.198e^{3}$	1	0.0001	Insignificant
REER overall	-1.041	$4.848e^{3}$	1	0.0002	Insignificant
Gov expenditures	-3.575	$2.395e^{4}$	1	0.0001	Insignificant
National savings	3.997	$9.109e^{3}$	1	0.0004	Insignificant
Domestic credit	4.461	$2.004e^{3}$	0.988	0.0022	Insignificant
Bank assets	-1.337	$1.538e^{3}$	0.999	0.0009	Insignificant
Pseudo R^2	1				
Out of sample correctness	0.7619				
In sample correctness	1				
Deviance	3.6966^{-8}				
Sensitivity	0.7924				
Specificity	0.5454				
Precision	0.8076				
Recall	0.7924				
PR AUC	0.5120				
ROC AUC	0.6972				
Brier Score	0.24				

From analysing the Western Europe data set considering mean imputation; There is no significant variable that can help in the prediction of sovereign debt. Based on the considered metric, the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 69.72%, see Table 5.16. The high standard error is a result of the Western Europe data set not having enough entries in the sample to construct a proper predictive model (Graham, 2009).

5.3.4 Comparison of Global and Regional Binary Logistic Model Results for Mean Imputation

Table 5.17: Binary logit models of sovereign debt from the year 1980-2005, where the missing values are filled using the average of the regions of which the country resides. Pseudo McFadden's R^2 and percentage of in and out of sample crisis predicted by each model ,global and regional are reported.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
Pseudo R^2	0.3995	0.4529	0.3269	0.6673	1
Out of sample correctness	0.9360	0.9662	0.9642	0.9120	0.7619
In sample correctness	0.8279	0.9153	0.7724	0.8524	1
Deviance	695.8678	120.2475	290.8733	82.2766	3.6966^{-8}
Sensitivity	0.9881	0.95891	0.9647	0.8235	0.7924
Specificity	0.36	1	0.8181	1	0.5454
Precision	0.9398	1	0.9761	1	0.8076
Recall	0.9881	0.95891	0.9647	0.8235	0.7924
PR AUC	0.5374	0.4834	0.4938	0.4387	0.5120
ROC AUC	0.6741	0.9794	0.8914	0.9118	0.6972
Brier Score	0.0683	0.0337	0.0520	0.1475	0.24
sample size (n)	988	260	312	182	234

The results for Table 5.17 are as follows: When considering the out-of-sample correctness, only the South and East Asia model, along with the Latin America model, scored better than the global model, which implies that only two regional models were able to develop models that could predict better results than the global model. For the in-sample correctness, Table 5.17 shows that the regional models, with the exception of the Latin America model, produced higher scores than the global model, suggesting that the global model and the Latin America model struggled to predict more than 85% of the outcomes considered in the data used to develop the model. The global model produced the highest deviance, suggesting that the regional models had a better goodness-of-fit. In terms of sensitivity and recall, the global model generated the best sensitivity score, which suggests that the model had the highest true positive prediction. When considering specificity, the regional models produced the highest score, suggesting that they generated the best true negative prediction rate. When looking at precision, the regional models, with the exception of the Western European model, produced the highest score, which suggests that the models could correctly classify a positive that is not a negative. Considering the PR AUC, the global model produced the highest score, implying that it did well in terms of classifying. The ROC AUC suggests that the regional models produced a higher score than the global model, which implies that they had the best performance measurements for classification at different thresholds. In terms of the Brier score, the regional models, with the exception of the Western European model and the Middle East and Africa models, generated a lower score, which means that it had the lowest mean squared error between the predicted probability and observed value. When look-

ing at the ROC AUC score on Table 5.17, the regional models performed better, making using regional models the best classification when mean imputation is considered for the logit classification method.

5.3.5 MICE Logit Models

Table 5.18: Binary logit model summary when applied to the global MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	0.4943	0.6565	0.3703		Insignificant
Total debt	0.0361	0.0051	< 0.0001	8.2490	Significant
Global interest	0.0264	0.0097	< 0.0001	2.5086	Significant
Foreign Exch reserves	-0.0502	0.0136	< 0.0001	5.6720	Significant
Trade openness	0.0002	0.0035	0.1532	0.2997	Insignificant
Export growth	-0.0081	0.0099	0.8857	0.9657	Insignificant
Current account	0.0370	0.0207	0.0774	3.9806	Insignificant
FDI	-0.1111	0.0617	0.0054	2.3116	Significant
Real GDP growth	-0.0219	0.0242	0.1207	0.4298	Insignificant
Inflation	0.0031	0.0043	0.0002	5.9023	Significant
REER overall	0.0017	0.0022	0.0732	0.6981	Insignificant
Gov expenditures	-0.1765	0.0310	< 0.0001	5.9657	Significant
National savings	-0.0663	0.0133	0.0001	3.9098	Significant
Domestic credit	0.0002	0.0087	< 0.0001	0.6688	Significant
Bank assets	-0.0214	0.0091	< 0.0001	2.0171	Significant
Pseudo R^2	0.3898				
Out of sample correctness	0.9323				
In sample correctness	0.8724				
Deviance	708.3074				
Sensitivity	0.9881				
Specificity	0.36				
Precision	0.9398				
Recall	0.9881				
PR AUC	0.5374				
ROC AUC	0.6741				
Brier Score	0.0683				

From analysing the global data set considering MICE; the significant variables show to be total debt, global interest, foreign exchange reserves, FDI, inflation, government expenditure, national savings, domestic credit, and bank assets. Based on the considered metric, the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 67.41%, see Table 5.18.

Table 5.19: Binary logit model summary when applied to the South and East Asia MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	−10.4528	2.9119	0.0003		Significant
Total debt	0.0362	0.0145	0.0009	2.5315	Significant
Global interest	0.1045	0.0839	0.0003	0.5041	Significant
Foreign Exch reserves	−0.0269	0.0222	0.6446	2.5830	Insignificant
Trade openness	−0.0119	0.0077	0.3724	0.1338	Insignificant
Export growth	−0.1204	0.0264	0.0282	3.5959	Insignificant
Current account	−0.0248	0.0499	0.2612	1.3833	Insignificant
FDI	−0.3166	0.2852	0.0053	2.9894	Significant
Real GDP growth	−0.0322	0.0568	0.9202	0.8244	Insignificant
Inflation	0.0220	0.0281	0.6587	1.4287	Insignificant
REER overall	−0.0192	0.0087	0.0003	1.7656	Significant
Gov expenditures	−0.2370	0.1283	0.0132	3.2407	Significant
National savings	0.0443	0.0564	0.3832	0.2315	Insignificant
Domestic credit	0.0499	0.0255	0.0512	1.6501	Insignificant
Bank assets	−0.0223	0.0219	0.3422	0.5701	Insignificant
Pseudo R^2	0.4212				
Out of sample correctness	0.9318				
In sample correctness	0.9038				
Deviance	125.1885				
Sensitivity	0.9589				
Specificity	0.8				
Precision	0.9589				
Recall	0.9589				
PR AUC	0.5				
ROC AUC	0.8794				
Brier Score	0.0681				

From analysing the South and East Asia data set considering MICE; the significant variables show to be total debt, global interest, FDI, REER overall, and government expenditure. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 87.94%, see Table 5.19.

Table 5.20: Binary logit model summary when applied to the Latin America MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	3.7169	1.4851	0.0123		Significant
Total debt	0.0212	0.0098	0.0007	1.7474	Significant
Global interest	0.0087	0.0030	0.8342	2.7833	Insignificant
Foreign Exch reserves	−0.0754	0.0247	0.0078	3.8396	Significant
Trade openness	−0.0005	0.0067	0.6909	0.0371	Insignificant
Export growth	0.0143	0.0171	0.0551	0.6455	Insignificant
Current account	−0.1273	0.0455	0.0580	1.3224	Insignificant
FDI	−0.2759	0.0877	0.2704	2.8749	Insignificant
Real GDP growth	−0.0291	0.0422	0.5792	1.4846	Insignificant
Inflation	0.0164	0.0076	0.0001	4.1138	Significant
REER overall	−0.0277	0.0069	< 0.0001	4.1797	Significant
Gov expenditures	−0.0968	0.0474	0.1080	0.4145	Insignificant
National savings	−0.0119	0.0364	0.0282	1.1129	Significant
Domestic credit	−0.0180	0.0215	0.2944	0.9468	Insignificant
Bank assets	0.01268	0.0290	0.1154	0.8421	Insignificant
Pseudo R^2	0.3070				
Out of sample correctness	1				
In sample correctness	0.7628				
Deviance	299.7058				
Sensitivity	0.9647				
Specificity	0.8182				
Precision	0.9761				
Recall	0.9647				
PR AUC	0.4938				
ROC AUC	0.8914				
Brier Score	0.0520				

From analysing the Latin America data set considering MICE; the significant variables show to be total debt, foreign exchange reserves, inflation, REER overall, and national savings. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 89.14%, see Table 5.20.

Table 5.21: Binary logit model summary when applied to the Middle East and Africa MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	4.9641	3.9743	0.2116		Insignificant
Total debt	0.0543	0.0271	0.6320	2.7234	Insignificant
Global interest	0.1182	0.0955	0.1864	0.7655	Insignificant
Foreign Exch reserves	−0.5254	0.1926	< 0.0001	2.9477	Significant
Trade openness	0.0205	0.0225	0.2448	1.8692	Insignificant
Export growth	0.0855	0.0377	0.6587	0.9265	Insignificant
Current account	0.0091	0.1158	0.0814	0.8231	Insignificant
FDI	−1.3748	0.3559	0.0306	2.6638	Significant
Real GDP growth	0.0271	0.0770	0.4931	0.6403	Insignificant
Inflation	0.2032	0.0520	0.0005	3.4522	Significant
REER overall	−0.0007	0.0054	0.2058	1.3172	Insignificant
Gov expenditures	−0.1138	0.1499	0.9783	1.5884	Insignificant
National savings	−0.0781	0.0482	0.0662	1.8194	Insignificant
Domestic credit	−0.0695	0.0250	0.0017	1.6939	Significant
Bank assets	0.0153	0.0308	0.8690	0.2528	Insignificant
Pseudo R^2	0.7074				
Out of sample correctness	0.9387				
In sample correctness	0.9230				
Deviance	72.15408				
Sensitivity	0.9411				
Specificity	1				
Precision	1				
Recall	0.9411				
PR AUC	0.4759				
ROC AUC	0.9706				
Brier Score	0.0491				

From analysing the Middle East and Africa data set considering MICE; the significant variables show to be foreign exchange reserves, FDI, inflation, and domestic credit. Based on the considered metric, the model is outstanding at predicting sovereign default since it generated a ROC AUC value of 97.06%, see Table 5.21.

Table 5.22: Binary logit model summary when applied to the Western Europe MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$1.609e^{2}$	$3.939e^{5}$	1		Insignificant
Total debt	0.5958	$9.755e^{3}$	1	0.0015	Insignificant
Global interest	5.412	$6.147e^{3}$	0.999	0.0005	Insignificant
Foreign Exch reserves	−0.3391	$1.285e^{3}$	1	0.0003	Insignificant
Trade openness	0.4729	$1.092e^{3}$	0.999	0.0000	Insignificant
Export growth	−0.2898	$4.820e^{3}$	1	0.0001	Insignificant
Current account	6.072	$1.461e^{4}$	0.999	0.0003	Insignificant
FDI	−0.5941	$9.167e^{3}$	0.999	0.0001	Insignificant
Real GDP growth	0.5270	2.078	1	0.0003	Insignificant
Inflation	−1.0830	$5.908e^{3}$	1	0.0002	Insignificant
REER overall	−0.6611	$3.724e^{3}$	1	0.0000	Insignificant
Gov expenditures	1.1180	$1.640e^{4}$	1	0.0002	Insignificant
National savings	−4.1670	$1.350e^{4}$	1	0.0003	Insignificant
Domestic credit	1.768	$1.669e^{3}$	0.999	0.0012	Insignificant
Bank assets	−0.5498	$8.654e^{2}$	0.999	0.0005	Insignificant
Pseudo R^2	1				
Out of sample correctness	0.5079				
In sample correctness	1				
Deviance	$7.4587e^{-9}$				
Sensitivity	0.5283				
Specificity	0.6364				
Precision	0.7778				
Recall	0.5283				
PR AUC	0.4242				
ROC AUC	0.5634				
Brier Score	0.44				

From analysing the Western Europe data set considering MICE; there is no significant variable that can help in the prediction of sovereign debt. Based on the considered metric, the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 56.34%, see Table 5.22.

5.3.6 Comparison of Global and Regional Binary Logistic Model Results for MICE

Table 5.23: Binary logit models of sovereign debt from the year 1980-2005, where the missing values are filled using the MICE technique. The Pseudo McFadden's $R^2$and percentage of out of sample crisis predicted by each model, global and regional are reported.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
Pseudo R^2	0.3898	0.4212	0.3070	0.7074	1
out of sample correctness	0.9323	1	1	0.9387	0.5079
In sample correctness	0.8724	0.9038	0.7628	0.9230	1
Deviance	708.3074	125.1885	299.7058	72.15408	$7.4587e^{-9}$
Sensitivity	0.9881	0.9589	0.9647	0.9411	0.5283
Specificity	0.36	0.8	0.8182	1	0.6364
Precision	0.9398	0.9589	0.9761	1	0.7778
Recall	0.9881	0.9589	0.9647	0.9411	0.5283
PR AUC	0.5374	0.5	0.4938	0.4759	0.4242
ROC AUC	0.6741	0.8794	0.8914	0.9706	0.5634
Brier Score	0.0683	0.0681	0.0520	0.0491	0.44
Sample size(n)	988	260	312	182	234

The results for Table 5.23 are as follows: When considering the out-of-sample correctness, the regional models, with the exception of the Western European model, generated higher scores than that of the global model, which implies that the regional models were able to develop a model that could predict better outcomes. For the in-sample correctness, Table 5.23 shows that the only regional model that could not generate an in-sample score better than the global model is the Latin America model, suggesting that it struggled to predict the outcome of the data used to develop the model. The global model produced the highest deviance, suggesting that the regional models had a better goodness-of-fit. In terms of sensitivity and recall, the global model generated the best sensitivity score, which suggests that the model had the highest true positive prediction rate. However, when considering specificity, the regional models produced the highest score, suggesting that they generated the best true negative prediction rate. When looking at precision, the regional models, with the exception of the Western European model, produced the highest score, which suggests that the models could correctly classify a positive that is not a negative. Considering the PR AUC, the global model produced the highest score, implying that it did well in terms of classifying. The ROC AUC suggests that the regional models, with the exception of the Western European model, produced a higher score than the global model, which implies that they had the best performance measurements for classification at different thresholds. In terms of the Brier score, the regional models, with the exception of the Western European model, generated a lower score, which means that it had the lowest mean squared error between the predicted probability and observed value. When looking at the ROC AUC score on Table 5.23, the

regional models, with the exception of the Western Europe model, performed better, making using regional models the best classification when MICE is considered for the logit classification method.

5.4 Probit Classifier Models

Similar to the logit classifier, all default events are treated as countries that went through a debt crisis are examined as individual crisis episodes. The binary dependent variable y_i is assigned one (1) for all crisis periods and zero (0) only during non-default times.

Five models are developed for each data set. The beta coefficients of the logged odds ratio, McFadden's Pseudo R^2, in and out of sample correctness, the deviance of the model, and metrics used in the evaluation of model performance for unbalanced data are applied. The dependent variable in these samples is the debt crisis indicator variables. The time frame that is used to build the models is from 1980 to 2005, and the period that is used for the out of sample testing is from 2006 to 2012. The performance of the models listed below are bucketed into performance groups, this is applied to better elaborate the outcome of the predictions. The following are the different buckets for the ROC AUC: 0% to 50%, suggest no discrimination and therefore poor, 51% to 69% represents slighlty accaptable which is considered moderately good, 70% to 80% is acceptable and therefore noted as good, 81% to 90% is excellent and anything greater than 90% is outstanding (Mandrekar, 2010). The main metric used to assess model prediction accuracy will be the ROC AUC, therefore conclusions will be based on the ROC AUC results. In the delevelopment classifiers, multicolinearity has been ignored. So interpretation of the significance of the independent variable in each model should be treated with caution.

5.4.1 Case Wise Deletion Probit Models

Table 5.24: Probit model summary when applied to the global case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	0.9893	1.4109	0.4831		Insignificant
Global interest	0.0212	0.0223	0.3424	4.1892	Insignificant
Foreign Exch reserves	−0.0494	0.0221	0.0255	3.3223	Significant
Trade openness	−0.0012	0.0048	0.8049	1.7043	Insignificant
Export growth	0.0003	0.0203	0.9844	0.4134	Insignificant
Current account	0.0234	0.0439	0.5928	0.7972	Insignificant
FDI	−0.2701	0.0963	0.0050	3.7308	Significant
Real GDP growth	0.1364	0.0629	0.0301	1.0789	Significant
Inflation	0.0453	0.0176	0.0102	4.5453	Significant
REER overall	−0.0038	0.0039	0.3271	0.4439	Insignificant
Gov expenditures	−0.1306	0.0575	0.0232	3.4807	Significant
National savings	−0.0207	0.0288	0.4732	0.9959	Insignificant
Domestic credit	0.0034	0.0109	0.7543	0.0239	Insignificant
Bank assets	0.0129	0.0139	0.3512	1.0244	Insignificant
Pseudo R^2	0.8422				
out of sample correctness	0.9097				
In sample correctness	0.8734				
Deviance	75.2976				
Sensitivity	0.9739				
Specificity	0.5				
Precision	0.9256				
Recall	0.9739				
PR AUC	0.5295				
ROC AUC	0.7369				
Brier Score	0.0902				

From analysing the global data set considering case-wise deletion; the significant variables show to be foreign exchange reserves, FDI, real GDP growth, inflation and government expenditure. Based on the considered metric, the model is good at predicting sovereign default since it generated a ROC AUC value of 73.69%, see Table 5.24.

Table 5.25: Probit model summary when applied to the South and East Asia case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	8.125	$1.608e^{5}$	1		Insignificant
Global interest	-2.836	$9.922e^{3}$	1	0.3939	Insignificant
Foreign Exch reserves	$-3.808e^{-2}$	$2.444e^{3}$	1	1.8691	Insignificant
Trade openness	$8.834e^{-2}$	$2.296e^{2}$	1	0.3598	Insignificant
Export growth	$7.159e^{-2}$	$1.386e^{3}$	1	1.4598	Insignificant
Current account	-1.640	$1.162e^{4}$	1	0.5043	Insignificant
FDI	$-7.595e^{-1}$	$8.267e^{3}$	1	2.6902	Insignificant
Real GDP growth	-3.016	$5.812e^{3}$	1	0.2095	Insignificant
Inflation	1.449	$5.011e^{3}$	1	0.6322	Insignificant
REER overall	$1.172e^{-1}$	$1.720e^{3}$	1	2.0610	Insignificant
Gov expenditures	$-8.384e^{-1}$	$7.763e^{3}$	1	1.4179	Insignificant
National savings	$6.286e^{-1}$	$4.110e^{3}$	1	2.4672	Insignificant
Domestic credit	$-5.025e^{-1}$	$2.598e^{3}$	1	0.8274	Insignificant
Bank assets	$3.489e^{-1}$	$2.738e^{3}$	1	1.9834	Insignificant
Pseudo R^2	1				
out of sample correctness	0.8571				
In sample correctness	0.75				
Deviance	$6.5366e^{-10}$				
Sensitivity	0.875				
Specificity	0.875				
Precision	0.9032				
Recall	0.875				
PR AUC	0.4909				
ROC AUC	0.8304				
Brier Score	0.1521				

From analysing the South and East Asia data set considering case-wise deletion; there is no significant variable that can help in the prediction of sovereign debt. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 83.04%, see Table 5.25.

Table 5.26: Probit model summary when applied to the Latin America case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	−1.1986	5.7811	0.8357		Insignificant
Global interest	0.0444	0.0381	0.2433	3.4673	Insignificant
Foreign Exch reserves	−0.0621	0.0561	0.2685	2.2962	Insignificant
Trade openness	0.0150	0.0215	0.4856	0.4112	Insignificant
Export growth	0.1260	0.0823	0.1260	2.0067	Insignificant
Current account	−0.2808	0.1502	0.0616	2.3592	Insignificant
FDI	0.0445	0.2018	0.8255	1.5512	Insignificant
Real GDP growth	0.3189	0.2119	0.1325	1.7222	Insignificant
Inflation	0.0648	0.0455	0.1540	3.9039	Insignificant
REER overall	−0.0233	0.0271	0.3902	0.0894	Insignificant
Gov expenditures	−0.2028	0.1422	0.1541	0.7902	Insignificant
National savings	−0.0336	0.1039	0.7466	1.2579	Insignificant
Domestic credit	−0.1401	0.0960	0.1447	1.0232	Insignificant
Bank assets	0.1375	0.0925	0.1373	1.2079	Insignificant
Pseudo R^2	0.85802				
out of sample correctness	0.9056				
In sample correctness	0.7555				
Deviance	30.0507				
Sensitivity	0.9259				
Specificity	0.8182				
Precision	0.9615				
Recall	0.9259				
PR AUC	0.4845				
ROC AUC	0.8721				
Brier Score	0.0923				

From analysing the Latin America data set considering case-wise deletion; there is no significant variable that can help in the prediction of sovereign debt. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 87.21%, see Table 5.26.

Table 5.27: Probit model summary when applied to the Middle East and Africa case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$-1.498e^{2}$	$3.575e^{5}$	1		Insignificant
Global interest	2.403	$3.197e^{3}$	0.9999	0.9601	Insignificant
Foreign Exch reserves	3.655	$2.101e^{4}$	1	1.0875	Insignificant
Trade openness	1.514	$1.774e^{3}$	0.9999	1.0228	Insignificant
Export growth	$-4.216e^{-1}$	$1.049e^{3}$	1	1.0457	Insignificant
Current account	1.277	$3.168e^{4}$	1	1.1220	Insignificant
FDI	-4.823	$3.933e^{4}$	1	0.9699	Insignificant
Real GDP growth	-2.574	$2.003e^{4}$	1	0.9990	Insignificant
Inflation	1.226	$8.577e^{3}$	1	0.9509	Insignificant
REER overall	$1.903e^{-3}$	$3.609e^{2}$	1	1.1354	Insignificant
Gov expenditures	2.532	$5.398e^{4}$	1	1.1028	Insignificant
National savings	$8.612e^{-1}$	$2.264e^{3}$	1	1.2238	Insignificant
Domestic credit	$2.893e^{-1}$	$2.619e^{3}$	1	1.0862	Insignificant
Bank assets	-1.473	$1.374e^{4}$	1	1.0500	Insignificant
Pseudo R^2	1				
out of sample correctness	0.8889				
In sample correctness	0.9655				
Deviance	$1.1393e^{-9}$				
Sensitivity	0.875				
Specificity	0.9047				
Precision	0.9130				
Recall	0.875				
PR AUC	0.4922				
ROC AUC	0.8899				
Brier Score	0.1111				

From analysing the Middle East and Africa data set considering case-wise deletion; there is no significant variable that can help in the prediction of sovereign debt. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 88.99%, see Table 5.27. The high standard error is a result of the deletion of entries which leads to the sample not closely representing the population (Afghari et al., 2019).

For the Western Europe case-wise deletion data set, there was no default observation in the test and training data set. As a result no model can be built.

5.4.2 Comparison of Global and Regional Binary Probit Model Results for Case-Wise Deletion

Table 5.28: Binary probit models of sovereign debt from the year 1980-2005, where case-wise deletion technique was used to compile the data. Pseudo McFadden's R^2 and percentage of out-of-sample crisis predicted by each model, global and regional are reported.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
Pseudo R^2	0.8422	1	0.85802	1	$3.350e^{-1}$
Out of sample correctness	0.8925	0.8571	0.9056	0.8095	0.6315
In sample correctness	0.8699	0.75	0.7555	0.7701	1
Deviance	75.2976	$6.5366e^{-10}$	30.0507	$1.1393e^{-9}$	$2.3194e^{-10}$
Sensitivity	0.9739	0.875	0.9259	0.875	1
Specificity	0.5	0.875	0.8182	0.9047	0
Precision	0.9256	0.9032	0.9615	0.9130	Nan
Recall	0.9739	0.875	0.9259	0.875	Nan
PR AUC	0.5295	0.4909	0.4845	0.4922	Nan
ROC AUC	0.7369	0.8304	0.8721	0.8899	0.5
Brier Score	0.0902	0.1521	0.0923	0.1111	0.3684
sample size (n)	569	160	180	87	143

The results for Table 5.28 are as follows: The Western Europe model will not be considered since not enough observations where available to develop an adequate model. When considering the out-of-sample correctness, only one regional model being the Latin America model scored better than the global model, which implies that the global model developed a model that could predict better results. For the in-sample correctness, Table 5.28 shows that the global model, produced a higher score than the regional models, suggesting that the regional models struggled to predict more than 85% of the outcomes considered in the data used to develop the model. The global model produced the highest deviance, suggesting that the regional models had a better goodness-of-fit. In terms of sensitivity and recall, the global model generated the best sensitivity score, which suggests that the model had the highest true positive prediction. When considering specificity, the regional models produced the highest score, suggesting that they generated the best true negative prediction rate. When looking at precision, the global model, with the exception of the Latin America model, produced the highest score, which suggests that the global model could correctly classify a positive which is really a positive. Considering the PR AUC, the global model produced the highest score, implying that it did well in terms of classifying. The ROC AUC suggests that the regional models produced a higher score than the global model, which implies that they had the best performance measurements for classification at different thresholds. In terms of the Brier score, the global model generated the lowest score, which means that it had the lowest mean squared error between the predicted probability and observed value. When looking at the ROC AUC score on Table 5.28, the global model did not performed better than

regional models, making using regional models the best classification method when case-wise deletion is considered for the probit classification method.

5.4.3 Mean Imputation Probit Models

Table 5.29: Probit model summary when applied to the global mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$3.763e^{-1}$	$3.942e^{-1}$	0.3397		Insignificant
Total debt	$1.301e^{-2}$	$2.652e^{-3}$	< 0.0001	4.9036	Significant
Global interest	$1.202e^{-2}$	$4.271e^{-3}$	0.0049	2.8137	Significant
Foreign Exch reserves	$-4.464e^{-2}$	$7.772e^{-3}$	< 0.0001	5.7433	Significant
Trade openness	$3.981e^{-3}$	$1.565e^{-3}$	0.0109	2.5441	Significant
Export growth	-$6.251e^{-5}$	$5.338e^{-3}$	0.9906	0.0117	Insignificant
Current account	$1.650e^{-2}$	$9.833e^{-3}$	0.0933	1.6778	Insignificant
FDI	$-8.683e^{-2}$	$2.885e^{-2}$	0.0026	3.0095	Significant
Real GDP growth	$-8.834e^{-3}$	$1.181e^{-2}$	0.4542	0.7483	Insignificant
Inflation	$1.832e^{-2}$	$3.843e^{-3}$	< 0.0001	4.7682	Significant
REER overall	$-2.869e^{-3}$	$1.344e^{-3}$	0.0327	2.1355	Significant
Gov expenditures	$-5.904e^{-2}$	$1.405e^{-2}$	< 0.0001	4.2018	Significant
National savings	$-2.204e^{-2}$	$7.444e^{-3}$	0.0030	2.9605	Significant
Domestic credit	$-7.058e^{-3}$	$3.825e^{-3}$	0.0649	1.8455	Insignificant
Bank assets	$-1.661e^{-3}$	$3.640e^{-3}$	0.6482	0.4562	Insignificant
Pseudo R^2	0.3868				
out of sample correctness	0.9398				
In sample correctness	0.8329				
Deviance	710.6309				
Sensitivity	0.9881				
Specificity	0.36				
Precision	0.9398				
Recall	0.9881				
PR AUC	0.5374				
ROC AUC	0.6741				
Brier Score	0.0683				

From analysing the global data set considering mean imputation; the significant variables show to be total debt, global interest, foreign exchange reserves, trade openness, FDI, inflation, REER overall, government expenditure, and national savings. Based on the considered metric, the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 67.41%, see Table 5.29.

Table 5.30: Probit model summary when applied to the South and East Asia mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	2.3540	1.7480	0.1780		Insignificant
Total debt	0.0250	0.0086	0.0037	2.9924	Significant
Global interest	0.0326	0.0298	0.2739	1.0058	Insignificant
Foreign Exch reserves	-0.0433	0.02011	0.0313	2.1584	Significant
Trade openness	0.0035	0.0037	0.3432	0.9817	Insignificant
Export growth	-0.0325	0.0120	0.0069	2.7310	Significant
Current account	0.0263	0.0226	0.2451	1.0544	Insignificant
FDI	-0.3381	0.1165	0.0037	3.2704	Significant
Real GDP growth	0.0066	0.0234	0.7780	0.1321	Insignificant
Inflation	0.0437	0.0066	0.0066	2.8170	Significant
REER overall	-0.0308	0.0132	0.0198	2.3212	Significant
Gov expenditures	-0.1697	0.0543	0.0017	3.1224	Significant
National savings	-0.0318	0.0255	0.2124	1.2468	Insignificant
Domestic credit	0.0271	0.0107	0.0121	2.5090	Significant
Bank assets	-0.0045	0.0074	0.5460	0.6037	Insignificant
Pseudo R^2	0.4525				
out of sample correctness	0.95505				
In sample correctness	0.9115				
Deviance	120.3343				
Sensitivity	0.9589				
Specificity	0.9375				
Precision	0.9859				
Recall	0.9589				
PR AUC	0.4887				
ROC AUC	0.9482				
Brier Score	0.0449				

From analysing the South and East Asia data set considering mean imputation; the significant variables show to be total debt, foreign exchange reserves, export growth, FDI, inflation, REER overall, government expenditure, and domestic credit. Based on the considered metric, the model is oustanding at predicting sovereign default since it generated a ROC AUC value of 94.82%, see Table 5.30.

Table 5.31: Probit model summary when applied to the Latin America mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	1.6698	0.8381	0.0463		Significant
Total debt	0.0013	0.006	0.8262	0.2195	Insignificant
Global interest	0.0115	0.0056	0.0403	2.0504	Significant
Foreign Exch reserves	-0.0505	0.0134	0.0001	3.7614	Significant
Trade openness	0.0042	0.0039	0.2814	1.0771	Insignificant
Export growth	0.0062	0.0086	0.4669	0.7275	Insignificant
Current account	0.0038	0.0244	0.8759	0.1561	Insignificant
FDI	-0.1192	0.0454	0.0086	2.6261	Significant
Real GDP growth	0.0089	0.0226	0.6935	0.3940	Insignificant
Inflation	0.0153	0.0042	0.0003	3.6000	Significant
REER overall	-0.0124	0.0038	0.0013	3.2099	Significant
Gov expenditures	-0.0268	0.0263	0.3072	1.0210	Insignificant
National savings	-0.0421	0.0194	0.0304	2.1648	Insignificant
Domestic credit	-0.0294	0.0113	0.0092	2.6031	Significant
Bank assets	0.0298	0.0141	0.0354	2.1036	Significant
Pseudo R^2	0.3235				
out of sample correctness	0.9479				
In sample correctness	0.7051				
Deviance	292.3744				
Sensitivity	0.9647				
Specificity	0.8182				
Precision	0.9761				
Recall	0.9647				
PR AUC	0.4938				
ROC AUC	0.8914				
Brier Score	0.0521				

From analysing the Latin America data set considering mean imputation; the significant variables show to be global interest, foreign exchange reserves, FDI, inflation, REER overall, domestic credit, and bank assets. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 89.14%, see Table 5.31.

Table 5.32: Probit model summary when applied to the Middle East and Africa mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	3.8418	2.4621	0.1186		Insignificant
Total debt	-0.0233	0.0207	0.2608	1.1244	Insignificant
Global interest	0.0670	0.0542	0.2164	1.2360	Insignificant
Foreign Exch reserves	-0.6054	0.1424	<0.0001	4.2515	Significant
Trade openness	0.0076	0.0133	0.5651	0.5752	Insignificant
Export growth	0.0280	0.0194	0.1494	1.4415	Insignificant
Current account	0.0887	0.0593	0.1345	1.4963	Insignificant
FDI	-0.6747	0.2353	0.0041	2.8677	Significant
Real GDP growth	0.0222	0.0394	0.5727	0.5640	Insignificant
Inflation	0.0891	0.0291	0.0022	3.0553	Significant
REER overall	-0.0073	0.0030	0.0154	2.4224	Significant
Gov expenditures	-0.0045	0.0904	0.9597	0.0505	Insignificant
National savings	-0.0055	0.0266	0.8348	0.2085	Insignificant
Domestic credit	-0.0446	0.0121	0.0002	3.6763	Significant
Bank assets	0.0224	0.0193	0.2479	1.1554	Insignificant
Pseudo R^2	0.6648				
o1ut of sample correctness	0.8775				
In sample correctness	0.9065				
Deviance	82.9021				
Sensitivity	0.8431				
Specificity	1				
Precision	1				
Recall	0.8431				
PR AUC	0.4441				
ROC AUC	0.9216				
Brier Score	0.1311				

From analysing the Middle East and Africa data set considering mean imputation; the significant variables show to be foreign exchange reserves, FDI, inflation, REER overall, and domestic credit. Based on the considered metric, the model is model outstanding at predicting sovereign default since it generated a ROC AUC value of 92.16%, see Table 5.32.

Table 5.33: Probit model summary when applied to the Western European mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$-1.079e^{2}$	$1.159e^{5}$	0.999		Insignificant
Total debt	$7.694e^{-1}$	$2.540e^{2}$	0.998	0.0030	Insignificant
Global interest	1.785	$1.118e^{3}$	0.999	0.0016	Insignificant
Foreign Exch reserves	$-5.609e^{-1}$	$4.163e^{2}$	0.999	0.0013	Insignificant
Trade openness	$-2.390e^{-1}$	$3.862e^{2}$	1	0.0006	Insignificant
Export growth	$-3.072e^{-1}$	$2.076e^{3}$	1	0.0001	Insignificant
Current account	$4.441e^{-1}$	$1.832e^{3}$	0.998	0.0024	Insignificant
FDI	-1.961	$4.364e^{3}$	1	0.0004	Insignificant
Real GDP growth	2.209	$4.414e^{3}$	1	0.0005	Insignificant
Inflation	$-2.103e^{-1}$	$1.393e^{3}$	1	0.0004	Insignificant
REER overall	$-3.173e^{-1}$	$8.662e^{2}$	1	0.0002	Insignificant
Gov expenditures	-1.010	$4.258e^{3}$	1	0.0007	Insignificant
National savings	1.169	$1.746e^{3}$	1	0.0002	Insignificant
Domestic credit	1.323	$3.606e^{2}$	0.997	0.0037	Insignificant
Bank assets	$-3.984e^{-1}$	$2.760e^{2}$	0.999	0.0014	Insignificant
Pseudo R^2	1				
out of sample correctness	0.7619				
In sample correctness	1				
Deviance	$4.2895e^{-8}$				
Sensitivity	0.7924				
Specificity	0.5455				
Precision	0.8076				
Recall	0.7924				
PR AUC	0.4943				
ROC AUC	0.6972				
Brier Score	0.28				

From analysing the Western Europe data set considering mean imputation; There is no significant variable that can help in the prediction of sovereign debt. Based on the considered metric, the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 69.72%, see Table 5.33.

5.4.4 Comparison of Global and Regional Binary Probit Model Results for Mean Imputation

Table 5.34: Binary probit models of sovereign debt from the year 1980-2005, where mean imputation technique was used to compile the data. Pseudo McFadden's R^2 and percentage of out-of-sample crises predicted by each model, global and regional are reported.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
Pseudo R^2	0.3995	0.4529	0.3269	0.6673	1
Out of sample correctness	0.9360	0.9662	0.9642	0.9120	0.7619
In sample correctness	0.8279	0.9153	0.7724	0.8524	0.9898
Deviance	695.8678	120.2475	290.8733	82.2766	3.6966^{-8}
Sensitivity	0.9881	0.95891	0.9647	0.8235	0.7924
Specificity	0.36	1	0.8181	1	0.5454
Precision	0.9398	1	0.9761	1	0.8076
Recall	0.9881	0.95891	0.9647	0.8235	0.7924
PR AUC	0.5374	0.4834	0.4938	0.4387	0.5120
ROC AUC	0.6741	0.9794	0.8914	0.9118	0.6972
Brier Score	0.0683	0.0337	0.0520	0.1475	0.24
Sample size (n)	988	260	312	182	234

The results for Table 5.34 are as follows: When considering the out-of-sample correctness, only the South and East Asia, along with the Latin America model, scored better than the global model, which implies that only two regional models were able to develop models that could predict better results. For the in-sample correctness, Table 5.34 shows that the regional models, with the exception of the Latin America model, produced higher scores than the global model, suggesting that the global model and the Latin America model struggled to predict more than 85% of the outcomes considered in the data used to develop the model. The global model produced the highest deviance, suggesting that the regional models had a better goodness-of-fit. In terms of sensitivity and recall, the global model generated the best sensitivity score, which suggests that the model had the highest true positive prediction. When considering specificity, the regional models produced the highest score, suggesting that they generated the best true negative prediction rate. When looking at precision, the regional models, with the exception of the Western European model, produced the highest score, which suggests that the models could correctly classify a positive which is really a positive. Considering the PR AUC, the global model produced the highest score, implying that it did well in terms of classifying. The ROC AUC suggests that the regional models produced a higher score than the global model, which implies that they had the best performance measurements for classification at different thresholds. In terms of the Brier score, the regional models, with the exception of the Western European model and the Middle East and Africa models, generated a lower score, which means that it had the lowest mean squared error between the predicted probability and observed value. When looking at the ROC AUC score on

Table 5.34, the regional models performed better, making using regional models the best classification when mean imputation is considered for the probit classification method.

5.4.5 MICE Probit Models

Table 5.35: Probit model summary when applied to the global MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	0.4434	0.3528	0.2088		Insignificant
Total debt	0.0197	0.0023	< 0.0001	8.4603	Significant
Global interest	0.0155	0.0046	0.0006	2.8345	Significant
Foreign Exch reserves	-0.0261	0.0050	0.0004	5.6856	Significant
Trade openness	0.0001	0.0017	0.3844	0.4151	Insignificant
Export growth	-0.0051	0.0054	0.5365	0.7564	Insignificant
Current account	0.0178	0.0112	0.0975	4.1608	Insignificant
FDI	-0.0647	0.0317	0.0034	2.4668	Significant
Real GDP growth	-0.0077	0.0132	0.6043	0.0797	Insignificant
Inflation	0.0015	0.0026	0.0014	6.0971	Significant
REER overall	0.0010	0.0011	0.5379	0.5399	Insignificant
Gov expenditures	-0.1065	0.0146	< 0.0001	6.5397	Significant
National savings	-0.03814	0.0076	< 0.0001	3.8969	Significant
Domestic credit	0.0024	0.0045	< 0.0001	0.9186	Significant
Bank assets	-0.0114	0.0047	< 0.0001	1.8569	Significant
Pseudo R^2	0.3796				
out of sample correctness	0.9323				
In sample correctness	0.8694				
Deviance	720.185				
Sensitivity	0.9881				
Specificity	0.36				
Precision	0.9398				
Recall	0.9881				
PR AUC	0.5374				
ROC AUC	0.6741				
Brier Score	0.0683				

From analysing the global data set considering MICE; the significant variables show to be total debt, global interest, foreign exch reserves, FDI, inflation, government expenditure, national savings, domestic credit, and bank assets. Based on the considered metric, the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 67.41%, see Table 5.35.

Table 5.36: Probit model summary when applied to the South and East Asia MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	−0.1940	1.5225	0.8985		Insignificant
Total debt	0.0211	0.0074	0.0042	2.5608	Significant
Global interest	0.0690	0.0391	0.0781	0.4728	Insignificant
Foreign Exch reserves	−0.0149	0.0136	0.2738	2.6257	Insignificant
Trade openness	0.0036	0.0036	0.3222	0.3025	Insignificant
Export growth	−0.0475	0.0143	0.0009	3.7037	Significant
Current account	0.0201	0.0266	0.4496	1.7502	Insignificant
FDI	−0.1371	0.077	0.0750	3.1023	Insignificant
Real GDP growth	−0.0470	0.0305	0.1240	0.9475	Insignificant
Inflation	0.0188	0.0152	0.2149	1.7168	Insignificant
REER overall	−0.0038	0.0077	0.6203	1.9568	Insignificant
Gov expenditures	−0.1798	0.0625	0.0040	3.3439	Significant
National savings	−0.0155	0.0251	0.5367	0.1370	Insignificant
Domestic credit	0.0313	0.0152	0.0400	1.6767	Significant
Bank assets	−0.0250	0.0129	0.0519	0.4718	Insignificant
Pseudo R^2	0.4220				
out of sample correctness	1				
In sample correctness	0.9038				
Deviance	125.0149				
Sensitivity	0.9589				
Specificity	0.8				
Precision	0.9589				
Recall	0.9589				
PR AUC	0.5				
ROC AUC	0.8794				
Brier Score	0.0681				

From analysing the South and East Asia data set considering MICE; the significant variables show to be total debt, export growth, government expenditures, and domestic credit. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 87.94%, see Table 5.36.

Table 5.37: Probit model summary when applied to the Latin America MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	2.6641	0.8560	0.0018		Significant
Total debt	0.0126	0.0042	0.1168	1.7474	Insignificant
Global interest	0.0048	0.0047	0.7951	2.7833	Insignificant
Foreign Exch reserves	−0.0444	0.0101	0.0093	3.8396	Significant
Trade openness	0.0001	0.0034	0.7335	0.0371	Insignificant
Export growth	0.0087	0.0092	0.5489	0.6455	Significant
Current account	−0.0789	0.0245	0.3520	1.3224	Insignificant
FDI	−0.1561	0.0464	0.0044	2.8749	Significant
Real GDP growth	−0.0178	0.0217	0.8788	1.4846	Insignificant
Inflation	0.0095	0.0044	< 0.0001	4.1138	Significant
REER overall	−0.0166	0.0037	0.0001	4.1797	Significant
Gov expenditures	−0.0561	0.0272	0.0305	0.4145	Significant
National savings	−0.0061	0.0208	0.0025	1.1129	Significant
Domestic credit	−0.0108	0.0093	0.5286	0.9468	Insignificant
Bank assets	0.0065	0.0126	0.5600	0.8421	Insignificant
Pseudo R^2	0.3080				
out of sample correctness	0.9881				
In sample correctness	0.6346				
Deviance	299.2989				
Sensitivity	0.9647				
Specificity	0.8182				
Precision	0.9761				
Recall	0.9647				
PR AUC	0.4938				
ROC AUC	0.8914				
Brier Score	0.0520				

From analysing the Latin America data set considering MICE; the significant variables show to be foreign exchange reserves, export growth, FDI, inflation, REER overall, government expenditures, and national savings. Based on the considered metric, the model is excellent at predicting sovereign default since it generated a ROC AUC value of 89.14%, see Table 5.37.

Table 5.38: Probit model summary when applied to the Middle East and Africa MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	−5.0346	2.9175	0.0844		Insignificant
Total debt	0.0716	0.0247	0.0038	3.2794	Significant
Global interest	0.0088	0.0631	0.8880	0.4080	Insignificant
Foreign Exch reserves	−0.3583	0.1207	0.0029	2.9785	Significant
Trade openness	0.02701	0.0166	0.1049	1.6883	Insignificant
Export growth	0.0886	0.0272	0.0011	0.8906	Significant
Current account	0.1266	0.0842	0.1329	0.6660	Insignificant
FDI	−0.8634	0.3587	0.0160	2.7635	Significant
Real GDP growth	0.0452	0.0533	0.3968	1.0182	Insignificant
Inflation	0.1780	0.0485	0.0002	3.7846	Significant
REER overall	0.0059	0.0051	0.2471	1.4844	Insignificant
Gov expenditures	0.0551	0.1087	0.6117	1.5407	Insignificant
National savings	−0.0065	0.0311	0.8341	1.5566	Insignificant
Domestic credit	−0.0163	0.0172	0.3451	1.3313	Insignificant
Bank assets	−0.0569	0.0246	0.0210	0.1248	Significant
Pseudo R^2	0.7122				
o1ut of sample correctness	0.9387				
In sample correctness	0.9230				
Deviance	70.9858				
Sensitivity	0.9411				
Specificity	0.9				
Precision	0.9795				
Recall	0.9411				
PR AUC	0.4834				
ROC AUC	0.9206				
Brier Score	0.0655				

From analysing the Middle East and Africa data set considering MICE; the significant variables show to be total debt, foreign exchange reserves, export growth, FDI, inflation, and bank assets. Based on the considered metric, the model is outstanding at predicting sovereign default since it generated a ROC AUC value of 92.06%, see Table 5.38.

Table 5.39: Probit model summary when applied to the Western European MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$-9.308e^{1}$	$6.647e^{4}$	0.999		Insignificant
Total debt	$-6.503e^{-2}$	$1.998e^{2}$	1	0.0024	Insignificant
Global interest	1.683	$8.303e^{2}$	0.998	0.0009	Insignificant
Foreign Exch reserves	-1.001	$3.055e^{2}$	0.997	0.0004	Insignificant
Trade openness	$-4.199e^{-1}$	$2.314e^{2}$	0.999	0.0001	Insignificant
Export growth	-1.564	$3.203e^{2}$	0.996	0.0003	Insignificant
Current account	9.60	$2.774e^{3}$	0.997	0.0011	Insignificant
FDI	-3.840	$9.355e^{2}$	0.997	0.0004	Insignificant
Real GDP growth	-5.252	$1.438e^{3}$	0.997	0.0005	Insignificant
Inflation	-6.153	$2.352e^{3}$	0.998	0.0000	Insignificant
REER overall	1.349	$7.259e^{2}$	0.999	0.0008	Insignificant
Gov expenditures	$-1.019e^{1}$	$1.623e^{3}$	0.995	0.0004	Insignificant
National savings	7.462	$2.163e^{3}$	0.997	0.0004	Insignificant
Domestic credit	$-2.753e^{-1}$	$2.264e^{2}$	0.999	0.0031	Insignificant
Bank assets	$6.872e^{-1}$	$2.548e^{2}$	0.998	0.0014	Insignificant
Pseudo R^2	1				
out of sample correctness	0.5079				
In sample correctness	1				
Deviance	$6.5475e^{-9}$				
Sensitivity	0.4905				
Specificity	0.6363				
Precision	0.764				
Recall	0.4905				
PR AUC	0.4175				
ROC AUC	0.5823				
Brier Score	0.466				

From analysing the Western Europe data set considering MICE; There is no significant variable that can help in the prediction of sovereign debt. Based on the considered metric, the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 58.23%, see Table 5.39.

5.4.6 Comparison of Global and Regional Binary Probit Model Results for MICE

Table 5.40: Binary probit models of sovereign debt from the year 1980-2005, where the missing values are filled using MICE. Pseudo McFadden's $R^2$and as well as metrics used for prediction performance are considered for the global and regional are reported.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
Pseudo R^2	0.3796	0.4220	0.3080	0.7122	1
Out of sample correctness	0.9323	1	0.9881	0.9387	0.5079
In sample correctness	0.8694	0.9038	0.6346	0.9230	1
Deviance	720.185	125.0149	299.2989	70.9858	$6.5475e^{-9}$
Sensitivity	0.9881	0.9589	0.9647	0.9411	0.4905
Specificity	0.36	0.8	0.8182	0.9	0.6363
Precision	0.9398	0.9589	0.9761	0.9795	0.764
Recall	0.9881	0.9589	0.9647	0.9411	0.4905
PR AUC	0.5374	0.5	0.4938	0.4834	0.4175
ROC AUC	0.6741	0.8794	0.8914	0.9206	0.5823
Brier Score	0.0683	0.0681	0.0520	0.0655	0.466
Sample size (n)	988	260	312	182	234

The results for Table 5.40 are as follows: When considering the out-of-sample correctness, the regional models, with the exception of the Western European model, generated higher scores than those of the global model, which implies that the regional models were able to develop a model that could predict better results. For the in-sample correctness, Table 5.40 shows that the only regional model that could not generate an in-sample score better than the global model is the Latin America model, suggesting that it struggled to predict the outcome of the data used to develop the model. The global model produced the highest deviance, suggesting that the regional models had a better goodness-of-fit. In terms of sensitivity and recall, the global model generated the best sensitivity score, which suggests that the model had the highest true positive prediction. When considering specificity, the regional models produced the highest score, suggesting that they generated the best true negative prediction rate. When looking at precision, the regional models, with the exception of the Western European model, produced the highest score, which suggests that the models could correctly classify a positive which is really a positive. Considering the PR AUC, the global model produced the highest score, implying that it did well in terms of classifying. The ROC AUC suggests that the regional models, with the exception of the Western European model, produced a higher score than the global model, which implies that they had the best performance measurements for classification at different thresholds. In terms of the Brier score, the regional models, with the exception of the Western European model, generated a lower score, which means that it had the lowest mean squared error between the predicted probability and observed value. When looking at the ROC AUC score on Table 5.40, the regional models, with the exception

of the Western Europe model, performed better, making using regional models the best classification when MICE is considered for the probit classification method.

5.5 Bayesian Additive Regression Trees Classifier

When developing EWSs using the BART classifier, five (5) models are developed for each data set. The RMSE, RRMSE, ROC curve, out and in-sample distribution plot, variable importance plot, RMSE distribution plots and accuracy metrics are used to evaluate the model performance for unbalanced data, see appendix C. The dependent variable in these samples is the debt crisis indicator variable.As before, the time frame that is used to build the models is from 1980 to 2005, and the period that is used for the out of sample testing is from 2006 to 2012. The performance of the models listed below are bucketed into performance groups, this is applied to better elaborate the outcome of the predictions. The following are the different buckets for the ROC AUC: 0% to 50%, suggest no discrimination and therefore poor, 51% to 69% represents slighlty accaptable which is considered moderately good, 70% to 80% is acceptable and therefore noted as good, 81% to 90% is excellent and anything greater than 90% is outstanding (Mandrekar, 2010). The main metric used to assess model prediction accuracy will be the ROC AUC, therefore conclusions will be based on the ROC AUC results.

5.5.1 Comparison of Global and Regional BART Model Results

The BART classifier failed to fit the Western European model and hence could not produce precision, recall, and Brier scores for that data set, this suggest that the BART classifier requires a clear balance between default and non-default observations to classify properly. The Western Europe data set will not be considered further.

Table 5.41: BART regional model comparison for the case-wise deletion data sets.

	World	South and East Asia	Latin America	Middle East and Africa	Western Europe
RMSE	1.9537	1.6915	1.4787	1.366	1.6466
ROC AUC	0.5	0.6667	0.9215	1	0.8809
Out of sample correctness	0.9256	0.8823	0.9622	0.8181	0.6315
In sample correctness	0.8717	0.875	0.7944	0.8505	0.9533
RMSE(0.5,0.75)	(1.8377, 2.2202)	(1.697, 1.9678)	(1.4359, 1.7069)	(1.18869, 1.4222)	(1.4524, 2.1725)
Sensitivity	0.9739	0.875	0.9444	1	1
Specificity	0.5	0.7142	0.8181	0.2857	0
Precision	0.9256	0.875	0.9622	0.6153	*Na*
Recall	0.9739	0.875	0.9444	1	*Na*
PR AUC	0.5295	0.5	0.4920	0.6753	*Na*
Brier Score	0.0902	0.1739	0.0769	0.3333	0.5161

The results for Table 5.41 are as follows: The global model generated the highest RMSE value compared to the regional models, suggesting that the global model has a higher standard deviation of the residuals. The model that produced the highest out-of-sample accuracy was the Latin America data set, followed by the global model, and then the other two regional models. The global model outperformed all of the regional models based on the in-sample accuracy metrics since it produced the highest score, see Table 5.41. The RMSE values at the different percentiles, as of Table 5.41, show that the global model generated higher RMSE values as compared to regional models, which implies that the global model had a lot of false predictions. The only regional model that generates a sensitivity and recall score greater than that of the global is the Middle East and Africa model. However, the Middle East and Africa failed to generate a higher specificity score, while the other two regional models scored higher than the global model. The global model has a higher precision score than two of the three regional models considered, with the Latin America model being the exception. The global model failed to produce a ROC AUC value better than all the regional models, and has the second best PR AUC score. The Middle East and Africa model has the highest PR AUC score. The Latin America model is the only model to produce a Brier score better than the global model. When concluding from the ROC AUC score, the regional models were better predictors than the global model.

Table 5.42: BART regional model comparison for the mean imputations data sets.

	World	South and East Asia	Latin America	Middle East and Africa	Western Europe
RMSE	1.7632	1.6692	1.2742	1.8706	2.6688
ROC AUC	0.6265	1	0.8914	0.9583	0.6061
Out of sample correctness	0.9398	0.9220	0.9761	0.9795	0.7936
In sample correctness	0.8117	0.8576	0.75	0.8186	0.9829
RMSE(0.5,0.75)	$(1.7446, 2.1128)$	$(1.6356, 1.9535)$	$(1.1108, 1.5061)$	$(1.8245, 2.4161)$	$(2.5505, 2.7630)$
Sensitivity	0.9841	0.9589	0.9647	0.9411	0
Specificity	0.4	0.9589	0.8181	0.9	1
Precision	0.9431	0.9210	0.9761	0.9795	*Na*
Recall	0.9841	0.9589	0.9647	0.9411	*Na*
PR AUC	0.5306	0.5189	0.4938	0.4834	*Na*
Brier Score	0.0683	0.1011	0.0520	0.0655	0.2933

The Western European model will not be considered since it failed to produce an adequate model. The following can be deduced from Table 5.42: The Middle East and African models have the highest RMSE, suggesting that the models have a higher standard deviation of the residuals, making the global model second worst. The regional models, except the South and East Asia model, predicted the out of sample data more accurately than the global model. The Latin America model is the only regional model that does not accurately predict the in-sample data when compared to the global model, since its in-sample score is less than the score produced by the global model. The RMSE values at the different percentiles as of Table 5.42, show that the global model generated higher RMSE values as

compared to regional models, with the exception of the Middle East and Africa model. The global model produced the highest sensitivity and recall but the lowest specificity. The global model produced a better precision score than the South and East Asia model. The regional models produced a better ROC AUC value than the global model. However, the global model produced a better PR AUC value than the regional models. The regional models have better a Brier score, except for the South and East Asia model. When concluding from the ROC AUC score, the regional models, with the excpetion of the Western Europe model, were better predictors than the global model.

Table 5.43: BART regional model comparison for the MICE data sets.

	World	South and East Asia	Latin America	Middle East and Africa	Western Europe
RMSE	1.8593	1.7787	0.9956	2.0580	2.7303
ROC AUC	0.7318	0.9887	0.81097	1	0.5970
Out of sample correctness	0.9323	0.9342	0.9761	0.9795	0.7460
In sample correctness	0.8147	0.8692	0.7980	0.8461	0.9829
RMSE(0.5,0.75)	$(1.7225, 2.2970)$	$(1.7624, 1.9499)$	$(0.77761.2196)$	$(1.9307, 2.4549)$	$(2.6330, 2.8524)$
Sensitivity	0.9762	0.9589	0.9529	0.9411	0
Specificity	0.4	0.6	0.8182	0.9	1
Precision	0.9427	0.9210	0.9759	0.9795	*Na*
Recall	0.9762	0.9589	0.9529	0.9411	*Na*
PR AUC	0.5243	0.5196	0.4879	0.4834	*Na*
Brier Score	0.0755	0.1022	0.0625	0.0655	0.2933

The global model has the second highest RMSE, suggesting that the global model has a higher standard deviation of the residuals, see Table 5.43. All regional models generated better accuracy rates than the global model in terms of the out-of-sample data. Only the Latin America model failed to surpass the global model's in-sample accuracy, suggesting that the model failed to predict the in-sample data. The RMSE values at the different percentiles, as of Table 5.43, show that the global model had the lowest mid-value but the second-highest seventy-fifth percentile score, whereas the Middle East and Africa's model had the highest value which suggests that at a certain percentile, the global model performed better than the regional models. The global model generated the highest sensitivity and recall scores but produced the lowest specificity score compared to the regional models. Except for the model from the Middle East and Africa, regional models outperformed the global model in terms of precision value and Brier score. The global model produced the highest PR AUC but also yielded the lowest ROC AUC. When concluding from the ROC AUC score, the regional models, with the excpetion of the Western Europe model, were better predictors than the global model.

5.6 Artificial Neural Network Classifier

In the development of sovereign EWS using ANN, one hidden layer was used to develop the different models. Guided by Dongare et al. (2012) study, five (5) different number of nodes were considered per data set, which range from six (6) to ten (10). The in-and out-of sample accuracy, sensitivity, specificity, precision, recall, Brier score, PR AUC, and ROC AUC metrics are reported in Appendix D. There are no default observations in the Western Europe case-wise deletion data, and hence no model was fitted. In order to compare ANN to other classifiers, the best model per number of nodes was chosen as per performance measurement criteria.The performances of the models listed below are bucketed into performance groups; this is done to better elaborate on the outcome of the predictions. The following are the different buckets for the ROC AUC: 0% to 50% suggest no discrimination and are therefore poor. 51% to 69% represents slightly acceptable, which is considered moderately good; 70% to 80% is acceptable and therefore noted as good, 81% to 90% is excellent; and anything greater than 90% is outstanding (Mandrekar, 2010). The main metric used to assess model prediction accuracy will be the ROC AUC; therefore, conclusions will be based on the ROC AUC results.

5.6.1 Comparison of Global and Regional ANN Model Results

There is no clear consistency across all the models (see Appendix D) on how many nodes are required in the hidden layer to predict sovereign default. The selection of the best number of nodes to use when developing an ANN classifier was based on looking at the highest ROC AUC score produced by each number of nodes per data set. The case-wise deletion Western Europe data set could not generate a model due to the lack of default observations; therefore, it was not considered.

Table 5.44: Artificial neural network classifier summary for the of the case-wise deletion data sets.

Region	World	South and East Asia	Latin America	Middle East and Africa	Western Europe
Number of nodes	7	8	8	7	8
In sample	0.6994	0.7941	0.1132	0.6222	0.6316
Out-of-sample	0.7024	0.5882	0.2264	0.8	0.6316
Sensitivity	0.75	0.5357	0.1961	1	1
Specificity	0.1111	0.8333	1	0.5714	0
Precision	0.9130	0.9375	1	0.7273	*Na*
Recall	0.75	0.5357	0.1961	1	*Na*
Brier Score	0.2780	0.2097	0.7736	0.2052	0.3570
PR AUC	0.4262	0.3874	0.3247	0.5882	*Na*
ROC AUC	0.6447	0.7813	0.5925	0.7857	0.5

For the case-wise deletion data, Table 5.44 shows that the optimum number of nodes needed to produce appropriate models for this missing data technique ranges from seven (7) to eight (8). This is because

the most appropriate models for the different data sets were developed using that range of nodes in the hidden layer. The global model has a higher in-sample correctness score than three (3) of the four (4) regional models, the exception being the South and East Asia model. Only the Middle East and Africa model gave a better out-of-sample score than the global model, suggesting that the global model correctly cassified more observations than the regional models when looking at the test set; The global model's sensitivity and recall rate outperformed those of two (2) regional models, Latin America and South and East Asia, implying that the global model had a better true positive prediction. The specificity rate of the global model was low compared to the three functional regional models, suggesting that it generated the worst false-positive prediction rate. The global model generated a good precision score, but was only better than the Middle East and Africa model, which suggests that all models could correctly classify a positive which is really a positive, with the exception of the Western Europe and the Middle East and Africa models. Brier scores for models of the regions of Latin America, and Western Europe performed poorly as compared to the global models. However, the global ROC AUC scores are only better than the Latin American and Western European models, which implies that the regional models had the best performance measurements for classification at different thresholds. When concluding from the ROC AUC score, the regional models, with the exception of the Latin American and Western European models, were better predictors than the global model.

Table 5.45: Artificial neural network summary for the of the mean imputation data sets.

Region	World	South and East Asia	Latin America	Middle East and Africa	Western Europe
Number of nodes	7	8	9	6	8
In sample	0.1278	0.3636	0.036	0.9796	0.7936
Out-of-sample	0.5075	0.8442	0.167	0.9796	0.7936
Sensitivity	0.512	0.8286	0.1463	0.9411	0.9433
Specificity	0.4375	1	1	0.9	0.4091
Precision	0.9343	1	1	0.9795	0.7976
Recall	0.512	0.8286	0.1463	0.9411	0.9433
Brier Score	0.2494	0.1553	0.8259	0.0650	0.2047
PR AUC	0.3685	0.4304	0.3196	0.4834	0.5711
ROC AUC	0.5902	0.8972	0.6823	0.9205	0.6762

For the mean imputed data, Table 5.45 shows that the optimum number of nodes needed to produce appropriate models for this missing data technique ranges from six (6) to nine (9).This is because the most appropriate models for the different data sets were developed using that range of nodes in the hidden layer. Although having low scores, the regional models, with the exception of the Latin America model, had better in-sample accuracy than the global model. Only the Latin America model had a lower out-of-sample accuracy than the global model. The Latin America model produced the lowest sensitivity and recall value compared to other regional models; the global model had the second-worst sensitivity and recall value. The regional models generated the best specificity and precision

values, except for the Western European model. The Latin American model has the highest Brier score, while other regional models scored lower than the global model. Only the Latin American model failed to outperform the regional model based on the PR AUC value. When looking at the ROC AUC, the regional model produced better scores, which implies that the regional models had the best performance measurements for classification at different thresholds. When concluding from the ROC AUC score, the regional models were better predictors than the global model.

Table 5.46: Artificial neural network summary for the of the MICE data sets.

Region	World	South and East Asia	Latin America	Middle East and Africa	Western Europe
Number of nodes	8	8	7	10	10
In sample	0.6691	0.3815	0.2619	0.1428	0.7936
Out-of-sample	0.9323	0.8421	0.1785	0.5714	0.7936
Sensitivity	0.9959	0.8309	0.5731	0.5625	0.9433
Specificity	0	1	1	1	0.4091
Precision	0.9358	1	1	1	0.7936
Recall	0.9959	0.8309	0.5731	0.5625	0.9433
Brier Score	0.0618	0.1562	0.4407	0.4351	0.2076
PR AUC	0.5648	0.4277	0.4197	0.3681	0.5711
ROC AUC	0.7110	0.9315	0.8882	0.9607	0.6762

For the MICE data, see Table 5.46, which shows that the optimum number of nodes needed to produce appropriate models for this missing data technique ranges from seven (7) to ten (10). This is because the most appropriate models for the different data sets were developed using that range of nodes in the hidden layer. The Western Europe model is the only regional model to outperform the global model for the in-sample data test. No regional model outperformed the global model when looking at the out-of-sample accuracy. No regional model generated sensitivity and recall scores that were better than the global model. The global model performed poorly in terms of the specificity score, with a zero score. The only model with a precision score lower than the global model is the Western Europe model. The global model produced a Brier score that was higher than all regional models. Except for the Western Europe model, the global model produced the best PR AUC score. The best ROC AUC score was produced by the regional models when compared to the global model, with the exception of the Western European model, which implies that the regional models had the best performance measurements for classification at different thresholds. When concluding from the ROC AUC score, the regional models, with the exception of the Latin America model, were better predictors than the global model.

5.7 Comparison of Classifiers across Data Sets

To assess the best classifier per data set, the evaluation will be based on which model scored the best ROC AUC score. Therefore, the model with the overall highest ROC AUC score will be considered the best model.

5.7.1 The Global Region

Table 5.47: Classifier accuracy summary for the case-wise deletion for the global region.

	Logit	Probit	BART	ANN
In sample	0.8611	0.8699	0.8717	0.6995
Out-of-sample	0.9256	0.8925	0.9256	0.7024
Sensitivity	0.9739	0.9739	0.9739	0.75
Specificity	0.5	0.5	0.5	0.111
Precision	0.9256	0.9256	0.9256	0.9130
Recall	0.9739	0.9739	0.9739	0.75
Brier Score	0.0902	0.0902	0.0902	0.2781
PR AUC	0.5295	0.5295	0.5295	0.4261
ROC AUC	0.7370	0.7369	0.9739	0.4739

For the case-wise deletion data set, applied to the global region, Table 5.47 indicates that the BART classifier performed the best, the logit classifier came in second, the probit classifier came in third, and the ANN classifier performed the poorest.

Table 5.48: Classifier accuracy summary for the mean imputation for the global region.

	Logit	Probit	BART	ANN
In sample	0.8279	0.8329	0.8117	0.1278
Out-of-sample	0.9360	0.9398	0.9398	0.5075
Sensitivity	0.9881	0.9881	0.996	0.512
Specificity	0.36	0.36	0.4	0.4375
Precision	0.9398	0.9398	0.9431	0.9343
Recall	0.9881	0.9881	0.996	0.512
Brier Score	0.0683	0.0683	0.0601	0.2494
PR AUC	0.5374	0.5374	0.5554	0.3685
ROC AUC	0.6741	0.6741	0.6355	0.5902

For the mean imputation data set, applied to the global region, Table 5.48 shows that the logit and probit produced the best results, followed by the BART and, in last place, the ANN.

Table 5.49: Classifier accuracy summary for the Mice for the global region.

	Logit	Probit	BART	ANN
In sample	0.8724	0.8694	0.8147	0.6691
Out-of-sample	0.9323	0.9323	0.9323	0.9323
Sensitivity	0.9881	0.9881	0.9762	0.9959
Specificity	0.36	0.36	0.4	0
Precision	0.9398	0.9398	0.9427	0.9358
Recall	0.9881	0.9881	0.9762	0.9959
Brier Score	0.0683	0.0683	0.0755	0.0618
PR AUC	0.5374	0.5374	0.5445	0.5648
ROC AUC	0.6741	0.6741	0.7295	0.7110

For the MICE data set, applied to the global region, Table 5.49 shows that the BART classifier produced the best results, followed by ANN in second place, and in last place, the probit and logit classifier.

5.7.2 South and East Asia

Table 5.50: Classifier accuracy summary for the case-wise deletion for the South and East Asia region.

	Logit	Probit	BART	ANN
In sample	0.8312	1	0.875	0.6764
Out-of-sample	0.8571	0.8571	0.8823	0.8529
Sensitivity	0.857	0.875	0.875	1
Specificity	0.875	0.875	0.7142	0.1667
Precision	0.9333	0.9032	0.875	0.8484
Recall	0.875	0.875	0.875	1
Brier Score	0.1304	0.1521	0.1739	0.1470
PR AUC	0.4824	0.4909	0.5	0.6148
ROC AUC	0.8661	0.8304	0.6667	0.7075

For the case-wise deletion world data set, applied to the South and East Asia region, Table 5.50 shows that the logit classifier performed best. The probit was the second (2nd) best classifier, the ANN classifier in third (3rd) and the BART classifier in fourth (4th).

Table 5.51: Classifier accuracy summary for the mean imputation for the South and East Asia region

	Logit	Probit	BART	ANN
In sample	0.9153	0.9115	0.8576	0.3636
Out-of-sample	1	1	0.9220	0.8442
Sensitivity	0.95891	0.9589	1	0.8286
Specificity	1	0.9375	0.9589	1
Precision	1	0.9859	0.9210	1
Recall	0.95891	0.9589	1	0.8286
Brier Score	0.0683	0.0449	0.0779	0.1553
PR AUC	0.5374	0.4887	0.5736	0.4304
ROC AUC	0.6741	0.9482	1	0.8972

For the mean imputation world data set, applied to the South and East Asia region, Table 5.51, shows that the classifier with the best predictive capability is the BART model, with the probit coming in second (2nd), ANN coming in third (3rd), and the logit coming in fourth (4th).

Table 5.52: Classifier accuracy summary for the MICE for the South and East Asia region.

	Logit	Probit	BART	ANN
In sample	0.9038	0.8694	0.8692	0.9078
Out-of-sample	1	0.9323	0.9887	0.9868
Sensitivity	0.9589	0.9881	0.9589	1
Specificity	0.8	0.36	0.6	0.8
Precision	0.9589	0.9398	0.9210	0.9861
Recall	0.9589	0.9881	0.9589	1
Brier Score	0.0681	0.0683	0.1022	0.0115
PR AUC	0.5	0.5374	0.5196	0.5102
ROC AUC	0.8794	0.6741	0.9887	0.8589

For the MICE world data set, applied to the South and East Asia region, Table 5.52, shows that the BART built classifier performed the best, while the competition for second place was close, the logit model took it. ANN came in third, and the probit came in fourth.

5.7.3 Latin America

Table 5.53: Classifier accuracy summary for the case-wise deletion for the Latin America region.

	Logit	Probit	BART	ANN
In sample	0.7778	0.7555	0.7944	0.3773
Out-of-sample	0.7924	0.9056	0.9622	0.5660
Sensitivity	0.9259	0.9259	0.9444	0.5490
Specificity	0.8182	0.8182	0.8182	1
Precision	0.9615	0.9615	0.9622	1
Recall	0.9259	0.9259	0.9444	0.5490
Brier Score	0.0923	0.0923	0.0769	0.4265
PR AUC	0.4845	0.4845	0.4920	0.3669
ROC AUC	0.8721	0.8721	0.9215	0.3779

For the case-wise deletion world data set, applied to the Latin America region, Table 5.53, shows that the BART scored the highest, second come the logit and probit, and last place goes to the ANN.

Table 5.54: Classifier accuracy summary for the mean for the Latin America region.

	Logit	Probit	BART	ANN
In sample	0.7724	0.7724	0.75	0.0833
Out-of-sample	0.9642	0.9642	0.9761	0.3333
Sensitivity	0.9647	0.9647	0.9647	0.3171
Specificity	0.8182	0.8182	0.8182	1
Precision	0.9761	0.9761	0.9761	1
Recall	0.9647	0.9647	0.9647	0.3171
Brier Score	0.0521	0.0521	0.0521	0.6726
PR AUC	0.4938	0.4938	0.4938	0.3366
ROC AUC	0.8914	0.8914	0.8914	0.6529

For the mean imputation world data set, applied to the Latin America region, see Table 5.54, the BART, logit, and probit all performed similarly. The ANN performed poorly compared to the other three.

Table 5.55: Classifier accuracy summary for the MICE for the Latin America region.

	Logit	Probit	BART	ANN
In sample	0.7628	0.6346	0.7980	0.3214
Out-of-sample	1	0.9881	0.9761	0.4285
Sensitivity	0.9647	0.9647	0.9529	0.4146
Specificity	0.8182	0.8182	0.8182	1
Precision	0.9761	0.9761	0.9759	1
Recall	0.9647	0.9647	0.9529	0.4146
Brier Score	0.0520	0.0520	0.0625	0.4187
PR AUC	0.4938	0.4938	0.4879	0.3479
ROC AUC	0.8914	0.8914	0.81097	0.5470

For the MICE world data set, applied to the Latin America region, see Table 5.55, The logit and probit classifiers performed best, followed by BART. The ANN did not perform well.

5.7.4 The Middle East and Africa

Table 5.56: Classifier accuracy summary for the case-wise deletion for the Middle East and Africa region.

	Logit	Probit	BART	ANN
In sample	0.7586	0.7701	0.8505	0.6222
Out-of-sample	1	0.8095	0.8181	0.8
Sensitivity	0.875	0.875	1	1
Specificity	0.9047	0.9047	0.5	0.5714
Precision	0.9130	0.9130	0.7778	0.7273
Recall	0.875	0.875	1	1
Brier Score	0.1111	0.1111	0.1818	0.2052
PR AUC	0.4922	0.4922	0.6753	0.5882
ROC AUC	0.8899	0.8899	1	0.7857

For the case-wise deletion world data set, applied to the Middle East and Africa region, Table 5.56 shows that the BART classifiers perform best, followed by the logit and probit. The ANN does not perform well.

Table 5.57: Classifier accuracy summary for the mean imputation for the Middle East and Africa region

	Logit	Probit	BART	ANN
In sample	0.9230	0.9065	0.8186	0.9796
Out-of-sample	0.8775	0.8775	0.9795	0.9796
Sensitivity	0.8235	0.8431	0.9411	0.9411
Specificity	1	1	0.9	0.9
Precision	1	1	0.9795	0.9795
Recall	0.8235	0.8431	0.9411	0.9411
Brier Score	0.1475	0.1311	0.0655	0.0650
PR AUC	0.4387	0.4441	0.4834	0.4834
ROC AUC	0.9118	0.9216	0.9583	0.9205

For the mean imputation world data set, applied to the Middle East and Africa region, Table 5.57 shows that the BART performed best, followed by ANN in second (2nd). The probit is ranked third (3rd), and the logit is ranked fourth (4th).

Table 5.58: Classifier accuracy summary for the MICE for the Middle East and Africa region.

	Logit	Probit	BART	ANN
In sample	0.9230	0.9230	0.8461	0.6122
Out-of-sample	0.9387	0.9387	0.9795	0.9183
Sensitivity	0.9411	0.9411	0.9411	0.9167
Specificity	1	0.9	0.9	1
Precision	1	0.9795	0.9795	1
Recall	0.9411	0.9411	0.9411	0.9167
Brier Score	0.0491	0.0655	0.0655	0.0616
PR AUC	0.4759	0.4834	0.4834	0.4511
ROC AUC	0.9706	0.9206	1	0.7450

For the MICE world data set, applied to the Middle East and Africa region, see Table 5.58, shows that the BART classifiers came first. The logit in second (2nd). The probit is in third and BART has marginally better results. The ANN was the least accurate classifier.

5.7.5 Western Europe

No model could be generated for the casewise deletion data, therefore no comparison exists due to lack of sufficient data.

Table 5.59: Classifier accuracy summary for the mean imputation for the Western Europe region.

	Logit	Probit	BART	ANN
In sample	1	1	0.9829	0.7936
Out-of-sample	0.7619	0.7619	0.7936	0.7936
Sensitivity	0.7924	0.7924	0	0.9433
Specificity	0.5454	0.5455	1	0.4091
Precision	0.8076	0.8076	*Na*	0.7976
Recall	0.7924	0.7924	*Na*	0.9433
Brier Score	0.24	0.28	0.2933	0.2047
PR AUC	0.5120	0.4943	*Na*	0.5711
ROC AUC	0.6972	0.6972	0.6061	0.6762

For the mean imputation world data set, applied to the Western European region, Table 5.59 shows that the BART model has missing values. This is because the European data set had little to no defaults in the test set and can be seen by the precision and recall value for the BART not being computed,i.e. Na. Therefore, the respective values could not be computed. The logit and probit classifiers generated the best score, the ANN came in second (2nd), and the BART came in last place.

Table 5.60: Classifier accuracy summary for the MICE for the Western Europe region.

	Logit	Probit	BART	ANN
In sample	1	1	0.9829	0.7936
Out-of-sample	0.5079	0.5079	0.7460	0.7936
Sensitivity	0.5283	0.4905	0	0.9433
Specificity	0.6364	0.6363	1	0.4091
Precision	0.7778	0.764	*Na*	0.7936
Recall	0.5283	0.4905	*Na*	0.9433
Brier Score	0.44	0.466	0.2933	0.2076
PR AUC	0.4242	0.4175	*Na*	0.5711
ROC AUC	0.5634	0.5823	0.5970	0.6762

For the MICE world data set, applied to the Western European region, Table 5.60 shows that the BART classifier has missing values, as the European data set had very few default observations in the test set. The ANN generated the best score, the BART in second (2nd), the probit in third (3rd), and the logit in fourth (4th).

5.8 Performance: Regional and Global Performance

The tables below summarize the performance of the various classifiers in each region for the various data sets. Tables 5.61 and 5.62 summarize the predictive performance of the various classifiers in

each region for the various data set. Detailed results can be found in Appendix B, when applied to the mentioned datasets. The best classifier per region and missing data technique will be labeled with an X. The X's are then summed up to compare which classifier had the highest number of times that it performed better than the other classifier. Western European models were not considered in the counting due to a lack of consistency driven by the data sets not having enough defaults to compute meaningful accuracy scores, which is why they are labelled with a red "X".

Table 5.61: Summary of the best classifier by data set and region. An "X" is assigned to the classifier with the best overall performance in that particular instance.

	Logit			Probit			BART			ANN		
	CWD	Mean	Mice	CWD	Mean	Mice	CWD	Mean	Mice	CWD	Mean	Mice
Global		X			X		X		X			
South and East Asia	X							X	X			
Latin America		X	X		X	X	X	X				
The Middle East and Africa							X	X	X			
Western Europe		X			X							X
Total	4			3			9			0		

When assessing the best classifier by data set and region, Table 5.61, suggests that the BART classifier outperformed all other classifiers in nine (9) of the twelve (12) considered data sets, making it the most appropriate classifier to make use of. The logit and probit were similar in terms of predictive performance. The ANN did poorly.

Table 5.62: Summary of the best 5-fold cross validation classifier by data set and region. An "X" is assigned to the classifier with the best overall performance in that particular instance.

	Logit			Probit			BART			ANN		
	CWD	Mean	Mice	CWD	Mean	Mice	CWD	Mean	Mice	CWD	Mean	Mice
Global		x	x	x	x	x						
South and East Asia		x			x		x				x	x
Latin America										x	x	x
The Middle East and Africa									x	x	x	
Western Europe			x			x		x		x		
Total	3			4			2			7		

When assessing the best classifier by data set and region having considered 5-fold cross-validation, Table 5.62, suggests that the ANN classifier outperformed all other classifiers in most of the data sets, followed by the logit and probit. The BART only outperformed the other models in two of the considered twelve (12) data sets. This demonstrates that when cross-validation is considered, the ANN classifier can produce adequate sovereign default EWS models.

Table 5.63: Tabular representation of global to regional model outcomes. In cases where a regional model performs better than the global model, only that model is assigned an "X" and if it happens that no regional model outperforms the global model, the "X" will be assigned to the global model. The results considered are those from the test set which ranges from 2006 to 2012. All conclusion are based on the ROC AUC.

	Method	Globe	South and East Asia	Latin America	The Middle East and Africa	Western Europe
Binary logit	CWD		X	X	X	
	Mean		X	X	X	X
	MICE		X	X	X	
Probit	CWD		X	X	X	
	Mean		X	X	X	X
	MICE		X	X	X	
BART	CWD		X	X	X	X
	Mean			X	X	
	MICE		X	X	X	
ANN	CWD		X	X	X	X
	Mean		X		X	X
	MICE		X		X	

From the results in Table 5.63, it can be concluded that the regional models performed better than the global model in terms of overall performance. The global model had times where it outperformed the regional model, but not in all instances. The most consistant model was the Middle East and Africa models.

Table 5.64: Tabular representation of global to regional model outcomes. In cases where a regional model performs better than the global model, only that model is assigned an "X" and if it happens that no regional model outperforms the global model, the "X" will be assigned to the global model. The results considered are those from the test set which ranges from 2006 to 2012. All conclusion are based on the ROC AUC.

	Method	Globe	South and East Asia	Latin America	The Middle East and Africa	Western Europe
Binary logit	CWD		X	X	X	
	Mean		X		X	X
	MICE		X	X	X	X
Probit	CWD		X	X	X	
	Mean		X	X	X	X
	MICE		X		X	
BART	CWD		X		X	
	Mean		X		X	X
	MICE		X		X	
ANN	CWD		X	X	X	
	Mean		X	X	X	
	MICE		X	X	X	

From the results in Table 5.64, it can be concluded that the regional models performed better than the global model in terms of overall performance. This is seen in the fact that it did not outperform all regional models in any data set. The most consistent models were those of Southeast Asia, the Middle East, and Africa.

Table 5.65: Variable importance in terms of model development for the BART classifier per data set. The most important variable is labelled one (1) and the least relevant fourteen (14). The Western Europe importance is not considered therefore coloured red, due to lack of data.

Variable	World	South and East Asia	Latin America	The Middle East and Africa	Western Europe
Total debt	3	11	3	4	1
Global interest	4	8	4	12	8
Foreign Exch reserves	1	1	1	1	11
Trade openness	7	7	7	11	10
Export growth	13	12	10	14	14
Current account	9	4	9	9	2
FDI	8	6	13	6	12
Real GDP growth	14	14	14	12	13
Inflation	2	3	2	2	5
REER overall	10	10	8	7	4
Gov expenditures	5	5	5	5	3
National savings	12	13	11	10	9
Domestic credit	11	9	12	8	7
Bank assets	6	2	6	3	6

Table 5.65 shows that the top three (3) best variables to consider when developing a BART EWS classification model are foreign exchange reserves, inflation, and total debt, while the top three (3) variables to not consider are real GDP growth, export growth, and national savings.

Table 5.66: Table representing variable importance in terms of model development for the cross validation data for the BART classifier per data set. The most important variable is labelled one (1) and the least relevant fourteen (14). The Western Europe importance is not considered therefore coloured red, due to lack of data.

Variable	World	South and East Asia	Latin America	The Middle East and Africa	Western Europe
Total debt	4	12	4	4	1
Global interest	5	7	3	12	6
Foreign Exch reserves	2	1	1	1	11
Trade openness	7	9	7	8	5
Export growth	13	13	10	14	14
Current account	8	6	9	11	3
FDI	9	4	13	9	9
Real GDP growth	14	11	14	13	13
Inflation	1	5	2	2	10
REER overall	10	14	8	7	7
Gov expenditures	6	2	6	5	4
National savings	12	10	11	10	12
Domestic credit	11	8	12	6	8
Bank assets	3	3	5	3	2

Table 5.66 shows that the top three (3) best variables to consider when developing a BART EWS classification model are foreign exchange reserves, inflation, and bank assets, while the top three (3) variables to not consider are real GDP growth, export growth, and national savings.

5.9 Additional Artificial Neural Networks Analysis

The ANN analysis in the previous section suggests that building a neural network classifier without cross-validation does not produce the best results. In this section, an analysis is done to determine if using the variables that are considered important can improve the predictive performance of ANN classifiers without cross-validation. The variables that were concluded to be the top five variables are foreign exchange reserves, inflation, total debt, global interest, and government expenditure.

Table 5.67: A summary comparison of the variable reduced ANN classifiers with 2-4 nodes and comparing those classifiers to the best classifier by data set and region. Case-wise deletion was considered here.

	2	3	4	Best
In sample	0.8769	0.8998	0.8910	0.6995
Out-of-sample	0.9173	0.9173	0.9090	0.7024
Sensitivity	0.9910	0.9910	0.9821	0.75
Specificity	0	0	0	0.111
Precision	0.925	0.9910	0.9243	0.9130
Recall	0.9910	0.925	0.9821	0.75
Brier Score	0.0826	0.0826	0.0909	0.2781
PR AUC	0.5668	0.5668	0.5550	0.4261
ROC AUC	0.4955	0.4955	0.4910	0.4739

The analysis in Table 5.67 shows that the neural network classifier that focuses on using the variables that are most important, performed better than the model that considers all variables; this was done for the global case-wise deletion data set. Therefore, one of the ways to improve results for the classifiers developed in this study is to reduce the number of variables in the study and focus on the top five (5) important variables used to build the model. Results are derived from observing the ROC AUC score.

Table 5.68: A summary comparison of the variable reduced ANN classifiers with 2-4 nodes and comparing those classifiers to the best classifier by data set and region. Mean imputation was considered here.

	2	3	4	Best
In sample	0.8633	0.8542	0.8623	0.1278
Out-of-sample	0.9473	0.9285	0.9323	0.5075
Sensitivity	0.996	0.972	0.976	0.512
Specificity	0.1875	0.25	0.25	0.4375
Precision	0.9503	0.9529	0.9531	0.9343
Recall	0.996	0.972	0.976	0.512
Brier Score	0.0526	0.0714	0.0676	0.2494
PR AUC	0.5445	0.5155	0.5191	0.3685
ROC AUC	0.5917	0.611	0.613	0.5902

The analysis in Table 5.68 shows that the neural network classifier that focuses on using the variables that are most important performed better than the model that considers all variables; this is done for the global mean imputation data set. Therefore, one of the ways to improve results for the classifiers developed in this study is to reduce the number of variables in the study and focus on the top five (5) important variables used to build the model. Results are derived from observing the ROC AUC score.

Table 5.69: A summary comparison of the variable reduced ANN classifiers with 2-4 nodes and comparing those classifiers to the best classifier by data set and region. MICE was considered here.

	2	3	4	Best
In sample	0.5751	0.5751	0.5751	0.6691
Out-of-sample	0.9097	0.9323	0.9323	0.9323
Sensitivity	0.956	0.988	0.992	0.9959
Specificity	0.1875	0.0625	0	0
Precision	0.9484	0.9427	0.9393	0.9358
Recall	0.956	0.988	0.992	0.9959
Brier Score	0.0813	0.0575	0.0632	0.0618
PR AUC	0.5058	0.5445	0.5554	0.5648
ROC AUC	0.5717	0.5252	0.496	0.7110

The analysis in Table 5.69 shows that the neural network classifier that focuses on using the variables that are most important performed worse than the model that considers all variables; this is for the global MICE data set. Therefore, when MICE is considered a missing data technique, no reduction is needed to improve the model's predictions. Results are derived from observing the ROC AUC score.

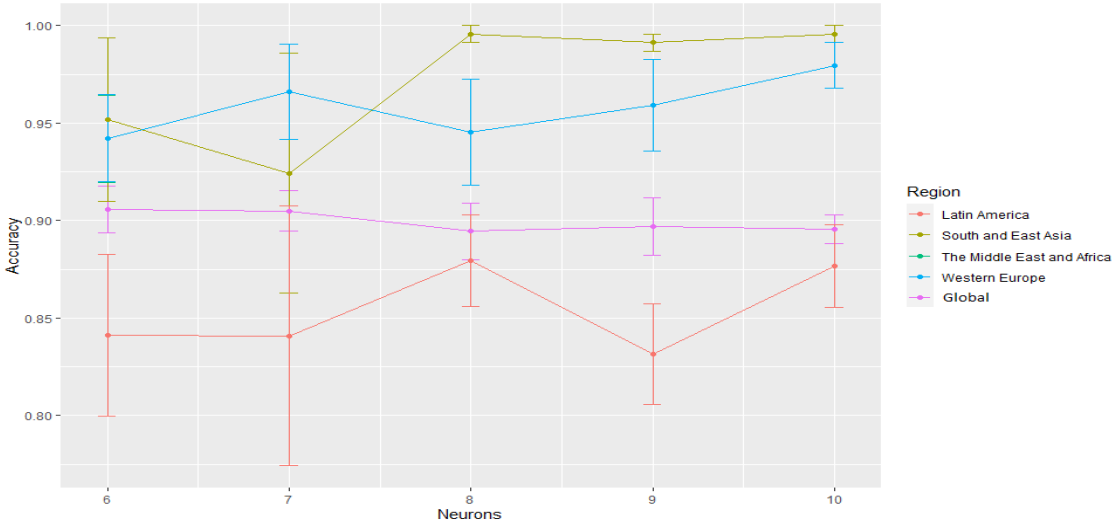

Figure 5.9: 5-fold accuracy to number of neurons plot for all regions.

The ANN accuracy plot for the different neurons, see Figure 5.9 , suggests that the global model had the least volatility when increasing the number of neurons in the classifier. The data set with the smallest standard error is the South and East Asia data set. The data set with the highest standard error is the Latin America data set. The global classifier outperformed the Latin America classifier.

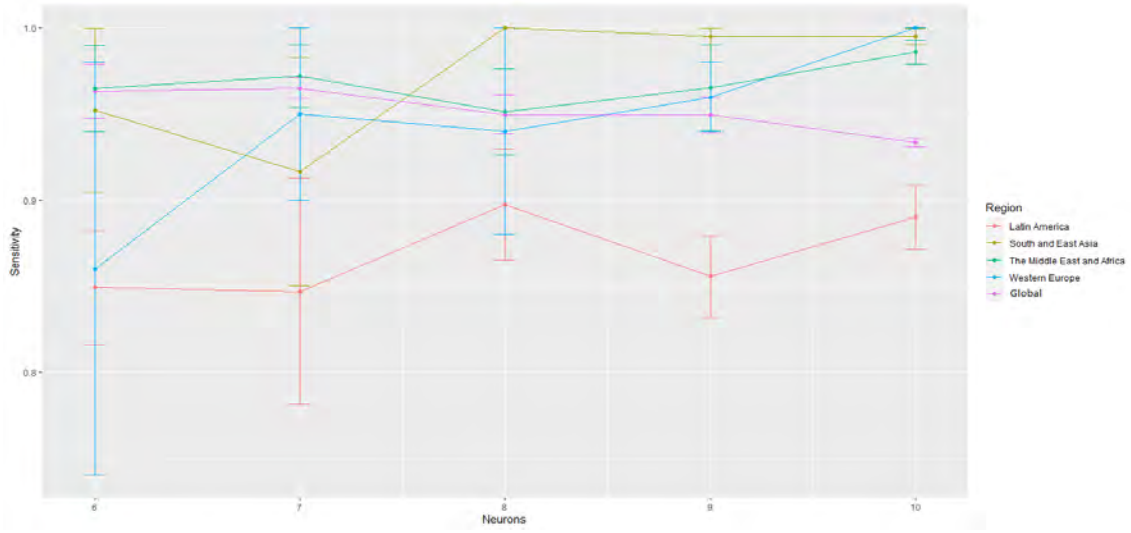

Figure 5.10: 5-fold sensitivity to number of neurons plot for all regions.

The sensitivity plot, see Figure 5.10, suggests that the sensitivity score of the global model decreases when the number of nodes in the hidden layer decreases. On average, the Latin America data set had the highest standard error, while the global model had the smallest.

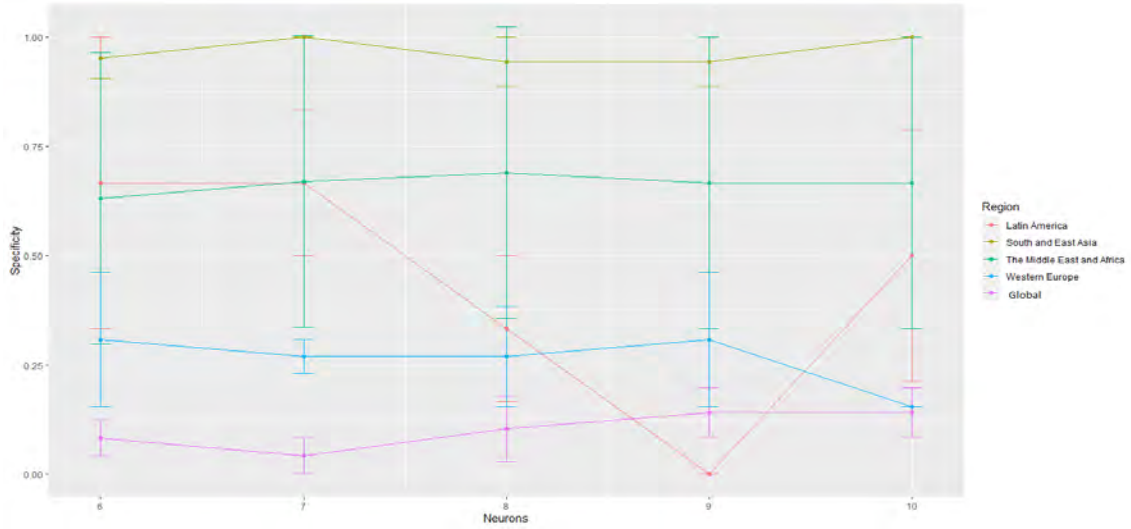

Figure 5.11: 5-fold specificity to number of neurons plot for all regions.

The specificity plot, see Figure 5.11, suggests that the model with the highest volatility as number of nodes in the hidden layer decrease is the Latin America model. The global model performed poorly.

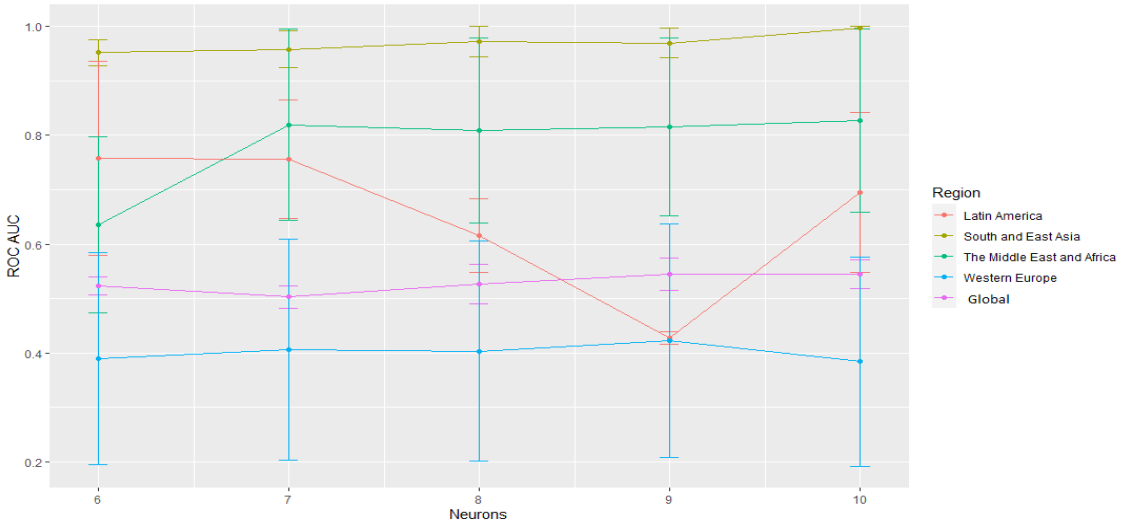

Figure 5.12: 5-fold ROC AUC to number of neurons plot for all regions.

The ROC AUC, see Figure 5.12, suggests that with the exception of the Western Europe classifier, the global model performed poorly. There is also a huge decline in the Latin America classifier from number of nodes seven (7) to nine (9). The Western Europe classifier had the highest average standard error.

Chapter 6

Conclusion

The purpose of this study was to develop classifiers of sovereign default and identify the feature variables that could be used to develop EWSs for sovereign default. In addition to the classifiers identified in the literature review, namely the ANNs, the binary logit, and the probit classifiers, BART was also applied to these data. Due to the high number of missing entries, three missing data methods were considered, namely: case-wise deletion, mean imputation and, MICE. Where there was enough data to improve the model's prediction accuracy, 5-fold cross-validation was used. The following conclusions can be drawn from the two cases: When 5-fold cross-validation was considered, ANNs produced the most consistent predictive performance (see Table 82, 83, 84, 85, 86, 87, 88, 89, and 90). For models generated without cross-validation, the BART classifier produced the most consistent predictive performance (Table 5.17, 5.23, 5.28, 5.34, 5.40, 5.42, 5.43, 5.44, 5.45, 5.46). In general, regional classifiers performed better than the global classifier. The results from Chapter 5 are briefly summarized below. Due to the highly imbalanced nature of the data, ROC AUC was primarily used to assess classifier performance, based on the interpretation of this metric by Mandrekar (2010).

6.1 Logit Classifier

Under the case-wise deletion data sets with and without cross-validation, the following tables show that no variables were concluded to be significant when generating EWS models. These tables include, Table 5.7, 5.8, 5.9, 5.10, as well as Tables 10, and 11 in Appendix B. This is due to the deletion of entries, which results in fewer default observations being available for the model to generate meaningful conclusions from the available data. For models generated without cross-validation, the following variables show to be significant contributors to sovereign default detection: foreign exchange reserves, FDI, inflation, REER overall, domestic credit, national savings, government expenditure, global interest, trade openness, export growth, and bank assets; and for models generated with cross-validation, the following variables show to be significant contributors to sovereign default detection: FDI, inflation,

foreign exchange reserves, REER, government expenditure, domestic credit, total debt, global interest, national savings, bank assets, current account, export growth, and trade openness. The resulting Western European models were poor detectors of sovereign default, which is a direct result of the lack of default observations in this data (see Table 5.16 and 5.22 aboveand Tables 11, 16, and 21 in Appendix B). Overall, the regional classifiers for both the models developed with and without cross-validation outperformed the global classifiers when considering all data techniques (see Table 5.11, 5.17, and 5.23 above and Tables 22, 23, 24, and 40 in Appendix B), supporting Dawood et al. (2017)'s finding that better predictions are generated if regional classifiers are developed rather than one general global classifier. The binary logit classifier is a good choice when generating EWS; its results are similar to those of the probit classifier. Although it is a good classifier, the ANN's classifier performed better when cross-validation was considered, and the BART performed better when cross-validation was not considered (see Tables 5.61 and 5.62 that summarize the best classifier per data set).

6.2 Probit Classifier

Under the case-wise deletion data sets with and without cross-validation, the following tables show that no variable was concluded to be significant when generating EWS models. These tables include, Table 5.25, 5.26, 5.27, as well as Tables 28, and 29 in Appendix B. Similar to the logit classifier, this results from too few default observations in the training data. For models generated without cross-validation, the following variables show to be significant contributors to sovereign default detection: foreign exchange reserves, FDI, inflation, total debt, REER overall, domestic credit, government expenditure, export growth, global interest, national savings, domestic credit, bank assets, and trade openness and for models generated with cross-validation, the following variables show to be significant contributors to sovereign default detection: FDI, foreign exchange, inflation, REER, total debt, global interest, government expenditure, national savings, domestic credit, bank assets, export growth, current account, trade openness, and real GDP growth. The resulting Western European models were poor detectors of sovereign default, which is a direct result of the lack of default observations in this data (see Table 5.33, 5.39 above and 29, 34, 39 in Appendix B). Overall, the regional classifiers for both the models developed with and without cross-validation outperformed the global classifiers when considering all data techniques (see Table 5.28, 5.34, 5.40, and 40 above and Table 41 in Appendix B), supporting Dawood et al. (2017)'s finding that better predictions are generated if regional classifiers are developed rather than one general global classifier. The probit classifier is a good choice when generating EWS; its results are similar to those of the logit classifier. Although it is a good classifier, the ANN's classifier performed better when cross-validation was considered, and the BART performed better when cross-validation was not considered (see Tables 5.61 and 5.62 that summarize the best classifier per data set).

6.3 BART Classifier

Due to the imbalanced data, the ROC AUC scores were used to determine the model's performance. Using this metric, the overall performance of regional models is better than that of a global model (see Table 5.41, 5.42, 5.43, as well as Tables 58, 59, and 60 in Appendix B). In addition, the global model generated higher RMSE scores compared to regional models (see Tables 5.41, 5.42, 5.43,as well as Tables 58, 59, and 60 in Appendix B), suggesting that it has more incorrect predictions than the regional models. This supports Dawood et al. (2017)'s findings, which state that predictions are better if regional classifiers are developed rather than one general global classifier. These data are sparse and often lack enough information to build a good model using BART. Tables 45, 50, 55, 57, 102, 107 in Appendix B show that BART, applying both with and without cross validation, failed to develop sufficiently good models to judge the difference between default and non-default periods. In addition, BART completely failed to generate any metric scores for Latin America models when cross-validation is considered. The overall rank importance of the variables considered in the study is shown in tables 5.65 and 5.66, with one (1) being the most important and fourteen (14) being the least important. The BART classifier performed the best as an EWS model, but when cross-validation is taken into account, the ANN performs the best.

6.4 Neural Network Classifier

The results from the ANN classifiers support the following conclusions: Overall, cross-validation improves the performance of ANN classifiers; see Section 5.7 for classifiers without cross-validation and Section B.5 for classifiers with cross-validation. The number of neurons in the hidden layer needed to build an adequate model ranges between seven (7) and ten (10) with cross-validation and between six (6) and nine (10) when cross-validation is not considered. Under cross-validation, it is clear that regional models are more appropriate to use in the generation of EWS (Table 76, 77, and 78). When cross-validation is not considered, the regional models performed best, but there were cases where one regional model would show poor performance (5.44, 5.45, and 5.46). Overall, this supports Dawood et al. (2017)'s findings, which state that predictions are better generated if regional classifiers are developed rather than one general global classifier.

6.5 Cross-Validation vs Non-Cross-Validation

When considering the 70/30 split between training and testing sets, the cross-validation method provided more significant results than the method without cross-validation. This is due to the method's ability to allow multiple training pieces via splits (Fushiki, 2011).

6.6 Variable Importance

Table 6.1 shows variables that have been identified as significant for the BART classifier's development.

Table 6.1: Official rank for the variable importance in default model development.

Rank	Non cross-validation	5-fold
1	Foreign Exch reserves	Inflation
2	Inflation	Foreign Exch reserves
3	Total debt	Bank assets
4	Global interest	Total debt
5	Gov expenditures	Global interest
6	Bank assets	Gov expenditures
7	Trade openness	Trade openness
8	FDI	Current account
9	Current account	FDI
10	REER overall	REER overval
11	Domestic credit	Domestic credit
12	National savings	National savings
13	Export growth	Export growth
14	Real GDP growth	Real GDP growth

Table 6.1 is a conclusive ranking of variables considered in this study. The order in which they appear indicates how relevant they are to model development when developing BART classifiers for cross-validations and even without using cross-validation.

6.7 Data Imputation

Three imputation techniques were applied in this study. The case-wise deletion technique produced the least competitive results, primarily as a result of removing a large percentage of the data available to estimate the various parameters' accuracy in the various models. A classifier built on the MICE data generated better results when compared to the classifiers built using the mean imputation data, as observed in the global model comparisons done in the study (compare Tables 5.48 and 5.49). MICE was the best method to use when tackling the missing data issue in this study.

6.8 Limitations of Research

The Western European model was particularly hard to develop, since there are a limited number of default observations. There were two variables not available in this analysis that were available to Dawood et al. (2017), namely M2/reserves, IMF credit, and government bank loans. The data sets

used in this analysis had many missing observations. Due to the likelihood of a sovereign default being very low, the data under consideration is extremely imbalanced. Some regional data sets were too small for the techniques to distinguish between default and non-default. The Western European data set was particularly hard to develop since there are a limited number of default observations. The small nature of these data has resulted in large standard errors (see, for example, Tables 5.16, 5.22, 5.25, 5.27, 5.33, and 5.39). This is also the primary cause of the differences observed between the 70/30 split and the cross-validation results.

6.9 Future Research

Only one hidden layer was present in the trained neural network classifiers; increasing the number of hidden layers in a neural network classifier may improve its performance. Ideally, future researchers could find the missing data, increasing the sample size would also give more accurate results. Finding data with more defaults may improve model performance, particularly for BART. Boosting was not considered in this analysis. Boosting is a standard method for improving any learning algorithm's accuracy. Future research could consider applying ensemble learners, for example, in random forests. MICE was applied in this study, future research could consider bagging and Multiple Imputation Ensembles (MIE) (Aleryani et al., 2020).

The time-series nature of these data has not been considered in this study. Appropriate methods could be applied to address these time series data, as, for example, suggested in Brownlee (2019). For example, the data could be set so that the current period is used to encode the class/dependent variable and the previous or past data, which may or may not include the current period feature variables, is used for the feature/independent variables. A sliding time window would then pass over the data set. It is unclear where this historical or timestamp-based data can be obtained, but this is worthy of future research.

References

Abraham, A. (2005). Artificial neural networks. *Handbook of Measuring System Design*.

Abramovitz, M. (1986). Simon kuznets 1901–1985. *The Journal of Economic History*, 46(1):241–246.

Abu-Nimeh, S., Nappa, D., Wang, X., and Nair, S. (2008a). Bayesian additive regression trees-based spam detection for enhanced email privacy. In *2008 Third International Conference on Availability, Reliability and Security*, pages 1044–1051. IEEE.

Abu-Nimeh, S., Nappa, D., Wang, X., and Nair, S. (2008b). A distributed architecture for phishing detection using Bayesian additive regression trees. In *2008 eCrime Researchers Summit*, pages 1–10. IEEE.

Afghari, A. P., Washington, S., Prato, C., and Haque, M. M. (2019). Contrasting case-wise deletion with multiple imputation and latent variable approaches to dealing with missing observations in count regression models. *Analytic Methods in Accident Research*, 24:100104.

Aldrich, J. H. and Nelson, F. D. (1984). *Linear probability, logit, and probit models*. Sage.

Aleryani, A., Wang, W., and De La Iglesia, B. (2020). Multiple imputation ensembles (MIE) for dealing with missing data. *SN Computer Science*, 1(3):1–20.

Allison, P. D. (2001). *Missing data*. Sage publications.

Altman, E. and Sabato, G. (2013). Modeling credit risk for SMEs: evidence from the US market. In *Managing and Measuring Risk: Emerging Global Standards and Regulations After the Financial Crisis*, pages 251–279. World Scientific.

Altman, N. and Krzywinski, M. (2017). Ensemble methods: bagging and random forests. *Nature Methods*, 14(10):933–935.

Andrews, D. W. and Monahan, J. C. (1992). An improved heteroskedasticity and autocorrelation consistent covariance matrix estimator. *Econometrica: Journal of the Econometric Society*, pages 953–966.

Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., and Ridella, S. (2012). The'k'in k-fold cross validation. In *ESANN*, pages 441–446.

Bai, Y. and Zhang, J. (2012). Duration of sovereign debt renegotiation. *Journal of International Economics*, 86(2):252–268.

Bennell, J. A., Crabbe, D., Thomas, S., and Ap Gwilym, O. (2006). Modelling sovereign credit ratings: Neural networks versus ordered probit. *Expert Systems with Applications*, 30(3):415–425.

Berry, W. D., DeMeritt, J. H., and Esarey, J. (2010). Testing for interaction in binary logit and probit models: is a product term essential? *American Journal of Political Science*, 54(1):248–266.

Bhaskaran, K. and Smeeth, L. (2014). What is the difference between missing completely at random and missing at random? *International Journal of Epidemiology*, 43(4):1336–1339.

Biau, G. and Scornet, E. (2016). A random forest guided tour. *Test*, 25(2):197–227.

Blackburn, S. (2005). *The Oxford Dictionary of Philosophy*. OUP Oxford.

Boyd, K., Eng, K. H., and Page, C. D. (2013). Area under the precision-recall curve: point estimates and confidence intervals. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 451–466. Springer.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Brownlee, J. (2019). Comparing classical and machine learning algorithms for time series forecasting. *Machine Learning Mastery, Australia*.

Buchheit, L. C. and Gulati, G. M. (2002). Sovereign bonds and the collective will. *Emory LJ*, 51:1317.

Buchheit, L. C. and Gulati, M. (2021). Avoiding a lost decadeâsovereign debt workouts in the post-covid era. *Capital Markets Law Journal*, 16(1):45–55.

Buuren, S. v. and Groothuis-Oudshoorn, K. (2010). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, pages 1–68.

Cakmakyapan, S., Goktas, A., et al. (2013). A comparison of binary logit and probit models with a simulation study. *Journal of Social and Economic statistics*, 2(1):1–17.

Calomiris, C. W. (2000). When Will Economics Guide, IMF and World Bank Reforms. *Cato J.*, 20:85.

Cao, P., Yang, J., Li, W., Zhao, D., and Zaiane, O. (2014). Ensemble-based hybrid probabilistic sampling for imbalanced data learning in lung nodule CAD. *Computerized Medical Imaging and Graphics*, 38(3):137–150.

Cao, Y., Liu, X., Zhai, J., and Hua, S. (2020). A two-stage Bayesian network model for corporate bankruptcy prediction. *International Journal of Finance & Economics*.

Cappa, E. P. and Cantet, R. J. C. (2008). Direct and competition additive effects in tree breeding: Bayesian estimation from an individual tree mixed model. *Silvae Genetica*.

Casella, G. and George, E. I. (1992). Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948.

Chipman, H. A., George, E. I., McCulloch, R. E., et al. (2010). BART: Bayesian Additive Regression Trees. *The Annals of Applied Statistics*, 4(1):266–298.

Choi, S. J., Gulati, M., and Posner, E. A. (2012). The evolution of contractual terms in sovereign bonds. *Journal of Legal Analysis*, 4(1):131–179.

Ciarlone, A. and Trebeschi, G. (2005). Designing an early warning system for debt crises. *Emerging Markets Review*, 6(4):376–395.

Cortes, C. and Mohri, M. (2003). AUC optimization vs. error rate minimization. *Advances in Neural Information Processing Systems*, 16:313–320.

Cox, D. R. and Snell, E. J. (2018). *Analysis of binary data*. Routledge.

Dakovic, R., Czado, C., and Berg, D. (2010). Bankruptcy Prediction in Norway: A Comparison Study. *Applied Economics Letters*, 17(17):1739–1746.

Dawood, M., Horsewood, N., and Strobel, F. (2017). Predicting sovereign debt crises: An early warning system approach. *Journal of Financial Stability*, 28:16–28.

Deininger, K. and Byerlee, D. (2011). *Rising global interest in farmland: can it yield sustainable and equitable benefits?* World Bank Publications.

Donders, A. R. T., Van Der Heijden, G. J., Stijnen, T., and Moons, K. G. (2006). A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, 59(10):1087–1091.

Dongare, A., Kharde, R., Kachare, A. D., et al. (2012). Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1):189–194.

Drews, C. D. and Greenland, S. (1990). The impact of differential recall on the results of case-control studies. *International Journal of Epidemiology*, 19(4):1107–1112.

Duan, T., Anand, A., Ding, D. Y., Thai, K. K., Basu, S., Ng, A., and Schuler, A. (2020). Ngboost: Natural gradient boosting for probabilistic prediction. In *International Conference on Machine Learning*, pages 2690–2700. PMLR.

Eaton, J. and Fernandez, R. (1995). Sovereign debt. *Handbook of International Economics*, 3:2031–2077.

Escanciano, J. C. (2006). A consistent diagnostic test for regression models using projections. *Econometric Theory*, pages 1030–1051.

Faraway, J. (2014). *Linear models with R*. CRC press.

Feng, D., Gouriéroux, C., and Jasiak, J. (2008). The ordered qualitative model for credit rating transitions. *Journal of Empirical Finance*, 15(1):111–130.

Fioramanti, M. (2008). Predicting sovereign debt crises using artificial neural networks: a comparative approach. *Journal of Financial Stability*, 4(2):149–164.

Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407.

Fuertes, A.-M. and Kalotychou, E. (2006). Early warning systems for sovereign debt crises: The role of heterogeneity. *Computational Statistics & Data Analysis*, 51(2):1420–1441.

Fuertes, A.-M. and Kalotychou, E. (2007). Optimal design of early warning systems for sovereign debt crises. *International Journal of Forecasting*, 23(1):85–100.

Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. *Statistics and Computing*, 21(2):137–146.

Goodhart, C. (2011). *The Basel Committee on Banking Supervision: a History of the Early Years 1974-1997*. Cambridge University Press.

Graham, J. (2009). Missing data analysis: Making it work in the real world. *Annual Review of Psychology*, 60:549–576.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Grunert, J., Norden, L., and Weber, M. (2005). The role of non-financial factors in internal credit ratings. *Journal of Banking & Finance*, 29(2):509–531.

Hand, D. J. (2009). Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning*, 77(1):103–123.

Harville, D. A. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, 72(358):320–338.

Hastie, T., Tibshirani, R., et al. (2000). Bayesian backfitting (with comments and a rejoinder by the authors. *Statistical Science*, 15(3):196–223.

Hausman, J. A. and Wise, D. A. (1978). A conditional probit model for qualitative choice: Discrete Decisions Recognizing Interdependence and Heterogeneous Preferences. *Econometrica: Journal of the Econometric Society*, pages 403–426.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Hebb, D. O. (1949). *The Organization of Behavior: a Neuropsychological Theory*. J. Wiley; Chapman & Hall.

Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier.

Hensher, D. and Greene, W. (2003). The mixed logit model: the state of practice. *Transportation*, 30(2):133–176.

Hensher, D., Jones, S., and Greene, W. (2007). An error component logit analysis of corporate bankruptcy and insolvency risk in Australia. *Economic Record*, 83(260):86–103.

Hill, J., Linero, A., and Murray, J. (2020). Bayesian additive regression trees: a review and look forward. *Annual Review of Statistics and Its Application*, 7:251–278.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.

Hung, K., Cheng, H. W., Chen, S., Huang, Y.-C., et al. (2013). Factors that affect credit rating: An application of ordered probit models. *Romanian Journal of Economic Forecasting*, 16(4):94–108.

Hunter, W. C., Kaufman, G. G., and Krueger, T. H. (2012). *The Asian financial crisis: origins, implications, and solutions*. Springer Science & Business Media.

Jacobs, M. et al. (2010). An empirical study of exposure at default. *Journal of Advanced Studies in Finance (JASF)*, 1(01):31–59.

Japkowicz, N. (2013). Assessment metrics for imbalanced learning. *Imbalanced Learning: Foundations, Algorithms, and Applications*, pages 187–206.

Jones, S. and Hensher, D. A. (2004). Predicting firm financial distress: A ixed logit model. *The accounting review*, 79(4):1011–1038.

Kapelner, A. and Bleich, J. (2013). bartMachine: Machine learning with Bayesian additive regression trees. *arXiv Preprint arXiv:1312.2171*.

Knoema (2021). knoema: Commodities data hub. `https://knoema.com/`, accessed May 2021.

Kotsiantis, S. B., Zaharakis, I., Pintelas, P., et al. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, 160(1):3–24.

Kukuk, M. and Rönnberg, M. (2013). Corporate credit default models: a mixed logit approach. *Review of Quantitative Finance and Accounting*, 40(3):467–483.

Laušev, J., Stojanović, A., and Todorović, N. (2011). Determinants of debt rescheduling in Eastern European countries. *Economic Annals*, 56(188):7–31.

Lemmens, A. and Croux, C. (2006). Bagging and boosting classification trees to predict churn. *Journal of Marketing Research*, 43(2):276–286.

Lennox, C. (1999). Identifying failing companies: a re-evaluation of the logit, probit and DA approaches. *Journal of Economics and Business*, 51(4):347–364.

Lestano, J. J. and Kuper, G. H. (2003). Indicators of Financial Crises do Work! An Early-Warning System for Six Asian Countries, Department of Economics, University of Groningen. *Paper Provided by EconWPA in its Series International Finance with*.

Luo, C., Wu, D., and Wu, D. (2017). A deep learning approach for credit scoring using credit default swaps. *Engineering Applications of Artificial Intelligence*, 65:465–470.

MachineLearningMastery (2021). ROC Curves and Precision-Recall Curves for Imbalanced Classification. URL(https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/).

Malmendier, U. (2018). Behavioral corporate finance. In *Handbook of Behavioral Economics: Applications and Foundations 1*, volume 1, pages 277–379. Elsevier.

Manasse, P. and Roubini, N. (2009). "Rules of thumb" for sovereign debt crises. *Journal of International Economics*, 78(2):192–205.

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9):1315–1316.

Marlin, B. (2008). *Missing data problems in machine learning*. PhD thesis, University of Toronto.

McCarthy, J., Minsky, M. L., and Rochester, N. (1956). The Dartmouth summer research project on artificial intelligence. *Artificial Intelligence: Past, Present, and Future*.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.

McFadden, D. L. (1984). Econometric analysis of qualitative response models. *Handbook of Econometrics*, 2:1395–1457.

Minsky, M. and Papert, S. (1969). An Introduction to Computational Geometry. *Cambridge tiass., HIT*.

Miu, P. and Ozdemir, B. (2006). Basel requirement of downturn LGD: Modeling and estimating PD and LGD correlations. *Journal of Credit Risk*, 2(2):43–68.

Montgomery, J. M. and Nyhan, B. (2010). Bayesian model averaging: Theoretical developments and practical applications. *Political Analysis*, pages 245–270.

Nalenz, M. and Villani, M. (2018). Tree ensembles with rule structured horseshoe regularization. *The Annals of Applied Statistics*, 12(4):2379–2408.

Narkhede, S. (2018). Understanding auc-roc curve. *Towards Data Science*, 26:220–227.

Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7:21.

Nathaniel, R. (1951). Signal Transmission Network. IMF Publication. US Patent 2,570,716.

O'Neill, D. (2014). Gross domestic product. In *Degrowth*, pages 131–136. Routledge.

Pastor, M. (1991). Bolivia: Hyperinflation, stabilisation, and beyond. *The Journal of Development Studies*, 27(2):211–237.

Patil, P. N., Dubey, S. K., Pinjari, A. R., Cherchi, E., Daziano, R., and Bhat, C. R. (2017). Simulation evaluation of emerging estimation techniques for multinomial probit models. *Journal of choice modelling*, 23:9–20.

Pereira, P., Silva, E. S., and Pereira, A. (2018). Operational risk management: The basel II. *The International Journal of Business Management and Technology*, 2.

Pescatori, A. and Sy, A. N. (2007). Are debt crises adequately defined? *IMF Staff Papers*, 54(2):306–337.

Peter, M. (2002). Estimating default probabilities of emerging market sovereigns: A new look at a not-so-new literature. *Available at SSRN 1017468*.

Porath, D. (2006). Estimating probabilities of default for German savings banks and credit cooperatives. *Schmalenbach Business Review*, 58(3):214–233.

Pratola, M. T. (2016). Efficient Metropolis–Hastings proposal mechanisms for Bayesian regression tree models. *Bayesian Analysis*, 11(3):885–911.

Radloff, S. (2019). *GLM Honours Course Notes*. Rhodes University.

Refaeilzadeh, P., Tang, L., and Liu, H. (2009). Cross-validation. *Encyclopedia of Database Systems*, 5:532–538.

Reusens, P. and Croux, C. (2017). Sovereign credit rating determinants: A comparison before and after the European debt crisis. *Journal of Banking & Finance*, 77:108–121.

Rosenblatt, F. (1961). Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY.

Royston, P., White, I. R., et al. (2011). Multiple imputation by chained equations (MICE): implementation in Stata. *J Stat Softw*, 45(4):1–20.

Rufibach, K. (2010). Use of Brier score to assess binary predictions. *Journal of Clinical Epidemiology*, 63(8):938–939.

Sathya, R., Abraham, A., et al. (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2):34–38.

Savona, R. and Vezzoli, M. (2015). Fitting and forecasting sovereign defaults using multiple risk signals. *Oxford Bulletin of Economics and Statistics*, 77(1):66–92.

Schimmelpfennig, A., Roubini, N., and Manasse, P. (2003). *Predicting Sovereign Debt Crises*. International Monetary Fund.

Sechrest, L. J. (1993). *Free Banking: Theory, History, and a Laissez-Faire Model*. Ludwig von Mises Institute.

Sheffi, Y., Hall, R., and Daganzo, C. (1982). On the Estimation of the Multinomial Probit Model. *Transportation Research Part A: General*, 16(5-6):447–456.

Slade, E. and Naylor, M. G. (2020). A fair comparison of tree-based and parametric methods in multiple imputation by chained equations. *Statistics in medicine*, 39(8):1156–1166.

Smith, A. F. and Roberts, G. O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(1):3–23.

Stern, S. (1992). A method for smoothing simulated moments of discrete probabilities in multinomial probit models. *Econometrica: Journal of the Econometric Society*, pages 943–952.

Sun, D. and Benekohal, R. (2007). A study on logit-based route choice models for stochastic user equilibrium problem. *Advances in Transportation Studies*, (11):33–48.

Sun, Y., Wong, A. K., and Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):687–719.

Sutcliffe, P. (2003). The standards programme of IFAC's public sector committee. *Public Money & Management*, 23(1):29–36.

The Global Economy (2021). The Global Economy: Business and economic data for 200 countries. `https://www.theglobaleconomy.com/`, accessed May 2021.

The World Group (2021). World Bank Open Data. `https://data.worldbank.org/`, accessed May 2021.

Trading Economics (2021). South Africa's GDP. `https://tradingeconomics.com/`, accessed May 2021.

Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge university press.

Trevino, L. and Thomas, S. (2000). Systematic differences in the determinants of foreign currency sovereign ratings by rating agency. *University of Southampton - Department of Accounting and Management Science, Papers*.

Trinkle, B. S. and Baldwin, A. A. (2007). Interpretable credit model development via artificial neural networks. *Intelligent Systems in Accounting, Finance & Management: International Journal*, 15(3-4):123–147.

Trustorff, J.-H., Konrad, P. M., and Leker, J. (2011). Credit risk prediction using support vector machines. *Review of Quantitative Finance and Accounting*, 36(4):565–581.

Ulam, S. (1958). John von Neumann 1903-1957. *Bulletin of the American Mathematical Society*, 64(3):1–49.

Van Buuren, S., Boshuizen, H. C., and Knook, D. L. (1999). Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in Medicine*, 18(6):681–694.

Van Gestel, T., Baesens, B., Van Dijcke, P., Garcia, J., Suykens, J. A. K., and Vanthienen, J. (2006). A process model to develop an internal rating system: sovereign credit ratings. *Decision Support Systems*, 42(2):1131–1151.

Vovk, V. (2015). The fundamental nature of the log loss function. In *Fields of Logic and Computation II*, pages 307–318. Springer.

Wang, S.-C. (2003). Artificial neural network. In *Interdisciplinary Computing in Java Programming*, pages 81–100. Springer.

Wang, W. and Lu, Y. (2018). Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. In *IOP Conference Series: Materials Science and Engineering*, volume 324, page 012049. IOP Publishing.

Wasserman, L. (2000). Bayesian model selection and model averaging. *Journal of Mathematical Psychology*, 44(1):92–107.

Wei, L. and Zhichao, Z. (2009). An analytical framework of China foreign exchange reserves: Based on financial stability and appropriate scale. *Economic Research Journal*, 8:27–36.

Widrow, B. and Hoff, M. E. (1960). Adaptive Switching Circuits. Technical report, Stanford Univ Ca Stanford Electronics Labs.

Widrow, B. and Lehr, M. A. (1990). 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442.

Wiginton, J. C. (1980). A note on the comparison of logit and discriminant models of consumer credit behavior. *Journal of Financial and Quantitative Analysis*, 15(3):757–770.

Willmott, C. and Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(1):79–82.

Witzany, J. (2011). Exposure at default modeling with default intensities. *Available at SSRN 2489672*.

Xia, F., Allen, J., Balaprakash, P., Brettin, T., Garcia-Cardona, C., Clyde, A., Cohn, J., Doroshow, J., Duan, X., Dubinkina, V., et al. (2021). A cross-study analysis of drug response prediction in cancer cell lines. *arXiv preprint arXiv:2104.08961*.

Yadav, N., Yadav, A., and Kumar, M. (2015). History of neural networks. In *An Introduction to Neural Network Methods for Differential Equations*, pages 13–15. Springer.

Yegnanarayana, B. (2009). *Artificial neural networks*. PHI Learning Pvt. Ltd.

Yeh, I.-C. and Lien, C.-h. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480.

Zarantonello, L., Jedidi, K., and Schmitt, B. H. (2013). Functional and experiential routes to persuasion: An analysis of advertising in emerging versus developed markets. *International Journal of Research in Marketing*, 30(1):46–56.

Zhang, J. L. and Härdle, W. K. (2010). The Bayesian additive classification tree applied to credit risk modelling. *Computational Statistics & Data Analysis*, 54(5):1197–1205.

Zou, Q., Xie, S., Lin, Z., Wu, M., and Ju, Y. (2016). Finding the best classification threshold in imbalanced classification. *Big Data Research*, 5:2–8.

Appendix A: A Linear Regression Analysis of the Sovereignty Data

Linear regression is not an appropriate model for this data, since the dependent variable is binary. Diagnostic plots confirm that the data do not conform to the model assumptions. This was done to support the reason behind using machine learning techniques instead of linear regression.

Escanciano (2006) notes that linear regression modeling is appropriate if the data meet the following requirements (in no particular order):

1. Linearity: The relationship between the mean of Y and X is linear.

2. Homoscedasticty: The variance of residuals is the same for all values of X.

3. Independence: Observations are distinct from one another.

4. Normality: For any value of X, Y is normally distributed.

Escanciano (2006) and Faraway (2014) note that for data to meet the linear regression model requirements, regression diagnostic plots and various summary statistics should be considered. A plot of the residuals, $e_i = y_i - \hat{y}_i$, versus the fitted or predicted values is utilized to check the assumptions of linearity and homoscedasticity. If the model does not meet the assumptions, it is expected to have a very large residual. To assess the linearity assumption, the residuals are not supposed to be too far from 0; standardized values less than negative-two (-2) or greater than two (2) are deemed problematic. In addition when observing the plot, there should be no pattern in the residuals and the points should be equally spread around the $e_i = 0$ line. A QQ-plot can be used to assess the normality of the residuals; by comparing the residual to ideal normal observations. Normality can be assumed if observations lie close to the forty-five degrees (45^o) line of the QQ-plot. The scale location plot, plots the square rooted standardized residual versus the predicted value. This plot is useful for assessing homoscedasticity. Which is violated if patterns in the residuals are assessed. The fourth plot, Cook's distance measures the influence of each observation on the regression coefficient. The Cook's measure distance statistic is a measure for each observation, in turn, of the extent of change in model estimates when that particular observation is omitted. If there are one or more observations close to or larger than Cook's distance, it requires investigation. The assumption of the random sample and independent observation is not included in the diagnostic plots, and it is an assumption tested by the study design.

Since the diagnostic plots consist of four diagrams, for the purpose of identification, those diagrams will be referred to as "panels." The top left graph is panel 1, the top left graph is panel 2, the bottom left graph is panel 3, and the bottom right graph is panel 4.

A.1 Regression Analysis for the Data

A.1.1 Global

Table 2: Variable mean and variance for crisis and non crisis cases along with the p-value of both the t-test and the F-test at a 95%, where $\alpha = 0.05$, for the global MICE data set.

Variable	$\bar{x}_{crisis}$	S^2_{crisis}	$\bar{x}_{non-crisis}$	$S^2_{non-crisis}$	p-value	
	268		720		Pr(>F)	Pr(>|t|)
Total debt	70.6703	1 299.613	56.6909	1 098.426	0.0901	<0.0001
Global interest	57.5888	82 876.51	16.4168	758.4936	<0.0001	0.0201
Foreign Exch reserves	5.6974	112.3725	26.1095	2 696.322	<0.0001	<0.0001
Trade openness	57.3001	1 572.811	73.7336	3 907.697	<0.0001	<0.0001
Export growth	5.0139	109.8782	7.0209	80.4892	0.0016	0.0057
Current account	-2.1486	44.6636	-1.1991	45.4254	0.8799	0.0481
FDI	1.1441	3.3868	2.8208	16.8986	<0.0001	<0.0001
Real GDP growth	2.9187	41.4257	4.0169	17.9408	<0.0001	0.0099
Inflation	133.7883	753 128.5	11.5523	5 450.793	<0.0001	0.0221
REER overall	109.4766	2609.485	115.5561	2876.94	0.3475	0.1015
Gov expenditures	12.0886	26.2352	15.1314	24.7281	0.5475	<0.0001
National savings	20.3452	82.3345	24.3716	97.5203	0.1028	<0.0001
Domestic credit	29.8438	594.9926	57.8911	1 113.297	<0.0001	<0.0001
Bank assets	32.3334	583.3359	63.9891	1250.762	<0.0001	<0.0001

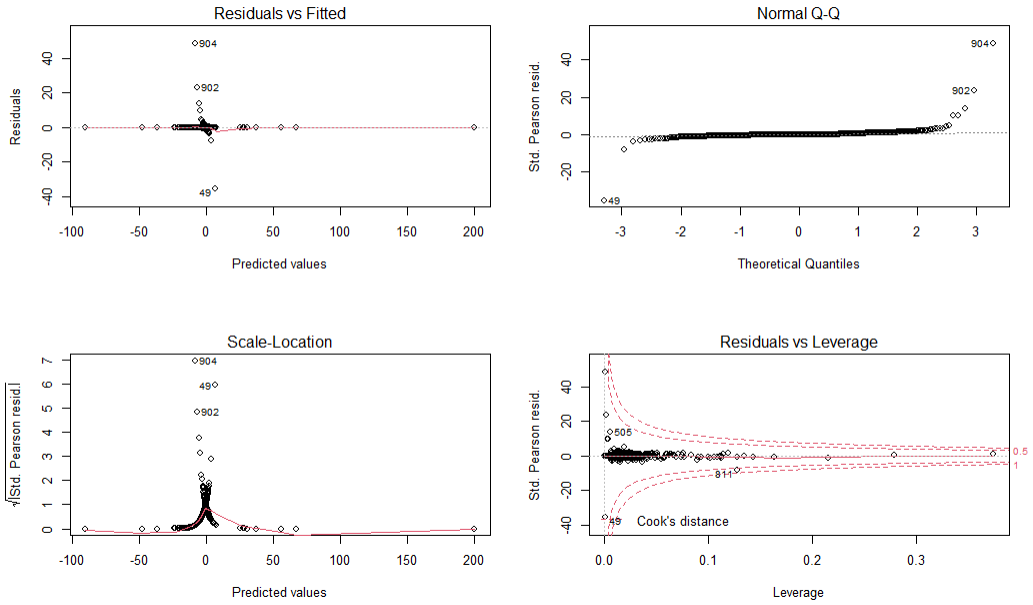

Figure 1: Diagnostics plots of the fitted GLM for the global MICE data.

The residual values exceed negative-two (-2) and two (2) respective of their distance from zero, and the points follow a clear pattern that looks to be a hyperbolic distribution, Figure 1 panel 1. It is clear that the values of the residuals run on the line but then change direction when moving along the line, creating an angle that is around forty-five degrees (45^o) of the trend line, QQ-plot panel 2, therefore normality can be assumed. The residuals have a pattern which implies that the linear regression line will have a high number of errors , see panel 3. Plot 4 indicates that residual values are close to Cook's distance but do not exceed Cook's distance. Therefore, further investigation is needed to further assess the data set. The regional model using the global MICE data set can not have a EWS model developed using regression since it failed to meet all requirements as stated above.

A.1.2 South and East Asia

Table 3: Variable mean and variance for crisis and non crisis cases along with the p-value of both the t-test and the F-test at a 95%, where $\alpha = 0.05$, for the South and East Asia MICE data set.

Variable	$\bar{x}_{crisis}$	S^2_{crisis}	$\bar{x}_{non-crisis}$	$S^2_{non-crisis}$	p-value	
	39		221		Pr(>F)	Pr(>|t|)
Total debt	95.1548	2 465.163	65.1512	1 781.767	0.1543	0.0007
Global interest	19.7662	96.6039	11.9418	29.2789	<0.0001	<0.0001
Foreign Exch reserves	10.1362	158.8064	34.7418	4 161.009	<0.0001	<0.0001
Trade openness	89.9861	4 276.408	108.2005	10 073.51	0.0020	0.1446
Export growth	7.0864	123.9955	10.8008	131.6212	0.8546	0.0585
Current account	-4.8040	154.8541	-0.9768	88.2053	0.0127	0.0703
FDI	1.2077	8.0909	3.3930	17.9852	0.0038	0.0001
Real GDP growth	1.3913	113.131	6.7038	14.8642	<0.0001	0.0032
Inflation	22.0480	595.6411	5.9541	66.0340	<0.0001	0.0002
REER overall	103.4546	312.4964	118.9922	2 175.185	<0.0001	0.0003
Gov expenditures	12.7755	30.9103	13.1686	21.1223	0.0947	0.675
National savings	22.4941	62.7370	30.1615	101.6861	0.0727	<0.0001
Domestic credit	54.2705	1 680.099	69.3894	1 146.927	0.0938	0.0325
Bank assets	58.6402	1 595.441	76.2071	1 484.385	0.7245	0.0129

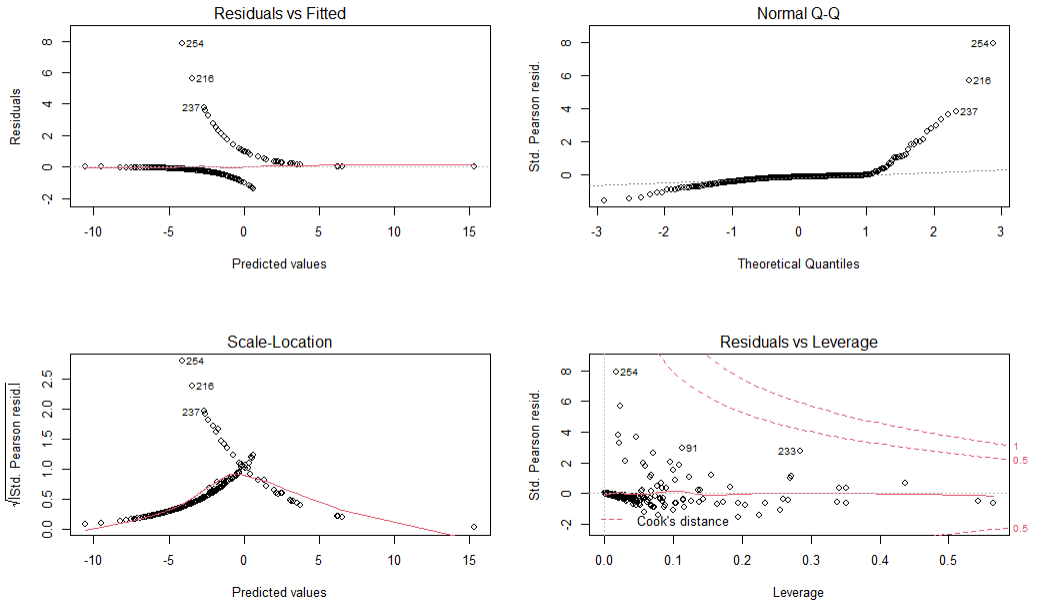

Figure 2: Diagnostics plots of the fitted GLM for the South and East Asia MICE data

The residual values exceed negative-two (-2) and two (2) respective of their distance from zero, and the points follow a clear pattern that looks to be a hyperbolic distribution, Figure 2 panel 1. It is clear

that the values of the residuals run on the line but then change direction when moving along the line, creating an angle that is around forty-five degrees (45^o) of the trend line, QQ-plot panel 2, therefore normality can be assumed. The residuals have a pattern which implies that the linear regression line will have a high number of errors , see panel 3. Plot 4 indicates values that are close to Cook's distance but do not exceed Cook's distance. Therefore, further investigation is needed to further assess the data set. The regional model using the South and East Asia Mice data set can not have a EWS model developed using regression since it failed to meet all requirements as stated above.

A.1.3 Latin America

Table 4: Variable mean and variance for crisis and non crisis cases along with the p-value of both the t-test and the F-test at a 95%, where $\alpha = 0.05$, for the Latin America MICE data set.

Variable	$\bar{x}_{crisis}$	S^2_{crisis}	$\bar{x}_{non-crisis}$	$S^2_{non-crisis}$	p-value	
	151		161		Pr(>F)	Pr(>|t|)
Total debt	3.3190	700.298	40.9601	466.6975	0.0117	0.3918
Global interest	181.4299	483 236.4	29.4835	422.767	<0.0001	0.0085
Foreign Exch reserves	4.4406	44.6661	12.0823	259.6315	<0.0001	<0.0001
Trade openness	52.3639	927.4293	59.3313	1 113.769	0.2577	0.0547
Export growth	6.2482	121.58	5.3371	80.3839	0.0101	0.4265
Current account	-2.2989	21.7330	-2.3862	22.7672	0.7748	0.8703
FDI	1.2087	3.9332	3.0373	6.4538	0.0023	<0.0001
Real GDP growth	2.6447	19.7994	3.4946	17.0843	0.3582	0.0823
Inflation	278.6461	1 472 545	13.9261	255.5777	<0.0001	0.0086
REER overall	98.8932	1 433.681	103.1489	795.47	0.0002	0.2648
Gov expenditures	10.6704	19.5401	12.2802	10.1194	<0.0001	0.0003
National savings	17.7024	37.2515	18.6901	34.7567	0.665	0.1478
Domestic credit	28.1702	380.1361	34.0119	424.1539	0.4983	0.0105
Bank assets	25.4229	220.0101	34.6362	337.6573	0.0083	<0.0001

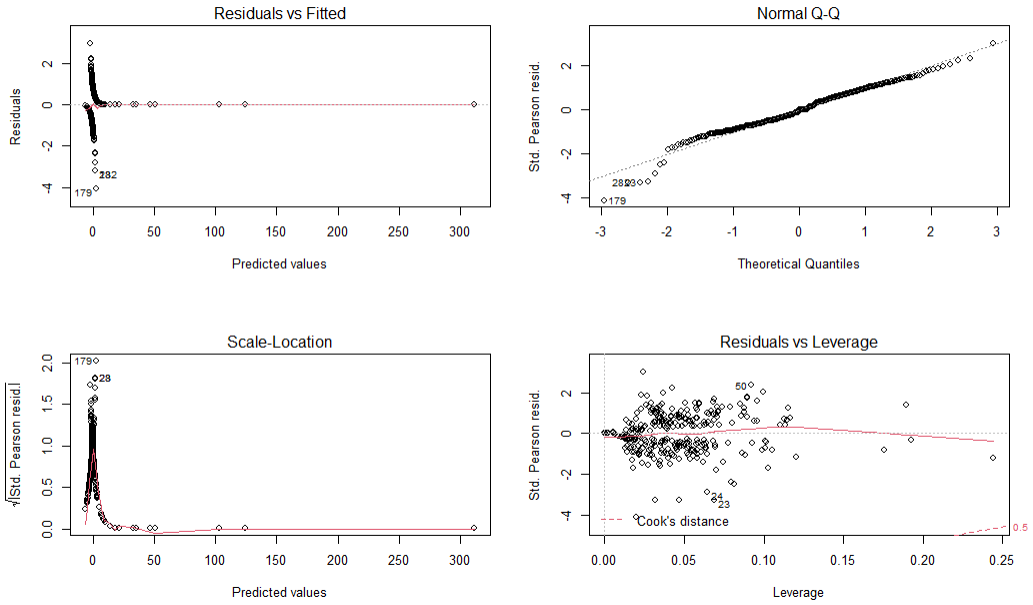

Figure 3: Diagnostics plots of the fitted GLM for the Latin America MICE data

The residual values exceed negative-two (-2) and two (2) respective of their distance from zero, and the points follow a clear pattern that looks to be a hyperbolic distribution, Figure 3 panel 1. It is clear that the values of the residuals run on the line but then change direction when moving along the line, creating an angle that is around forty-five degrees (45^o) of the trend line, QQ-plot panel 2, therefore normality can be assumed. The residuals have a pattern which implies that the linear regression line will have a high number of errors , see panel 3. Plot 4 indicates values that are close to Cook's distance but do not exceed Cook's distance. Therefore, further investigation is needed to further assess the data set. The regional model using the Latin America data set can not have a EWS model developed using regression since it failed to meet all requirements as stated above.

A.1.4 The Middle East and Africa

Table 5: Variable mean and variance for crisis and non crisis cases along with the p-value of both the t-test and the F-test at a 95%, where $\alpha = 0.05$, for the Middle East and Africa MICE data set.

Variable	$\bar{x}_{crisis}$	S^2_{crisis}	$\bar{x}_{non-crisis}$	$S^2_{non-crisis}$	p-value	
Sample size	76		106		Pr(>F)	Pr(>|t|)
Total debt	77.6835	544.2065	50.2742	449.9918	0.3671	<0.0001
Global interest	15.7929	25.1622	13.1843	26.0392	0.8835	0.0007
Foreign Exch reserves	1.8632	4.1977	7.8822	80.9694	<0.0001	<0.0001
Trade openness	50.9846	199.4035	54.5758	370.3034	0.0051	0.1483
Export growth	7.3625	113.9226	4.3376	60.0353	0.0024	0.0379
Current account	-1.6109	16.8412	0.1725	33.2134	0.0022	0.0157
FDI	0.8962	1.3977	1.2323	1.1904	0.4453	0.0532
Real GDP growth	2.8760	13.3147	3.2997	15.8405	0.4288	0.4587
Inflation	14.7322	252.7233	7.5359	26.0581	<0.0001	0.0002
REER overall	132.0403	3 729.291	161.7244	10 481.1	<0.0001	0.0155
Gov expenditures	13.5244	28.7008	14.2817	27.0358	0.7708	0.3435
National savings	25.4887	144.399	28.1555	229.4227	0.0353	0.1879
Domestic credit	21.1426	410.2567	50.8006	1 246.683	<0.0001	<0.0001
Bank assets	26.9133	398.4785	49.0207	406.0923	0.9399	<0.0001

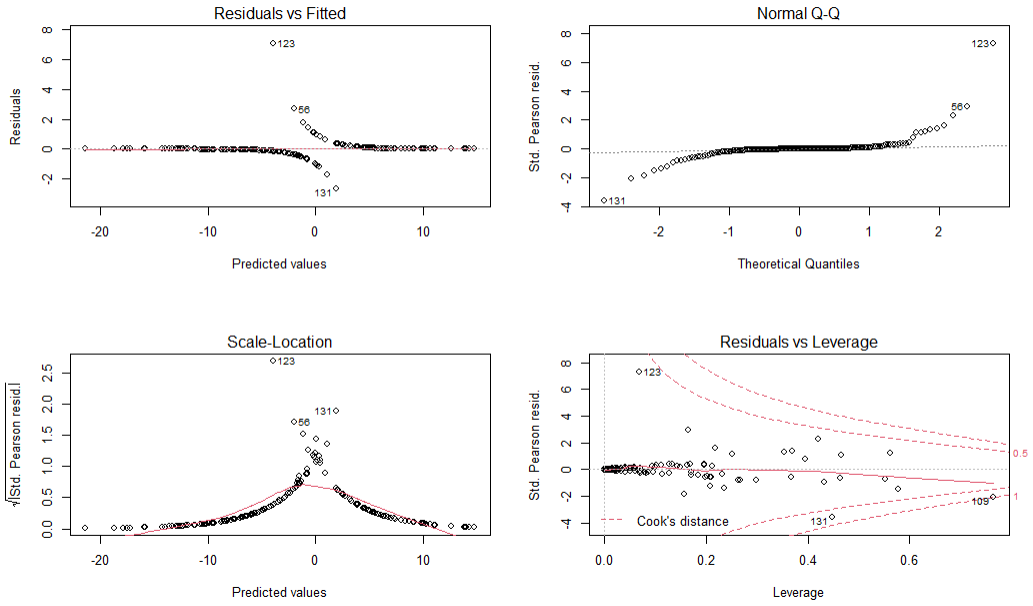

Figure 4: Diagnostics plots of the fitted GLM for the Middle East and Africa MICE data

The residual values exceed negative-two (-2) and two (2) respective of their distance from zero, and the points follow a clear pattern that looks to follow a be distribution, Figure 4 panel 1. It is clear that

the values of the residuals run on the line but then change direction when moving along both directions of the line, creating an angle that is around forty-five degrees (45^o) of the trend line, QQ-plot panel 2, therefore normality can be assumed. The residuals have a pattern which implies that the linear regression line will have a high number of errors , see panel 3. Plot 4 indicates values that are not only close to Cook's distance but do exceed Cook's distance. Therefore, further investigation is needed to further assess the data set. The regional model using the Middle East and Africa MICE data set can not have a EWS model developed using regression since it failed to meet all requirements as stated above.

A.1.5 Western Europe

Table 6: Variable mean and variance for crisis and non crisis cases along with the p-value of both the t-test and the F-test at a 95%, where $\alpha = 0.05$, for the Western Europe MICE data set.

Variable	$\bar{x}_{crisis}$	S^2_{crisis}	$\bar{x}_{non-crisis}$	$S^2_{non-crisis}$	p-value	
Sample size	4		230		Pr(>F)	Pr(>|t|)
Total debt	125.0947	560.2516	65.985	959.3541	0.7479	0.0139
Global interest	15.6775	64.2254	12.2044	40.1827	0.3812	0.4508
Foreign Exch reserves	19.0400	42.6842	34.7829	952.875	0.0252	0.0069
Trade openness	97.8336	859.5959	66.4072	1 081.242	0.9955	0.1203
Export growth	4.0881	23.8952	6.0596	33.1374	0.9194	0.4812
Current account	4.0865	1.6934	-1.0811	12.4494	0.1231	0.002
FDI	3.4204	3.8854	3.1070	36.2499	0.0883	0.7825
Real GDP growth	1.7338	5.5805	2.6862	5.2475	0.7308	0.4806
Inflation	5.025	23.5491	6.080	32.4113	0.9259	0.6952
REER overall	95.8127	210.0651	102.1545	245.9441	0.9313	0.4477
Gov expenditures	19.1750	15.0457	19.2801	10.6484	0.4795	0.9604
National savings	26.8078	2.9807	22.3913	12.9454	0.2495	0.0112
Domestic credit	62.6225	512.1578	66.7067	815.8977	0.8044	0.7442
Bank assets	118.1175	1436.66	80.7943	803.839	0.3007	0.1431

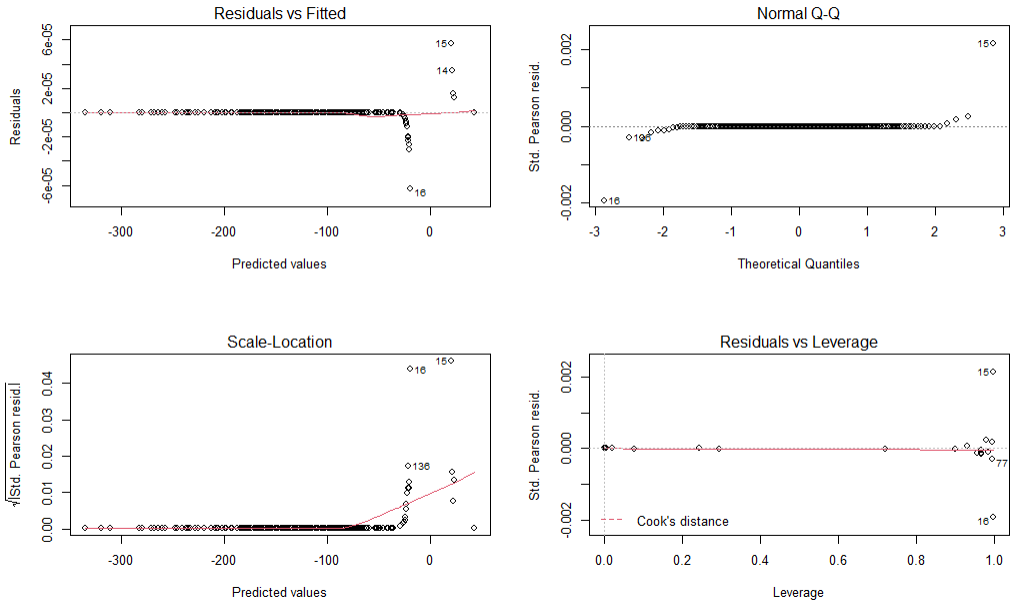

Figure 5: Diagnostics plots of the fitted GLM for the Western Europe MICE data

The residual values do not exceed negative-two (-2) and two (2) respective of their distance from zero, and the points follow a clear pattern that looks to be a hyperbolic distribution, Figure 5 panel 1. It is clear that the values of the residuals run on the line but then change direction when moving along both ends of the line, creating an angle that is around forty-five degrees (45^o) from the trend line, QQ-plot panel 2, therefore normality can be assumed. The residuals have a pattern which implies that the linear regression line will have a high number of errors , see panel 3. Plot 4 clearly indicates values that are not only close to Cook's distance but do exceed Cook's distance. Therefore, further investigation is needed to further assess the data set. The regional model using the Western European MICE data set can not have a EWS model developed using regression since it failed to meet all requirements as stated above.

A.1.6 Conclusion

To further conclude, as seen from above, the data used to derive the models does not fully satisfy regression diagnostic requirements, making the data unsuitable for work since the dependent variable is binary. Therefore, models that are more suitable for binary data are going to be used. More information on why regression does not work on binary data is seen in Cox and Snell (2018).

Appendix B: Cross-Validation Classification

This chapter is done in addition to Chapter 5. Here, the same data sets and classifiers are used; the only difference is that cross-validation is considered in the model development stages. As per chapter 5, while developing these classifiers, multicolinearity has been ignored. As a result, interpretation of the significance of the independent variables in each model should be treated with caution.

B.1 5-fold Cross Validation: Logistic Regression

The 5-fold cross-validation for different models is considered as follows as detailed by Fushiki (2011); the initial step is to partition the data into training and testing data sets. This separation is done based on the Dawood et al. (2017) study. After a training model is specified and the fold number used for the model development process is chosen, a package called "caret is used to initially" build the model. For this, a function in caret called "train-control" will aid in the specification of the number of folds. A model is then specified with the training set. A summary statistics is then observed, which results from the average of the estimates of the variables as a result of k-fold cross-validation outputs. The variable kappa is a classification index that, unlike accuracy, considers and accounts for the baseline probability from the null model. The model summary results from an average of the observed estimates, and both in-sample and out-of-sample statistics are considered in the products, as well as all metrics considered for performance evaluation in the previous chapters.The following are the different buckets for the ROC AUC: poor 0% to 50%, suggest no discrimination and therefore poor, 51%-69% slighlty accaptable which is considered moderately good, 70% to 80% is acceptable and therefore noted as good, 81% to 90% is excellent and anoything greater than 90% is outstanding (Mandrekar, 2010). The main metric used to assess model prediction accuracy will be the ROC AUC, therefore conclusions will be based on the ROC AUC results

B.1.1 5-Fold Cross-Validation Case-Wise Deletion Logit Models

Table 7: 5-fold cross-validation binary logit model summary when applied to the global case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	1.1931	1.0632	0.2617		Insignificant
Global interest	-0.0640	0.0152	<0.001	92.124	Significant
Foreign Exch reserves	0.0548	0.0165	0.0008	72.951	Significant
Trade openness	-0.0067	0.0039	0.0883	37.164	Insignificant
Export growth	-0.0064	0.0155	0.6793	8.614	Insignificant
Current account	0.0288	0.0361	0.4253	17.103	Insignificant
FDI	0.3364	0.0901	0.0002	81.984	Significant
Real GDP growth	-0.0458	0.0424	0.2806	23.332	Insignificant
Inflation	-0.0752	0.0165	<0.001	100.000	Significant
REER overall	0.0013	0.0031	0.6570	9.289	Insignificant
Gov expenditures	0.1470	0.0422	0.0005	76.454	Significant
National savings	0.0208	0.0209	0.3193	21.496	Insignificant
Domestic credit	-0.0002	0.0102	0.9809	0	Insignificant
Bank assets	-0.0130	0.0127	0.3056	22.127	Insignificant
out of sample correctness	0.9008				
In sample correctness	0.8769				
Deviance	535.17				
Sensitivity	0.9732				
Specificity	0				
Precision	0.9237				
Recall	0.9732				
PR AUC	0.5446				
ROC AUC	0.4866				
Brier Score	0.0991				

From analysing the global data set considering case-wise deletion; the top five (5) important variables when developing an EWS for this data set show to be inflation, global interest, FDI , government expenditure, and FX reserves. Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 48.66%, see Table 7.

Table 8: 5-fold cross-validation binary logit model summary when applied to the South and East Asia case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-9.9710	6.0672	0.1003		Insignificant
Global interest	-0.0776	0.1801	0.6666	7.185	Insignificant
Foreign Exch reserves	0.0859	0.0464	0.0643	66.383	Insignificant
Trade openness	0.0043	0.0149	0.7707	1.371	Insignificant
Export growth	0.0543	0.0395	0.1697	46.485	Insignificant
Current account	0.0587	0.1149	0.6095	10.519	Insignificant
FDI	0.8517	0.3206	0.0079	100.000	Significant
Real GDP growth	0.0315	0.1219	0.7960	0.000	Insignificant
Inflation	-0.0698	0.1045	0.5040	17.082	Insignificant
REER overall	0.0536	0.0262	0.0408	74.505	Significant
Gov expenditures	0.1681	0.1265	0.1841	44.607	Insignificant
National savings	0.2231	0.0945	0.0182	87.673	Significant
Domestic credit	0.0399	0.0487	0.4116	23.461	Insignificant
Bank assets	-0.0993	0.0515	0.0542	69.506	Insignificant
out of sample correctness	0.9412				
In sample correctness	0.925				
Deviance	135.27				
Sensitivity	1				
Specificity	0.6667				
Precision	0.9333				
Recall	1				
PR AUC	0.5354				
ROC AUC	0.8333				
Brier Score	0.0588				

From analysing the South and East Asia data set considering case-wise deletion; the top five (5) important variables when developing an EWS for this data set show to be FDI, national savings, REER overall, bank assets and FX reserves. Based on the metrics the model is excellent at predicting sovereign default since it generated a ROC AUC value of 83.33%, see Table 8.

Table 9: 5-fold cross-validation binary logit model summary when applied to the Latin America case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	5.9459	3.2940	0.0710		Insignificant
Global interest	-0.0764	0.0226	0.0007	93.17	Significant
Foreign Exch reserves	0.0859	0.0376	0.0223	62.59	Insignificant
Trade openness	-0.0035	0.0125	0.7791	6.42	Insignificant
Export growth	-0.0684	0.0364	0.0602	51.22	Insignificant
Current account	0.1728	0.0746	0.0205	63.47	Significant
FDI	0.2176	0.1283	0.0900	46.06	Insignificant
Real GDP growth	-0.1323	0.0780	0.0898	46.09	Insignificant
Inflation	-0.0862	0.0238	0.0002	100.00	Significant
REER overall	0.0007	0.0154	0.9590	0.00	Insignificant
Gov expenditures	0.0767	0.0855	0.3695	23.71	Insignificant
National savings	-0.0884	0.0720	0.2194	32.97	Insignificant
Domestic credit	0.0432	0.0388	0.2650	29.80	Insignificant
Bank assets	-0.0609	0.0472	0.1967	34.73	Insignificant
out of sample correctness	0.9434				
In sample correctness	0.7944				
Deviance	227.73				
Sensitivity	0.9804				
Specificity	0				
Precision	0.9615				
Recall	0.9803				
PR AUC	0.5187				
ROC AUC	0.4902				
Brier Score	0.0566				

From analysing the Latin America data set considering case-wise deletion; the top five (5) important variables when developing an EWS for this data set show to be inflation, global interest, current account, FX reserves, and export growth. Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 49.02%, see Table 9.

Table 10: 5-fold cross-validation binary logit model summary when applied to the Middle East and Africa case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	22.2753	47.5996	0.640		Insignificant
Global interest	-1.6163	1.9969	0.418	12.797	Insignificant
Foreign Exch reserves	19.6494	21.7680	0.367	54.588	Insignificant
Trade openness	-4.6160	5.4341	0.396	30.743	Insignificant
Export growth	1.0030	1.1601	0.387	37.498	Insignificant
Current account	-4.4610	4.7947	0.352	67.012	Insignificant
FDI	8.7063	10.8765	0.423	8.796	Insignificant
Real GDP growth	4.0582	4.9227	0.410	19.507	Insignificant
Inflation	-0.7654	0.9802	0.435	0.000	Insignificant
REER overall	-0.1065	0.1135	0.348	70.685	Insignificant
Gov expenditures	8.3547	9.1516	0.361	59.181	Insignificant
National savings	1.0350	1.0308	0.315	100.000	Insignificant
Domestic credit	-0.6904	0.7581	0.362	58.223	Insignificant
Bank assets	1.3920	1.5966	0.383	40.805	Insignificant
out of sample correctness	0.9697				
In sample correctness	0.9770				
Deviance	91.326				
Sensitivity	1				
Specificity	0.9166				
Precision	0.9545				
Recall	1				
PR AUC	0.5124				
ROC AUC	0.9583				
Brier Score	0.0303				

From analysing the Middle East and Africa data set considering case-wise deletion; the top five (5) important variables when developing an EWS for this data set show to be inflation, global interest, FDI, , government expenditure, and FX reserves. Based on the metrics the model is outstanding at predicting sovereign default since it generated a ROC AUC value of 95.83%, see Table 10.

Table 11: 5-fold cross-validation binary logit model summary when applied to the Western Europe case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$-2.372e^{2}$	$1.861e^{6}$	1		Insignificant
Global interest	$5.267e^{-1}$	$1.747e^{4}$	1	2.4490	Insignificant
Foreign Exch reserves	-2.261	$3.137e^{3}$	0.999	100.0000	Insignificant
Trade openness	-1.727	$5.220e^{3}$	1	44.9031	Insignificant
Export growth	-5.375	$1.565e^{4}$	1	46.6902	Insignificant
Current account	6.744	$2.195e^{4}$	1	41.5723	Insignificant
FDI	$8.828e^{-1}$	$5.326e^{4}$	1	0.5313	Insignificant
Real GDP growth	$2.457e^{1}$	$5.995e^{4}$	1	56.0668	Insignificant
Inflation	3.249	$3.140e^{4}$	1	12.8012	Insignificant
REER overall	3.706	$1.381e^{4}$	1	36.0768	Insignificant
Gov expenditures	$6.687e^{-1}$	$5.220e^{4}$	1	0.0000	Insignificant
National savings	-2.099	$5.237e^{4}$	1	3.8499	Insignificant
Domestic credit	-3.936	$1.037e^{4}$	1	51.7923	Insignificant
Bank assets	4.957	$1.273e^{4}$	1	53.1706	Insignificant
out of sample correctness	0.6315				
In sample correctness	1				
Deviance	$5.6574e$				
Sensitivity	1				
Specificity	0				
Precision	Nan				
Recall	Nan				
PR AUC	Nan				
ROC AUC	Nan				
Brier Score	0.3684				

From analysing the Western European data set considering case-wise deletion; the top five (5) important variables when developing an EWS for this data set show to be FX reserves, real GDP growth, bak assets, domestic credit, and export growth. Based on the metrics the model did not generate a ROC AUC score since there were not enough defaults for the model to generate scores, see Table 11.

B.1.2 5-Fold Cross-Validation Mean Imputation Logit Models

Table 12: 5-fold cross-validation binary logit model summary when applied to the global mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-0.7522	0.7077	0.2878		Insignificant
Total debt	-0.0251	0.0050	<0.001	84.834	Significant
Global interest	-0.0175	0.0075	0.0199	38.747	Significant
Foreign Exch reserves	0.0954	0.0162	<0.001	100.000	Significant
Trade openness	-0.0076	0.0029	0.0094	43.324	Significant
Export growth	0.0007	0.0093	0.9398	0.000	Insignificant
Current account	-0.0299	0.0177	0.0907	27.804	Significant
FDI	0.1599	0.0541	0.0031	49.518	Significant
Real GDP growth	0.0037	0.0200	0.8514	1.923	Insignificant
Inflation	-0.0365	0.0073	<0.001	83.773	Significant
REER overall	0.0048	0.0024	0.0472	32.835	Significant
Gov expenditures	0.0963	0.0255	0.0001	63.547	Significant
National savings	0.0416	0.0133	0.0018	52.385	Significant
Domestic credit	0.0149	0.0070	0.0334	35.289	Significant
Bank assets	0.0058	0.0068	0.3905	13.475	Insignificant
out of sample correctness	0.9361				
In sample correctness	0.8279				
Deviance	1158.91				
Sensitivity	0.996				
Specificity	0				
Precision	0.9396				
Recall	0.996				
PR AUC	0.5615				
ROC AUC	0.498				
Brier Score	0.0639				

From analysing the global data set considering mean imputation; the top five (5) important variables when developing an EWS for this data set show to be FX reserves, total debt, inflation, government expenditure, and national savings. Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 49.8%, see Table 12.

Table 13: 5-fold cross-validation binary logit model summary when applied to the South and East Asia mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-4.7961	3.2227	0.1367		Insignificant
Total debt	-0.0438	0.0163	0.0074	89.829	Significant
Global interest	-0.0473	0.0515	0.3576	27.382	Insignificant
Foreign Exch reserves	0.0890	0.0396	0.0248	74.414	Significant
Trade openness	-0.0073	0.0073	0.3155	30.355	Insignificant
Export growth	0.0564	0.0215	0.0089	87.535	Significant
Current account	-0.0329	0.0422	0.4352	22.419	Insignificant
FDI	0.6625	0.2234	0.0030	100.000	Significant
Real GDP growth	-0.0065	0.04404	0.8816	0.000	Insignificant
Inflation	-0.0710	0.0288	0.0137	82.191	Significant
REER overall	0.0555	0.0250	0.0264	73.553	Significant
Gov expenditures	0.3214	0.1084	0.0030	99.983	Significant
National savings	0.0539	0.0476	0.2579	34.884	Insignificant
Domestic credit	-0.0458	0.0200	0.0224	75.757	Significant
Bank assets	0.0055	0.0138	0.6889	8.928	Insignificant
out of sample correctness	1				
In sample correctness	0.9153				
Deviance	219.81				
Sensitivity	1				
Specificity	1				
Precision	1				
Recall	1				
PR AUC	0.5				
ROC AUC	1				
Brier Score	0				

From analysing the South and East Asia data set considering mean imputation; the top five (5) important variables when developing an EWS for this data set show to be FDI, government expenditure, total debt, export growth and inflation. Based on the metrics the model is outstanding at predicting sovereign default since it generated a ROC AUC value of 100%, see Table 13.

Table 14: 5-fold cross-validation binary logit model summary when applied to the Latin America mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-3.0918	1.4366	0.0313		Significant
Total debt	0.0001	0.0101	0.9911	0.000	Insignificant
Global interest	-0.0189	0.0097	0.0516	52.306	Insignificant
Foreign Exch reserves	0.0873	0.0238	0.0002	98.649	Significant
Trade openness	-0.0046	0.0068	0.4972	18.046	Insignificant
Export growth	-0.0125	0.0148	0.3981	22.538	Insignificant
Current account	-0.0101	0.0420	0.8097	6.209	Insignificant
FDI	0.2283	0.0797	0.0041	77.083	Significant
Real GDP growth	-0.0240	0.0391	0.5385	16.322	Insignificant
Inflation	-0.0300	0.0080	0.0002	100.000	Significant
REER overall	0.0222	0.0068	0.0010	88.295	Significant
Gov expenditures	0.0463	0.0446	0.2990	27.768	Insignificant
National savings	0.0749	0.0333	0.02440	60.528	Significant
Domestic credit	0.0488	0.0199	0.0143	65.889	Significant
Bank assets	-0.0554	0.0251	0.0275	59.258	Significant
out of sample correctness	0.9643				
In sample correctness	0.7433				
Deviance	432.20				
Sensitivity	0.9878				
Specificity	0				
Precision	0.9759				
Recall	0.9878				
PR AUC	0.5131				
ROC AUC	0.4939				
Brier Score	0.0357				

From analysing the Latin America data set considering mean imputation; the top five (5) important variables when developing an EWS for this data set show to be inflation, FX reserves, REER overall, FDI, and domestic credit. Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 49.39%, see Table 14.

Table 15: 5-fold cross-validation binary logit model summary when applied to the Middle East and Africa mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-7.1731	4.6532	0.1231		Insignificant
Total debt	0.0320	0.0396	0.4195	17.476	Insignificant
Global interest	-0.0953	0.0993	0.3369	21.643	Insignificant
Foreign Exch reserves	1.0874	0.2833	0.0001	100.000	Significant
Trade openness	-0.0184	0.0247	0.4569	15.749	Insignificant
Export growth	-0.0561	0.0352	0.1111	38.873	Insignificant
Current account	-0.1692	0.1138	0.1370	35.984	Insignificant
FDI	1.1114	0.4391	0.0113	64.413	Significant
Real GDP growth	-0.0119	0.0721	0.8685	0.000	Insignificant
Inflation	-0.1600	0.0574	0.0053	71.334	Significant
REER overall	0.0128	0.0058	0.0267	55.803	Significant
Gov expenditures	0.0471	0.168	0.7797	3.110	Insignificant
National savings	0.0255	0.0507	0.61486	9.195	Insignificant
Domestic credit	0.0762	0.0230	0.0009	85.667	Significant
Bank assets	-0.036	0.0359	0.3097	23.153	Insignificant
out of sample correctness	0.8776				
In sample correctness	0.9231				
Deviance	247.338				
Sensitivity	0.875				
Specificity	1				
Precision	1				
Recall	0.875				
PR AUC	0.4359				
ROC AUC	0.9375				
Brier Score	0.1224				

From analysing the Middle East and Africa data set considering mean imputation; the top five (5) important variables when developing an EWS for this data set show to be FX reserves, domestic credit, inflation, FDI, and REER overall. Based on the metrics the model is outstanding at predicting sovereign default since it generated a ROC AUC value of 93.75%, see Table 15.

Table 16: 5-fold cross-validation binary logit model summary when applied to the Western Europe mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$3.611e^{2}$	$5.724e^{5}$	0.999		Insignificant
Total debt	-2.603	$1.247e^{3}$	0.998	93.5263	Insignificant
Global interest	-5.903	$6.572e^{3}$	0.999	37.8024	Insignificant
Foreign Exch reserves	1.952	$2.390e^{3}$	0.999	33.9751	Insignificant
Trade openness	$8.257e^{-1}$	$2.329e^{3}$	1	12.3311	Insignificant
Export growth	$9.582e^{-1}$	$1.048e^{4}$	1	0.0000	Insignificant
Current account	$-1.509e^{1}$	$9.977e^{3}$	0.999	66.5968	Insignificant
FDI	6.829	$2.654e^{4}$	1	7.7722	Insignificant
Real GDP growth	-7.065	$2.371e^{4}$	1	9.6805	Insignificant
Inflation	$8.041e^{-1}$	$7.198e^{3}$	1	0.9498	Insignificant
REER overall	1.041	$4.848e^{3}$	1	5.7795	Insignificant
Gov expenditures	3.575	$2.395e^{4}$	1	2.7098	Insignificant
National savings	-3.997	$9.109e^{3}$	1	16.2773	Insignificant
Domestic credit	-4.461	$2.004e^{3}$	0.998	100.000	Insignificant
Bank assets	1.337	$1.538e^{3}$	0.999	36.4684	Insignificant
out of sample correctness	0.7619				
In sample correctness	1				
Deviance	$4.0483e^{1}$				
Sensitivity	0.9				
Specificity	0.2307				
Precision	0.8182				
Recall	0.9				
PR AUC	0.5462				
ROC AUC	0.5653				
Brier Score	0.2381				

From analysing the Western European data set considering mean imputation; the top five (5) important variables when developing an EWS for this data set show to be domestic credit, total debt, current account, global interest, and bank assets. Based on the metrics the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 56.53%, see Table 16.

B.1.3 5-Fold Cross-Validation MICE Logit Models

Table 17: 5-fold cross-validation binary logit model summary when applied to the global MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-2.5281	0.4840	<0.001		Significant
Total debt	-0.0372	0.0043	<0.001	100.000	Significant
Global interest	-0.0021	0.0021	0.3158	9.734	Insignificant
Foreign Exch reserves	0.0524	0.0122	<0.001	48.530	Significant
Trade openness	0.0017	0.0030	0.5763	4.470	Insignificant
Export growth	-0.0017	0.0096	0.8562	0.000	Insignificant
Current account	-0.0132	0.0210	0.5277	5.332	Insignificant
FDI	0.1597	0.0552	0.0038	32.087	Significant
Real GDP growth	0.0279	0.0208	0.1806	13.711	Insignificant
Inflation	-0.0021	0.0013	0.1035	17.137	Insignificant
REER overall	-0.0013	0.0020	0.5141	5.580	Insignificant
Gov expenditures	0.2043	0.0272	<0.0001	86.532	Significant
National savings	0.0635	0.0124	<0.001	58.455	Significant
Domestic credit	0.0129	0.0078	0.1011	17.275	Insignificant
Bank assets	0.0346	0.0084	<0.001	46.455	Significant
out of sample correctness	0.9436				
In sample correctness	0.8684				
Deviance	1144.97				
Sensitivity	0.996				
Specificity	0.125				
Precision	0.9467				
Recall	0.996				
PR AUC	0.5498				
ROC AUC	0.5605				
Brier Score	0.0563				

From analysing the global data set considering MICE; the top five (5) important variables when developing an EWS for this data set show to be total debt, government, national savings, FX reserves, and bank assets. Based on the metrics the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 56.05%, see Table 17.

Table 18: 5-fold cross-validation binary logit model summary when applied to the South and East Asia MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	6.9514	2.6652	0.0091		Significant
Total debt	-0.0724	0.0182	<0.001	100.000	Significant
Global interest	-0.2328	0.0697	0.0008	80.442	Significant
Foreign Exch reserves	-0.0114	0.0034	0.0010	78.922	Significant
Trade openness	-0.0056	0.0074	0.4477	0.000	Insignificant
Export growth	0.1040	0.0314	0.0009	79.396	Significant
Current account	-0.0525	0.0611	0.3900	3.129	Insignificant
FDI	0.5243	0.2170	0.0156	51.710	Significant
Real GDP growth	-0.0731	0.0566	0.1967	16.592	Insignificant
Inflation	-0.0873	0.0359	0.0151	52.128	Significant
REER overall	-0.0256	0.0126	0.0428	39.502	Significant
Gov expenditures	0.4112	0.1275	0.0012	76.888	Significant
National savings	0.0557	0.0527	0.2902	9.310	Insignificant
Domestic credit	-0.0798	0.0300	0.0078	59.304	Significant
Bank assets	0.0612	0.0264	0.0205	48.556	Significant
out of sample correctness	0.9605				
In sample correctness	0.8923				
Deviance	223.25				
Sensitivity	1				
Specificity	0.5				
Precision	0.9589				
Recall	1				
PR AUC	0.5325				
ROC AUC	0.75				
Brier Score	0.0394				

From analysing the South and East Asia data set considering MICE; the top five (5) important variables when developing an EWS for this data set show to be total debt, global interest, export growth, FX reseres, and government expenditure. Based on the metrics the model is good at predicting sovereign default since it generated a ROC AUC value of 75%, see Table 18.

Table 19: 5-fold cross-validation binary logit model summary when applied to the Latin America MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-0.8646	1.4089	0.5394		Insignificant
Total debt	-0.0433	0.0095	<0.001	100.000	Significant
Global interest	-0.0242	0.0086	0.0048	56.657	Significant
Foreign Exch reserves	0.0298	0.0191	0.1193	25.001	Insignificant
Trade openness	-0.0097	0.0066	0.1459	22.392	Insignificant
Export growth	-0.0134	0.0160	0.4011	6.911	Insignificant
Current account	0.1775	0.0493	0.0003	76.292	Significant
FDI	0.2308	0.0886	0.0092	51.337	Significant
Real GDP growth	-0.0319	0.0407	0.4330	5.511	Insignificant
Inflation	-0.0192	0.0078	0.0138	47.733	Significant
REER overall	0.0282	0.0066	<0.001	92.197	Significant
Gov expenditures	0.0873	0.0465	0.0608	32.984	Insignificant
National savings	-0.0196	0.0347	0.5719	0.000	Insignificant
Domestic credit	-0.0494	0.0232	0.0336	39.275	Significant
Bank assets	0.0835	0.0312	0.0075	53.093	Significant
out of sample correctness	0.7980				
In sample correctness	0.7722				
Deviance	432.47				
Sensitivity	1				
Specificity	0				
Precision	Nan				
Recall	Nan				
PR AUC	Nan				
ROC AUC	0.5				
Brier Score	0.0238				

From analysing the Latin America data set considering MICE; the top five (5) important variables when developing an EWS for this data set show to be total debt, REER overal, current account, global intrest, and bank assets . Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 50%, see Table 19.

Table 20: 5-fold cross-validation binary logit model summary when applied to the Middle East and Africa MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-5.3873	10.1338	0.5949		Insignificant
Total debt	-0.4085	0.1365	0.0027	93.14	Insignificant
Global interest	-0.1073	0.1765	0.5430	0.00	Insignificant
Foreign Exch reserves	1.7210	0.5559	0.0019	97.17	Significant
Trade openness	-0.3320	0.1132	0.0033	90.79	Significant
Export growth	0.1949	0.0964	0.0432	55.21	Significant
Current account	-1.3107	0.4137	0.0015	100.00	Significant
FDI	3.4314	1.1267	0.0023	95.20	Significant
Real GDP growth	0.2946	0.1580	0.0622	49.07	Insignificant
Inflation	-0.6399	0.2394	0.0075	80.64	Significant
REER overall	-0.0559	0.0206	0.0067	82.12	Significant
Gov expenditures	1.7170	0.5799	0.0030	91.90	Significant
National savings	0.8051	0.2675	0.0026	93.78	Significant
Domestic credit	0.0859	0.0512	0.0935	41.75	Insignificant
Bank assets	0.1250	0.0987	0.2055	25.69	Insignificant
out of sample correctness	0.9591				
In sample correctness	0.9395				
Deviance	247.981				
Sensitivity	0.9791				
Specificity	0				
Precision	0.9791				
Recall	0.9791				
PR AUC	0.5				
ROC AUC	0.4895				
Brier Score	0.0408				

From analysing the Middle East and Africa data set considering MICE; the top five (5) important variables when developing an EWS for this data set show to be current account, FX reserves, FDI, national savings, and total debt . Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 48.95%, see Table 20.

Table 21: 5-fold cross-validation binary logit model summary when applied to the Western Europe MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$3.215e^{2}$	$4.769e^{5}$	1		Insignificant
Total debt	$-2.192e^{-1}$	$1.866e^{3}$	1	11.019	Insignificant
Global interest	-9.772	$1.319e^{4}$	1	78.384	Insignificant
Foreign Exch reserves	$2.125e^{-1}$	$1.333e^{3}$	1	15.558	Insignificant
Trade openness	$-9.138e^{-1}$	$9.711e^{2}$	0.999	100.000	Insignificant
Export growth	3.167	$8.114e^{3}$	1	40.498	Insignificant
Current account	1.998	$9.234e^{3}$	1	21.710	Insignificant
FDI	1.497	$5.562e^{3}$	1	27.413	Insignificant
Real GDP growth	$-8.706e^{-1}$	$1.868e^{4}$	1	3.368	Insignificant
Inflation	8.986	$1.433e^{4}$	0.999	66.067	Insignificant
REER overall	$3.543e^{-1}$	$3.900e^{3}$	1	8.147	Insignificant
Gov expenditures	$-2.145e^{-1}$	$1.390e^{4}$	1	0.000	Insignificant
National savings	-3.971	$1.519e^{4}$	1	26.569	Insignificant
Domestic credit	$9.877e^{-1}$	$1.334e^{3}$	0.999	78.339	Insignificant
Bank assets	$5.338e^{-2}$	$9.365e^{2}$	1	4.491	Insignificant
out of sample correctness	0.7301				
In sample correctness	1				
Deviance	$4.0483e^{1}$				
Sensitivity	0.88				
Specificity	0.1538				
Precision	0.8				
Recall	0.88				
PR AUC	0.5462				
ROC AUC	0.5169				
Brier Score	0.2698				

From analysing the Western European data set considering MICE;the top five (5) important variables when developing an EWS for this data set show to be trade openness, global interest, domestic credit, inflation, and export growth. Based on the metrics the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 51.69%, see Table 21.

B.1.4 Performance comparison

Table 22: Case-wise deletion 5-fold cross-validation logit model summary.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
Out of sample correctness	0.9008	0.9412	0.9434	0.9697	0.6315
In sample correctness	0.8769	0.925	0.7944	0.9770	1
Deviance	535.17	135.27	227.73	91.326	$5.6574e$
Sensitivity	0.9732	1	0.9804	1	1
Specificity	0	0.6667	0	0.9166	0
Precision	0.9237	0.9333	0.9615	0.9545	Nan
Recall	0.9732	1	0.9803	1	Nan
PR AUC	0.5446	0.5354	0.5187	0.5124	Nan
ROC AUC	0.4866	0.8333	0.4902	0.9583	Nan
Brier Score	0.0991	0.0588	0.0566	0.0303	0.3684
sample size (n)	569	160	180	87	143

From the regional and global model comparison in Table 22, what can be concluded is that with the exception of the Western European, the regional models performed better than the global models, this is concluded by observing the ROC AUC score.

Table 23: mean imputation 5-fold cross-validation logit model summary.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
out of sample correctness	0.9361	1	0.9643	0.8776	0.7619
In sample correctness	0.8279	0.9153	0.7433	0.9231	1
Deviance	1158.91	219.81	432.20	247.338	$4.0483e^{1}$
Sensitivity	0.996	1	0.9878	0.875	0.9
Specificity	0	1	0	1	0.2307
Precision	0.9396	1	0.9759	1	0.8182
Recall	0.996	1	0.9878	0.875	0.9
PR AUC	0.5615	0.5	0.5131	0.4359	0.5462
ROC AUC	0.498	1	0.4939	0.9375	0.5653
Brier Score	0.0639	0	0.0357	0.1224	0.2381
sample size (n)	988	260	312	182	234

From the regional and global model comparison in Table 23, what can be concluded is that with the exception of the Latin America model, the regional models performed better than the global models, this is concluded by observing the ROC AUC.

Table 24: MICE 5-fold cross-validation logit model summary.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
out of sample correctness	0.9361	1	0.9643	0.8776	0.7619
In sample correctness	0.8279	0.9153	0.7433	0.9231	1
Deviance	1158.91	219.81	432.20	247.338	$4.0483e^{1}$
Sensitivity	0.996	1	0.9878	0.875	0.9
Specificity	0	1	0	1	0.2307
Precision	0.9396	1	0.9759	1	0.8182
Recall	0.996	1	0.9878	0.875	0.9
PR AUC	0.5615	0.5	0.5131	0.4359	0.5462
ROC AUC	0.498	1	0.4939	0.9375	0.5653
Brier Score	0.0639	0	0.0357	0.1224	0.2381
sample size (n)	988	260	312	182	234

From the regional and global model comparison in Table 24, what can be concluded is that with the exception of the Latin America model, the regional models performed better than the global models, this is concluded by observing the ROC AUC.

B.2 5-fold Cross Validation: Probit Model

In understanding the performance of the models listed below, bucketing is done to better elaborate the understanding. The following are the different buckets for the ROC AUC: poor 0% to 50%, suggest no discrimination and therefore poor, 51%-69% slighlty accaptable which is considered moderately good, 70% to 80% is acceptable and therefore noted as good, 81% to 90% is excellent and anoything greater than 90% is outstanding (Mandrekar, 2010). The main metric used to assess model prediction accuracy will be the ROC AUC, therefore conclusions will be based on the ROC AUC results

B.2.1 5-Fold Cross-Validation Case-Wise Deletion Probit Models

Table 25: 5-fold cross-validation binary probit model summary when applied to the global case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	0.6015	0.5859	0.3045		Insignificant
Global interest	-0.0344	0.0083	<0.0001	85.935	Significant
Foreign Exch reserves	0.0316	0.0090	0.0004	72.437	Significant
Trade openness	-0.0035	0.0020	0.0881	34.651	Insignificant
Export growth	-0.0035	0.0089	0.6934	7.019	Insignificant
Current account	0.01054	0.0194	0.5885	10.112	Insignificant
FDI	0.1568	0.0455	0.0005	71.217	Significant
Real GDP growth	-0.0206	0.0236	0.3819	17.137	Insignificant
Inflation	-0.0445	0.0092	<0.0001	100.000	Significant
REER overall	0.0008	0.0016	0.6216	9.114	Insignificant
Gov expenditures	0.0883	0.0233	0.0001	78.495	Significant
National savings	0.0153	0.0118	0.1944	26.055	Insignificant
Domestic credit	0.0003	0.0054	0.9512	0.000	Insignificant
Bank assets	-0.0094	0.0067	0.1649	27.975	Insignificant
out of sample correctness	0.9091				
In sample correctness	0.8769				
Deviance	535.17				
Sensitivity	0.9821				
Specificity	0				
Precision	0.9243				
Recall	0.9821				
PR AUC	0.5551				
ROC AUC	0.4911				
Brier Score	0.0910				

From analysing the global data set considering case-wise deletion; the top five (5) important variables when developing an EWS for this data set show to be inflation, global interest, government expenditure, FX reserves, and FDI . Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 49.11%, see Table 25.

Table 26: 5-fold cross-validation binary probit model summary when applied to the South and East Asia case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-5.8524	3.4951	0.0940		Insignificant
Global interest	-0.0397	0.1008	0.6936	7.433	Insignificant
Foreign Exch reserves	0.0500	0.0267	0.0616	66.902	Insignificant
Trade openness	0.0031	0.0086	0.7189	6.061	Insignificant
Export growth	0.0330	0.0226	0.1443	50.401	Insignificant
Current account	0.0328	0.0651	0.6140	11.885	Insignificant
FDI	0.4925	0.1831	0.0071	100.000	Significant
Real GDP growth	0.0147	0.0705	0.8340	0.000	Insignificant
Inflation	-0.0375	0.0593	0.5272	17.040	Insignificant
REER overall	0.0304	0.0147	0.0393	74.637	Significant
Gov expenditures	0.0990	0.0698	0.1562	48.713	Insignificant
National savings	0.1305	0.0529	0.0136	91.013	Significant
Domestic credit	0.0226	0.0273	0.4080	24.909	Insignificant
Bank assets	-0.0576	0.0290	0.0473	71.508	Significant
out of sample correctness	0.9412				
In sample correctness	0.925				
Deviance	135.267				
Sensitivity	1				
Specificity	0.6667				
Precision	0.9333				
Recall	1				
PR AUC	0.5354				
ROC AUC	0.8333				
Brier Score	0.0588				

From analysing the South and East Asia data set considering case-wise deletion; the top five (5) important variables when developing an EWS for this data set show to be FDI, national savings, REER overall, bank assets and FX reserves. Based on the metrics the model is excellent at predicting sovereign default since it generated a ROC AUC value of 83.33%, see Table 26.

Table 27: 5-fold cross-validation binary probit model summary when applied to the Latin America case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	3.7499	1.8379	0.0413		Significant
Global interest	-0.0433	0.0125	0.0005	88.553	Significant
Foreign Exch reserves	0.0475	0.0207	0.0216	57.853	Significant
Trade openness	-0.0028	0.0068	0.6809	8.436	Insignificant
Export growth	-0.0400	0.0199	0.0447	50.263	Significant
Current account	0.1016	0.0430	0.0183	59.505	Significant
FDI	0.1092	0.0704	0.1208	38.321	Insignificant
Real GDP growth	-0.0753	0.0437	0.0850	42.806	Insignificant
Inflation	-0.0522	0.0133	<0.0001	100.000	Significant
REER overall	-0.0007	0.0086	0.9287	0.000	Insignificant
Gov expenditures	0.0392	0.0496	0.4294	18.373	Insignificant
National savings	0.0392	0.0410	0.2084	30.632	Insignificant
Domestic credit	0.0227	0.0222	0.3062	24.480	Insignificant
Bank assets	-0.0324	0.0268	0.2270	29.323	Insignificant
out of sample correctness	0.9434				
In sample correctness	0.8222				
Deviance	227.73				
Sensitivity	0.9803				
Specificity	0				
Precision	0.9615				
Recall	0.9803				
PR AUC	0.5187				
ROC AUC	0.4901				
Brier Score	0.0566				

From analysing the Latin America data set considering case-wise deletion; the top five (5) important variables when developing an EWS for this data set show to be inflation, global interest, current account, FX reserves, and export growth. Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 49.01%, see Table 27.

Table 28: 5-fold cross-validation binary probit model summary when applied to the Middle East and Africa case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	11.8346	22.1969	0.594		Insignificant
Global interest	-0.8946	0.9318	0.337	3.373	Insignificant
Foreign Exch reserves	10.9756	10.0930	0.277	50.044	Insignificant
Trade openness	-2.5737	2.5163	0.306	26.352	Insignificant
Export growth	0.5629	0.5383	0.296	34.759	Insignificant
Current account	-2.4943	2.2231	0.262	62.705	Insignificant
FDI	4.8822	5.0338	0.332	6.972	Insignificant
Real GDP growth	2.2732	2.2755	0.318	17.623	Insignificant
Inflation	-0.4314	0.4537	0.342	0.000	Insignificant
REER overall	-0.0599	0.0528	0.256	67.616	Insignificant
Gov expenditures	4.6657	4.2307	0.270	55.671	Insignificant
National savings	0.5868	0.4794	0.221	100.000	Insignificant
Domestic credit	-0.3910	0.3599	0.277	49.576	Insignificant
Bank assets	0.7879	0.7504	0.294	36.332	Insignificant
out of sample correctness	0.9697				
In sample correctness	0.9770				
Deviance	91.326				
Sensitivity	1				
Specificity	0.9166				
Precision	0.9545				
Recall	1				
PR AUC	0.5124				
ROC AUC	0.9583				
Brier Score	0.0303				

From analysing the Middle East and Africa data set considering case-wise deletion; the top five (5) important variables when developing an EWS for this data set show to be national savings, REER overall, current account, and government expenditure. Based on the metrics the model is outstanding at predicting sovereign default since it generated a ROC AUC value of 95.83%, see Table 28.

Table 29: 5-fold cross-validation binary probit model summary when applied to the Western Europe case-wise deletion data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$-7.633e^{1}$	$3.169e^{5}$	1		Insignificant
Global interest	$1.038e^{-1}$	$4.286e^{3}$	1	1.973	Insignificant
Foreign Exch reserves	$-6.643e^{-1}$	$5.486e^{2}$	1	100.000	Insignificant
Trade openness	$-4.891e^{-1}$	$9.780e^{2}$	1	41.277	Insignificant
Export growth	-1.545	$2.629e^{3}$	1	48.515	Insignificant
Current account	1.850	$5.495e^{3}$	1	27.781	Insignificant
FDI	$4.311e^{-2}$	$1.145e^{4}$	1	0.283	Insignificant
Real GDP growth	7.054	$1.034e^{4}$	0.999	56.313	Insignificant
Inflation	$8.705e^{-1}$	$8.083e^{3}$	1	8.867	Insignificant
REER overall	1.149	$2.458e^{3}$	1	38.596	Insignificant
Gov expenditures	$-4.390e^{-1}$	$1.297e^{4}$	1	0.000	Insignificant
National savings	$-3.805e^{-1}$	$1.240e^{4}$	1	2.507	Insignificant
Domestic credit	-1.135	$2.630e^{3}$	1	35.627	Insignificant
Bank assets	1.432	$3.076e^{3}$	1	38.413	Insignificant
out of sample correctness	0.6315				
In sample correctness	1				
Deviance	$5.6574e$				
Sensitivity	1				
Specificity	0				
Precision	Nan				
Recall	Nan				
PR AUC	Nan				
ROC AUC	Nan				
Brier Score	0.3684				

From analysing the Western European data set considering case-wise deletion; the top five (5) important variables when developing an EWS for this data set show to be FX reserves, real GDP growth, bak assets, domestic credit, and export growth. Based on the metrics the model did not generate a ROC AUC score since there were not enough defaults for the model to generate scores, see Table 29.

B.2.2 5-Fold Cross-Validation Mean Imputation Probit Models

Table 30: 5-fold cross-validation binary probit model summary when applied to the global mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$-3.763e^{-1}$	$3.942e-01$	0.3397		Insignificant
Total debt	$-1.301e^{-2}$	$2.652e^{-3}$	< 0.0001	85.348	Significant
Global interest	$-1.202e^{-2}$	$4.271e^{-3}$	0.0049	48.886	Significant
Foreign Exch reserves	$4.464e^{-2}$	$7.772e^{-3}$	< 0.0001	100.000	Significant
Trade openness	$-3.981e^{-3}$	$1.565e^{-3}$	0.0109	44.183	Significant
Export growth	$6.251e^{-5}$	$5.338e^{-3}$	0.9906	0.000	Insignificant
Current account	$-1.650e^{-2}$	$9.833e^{-3}$	0.0933	29.068	Insignificant
FDI	$8.683e^{-2}$	$2.885e^{-2}$	0.0026	52.302	Significant
Real GDP growth	$8.834e^{-3}$	$1.181e^{-2}$	0.4542	12.851	Insignificant
Inflation	$-1.832e^{-2}$	$3.843e^{-3}$	< 0.0001	82.986	Significant
REER overall	$2.869e^{-3}$	$1.344e^{-3}$	0.0327	37.054	Significant
Gov expenditures	$5.904e^{-2}$	$1.405e^{-2}$	< 0.0001	73.105	Significant
National savings	$2.204e^{-2}$	$7.444e^{-3}$	0.0031	51.447	Significant
Domestic credit	$7.058e^{-3}$	$3.825e^{-3}$	0.0649	31.994	Insignificant
Bank assets	$1.661e^{-3}$	$3.640e^{-3}$	0.6482	7.756	Insignificant
out of sample correctness	0.9398				
In sample correctness	0.8238				
Deviance	1158.91				
Sensitivity	1				
Specificity	0				
Precision	Nan				
Recall	Nan				
PR AUC	Nan				
ROC AUC	0.5				
Brier Score	0.0601				

From analysing the global data set considering mean imputation; the top five (5) important variables when developing an EWS for this data set show to be FX reserves, total debt, inflation, government expenditure, and national savings. Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 50%, see Table 30.

Table 31: 5-fold cross-validation binary probit model summary when applied to the South and East Asia mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-2.3540	1.7481	0.1781		Insignificant
Total debt	-0.0250	0.0086	0.0037	92.18	Significant
Global interest	-0.0326	0.0298	0.2739	28.59	Insignificant
Foreign Exch reserves	0.0433	0.0201	0.0313	65.88	Significant
Trade openness	-0.0035	0.0037	0.3432	23.45	Insignificant
Export growth	0.0325	0.0120	0.0069	85.15	Significant
Current account	-0.0263	0.0226	0.2451	30.99	Insignificant
FDI	0.3380	0.1165	0.0037	92.19	Significant
Real GDP growth	-0.0066	0.0234	0.7781	0.00	Insignificant
Inflation	-0.0437	0.0161	0.0066	85.57	Significant
REER overall	0.0308	0.0132	0.0198	72.10	Significant
Gov expenditures	0.1697	0.0543	0.0017	100.00	Significant
National savings	0.0318	0.0255	0.2124	33.97	Insignificant
Domestic credit	-0.0270	0.0107	0.0121	78.40	Significant
Bank assets	0.0044	0.0074	0.5460	11.33	Insignificant
out of sample correctness	1				
In sample correctness	0.9115				
Deviance	219.81				
Sensitivity	1				
Specificity	1				
Precision	1				
Recall	1				
PR AUC	0.5				
ROC AUC	1				
Brier Score	0				

From analysing the South and East Asia data set considering mean imputation; the top five (5) important variables when developing an EWS for this data set show to be government expenditure, FDI, total debt, inflation, and export growth. Based on the metrics the model is outstanding at predicting sovereign default since it generated a ROC AUC value of 100%, see Table 31.

Table 32: 5-fold cross-validation binary probit model summary when applied to the Latin America mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-1.6698	0.8381	0.0463		Significant
Total debt	-0.0013	0.0060	0.8262	1.759	Insignificant
Global interest	-0.0115	0.0056	0.0403	52.543	Significant
Foreign Exch reserves	0.0505	0.0134	0.0001	100.000	Significant
Trade openness	-0.0042	0.0039	0.2814	25.545	Insignificant
Export growth	-0.0062	0.0086	0.4669	15.849	Insignificant
Current account	-0.0038	0.0244	0.8759	0.000	Insignificant
FDI	0.1192	0.0454	0.0086	68.510	Significant
Real GDP growth	-0.0089	0.0226	0.6935	6.600	Insignificant
Inflation	-0.0153	0.0042	0.0003	95.522	Significant
REER overall	0.0124	0.0038	0.0013	84.704	Significant
Gov expenditures	0.0268	0.0263	0.3072	23.990	Insignificant
National savings	0.0421	0.0194	0.0304	55.716	Significant
Domestic credit	0.0294	0.0113	0.0092	67.871	Significant
Bank assets	-0.0298	0.0141	0.0354	54.019	Significant
out of sample correctness	0.9643				
In sample correctness	0.7724				
Deviance	432.20				
Sensitivity	0.9878				
Specificity	0				
Precision	0.9759				
Recall	0.9878				
PR AUC	0.5131				
ROC AUC	0.4939				
Brier Score	0.0357				

From analysing the Latin America data set considering mean imputation; the top five (5) important variables when developing an EWS for this data set show to be FX reserves, inflation, REER overall, FDI, and domestic credit. Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 49.39%, see Table 32.

Table 33: 5-fold cross-validation binary probit model summary when applied to the Middle East and Africa mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-3.8418	2.4621	0.1186		Insignificant
Total debt	0.0233	0.0207	0.2608	25.564	Insignificant
Global interest	-0.0671	0.0542	0.2164	28.219	Insignificant
Foreign Exch reserves	0.6054	0.1424	<0.0001	100.000	Significant
Trade openness	-0.0076	0.0133	0.5651	12.489	Insignificant
Export growth	-0.0280	0.0194	0.1494	33.112	Insignificant
Current account	-0.0887	0.0593	0.1345	34.417	Insignificant
FDI	0.6747	0.2353	0.0041	67.060	Significant
Real GDP growth	-0.0222	0.0394	0.5727	12.225	Insignificant
Inflation	-0.0891	0.0291	0.0022	71.526	Significant
REER overall	0.0073	0.0030	0.0154	56.460	Significant
Gov expenditures	0.0045	0.0904	0.9597	0.000	Insignificant
National savings	0.0055	0.0266	0.8348	3.761	Insignificant
Domestic credit	0.0446	0.0121	0.0002	86.308	Significant
Bank assets	-0.0224	0.0193	0.2479	26.301	Insignificant
out of sample correctness	0.8776				
In sample correctness	0.9065				
Deviance	247.338				
Sensitivity	0.875				
Specificity	1				
Precision	1				
Recall	0.875				
PR AUC	0.4359				
ROC AUC	0.9375				
Brier Score	0.1224				

From analysing the Middle East and Africa data set considering mean imputation; the top five (5) important variables when developing an EWS for this data set show to be FX reserves, domestic credit, inflation, FDI, and REER overall. Based on the metrics the model is outstanding at predicting sovereign default since it generated a ROC AUC value of 93.75%, see Table 33.

Table 34: 5-fold cross-validation binary probit model summary when applied to the Western Europe mean imputation data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$1.079e^{2}$	$1.159e^{5}$	0.999		Insignificant
Total debt	$-7.694e^{-1}$	$2.540e^{2}$	0.998	81.8488	Insignificant
Global interest	-1.785	$1.118e^{3}$	0.999	41.1335	Insignificant
Foreign Exch reserves	$5.609e^{-1}$	$4.163e^{2}$	0.999	34.0717	Insignificant
Trade openness	$2.390e^{-1}$	$3.862e^{2}$	1	13.3780	Insignificant
Export growth	$3.072e^{-1}$	$2.076e^{3}$	1	0.000	Insignificant
Current account	-4.441	$1.832e^{3}$	0.998	64.6472	Insignificant
FDI	1.961	$4.364e^{3}$	1	8.5629	Insignificant
Real GDP growth	-2.209	$4.414e^{3}$	1	10.0145	Insignificant
Inflation	$2.103e^{-1}$	$1.393e^{3}$	1	0.0849	Insignificant
REER overall	$3.173e^{-1}$	$8.662e^{2}$	1	6.2024	Insignificant
Gov expenditures	-1.169	$1.746e^{3}$	0.999	2.5353	Insignificant
National savings	1.010	$4.258e^{3}$	1	14.8129	Insignificant
Domestic credit	-1.323	$3.606e^{2}$	0.997	100.000	Insignificant
Bank assets	$3.984e^{-1}$	$2.760e^{2}$	0.999	36.8071	Insignificant
out of sample correctness	0.7619				
In sample correctness	1				
Deviance	$4.0483e^{3}$				
Sensitivity	0.9				
Specificity	0.2307				
Precision	0.8182				
Recall	0.9				
PR AUC	0.5462				
ROC AUC	0.5653				
Brier Score	0.2381				

From analysing the Western European data set considering mean imputation; the top five (5) important variables when developing an EWS for this data set show to be domestic credit, total debt, current account, global interest, and bank assets. Based on the metrics the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 56.53%, see Table 34.

B.2.3 5-Fold Cross-Validation MICE Probit Models

Table 35: 5-fold cross-validation binary probit model summary when applied to the global MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	1.4879	0.2742	<0.0001		Significant
Total debt	-0.0207	0.0023	<0.0001	100.000	Significant
Global interest	-0.0013	0.0011	0.2602	11.713	Insignificant
Foreign Exch reserves	0.0269	0.0061	<0.0001	48.804	Significant
Trade openness	0.0012	0.0016	0.4638	7.314	Insignificant
Export growth	-0.0004	0.0055	0.9373	0.000	Insignificant
Current account	-0.0094	0.0117	0.4227	8.088	Insignificant
FDI	0.0899	0.0295	0.0023	33.155	Significant
Real GDP growth	0.0190	0.0120	0.1154	16.728	Insignificant
Inflation	-0.0009	0.0006	0.1528	15.108	Insignificant
REER overall	-0.0007	0.0011	0.5371	6.024	Insignificant
Gov expenditures	0.1232	0.0149	<0.0001	91.107	Significant
National savings	0.0373	0.0069	<0.0001	58.890	Significant
Domestic credit	-0.0071	0.0041	0.0875	18.232	Insignificant
Bank assets	0.0166	0.0044	0.0001	40.964	Significant
out of sample correctness	0.9436				
In sample correctness	0.8684				
Deviance	1144.97				
Sensitivity	0.996				
Specificity	0.125				
Precision	0.9467				
Recall	0.996				
PR AUC	0.5498				
ROC AUC	0.5605				
Brier Score	0.0563				

From analysing the global data set considering MICE; the top five (5) important variables when developing an EWS for this data set show to be total debt, government, national savings, FX reserves, and bank assets. Based on the metrics the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 56.05%, see Table 35.

Table 36: 5-fold cross-validation binary probit model summary when applied to the South and East Asia MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	4.0596	1.4342	0.0046		Significant
Total debt	-0.0414	0.0097	<0.0001	100.00	Significant
Global interest	-0.1399	0.0378	0.0002	85.04	Significant
Foreign Exch reserves	-0.0063	0.0019	0.0008	75.37	Significant
Trade openness	-0.0021	0.0039	0.5838	0.00	Insignificant
Export growth	0.0565	0.0165	0.0006	77.33	Significant
Current account	-0.0423	0.0326	0.1944	20.24	Insignificant
FDI	0.2573	0.1128	0.0225	46.78	Significant
Real GDP growth	-0.0415	0.0315	0.1881	20.74	Insignificant
Inflation	-0.0501	0.0197	0.0109	53.88	Significant
REER overall	-0.0146	0.0068	0.0318	43.14	Significant
Gov expenditures	0.2277	0.0668	0.0006	77.10	Significant
National savings	0.0345	0.0282	0.2216	18.20	Insignificant
Domestic credit	-0.0466	0.0159	0.0033	64.34	Significant
Bank assets	0.0357	0.0141	0.0111	53.80	Significant
out of sample correctness	0.9605				
In sample correctness	0.8923				
Deviance	223.25				
Sensitivity	1				
Specificity	0.5				
Precision	0.9589				
Recall	1				
PR AUC	0.5325				
ROC AUC	0.75				
Brier Score	0.0394				

From analysing the South and East Asia data set considering MICE; the top five (5) important variables when developing an EWS for this data set show to be total debt, export growth, global interest, FX reseres, and government expenditure . Based on the metrics the model is good at predicting sovereign default since it generated a ROC AUC value of 75%, see Table 36.

Table 37: 5-fold cross-validation binary probit model summary when applied to the Latin America MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-0.5534	0.8085	0.4936		Insignificant
Total debt	-0.0247	0.0053	<0.0001	100.000	Significant
Global interest	-0.0139	0.0049	0.0044	57.786	Significant
Foreign Exch reserves	0.0175	0.0107	0.1022	28.746	Insignificant
Trade openness	-0.0059	0.0038	0.1234	26.502	Insignificant
Export growth	-0.0077	0.0092	0.4040	9.568	Insignificant
Current account	0.1033	0.0278	0.0002	78.708	Significant
FDI	0.1380	0.0514	0.0072	53.927	Significant
Real GDP growth	-0.0221	0.0234	0.3456	12.174	Insignificant
Inflation	-0.0111	0.0044	0.0114	50.203	Significant
REER overall	0.0165	0.0037	<0.0001	95.520	Significant
Gov expenditures	0.0470	0.0267	0.0788	31.712	Insignificant
National savings	-0.0086	0.0198	0.6632	0.000	Insignificant
Domestic credit	-0.0261	0.0130	0.0454	37.533	Significant
Bank assets	0.0468	0.0175	0.0076	53.543	Significant
out of sample correctness	0.9761				
In sample correctness	0.7980				
Deviance	432.47				
Sensitivity	1				
Specificity	0				
Precision	Nan				
Recall	Nan				
PR AUC	Nan				
ROC AUC	0.5				
Brier Score	0.0238				

From analysing the Latin America data set considering MICE; the top five (5) important variables when developing an EWS for this data set show to be total debt, REER overal, current account, global intrest, and bank assets . Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 50%, see Table 37.

Table 38: 5-fold cross-validation binary probit model summary when applied to the Middle East and Africa MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	-3.0101	5.6044	0.5912		Insignificant
Total debt	-0.2225	0.0681	0.0011	94.52	Significant
Global interest	-0.0516	0.0977	0.5971	0.00	Insignificant
Foreign Exch reserves	0.9304	0.2744	0.0006	98.68	Significant
Trade openness	-0.1835	0.0577	0.0014	91.38	Significant
Export growth	0.0956	0.0489	0.0508	49.11	Significant
Current account	-0.7065	0.2061	0.0006	100.00	Significant
FDI	1.9298	0.5813	0.0009	96.25	Significant
Real GDP growth	0.1723	0.0845	0.0415	52.07	Significant
Inflation	-0.3426	0.1219	0.0049	78.70	Significant
REER overall	-0.0299	0.0103	0.0039	81.16	Significant
Gov expenditures	0.94191	0.2953	0.0014	91.74	Significant
National savings	0.4343	0.1333	0.0011	94.06	Significant
Domestic credit	0.0461	0.0277	0.0962	39.14	Insignificant
Bank assets	0.0688	0.0542	0.2044	25.54	Insignificant
out of sample correctness	0.9591				
In sample correctness	0.9395				
Deviance	247.981				
Sensitivity	0.9791				
Specificity	0				
Precision	0.9791				
Recall	0.9791				
PR AUC	0.5				
ROC AUC	0.4895				
Brier Score	0.0408				

From analysing the Middle East and Africa data set considering MICE; the top five (5) important variables when developing an EWS for this data set show to be current account, FX reserves, FDI, national savings, and total debt . Based on the metrics the model is poor at predicting sovereign default since it generated a ROC AUC value of 48.95%, see Table 38.

Table 39: 5-fold cross-validation binary probit model summary when applied to the Western Europe MICE data set.

	Estimate	Standard Error	Pr(>|z|)	Importance	Significance
Intercept	$9.345e^{1}$	$8.412e^{4}$	0.999		Insignificant
Total debt	$-5.895e^{-2}$	$3.424e^{2}$	1	8.846	Insignificant
Global interest	-2.849	$2.327e^{3}$	0.999	78.656	Insignificant
Foreign Exch reserves	$6.341e^{-2}$	$2.528e^{2}$	1	14.062	Insignificant
Trade openness	$-2.668e^{-1}$	$1.725e^{2}$	0.999	100.000	Insignificant
Export growth	$9.393e^{-1}$	$1.434e^{3}$	0.999	40.871	Insignificant
Current account	$5.371e^{-1}$	$1.647e^{3}$	1	19.060	Insignificant
FDI	$4.414e^{-1}$	$1.137e^{3}$	1	23.186	Insignificant
Real GDP growth	$-3.115e^{-1}$	$3.296e^{3}$	1	3.697	Insignificant
Inflation	2.592	$2.706e^{3}$	0.999	60.980	Insignificant
REER overall	$1.007e^{-1}$	$6.899e^{2}$	1	7.111	Insignificant
Gov expenditures	$-8.914e^{-2}$	$2.297e^{3}$	1	0.000	Insignificant
National savings	-1.145	$2.569e^{3}$	1	26.999	Insignificant
Domestic credit	$-2.908e^{-1}$	$2.389e^{2}$	0.999	78.180	Insignificant
Bank assets	$2.063e^{-2}$	$1.609e^{2}$	1	5.930	Insignificant
out of sample correctness	0.7301				
In sample correctness	1				
Deviance	$4.0483e^{1}$				
Sensitivity	0.88				
Specificity	0.1538				
Precision	0.8				
Recall	0.88				
PR AUC	0.5462				
ROC AUC	0.5169				
Brier Score	0.2698				

From analysing the Western European data set considering MICE; the top five (5) important variables when developing an EWS for this data set show to be trade openness, global interest, domestic credit, inflation, and export growth. Based on the metrics the model is moderately good at predicting sovereign default since it generated a ROC AUC value of 51.69%, see Table 39.

B.2.4 Performance Comparison

Table 40: Case-wise deletion 5-fold cross-validation probit model summary.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
Out of sample correctness	0.9091	0.9412	0.9434	0.9697	0.6315
In sample correctness	0.8769	0.925	0.8222	0.9770	1
Deviance	535.17	135.267	227.73	91.326	$5.6574e$
Sensitivity	0.9821	1	0.9803	1	1
Specificity	0	0.6667	0	0.9166	0
Precision	0.9243	0.9333	0.9615	0.9545	Nan
Recall	0.9821	1	0.9803	1	Nan
PR AUC	0.5551	0.5354	0.5187	0.5124	Nan
ROC AUC	0.4911	0.8333	0.4901	0.9583	Nan
Brier Score	0.0910	0.0588	0.0566	0.0303	0.3684
sample size (n)	569	160	180	87	143

From the regional and global model comparison in Table 40 the following can be concluded; the Western European model could not generate metrics scores because of not having enough default observations in the test set. The only regional model that failed to surpace the global model was the Latin America model, therefore making the regional models still more predictive. This is concluded by observing the ROC AUC.

Table 41: mean imputation 5-fold cross-validation probit model summary.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
out of sample correctness	0.9398	1	0.9643	0.8776	0.7619
In sample correctness	0.8238	0.9115	0.7724	0.9065	1
Deviance	1158.91	219.81	432.20	247.338	$4.0483e^3$
Sensitivity	1	1	0.9878	0.875	0.9
Specificity	0	1	0	1	0.2307
Precision	Nan	1	0.9759	1	0.8182
Recall	Nan	1	0.9878	0.875	0.9
PR AUC	Nan	0.5	0.5131	0.4359	0.5462
ROC AUC	0.5	1	0.4939	0.9375	0.5653
Brier Score	0.0601	0	0.0357	0.1224	0.2381
sample size (n)	988	260	312	182	234

From the regional and global model comparison in Table 41, what can be concluded is that with the exception of the Latin America model, the regional models performed better than the global models, this is concluded by observing the ROC AUC.

Table 42: MICE 5-fold cross-validation probit model summary.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
out of sample correctness	0.9436	0.9605	0.9761	0.9591	0.7301
In sample correctness	0.8684	0.8923	0.7980	0.9395	1
Deviance	1144.97	223.25	432.47	247.981	$4.0483e^{1}$
Sensitivity	0.996	1	1	0.9791	0.88
Specificity	0.125	0.5	0	0	0.1538
Precision	0.9467	0.9589	Nan	0.9791	0.8
Recall	0.996	1	Nan	0.9791	0.88
PR AUC	0.5498	0.5325	Nan	0.5	0.5462
ROC AUC	0.5605	0.75	0.5	0.4895	0.5169
Brier Score	0.0563	0.0394	0.0238	0.0408	0.2698
sample size (n)	988	260	312	182	234

From the regional and global model comparison in Table 42, what can be concluded is that the global model performed better than three (3) regional models; Latin America, the Middle East and Africa, and the Western European model. This implies that in the case of using MICE for the probit, with cross-validation, not all regional models perform better than the global model. This is concluded by observing the ROC AUC.

B.3 5-fold Cross Validation: BART

In the same way, the BART model was build using cross-validation with the exact features and metrics considered in the previous chapter. In this analysis, a different package was used by the name of "bartMachine". This section looks at the models built using 5 fold cross validation for the different data sets, the summary of the model is details in Chapter 5 of the study, here we have the model additional model figures.

B.3.1 5-Fold Cross-Validation Case-Wise Deletion BART Models

B.3.1.1 Global

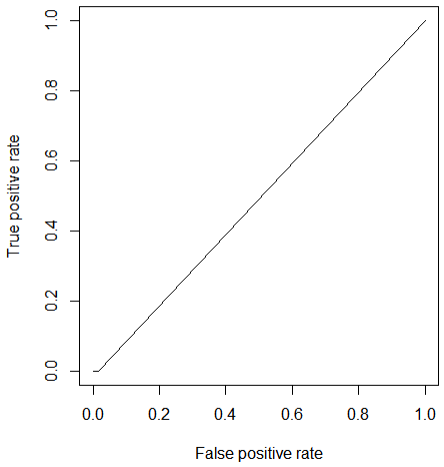

Figure 6: Area under the curve graph applied to case-wise deletion global data.

From the observation of Figure 6 and the calculation in Table 43, it can be concluded that the classifier built using the case wise deletion global data set produced a non-skilled classifier since the ROC AUC value is below 50% and the curve falls below the diagonal axis.

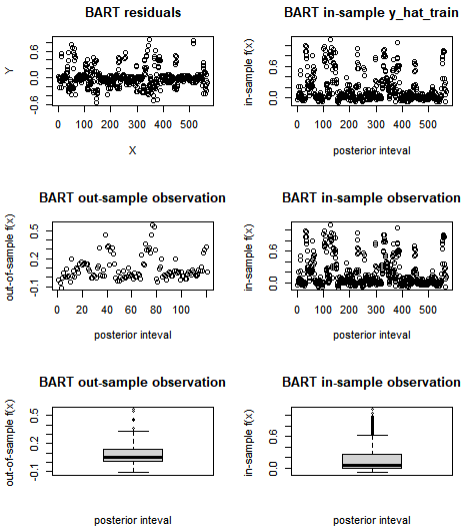

Figure 7: Plots and box plots of the model output results for training and testing case-wise deletion global data set.

From Figure 7, the following can be concluded: the BART residual plot shows that the majority of the residual values lie close to the zero (0) line, which shows that the number of closely correct predictions

is large. The BART out-sample and in-sample plots represent the predicted observations from the developed classifier using the training set, and they both illustrate that the majority of the observations from the input data are non-default states of the sovereign. The box plots in Figure 7 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults. In addition, the defaults are seen as outliers.

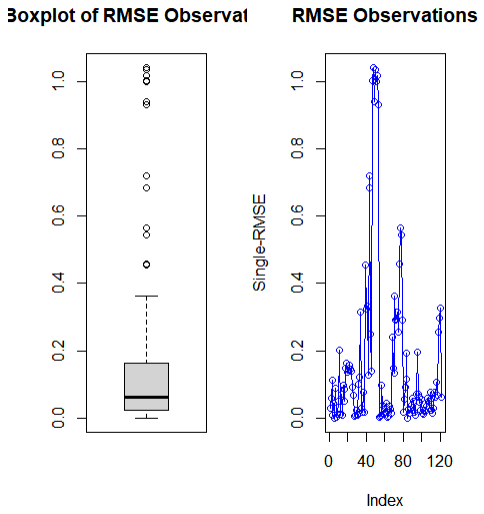

Figure 8: RMSE box plot and line plot for case-wise deletion global data.

From Figure 8, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.4 are seen as outliers.

Table 43: 5-fold cross-validation BART model summary when applied to the global case-wise deletion data set.

RMSE	0.3015
RRMSE	1.6583
ROC AUC	0.4910
Out of sample correctness	0.9091
In sample correctness	0.9648
RMSE(0.5,0.75)	(0.0630, 0.1625)
Sensitivity	0.9821
Specificity	0
Precision	0.9243
Recall	0.9821
PR AUC	0.5550
Brier Score	0.0909

From the observations in Table 43, the following can be concluded: The out-of-sample-correctness

values show that the classifer scored 90.91% of all observations. The ROC AUC value shows that the classifier is not skillful as an EWS since the value is below 0.5. The sensitivity and recall scores illustrate that the classifier did a good job at predicting non-default times as compared to default times since the specificity score was zero (0). The precision score also shows that the model performed well since it correctly identified 92.43% of non-default states.

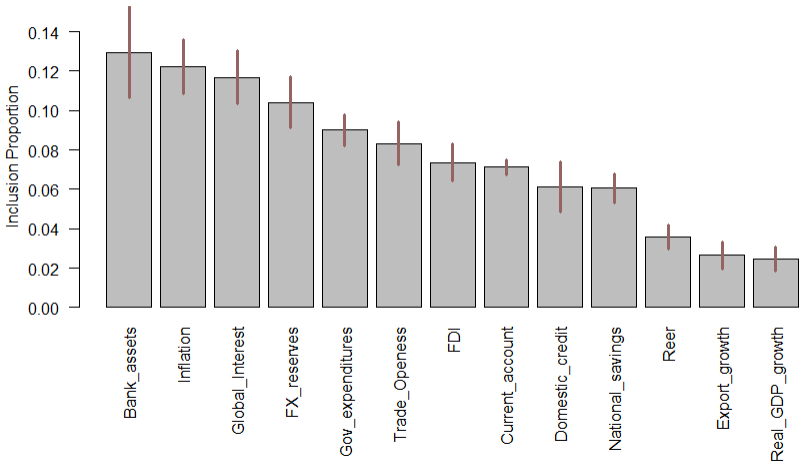

Figure 9: 5-fold cross validation BART variable importance for the global case wise deletion data set.

From Figure 9, it can be concluded that when considering case wise deletion for the global data set, the top five variables in terms of inclusion proportion are bank assets, inflation, global interest, FX reserves, and government expenditure.

B.3.1.2 South and East Asia

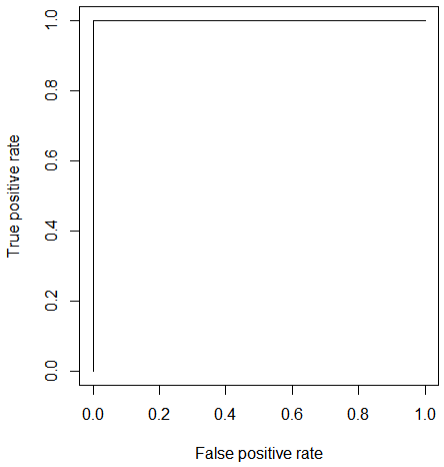

Figure 10: Area under the curve graph applied to case-wise deletion South and East Asia data set.

From the observation of Figure 10 and calculation in Table 44, it can be concluded that the classifier built using the case wise deletion South and East Asia data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches over the diagonal line centre.

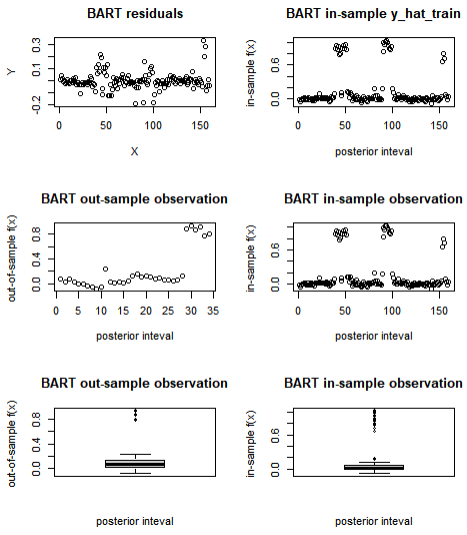

Figure 11: Plots and box plots of the model output results for training and testing case-wise deletion South and East Asia data set.

From Figure 11 the following can be concluded: the BART residual plot shows that the majority of the residual values lie on the zero (0) line, which shows that the number of closely correct predictions is

large. The BART out-sample and in-sample plots represent the predicted observations from the developed classifier using the training set, and they both illustrate that the majority of the observations from the input data are non-default states of the sovereign. The box plots in Figure 11 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults. In addition, the defaults are seen as outliers.

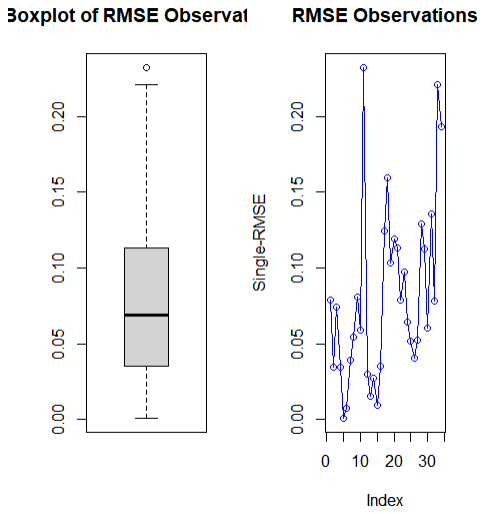

Figure 12: RMSE box plot and line plot for case-wise deletion South and East Asia data.

From Figure 12, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.2 are seen as outliers.

Table 44: 5-fold cross-validation BART model summary when applied to the South and East Asia case-wise deletion data set.

RMSE	0
RRMSE	0
ROC AUC	1
Out of sample correctness	1
In sample correctness	1
RMSE(0.5,0.75)	(0.06920.1132)
Sensitivity	1
Specificity	1
Precision	1
Recall	1
PR AUC	0.5
Brier Score	0

From the observations in Table 44, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to perfectly predict all the observations. The ROC AUC value shows that the classifier is ideal as an EWS since the value is above 50% and has a 100% value. The sensitivity and recall scores illustrate that the classifier did a good job at predicting non-default times as well as specificity since they all scored a perfect 100%. The precision score also shows that the model performed well since it correctly identified 100% of non-default states.

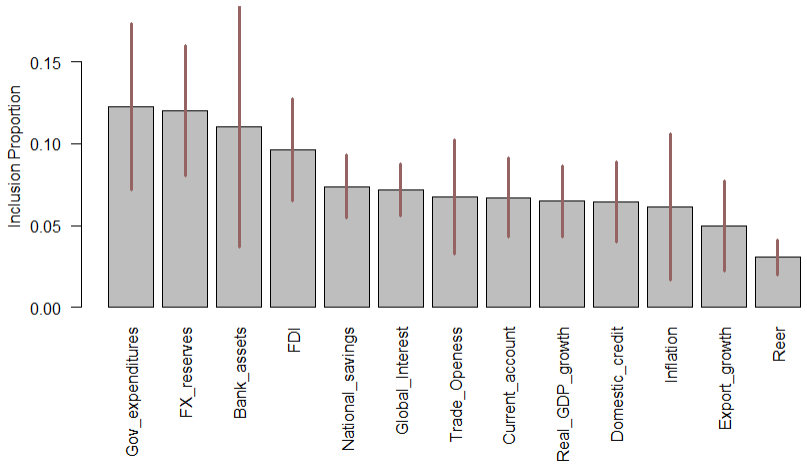

Figure 13: 5-fold cross validation BART variable importance for the South and East Asia case wise deletion data set.

From Figure 13 it can be concluded that when considering case wise deletion for the South and East Asia data set, the top five variables in terms of inclusion proportion are government expenditure, FX reserves, bank assets, FDI, and national savings.

B.3.1.3 Latin America

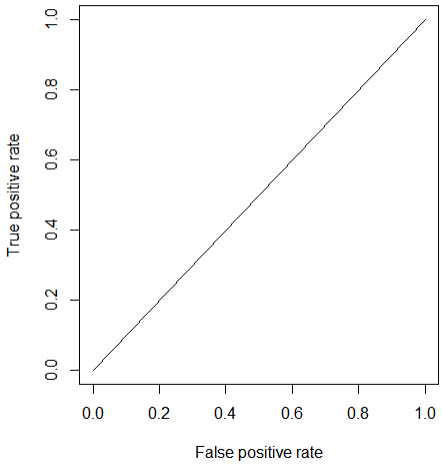

Figure 14: Area under the curve graph applied to case-wise deletion Latin America data set.

From the observation of Figure 14 and calculation in Table 45, it can be concluded that the classifier built using the case wise deletion Latin America data set produced a somewhat skilled classifier since the since ROC AUC value is 50% and the curve runs of the the diagonal centre line.

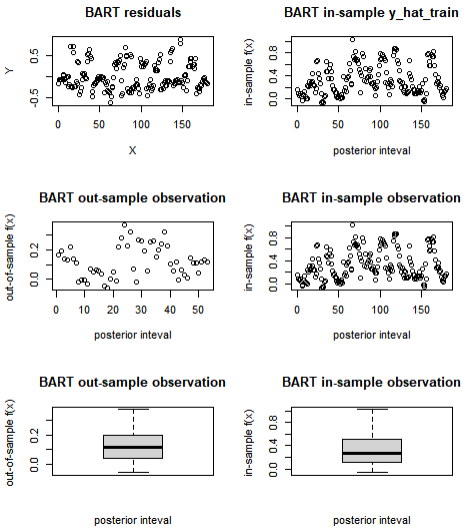

Figure 15: Plots and box plots of the model output results for training and testing case-wise deletion Latin America data set.

From Figure 15 the following can be concluded: The BART residual plot shows that the majority of the residual values are scattered around the zero (0) line, which shows that the number of correct predictions was off by at least an interval of 0.1 since in that interval is where the majority of errors lie. The

BART out-sample and in-sample plots represent the predicted observations from the developed classifier using the training set, and they both illustrate that all of the observations from the input data are non-default states of the sovereign, therefore showing that the classifier failed to detect defaults. The box plots in Figure 15 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults. The classifier was only able to detect defaults for the in-sample data.

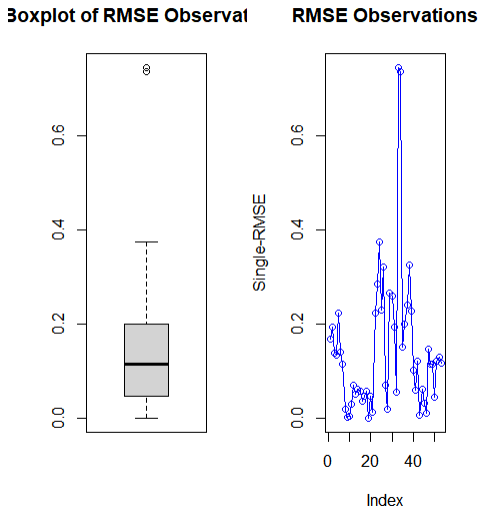

Figure 16: RMSE box plot and line plot for case-wise deletion Latin America data.

From Figure 16, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.4 are seen as outliers.

Table 45: 5-fold cross-validation BART model summary when applied to the Latin America case-wise deletion data set.

RMSE	0.2152
RRMSE	0.2416
ROC AUC	0.5
Out of sample correctness	0.9622
In sample correctness	0.9166
RMSE(0.5,0.75)	(0.1208,0.2158)
Sensitivity	1
Specificity	0
Precision	Nan
Recall	Nan
PR AUC	Nan
Brier Score	0.0377

From the observations in Table 45, the following can be concluded: The classifier outputs can not be summarised since they did not have enough defaults to conclude adequate statistics.

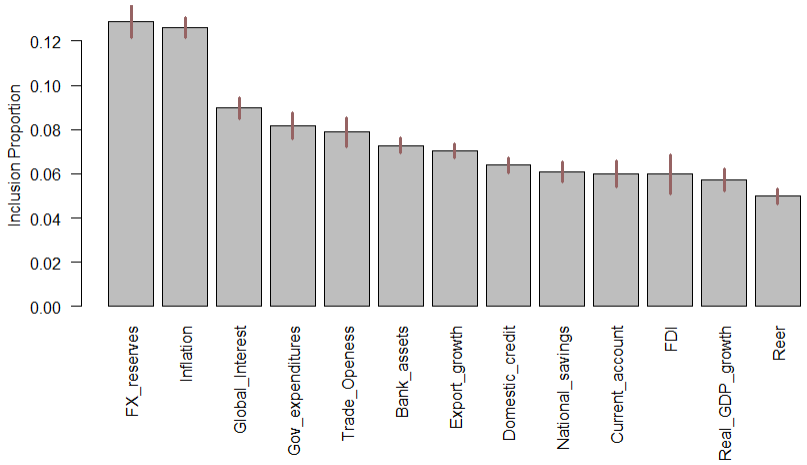

Figure 17: 5-fold cross validation BART variable importance for the Latin America case wise deletion data set.

From Figure 17 it can be concluded that when considering case wise deletion for the Latin America data set, the top five variables in terms of inclusion proportion are FX reserves, inflation, global interest, government expenditure , and trade openess.

B.3.1.4 The Middle East and Africa

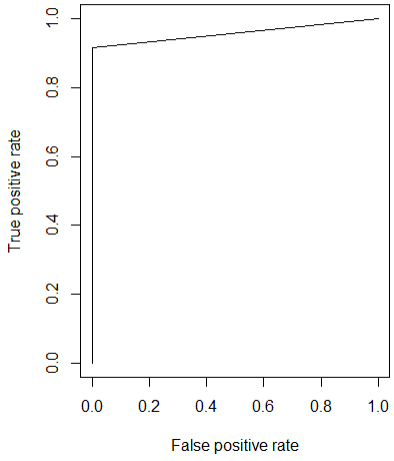

Figure 18: Area under the curve graph applied to case-wise deletion the Middle East and Africa data set.

From the observation of Figure 18 and calculation in Table 46, it can be concluded that the classifier built using the the case wise deletion Middle East and Africa data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal centre line.

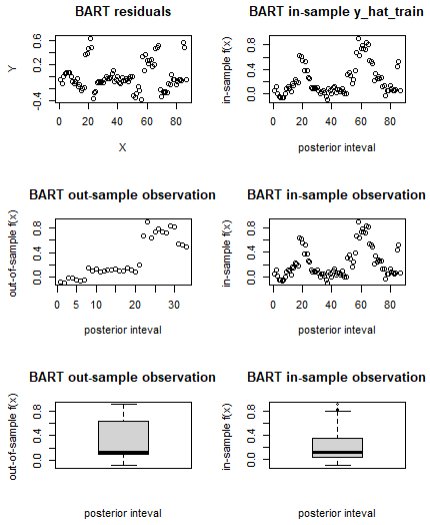

Figure 19: Plots and box plots of the model output results for training and testing case-wise deletion Middle East and Africa data set.

From Figure 19 the following can be concluded: The BART residual plot shows that the majority of the residual values are scattered around the zero (0) line, which shows that the number of correct predictions was off by at least an interval of 0.2 since in that interval is where the majority of errors lie. The BART out-sample and in-sample plots represent the predicted observations from the developed classifier using the training set, and they both illustrate that the majority of the observations from the input data are non-default states of the sovereign. The box plots in Figure 19 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults. In addition, the defaults are seen as outliers.

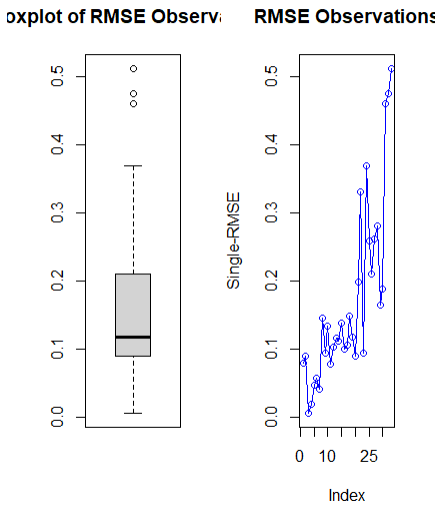

Figure 20: RMSE box plot and line plot for case-wise deletion Middle East and Africa data.

From Figure 20, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.4 are seen as outliers.

Table 46: 5-fold cross-validation BART model summary when applied to the Middle East and Africa case-wise deletion data set.

RMSE	0.2140
RRMSE	0.1262
ROC AUC	0.9583
Out of sample correctness	0.9696
In sample correctness	0.9655
RMSE(0.5,0.75)	(0.1182,0.2102)
Sensitivity	1
Specificity	0.9166
Precision	0.9545
Recall	1
PR AUC	0.5124
Brier Score	0.0303

From the observations in Table 46, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 96.96% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 95.83%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 100% at predicting non-default times and having a specificity score of 91.66%, which shows that it did a good job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 95.45% of non-default states.

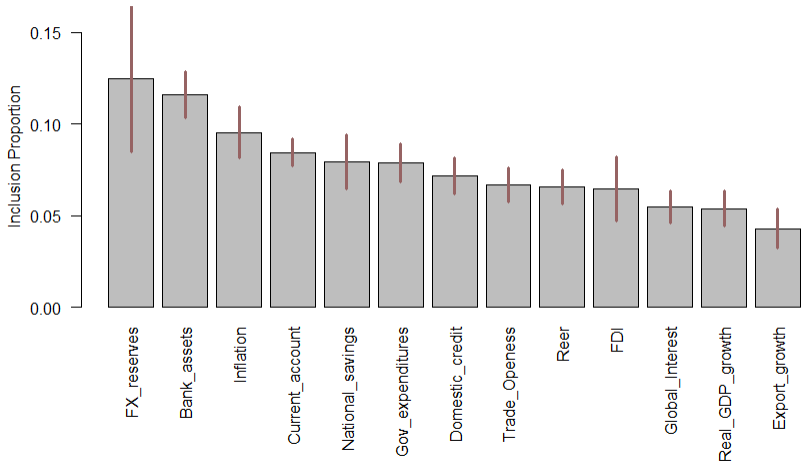

Figure 21: 5-fold cross validation BART variable importance for the Middle East and Africa case wise deletion data set.

From Figure 21 it can be concluded that when considering case wise deletion for the Middle East and Africa data set, the top five variables in terms of inclusion proportion are FX reserves, bank assets, inflation, current account, and national savings.

B.3.1.5 Western Europe

Table 47: 5-fold cross-validation BART model summary when applied to the Western Europe case-wise deletion data set.

RMSE	Nan
RRMSE	Nan
ROC AUC	Nan
Out of sample correctness	Nan
In sample correctness	Nan
RMSE(0.5,0.75)	(Nan,Nan)
Sensitivity	Nan
Specificity	Nan
Precision	Nan
Recall	Nan
PR AUC	Nan
Brier Score	Nan

For this data set, there was no default observation in the training data set. Therefore, no model could be built since it could not be distinguished between what inputs contributed to a default.

B.3.2 5-Fold Cross-Validation Mean Imputation BART Models

B.3.2.1 Global

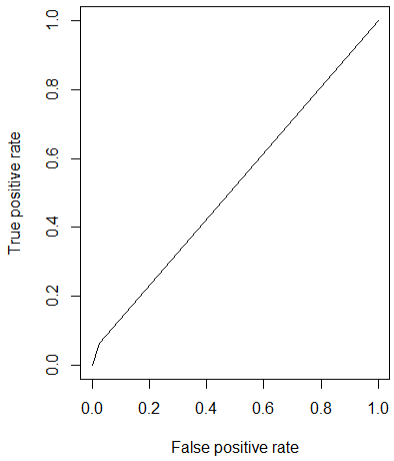

Figure 22: Area under the curve graph applied for the mean imputed global data set.

From the observation of Figure 22 and calculation in Table 48, it can be concluded that the classifier built using the mean imputed global data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal centre line.

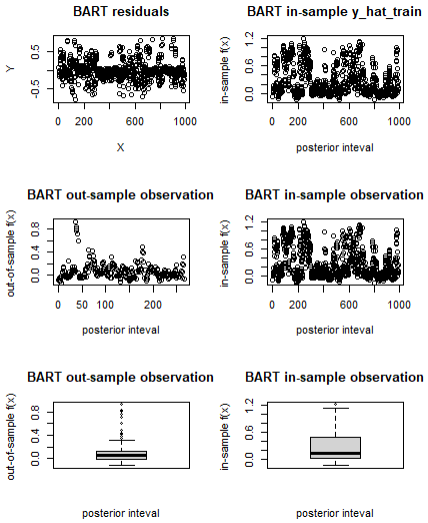

Figure 23: Plots and box plots of the model output results for training and testing mean imputed global data set.

From Figure 23 the following can be concluded: The BART residual plot shows that although there is scattering, the majority of the residual values lie on the zero (0) line, which shows that the number of

closely correct predictions is large. The BART out-sample and in-sample plots represent the predicted observations from the developed classifier using the training set, and the out-sample plot illustrates that the majority of the observations from the input data are non-default states, while the in-sample shows a clear distribution between defaults and non-defaults. The box plots in Figure 23 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults.

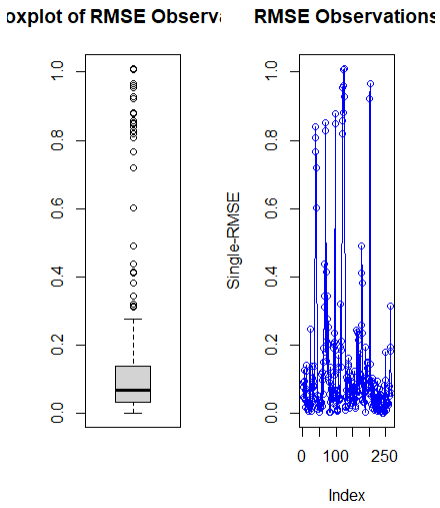

Figure 24: RMSE box plot and line plot for mean imputed global data.

From Figure 24, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.3 are seen as outliers.

Table 48: 5-fold cross-validation BART model summary when applied to the global mean imputation data set.

RMSE	0.2716
RRMSE	0.2040
ROC AUC	0.5192
Out of sample correctness	0.9210
In sample correctness	0.9352
RMSE(0.5,0.75)	(0.0669,0.1388)
Sensitivity	0.976
Specificity	0.0625
Precision	0.9420
Recall	0.976
PR AUC	0.5307
Brier Score	0.0789

From the observations in Table 48, the following can be concluded: The out-of-sample-correctness

values show that the classifier managed to predict about 92.10% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 51.92%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 97.6% at predicting non-default times and having a specificity score of 6.25%, which shows that it did not job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 94.20% of non-default states.

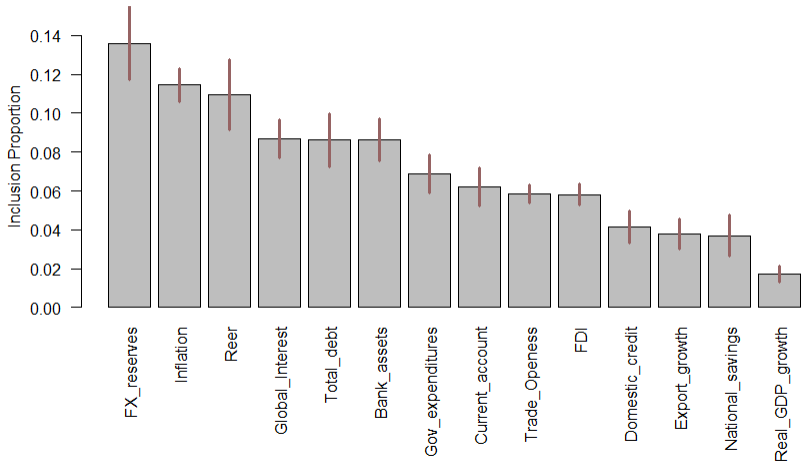

Figure 25: 5-fold cross validation BART variable importance for the global mean by region data set.

From Figure 25 it can be concluded that when considering mean imputation for the global data set, the top five variables in terms of inclusion proportion are FX reserves, inflation, REER, global interest, and total debt.

B.3.2.2 South and East Asia

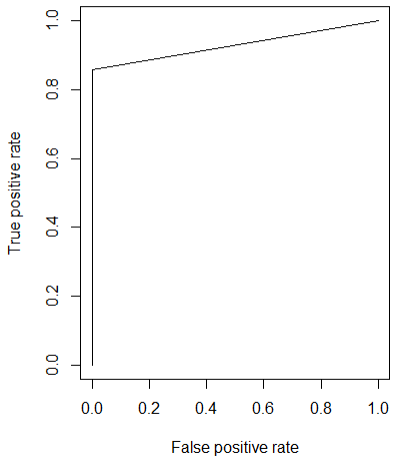

Figure 26: Area under the curve graph applied for the mean imputed South and East Asia data set.

From the observation of Figure 26 and calculation in Table 49, it can be concluded that the classifier built using the mean imputed South and East Asia data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal centre line.

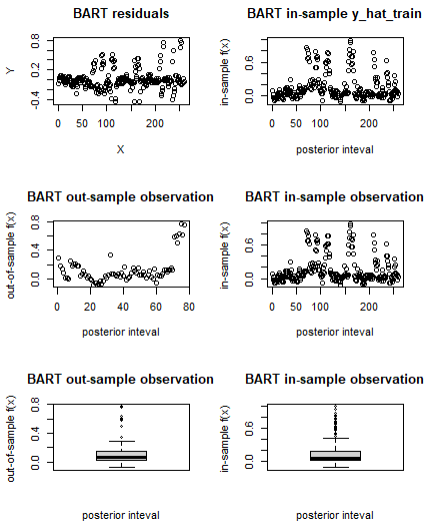

Figure 27: Plots and box plots of the model output results for training and testing mean imputed Middle East and Asia data set.

From Figure 27 the following can be concluded:The BART residual plot shows that the majority of the residual values are scattered around the zero (0) line, which shows that the number of correct predictions was off by at least an interval width of 0.2. The BART out-sample and in-sample plots represent

the predicted observations from the developed classifier using the training set, and they both illustrate that the majority of the observations from the input data are non-default states of the sovereign. The box plots in Figure 27 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults. In addition, the defaults are seen as outliers.

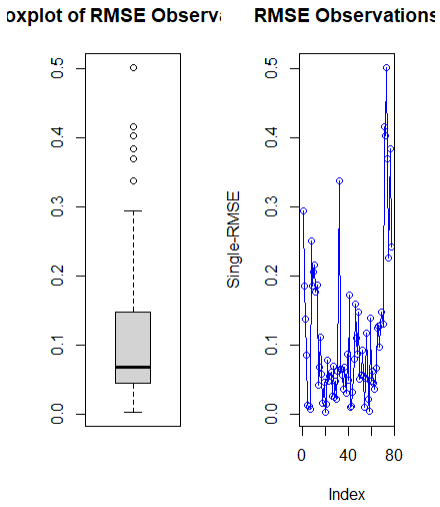

Figure 28: RMSE box plot and line plot for mean imputed South and East Asia data.

From Figure 28, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.3 are seen as outliers.

Table 49: 5-fold cross-validation BART model summary when applied to the South and East Asia mean imputation data set.

RMSE	0.1550
RRMSE	0.9412
ROC AUC	0.9285
Out of sample correctness	0.9480
In sample correctness	0.9692
RMSE(0.5,0.75)	(0.0682,0.1480)
Sensitivity	1
Specificity	0.8571
Precision	0.9859
Recall	1
PR AUC	0.5089
Brier Score	0.0129

From the observations in Table 49, the following can be concluded: The out-of-sample-correctness

values show that the classifier managed to predict about 94.80% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 92.85%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 100% at predicting non-default times and having a specificity score of 85.71%, which shows that it did a good job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 98.59% of non-default states.

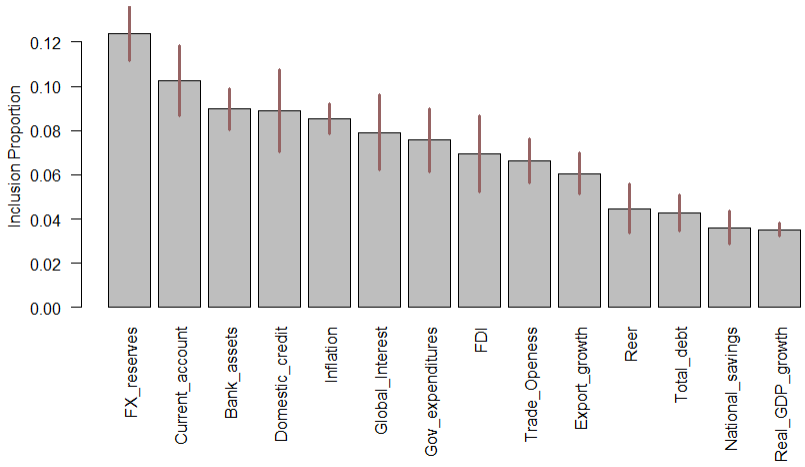

Figure 29: 5-fold cross validation BART variable importance for the South and East Asia mean by region data set.

From Figure 29 it can be concluded that when considering mean imputation for the South and East Asia data set, the top five variables in terms of inclusion proportion are FX reserves, current accounts, bank assets, domestic credit, and inflation.

B.3.2.3 Latin America

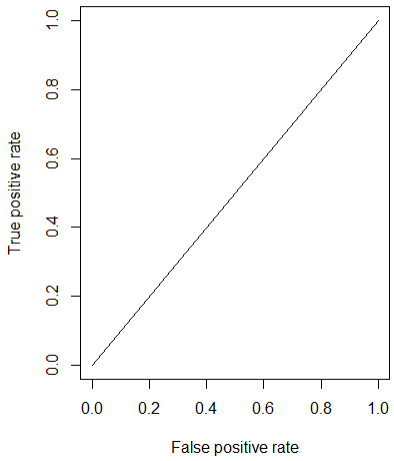

Figure 30: Area under the curve graph for the mean imputed Latin America data set.

From the observation of Figure 30 and calculation in Table 50, it can be concluded that the classifier built using the mean imputed Latin America data set produced a somewhat skilled classifier since the since ROC AUC value is 50% and the curve runs on the diagonal centre line.

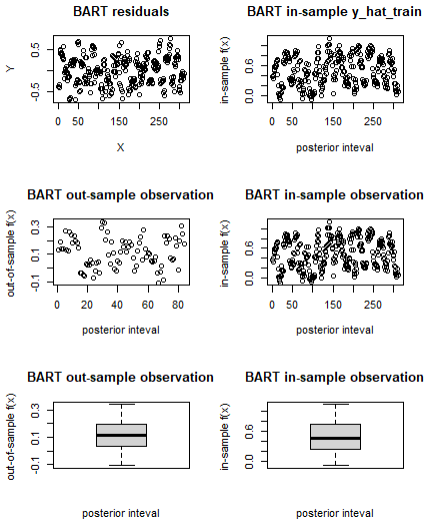

Figure 31: Plots and box plots of the model output results for training and testing mean imputed Latin America data set.

From Figure 31 the following can be concluded: The BART residual plot shows that the majority of the residual values are scattered around the zero (0) line, which shows that the classifier did not perform well since the scattering of the plot falls within the 0.5 interval. The BART out-sample and in-sample

plots represent the predicted observations from the developed classifier using the training set, and they both illustrate that all of the observations from the input data are non-default states of the sovereign, therefore showing that the classifier failed to detect defaults in terms of the out-of-sample data . The box plots on Figure 31 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults. The classifier failed to detect out of sample observation defaults.

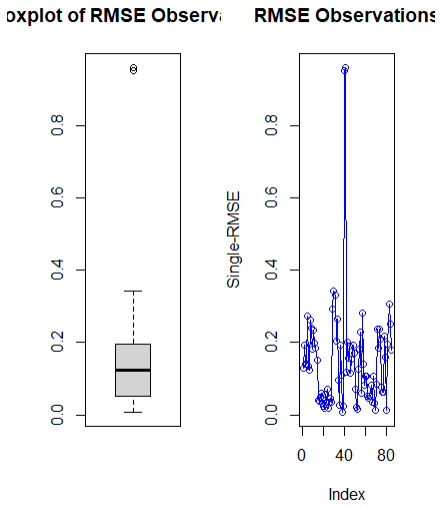

Figure 32: RMSE box plot and line plot for mean imputed Latin America data.

From Figure 32, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.4 are seen as outliers and since only two values are seen as outliers.

Table 50: 5-fold cross-validation BART model summary when applied to the Latin America mean imputation data set.

RMSE	0.2137
RRMSE	0.2070
ROC AUC	0.5
Out of sample correctness	0.9761
In sample correctness	0.875
RMSE(0.5,0.75)	(0.1237,0.1962)
Sensitivity	1
Specificity	0
Precision	Nan
Recall	Nan
PR AUC	Nan
Brier Score	0.0238

From the observations in Table 50, the following can be concluded: The classifier outputs can not be summarised since they did not have enough defaults to conclude adequate statistics.

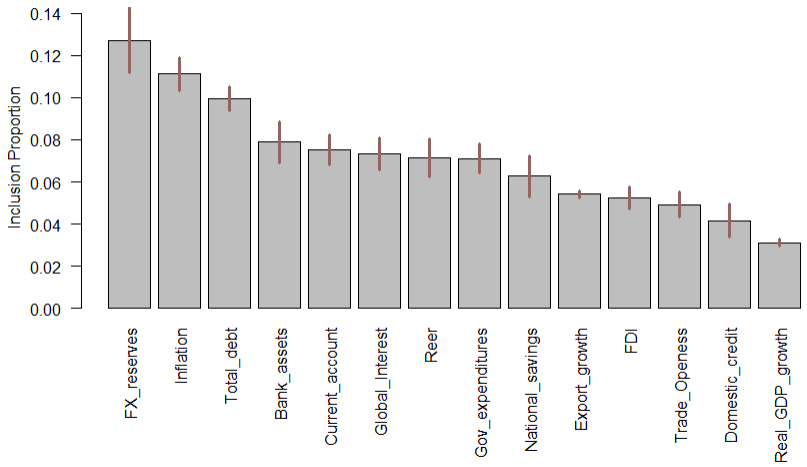

Figure 33: 5-fold cross validation BART variable importance for the Latin America mean by region data set.

From Figure 33 it can be concluded that when considering mean imputation for the Latin America data set, the top five variables in terms of inclusion proportion are FX reserves, inflation, total debt, bank assets, and current account.

B.3.2.4 The Middle East and Africa

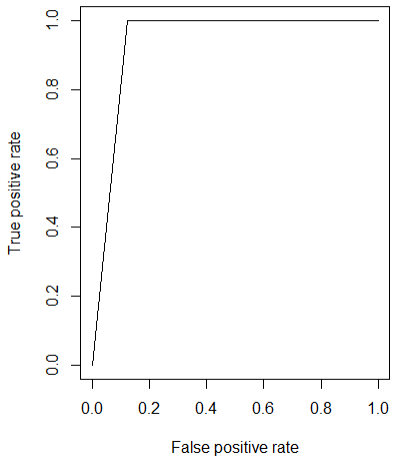

Figure 34: Area under the curve graph applied for the mean imputed Middle East and Africa data set.

From the observation of Figure 34 and calculation in Table 51, it can be concluded that the classifier built using the mean imputed Middle East and Africa data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches over the diagonal centre line.

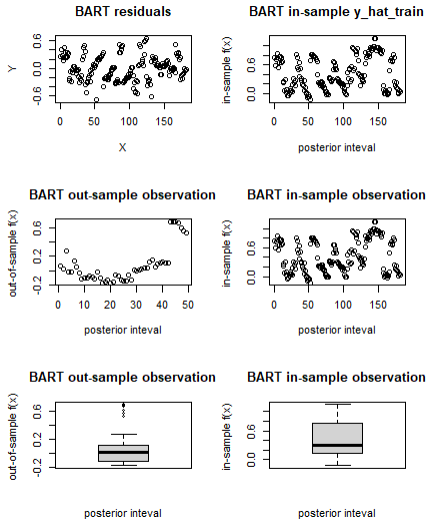

Figure 35: Plots and box plots of the model output results for training and testing mean imputed Middle East and Africa data set.

From Figure 35 the following can be concluded: The BART residual plot shows that the majority of the residual values are scattered around the zero (0) line, which shows that the classifier did not perform at its best since the scattering of the plot falls within the 0.4 interval. The BART out-sample and in-sample plots represent the predicted observations from the developed classifier using the training set, and they both illustrate that most of the observations from the input data are non-default states of the sovereign. The box plots in Figure 35 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults.

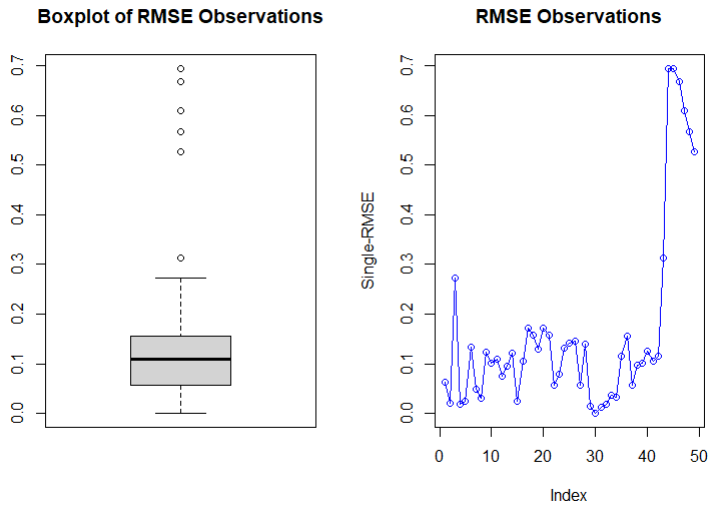

Figure 36: RMSE box plot and line plot for mean imputed Middle East and Africa data.

From Figure 36, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.3 are seen as outliers.

Table 51: 5-fold cross-validation BART model summary when applied to the Middle East and Africa mean imputation data set.

RMSE	0.2464
RRMSE	0.4785
ROC AUC	0.9375
Out of sample correctness	0.8775
In sample correctness	0.9230
RMSE(0.5,0.75)	(0.1087,0.1550)
Sensitivity	0.875
Specificity	1
Precision	1
Recall	0.875
PR AUC	0.4359
Brier Score	0.1224

From the observations in Table 51, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 87.75% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 93.75%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 87.5% at predicting non-default times and having a specificity score of 100%, which shows that it did a good job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 100% of non-default states.

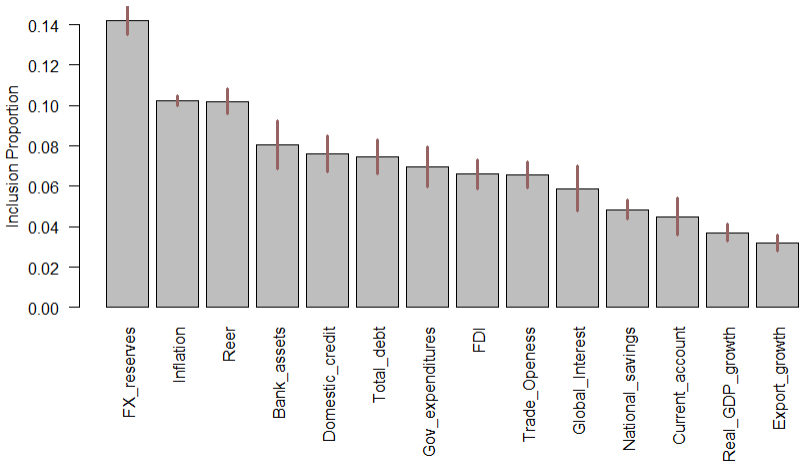

Figure 37: 5-fold cross validation BART variable importance for the Middle East and Africa mean by region data set.

From Figure 37 it can be concluded that when considering mean imputation for the Middle East and Africa data set, the top five variables in terms of inclusion proportion are FX reserves, inflation, REER, bank assets, and domestic credit.

B.3.2.5 Western Europe

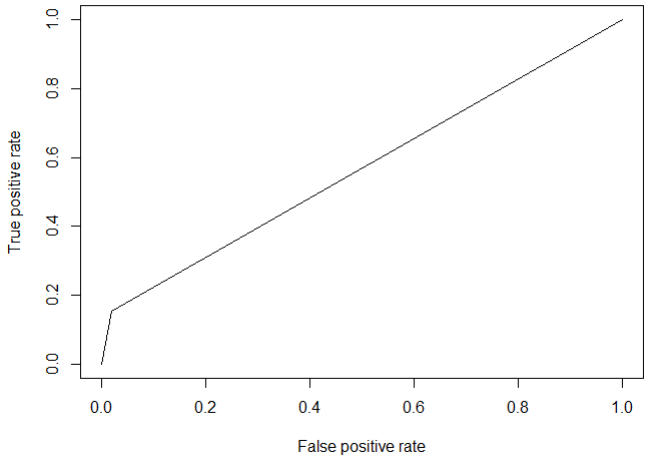

Figure 38: Area under the curve graph applied for the mean imputed Wesern Europe data set.

From the observation of Figure 38 and calculation in Table 52, it can be concluded that the classifier built using the mean imputed Wesern Europe data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches over the diagonal centre line.

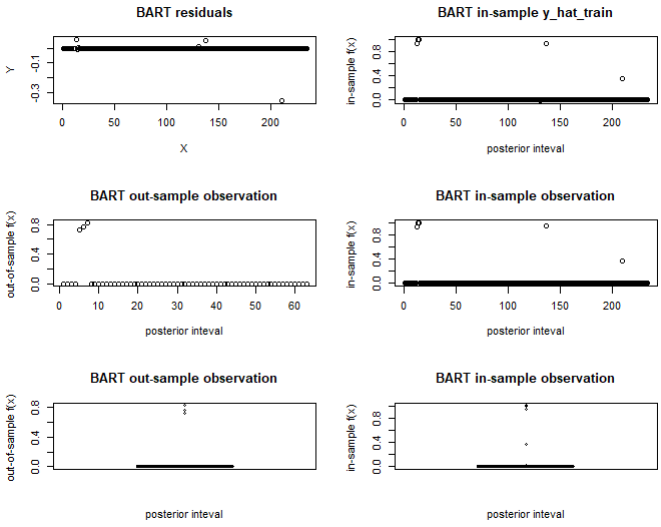

Figure 39: Plots and box plots of the model output results for training and testing mean imputed Western European data set.

Figure 39 is not included in the summary since not enough default observations were present in the data to make a conclusion.

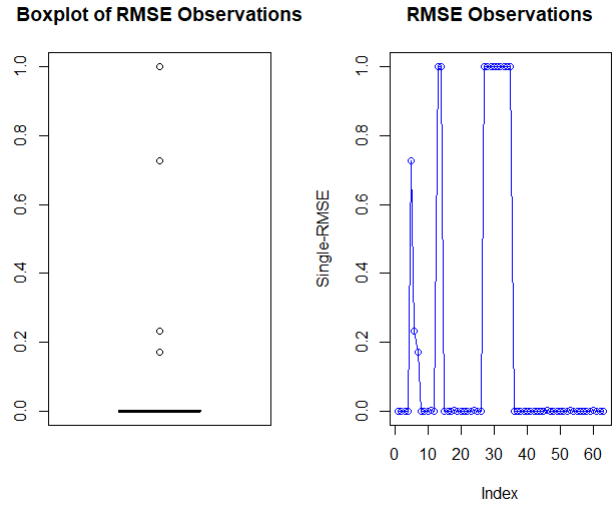

Figure 40: RMSE box plot and line plot for mean imputed Western European data.

From Figure 40, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.1 are seen as outliers.

Table 52: 5-fold cross-validation BART model summary when applied to the Western Europe mean imputation data set.

RMSE	0.4292
RRMSE	1.4703
ROC AUC	0.5669
Out of sample correctness	0.8095
In sample correctness	1
RMSE(0.5,0.75)	$(2.3040e^{-4}, 1.4192e^{-3})$
Sensitivity	0.98
Specificity	0.1538
Precision	0.8166
Recall	0.98
PR AUC	0.6136
Brier Score	0.1904

From the observations in Table 52, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 80.95% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 56.69%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 98% at predicting non-default times and having a specificity score of 15.38%, which shows that it did a bad job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 81.66% of non-default states.

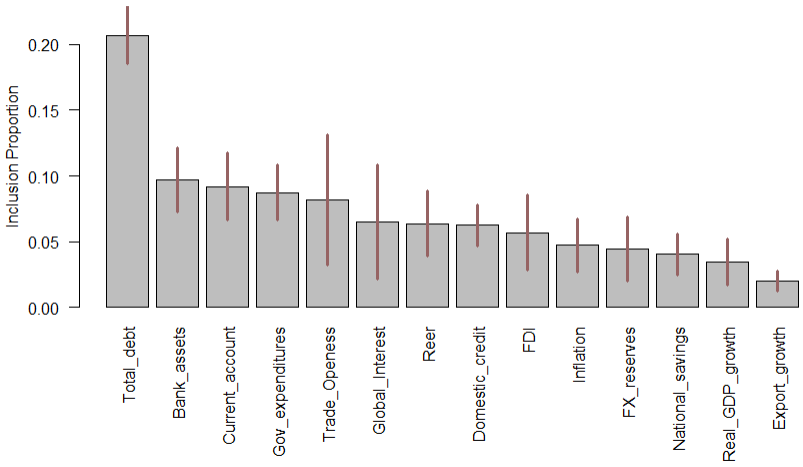

Figure 41: 5-fold cross validation BART variable importance for the Western Europe mean by region data set.

From Figure 41 it can be concluded that when considering mean imputation for the Western European data set, the top five variables in terms of inclusion proportion are total debt, bank assets, current account, government expenditure, and trade openess.

B.3.3 5-Fold Cross-Validation MICE BART Models

B.3.3.1 Global

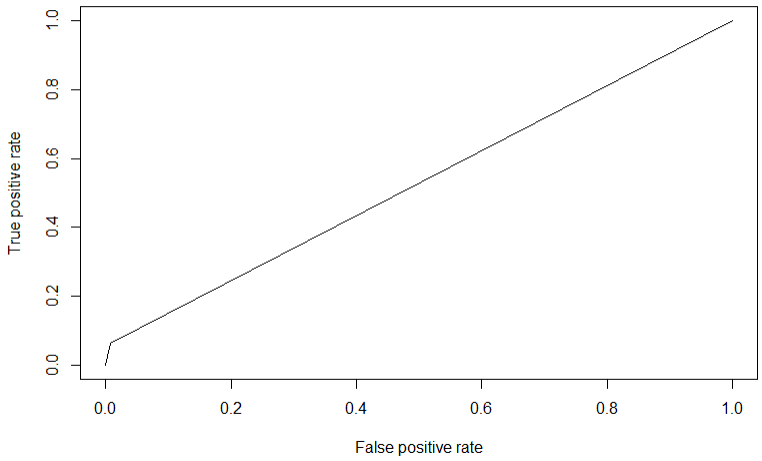

Figure 42: Area under the curve graph applied to the MICE global data set.

From the observation of Figure 42 and calculation in Table 53, it can be concluded that the classifier built using the MICE global data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches over the diagonal centre line.

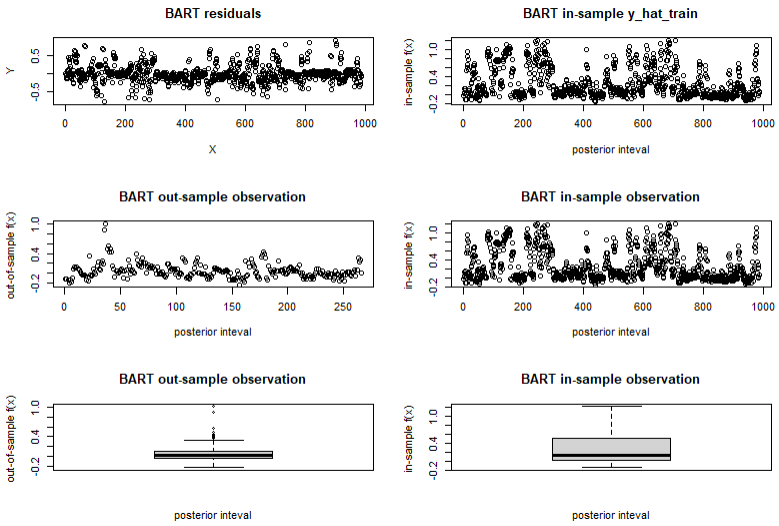

Figure 43: Plots and box plots of the model output results for training and testing MICE global data set.

From Figure 43 the following can be concluded: The BART residual plot shows that the majority of the residual values lie on the zero (0) line, which shows that the number of closely correct predictions is

large. The BART out-sample and in-sample plots represent the predicted observations from the developed classifier using the training set, and they both illustrate that the majority of the observations from the input data are non-default states of the sovereign. The box plots in Figure 43 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults. In addition, the defaults are seen as outliers.

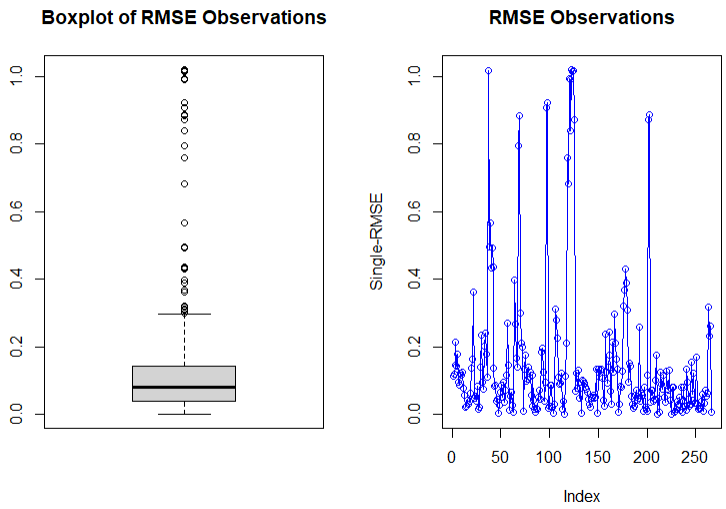

Figure 44: RMSE box plot and line plot for case-wise deletion global data.

From Figure 44, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.3 are seen as outliers.

Table 53: 5-fold cross-validation BART model summary when applied to the global MICE data set.

RMSE	0.2625
RRMSE	0.3306
ROC AUC	0.5272
Out of sample correctness	0.9361
In sample correctness	0.9402
RMSE(0.5,0.75)	(0.0799, 0.1409)
Sensitivity	0.992
Specificity	0.0625
Precision	0.9429
Recall	0.992
PR AUC	0.5498
Brier Score	0.0639

From the observations in Table 53, the following can be concluded: The out-of-sample-correctness

values show that the classifier managed to predict about 93.61.% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 52.72%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 92.2% at predicting non-default times and having a specificity score of 6.25%, which shows that it did bad a good job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 94.29% of non-default states.

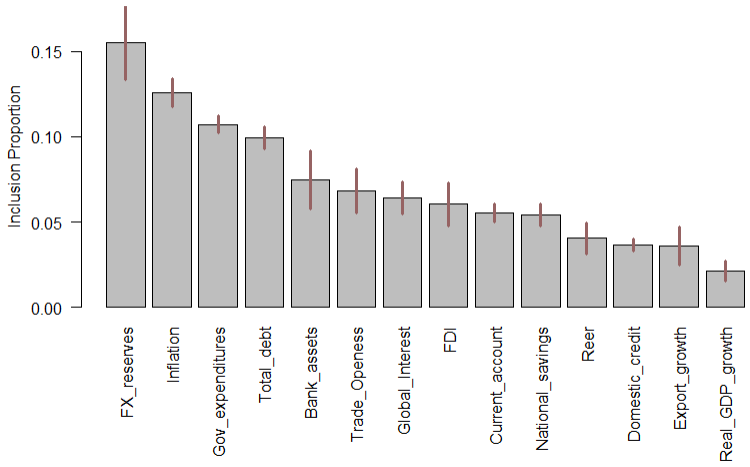

Figure 45: 5-fold cross validation BART variable importance for the global MICE data set.

From Figure 45 it can be concluded that when considering MICE for the global data set, the top five variables in terms of inclusion proportion are FX reserves, inflation, government expenditure, total debt, and bank assets.

B.3.3.2 South and East Asia

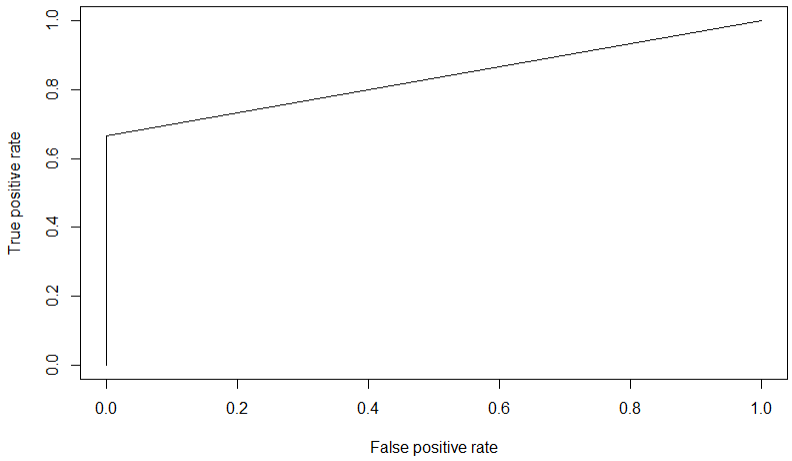

Figure 46: Area under the curve graph applied to the MICE South and East Asia data set.

From the observation of Figure 46 and calculation in Table 54, it can be concluded that the classifier built using the MICE South and East Asia data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches over the diagonal centre line.

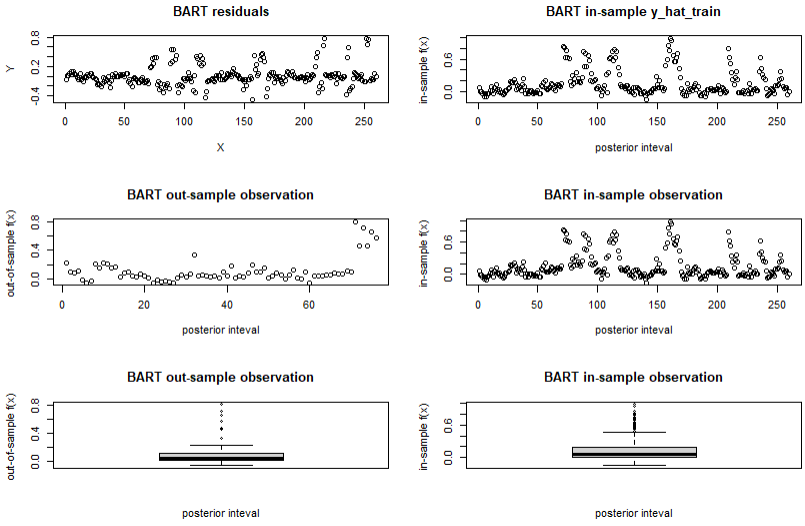

Figure 47: Plots and box plots of the model output results for training and testing MICE South and East Asia data set.

From Figure 47, the following can be concluded: The BART residual plot shows that the majority of the residual values lie close to the zero (0) line, which shows that the number of closely correct predictions is large. The BART out-sample and in-sample plots represent the predicted observations

from the developed classifier using the training set, and they both illustrate that the majority of the observations from the input data are non-default states of the sovereign. The box plots in Figure 47 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults. In addition, the defaults are seen as outliers.

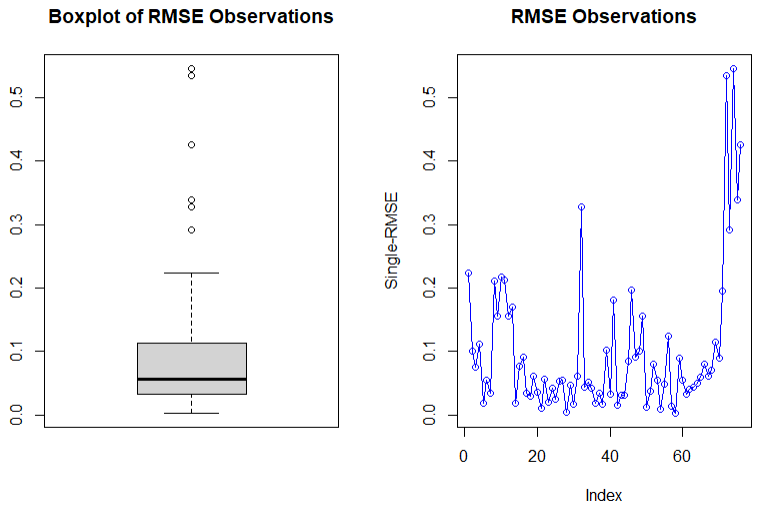

Figure 48: RMSE box plot and line plot for case-wise deletion global data.

From Figure 48, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.2 are seen as outliers.

Table 54: 5-fold cross-validation BART model summary when applied to the South and East Asia MICE data set.

RMSE	0.1486
RRMSE	0.1613
ROC AUC	0.8333
Out of sample correctness	0.9736
In sample correctness	0.9692
RMSE(0.5,0.75)	(0.0560,0.1122)
Sensitivity	1
Specificity	0.6667
Precision	0.9722
Recall	1
PR AUC	0.5203
Brier Score	0.0263

From the observations in Table 54, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 97.36% of the observations. The ROC AUC

value shows that the classifier is good as an EWS since the value is above 50%, being 83.33%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 100% at predicting non-default times and having a specificity score of 66.67%, which shows that it did a descent job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 97.22% of non-default states.

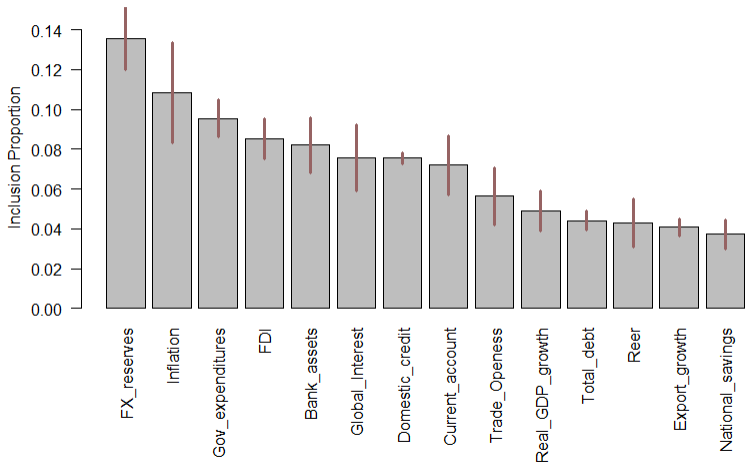

Figure 49: 5-fold cross validation BART variable importance for the South and East Asia MICE data set.

From Figure 49 it can be concluded that when considering MICE for the South and East Asia data set, the top five variables in terms of inclusion proportion are FX reserves, inflation, government expenditure, FDI, and bank assets.

B.3.3.3 Latin America

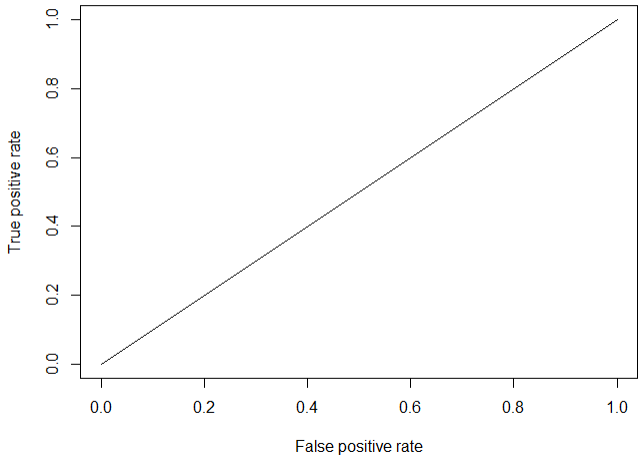

Figure 50: Area under the curve graph applied to the MICE Latin America data set.

From the observation of Figure 50 and calculation in Table 55, it can be concluded that the classifier built using the MICE Latin America data set produced a somwhat skilled classifier since the since ROC AUC value is 50% and the curve runs on the diagonal centre line.

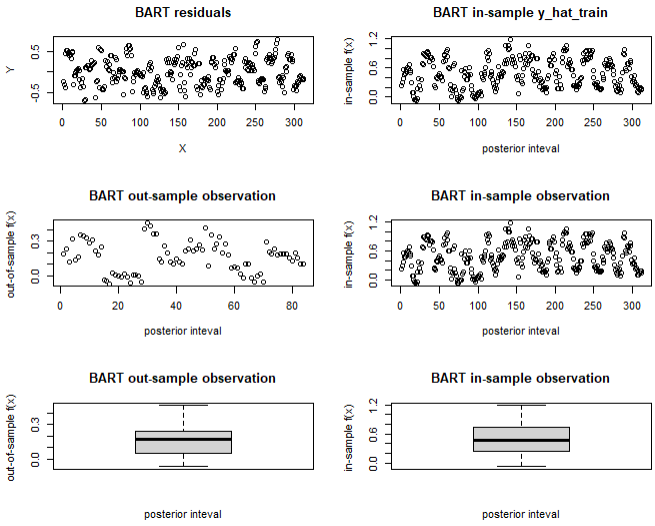

Figure 51: Plots and box plots of the model output results for training and testing MICE Latin America data set.

From Figure 51 the following can be concluded: The BART residual plot shows that the majority of the residual values are scattered around the zero (0) line, which shows that the classifier did not perform well since the scattering of the plot falls within the 0.5 interval. The BART out-sample and in-sample plots represent the predicted observations from the developed classifier using the training set, and they

both illustrate that all of the observations from the input data are non-default states of the sovereign, therefore showing that the classifier failed to detect defaults in terms of the out of sample data. The box plots in Figure 51 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults. The classifier failed to detect out of sample observations defaults.

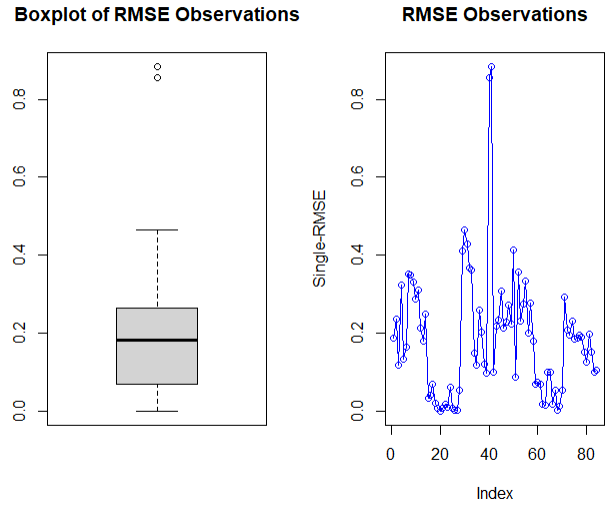

Figure 52: RMSE box plot and line plot for case-wise deletion global data.

From Figure 52, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.5 are seen as outliers.

Table 55: 5-fold cross-validation BART model summary when applied to the Latin America MICE data set.

RMSE	0.2471
RRMSE	0.1682
ROC AUC	0.5
Out of sample correctness	0.9761
In sample correctness	0.8878
RMSE(0.5,0.75)	(0.1834,0.2624)
Sensitivity	1
Specificity	0
Precision	Nan
Recall	Nan
PR AUC	Nan
Brier Score	0.0238

From the observations in Table 55, the following can be concluded: The classifier outputs can not be summarised since they did not have enough defaults to conclude adequate statistics.

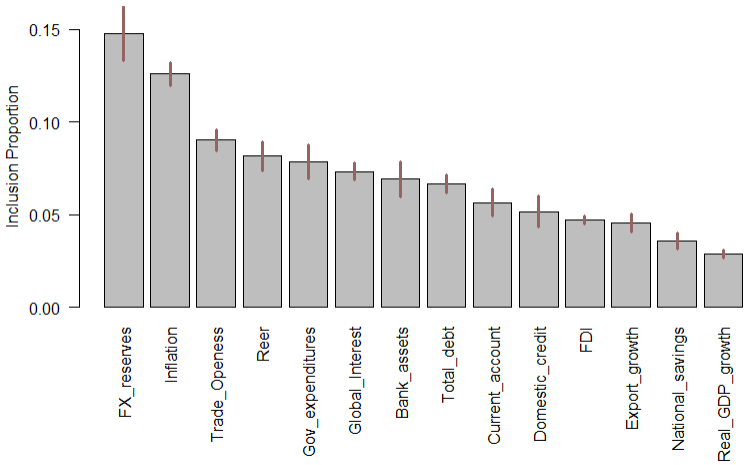

Figure 53: 5-fold cross validation BART variable importance for the Latin America MICE data set.

From Figure 53 it can be concluded that when considering MICE for the Latin America data set, the top five variables in terms of inclusion proportion are FX reserves, inflation, trade openess, REER, and government expenditure.

B.3.3.4 The Middle East and Africa

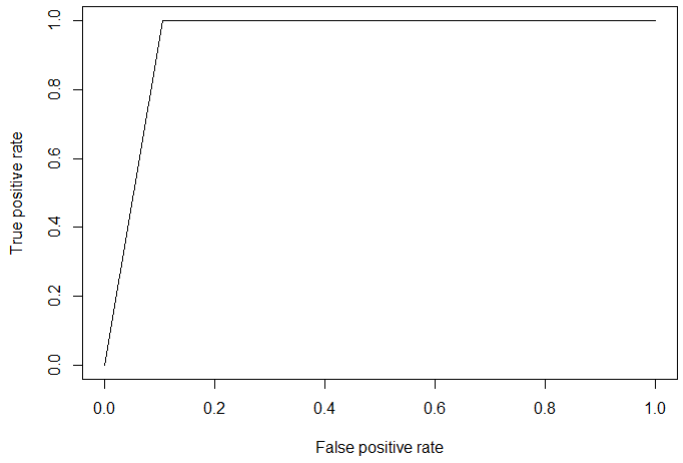

Figure 54: Area under the curve graph applied to the MICE Middle East and Africa data set.

From the observation of Figure 54 and calculation in Table 56, it can be concluded that the classifier built using the MICE Middle East and Africa data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches over the diagonal centre line.

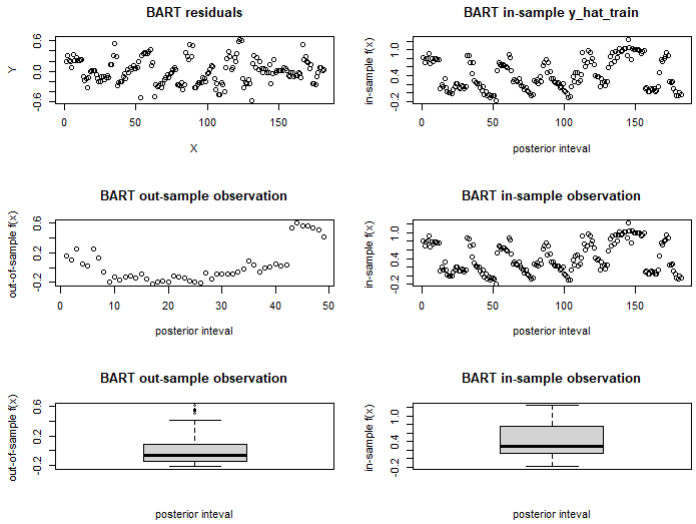

Figure 55: Plots and box plots of the model output results for training and testing MICE Middle East and Africa data set.

From Figure 55 the following can be concluded: The BART residual plot shows that the majority of the residual values are scattered around the zero (0) line, which shows that the classifier did not perform at its best since the scattering of the plot falls within the 0.5 interval. The BART out-sample and in-sample plots represent the predicted observations from the developed classifier using the training set. The out-sample illustrates that most of the observations from the input data are non-default states of the sovereign, and the in-sample shows a distribution of both default and non-default states. The box plots in Figure 55 for the out-sample and in-sample observations show a summarised distribution of the outputs, which also illustrates that the majority of the observations are non-defaults compared to defaults.

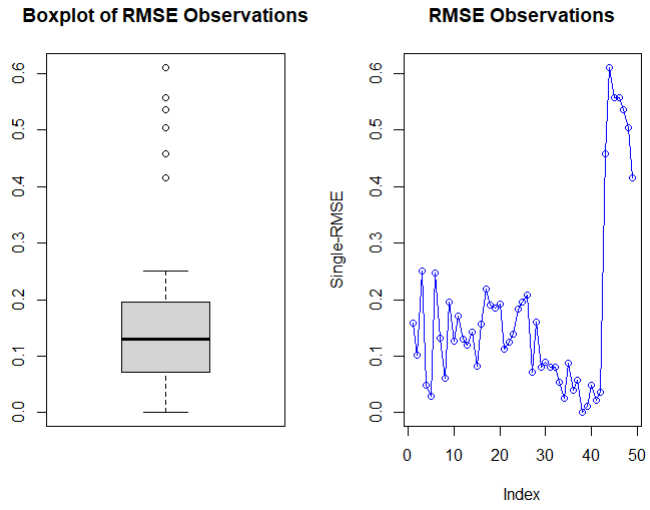

Figure 56: RMSE box plot and line plot for case-wise deletion global data.

From Figure 8, the following can be concluded: The model generated outputs close to zero (0), implying that the majority of the observations generated by the classifier were correct. Therefore, the classifier performed well since the RMSE values greater than 0.25 are seen as outliers.

Table 56: 5-fold cross-validation BART model summary when applied to the Middle East and Africa MICE data set.

RMSE	0.2332
RRMSE	1.2471
ROC AUC	0.9479
Out of sample correctness	0.8979
In sample correctness	0.9505
RMSE(0.5,0.75)	(0.1295,0.1958)
Sensitivity	0.8958
Specificity	1
Precision	1
Recall	0.8958
PR AUC	0.4431
Brier Score	0.1020

From the observations in Table 56, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 89.79% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 94.79%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 89.58% at predicting non-default times and having a specificity score of 100%, which shows that it did a good job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 100% of non-default states.

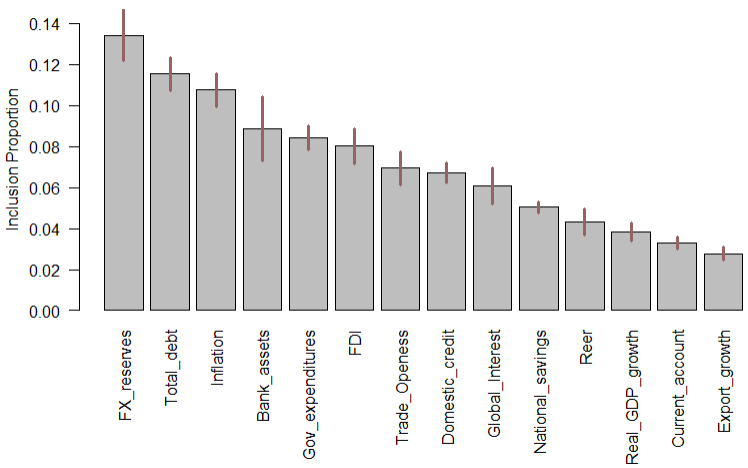

Figure 57: 5-fold cross validation BART variable importance for the Middle East and Africa MICE data set.

From Figure 57 it can be concluded that when considering MICE for the Middle East and Africa data set, the top five variables in terms of inclusion proportion are FX reserves, total debt, inflation, bank assets, and government expenditure.

B.3.3.5 Western Europe

Table 57: 5-fold cross-validation BART model summary when applied to the Western Europe MICE data set.

RMSE	Nan
RRMSE	Nan
ROC AUC	Nan
Out of sample correctness	Nan
In sample correctness	Nan
RMSE(0.5,0.75)	(Nan,Nan)
Sensitivity	Nan
Specificity	Nan
Precision	Nan
Recall	Nan
PR AUC	Nan
Brier Score	Nan

For this data set, there was no default observation in the training data set. Therefore, no model could be built since it could not be distinguished between what inputs contributed to a default.

B.3.4 Model Comparison

In this section, the regional models are being compared to the global models that were developed using 5 fold cross validation for BART.

Table 58: BART regional model comparison for the case-wise deletion data.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
RMSE	0.3015	0	0.2152	0.2140	Nan
RRMSE	1.6583	0	0.2416	0.1262	Nan
ROC AUC	0.4910	1	0.5	0.9583	Nan
Out of sample correctness	0.9091	1	0.9622	0.9696	Nan
In sample correctness	0.9648	1	0.9166	0.9655	Nan
RMSE(0.5,0.75)	(0.0630, 0.1625)	(0.06920.1132)	(0.1208,0.2158)	(0.1182,0.2102)	(Nan,Nan)
Sensitivity	0.9821	1	1	1	Nan
Specificity	0	1	0	0.9166	Nan
Precision	0.9243	1	Nan	0.9545	Nan
Recall	0.9821	1	Nan	1	Nan
PR AUC	0.5550	0.5	Nan	0.5124	Nan
Brier Score	0.0909	0	0.0377	0.0303	Nan

From the regional and global model comparison in Table 58, what can be concluded is that both the Western Europe and Latin America model had missing values in the metric scores, thereforeare the models are not considered. The two regional models that are left have ROC AUC scores higher than the global model, therefore the regional models performed better.

Table 59: BART regional model comparison for the case-wise deletion data.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
RMSE	0.2716	0.1550	0.2137	0.2464	0.4292
RRMSE	0.2040	0.1412	0.2070	0.4785	1.4703
ROC AUC	0.5192	0.9285	0.5	0.9375	0.5669
Out of sample correctness	0.9210	0.9480	0.9761	0.8775	0.8095
In sample correctness	0.9352	0.9692	0.875	0.9230	1
RMSE(0.5,0.75)	(0.0669,0.1388)	(0.0682,0.1480)	(0.1237,0.1962)	(0.1087,0.1550)	$(2.3040e^{-4}, 1.4192e^{-3})$
Sensitivity	0.976	1	1	0.875	0.98
Specificity	0.0625	0.8571	0	1	0.1538
Precision	0.9420	0.9859	Nan	1	0.8166
Recall	0.976	1	Nan	0.875	0.98
PR AUC	0.5307	0.5089	Nan	0.4359	0.6136
Brier Score	0.0789	0.0129	0.0238	0.1224	0.1904

From the regional and global model comparison in Table 59, what can be concluded is that the Latin America model had missing values in the metric scores, thereforeare the model is not considered in the comparison. The two regional models that are left have ROC AUC scores higher than the global model, therefore the regional models performed better.

Table 60: BART regional model comparison for the case-wise deletion data.

	Global	South and East Asia	Latin America	The Middle East and Africa	Western Europe
RMSE	0.2625	0.1486	0.2471	0.2332	Nan
RRMSE	0.3306	0.1613	0.1682	1.2471	Nan
ROC AUC	0.5272	0.8333	0.5	0.9479	Nan
Out of sample correctness	0.9361	0.9736	0.9761	0.8979	Nan
In sample correctness	0.9402	0.9692	0.8878	0.9505	Nan
RMSE(0.5,0.75)	(0.0799, 0.1409)	(0.0560,0.1122)	(0.1834,0.2624)	(0.1295,0.1958)	(Nan,Nan)
Sensitivity	0.992	1	1	0.8958	Nan
Specificity	0.0625	0.6667	0	1	Nan
Precision	0.9429	0.9722	Nan	1	Nan
Recall	0.992	1	Nan	0.8958	Nan
PR AUC	0.5498	0.5203	Nan	0.4431	Nan
Brier Score	0.0639	0.0263	0.0238	0.1020	Nan

From the regional and global model comparison in Table 60, what can be concluded is that both the Western Europe and Latin America model had missing values in the metric scores, thereforeare the models are not considered. The two regional models that are left have ROC AUC scores higher than the global model, therefore the regional models performed better.

B.4 5-fold Cross Validation: Neural Network

When building the neural network cross-validation models, the same metrics were considered as well as the same number of neurons. The package considered in the development of the model is the “caret” package. Since different node classifiers were developed for each data set, the best classifiers will be selected by assessing the ROC AUC value.

B.4.1 5-Fold Cross-Validation Case Wise Deletion ANN Models

Table 61: 5-fold cross-validation ANN model summary when applied to the global case wise deletion data set.

Number of nodes	6	7	8	9	10
In sample	0.9824	0.9894	0.9753	0.9947	0.9929
Out-of-sample	0.9091	0.8842	0.8677	0.8677	0.9090
Sensitivity	0.98214	0.9553	0.9375	0.9285	0.9732
Specificity	0	0	0	0.1111	0.1111
Precision	0.9243	0.9224	0.9210	0.9285	0.9316
Recall	0.9821	0.9553	0.9375	0.9285	0.9732
Brier Score	0.0909	0.1157	0.1322	0.1322	0.0909
PR AUC	0.5551	0.5270	0.5124	0.5	0.5354
ROC AUC	0.4910	0.4776	0.4687	0.5198	0.5421

From the analysis shown in Table 61, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is ten (10).

Table 62: 5-fold cross-validation ANN model summary when applied to the South and East Asia case wise deletion data set.

Number of nodes	6	7	8	9	10
In sample	0.9812	1	0.9812	0.9812	0.9812
Out-of-sample	1	0.9705	1	1	1
Sensitivity	1	0.9642	1	1	1
Specificity	1	1	1	1	1
Precision	1	1	1	1	1
Recall	1	0.9642	1	1	1
Brier Score	0	0.0294	0	0	0
PR AUC	0.5	0.4853	0.5	0.5	0.5
ROC AUC	1	0.9821	1	1	1

From the analysis shown in Table 62, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is six (6), eight (8), nine (9), and ten (10).

Table 63: 5-fold cross-validation ANN model summary when applied to the Latin America case wise deletion data set.

Number of nodes	6	7	8	9	10
In sample	0.9889	1	0.9944	0.9944	1
Out-of-sample	0.7735	0.8679	0.9245	0.7924	
Sensitivity	0.8039	0.8823	0.9607	0.8235	0.9019
Specificity	0	0.5	0	0	0
Precision	0.9534	0.9782	0.9607	0.9545	0.9583
Recall	0.8039	0.8823	0.9607	0.8235	0.9019
Brier Score	0.2264	0.1321	0.0754	0.2075	0.1320
PR AUC	0.4277	0.4474	0.5	0.4338	0.4641
ROC AUC	0.4019	0.6911	0.4803	0.4117	0.4509

From the analysis shown in Table 63, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is seven (7).

Table 64: 5-fold cross-validation ANN model summary when applied to the Middle East and Africa data set.

Number of nodes	6	7	8	9	10
In sample	0.9885	1	1	1	1
Out-of-sample	0.9696	1	1	1	1
Sensitivity	1	1	1	1	1
Specificity	0.9167	1	1	1	1
Precision	0.9545	1	1	1	1
Recall	1	1	1	1	1
Brier Score	0.0303	0	0	0	0
PR AUC	0.5124	0.5	0.5	0.5	0.5
ROC AUC	0.9583	1	1	1	1

From the analysis shown in Table 64, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is sevn (7), eight (8), nine (9), and ten (10).

Table 65: 5-fold cross-validation ANN model summary when applied to the Western Europe case wise deletion data set.

Number of nodes	6	7	8	9	10
In sample	Nan	Nan	Nan	Nan	Nan
Out-of-sample	Nan	Nan	Nan	Nan	Nan
Sensitivity	Nan	Nan	Nan	Nan	Nan
Specificity	Nan	Nan	Nan	Nan	Nan
Precision	Nan	Nan	Nan	Nan	Nan
Recall	Nan	Nan	Nan	Nan	Nan
Brier Score	Nan	Nan	Nan	Nan	Nan
PR AUC	Nan	Nan	Nan	Nan	Nan
ROC AUC	Nan	Nan	Nan	Nan	Nan

For this data set, there was no default observation in the training data set. Therefore, no model could be built since it could not be distinguished between what inputs contributed to a default.

B.4.2 5-Fold Cross-Validation Mean Imputation ANN Models

Table 66: 5-fold cross-validation ANN model summary when applied to the global mean imputation data set.

Number of nodes	6	7	8	9	10
In sample	0.9392	0.9483	0.9746	0.9585	0.9696
Out-of-sample	0.8834	0.9135	0.8984	0.9060	0.8947
Sensitivity	0.932	0.964	0.94	0.96	0.936
Specificity	0.125	0.125	0.25	0.0625	0.25
Precision	0.9433	0.9450	0.9514	0.9411	0.9512
Recall	0.932	0.964	0.94	0.96	0.936
Brier Score	0.1165	0.0864	0.1015	0.0939	0.1052
PR AUC	0.4919	0.5155	0.4919	0.5155	0.4895
ROC AUC	0.5285	0.5445	0.595	0.5112	0.593

From the analysis shown in Table 66, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is ten (10).

Table 67: 5-fold cross-validation ANN model summary when applied to the South and East Asia mean imputation data set.

Number of nodes	6	7	8	9	10
In sample	0.9884	0.9884	0.9923	1	0.9884
Out-of-sample	0.9870	1	1	0.9870	1
Sensitivity	1	1	1	0.9857	1
Specificity	0.8571	1	1	1	1
Precision	0.9859	1	1	1	1
Recall	1	1	1	0.9857	1
Brier Score	0.0129	0	0	0.0129	0
PR AUC	0.5089	0.5	0.5	0.4918	0.5
ROC AUC	0.9285	1	1	0.9928	1

From the analysis shown in Table 67, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is eight (8) only since the in-sample score make the number of nodes better than otherss.

Table 68: 5-fold cross-validation ANN model summary when applied to the Latin America mean imputation data set.

Number of nodes	6	7	8	9	10
In sample	0.9807	0.9839	0.9871	0.9967	1
Out-of-sample	0.8333	0.7142	0.8452	0.8214	0.9167
Sensitivity	0.8292	0.7195	0.8536	0.8414	0.9146
Specificity	1	0.5	0.5	0	1
Precision	1	0.9833	0.9859	0.9718	1
Recall	0.8292	0.7195	0.8536	0.8414	0.9146
Brier Score	0.1667	0.2857	0.1547	0.1785	0.0833
PR AUC	0.4223	0.3986	0.4332	0.4332	0.4507
ROC AUC	0.9146	0.6097	0.6768	0.4207	0.9573

From the analysis shown in Table 68, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is ten (10).

Table 69: 5-fold cross-validation ANN model summary when applied to the Middle East and Africa mean imputation data set.

Number of nodes	6	7	8	9	10
In sample	1	1	0.9945	0.9945	0.9945
Out-of-sample	0.8979	0.9795	0.9183	0.9183	0.9795
Sensitivity	0.9167	0.9791	0.9167	0.9166	0.9791
Specificity	0	1	1	1	1
Precision	0.9777	1	1	1	1
Recall	0.9167	0.9791	0.9167	0.9166	0.9791
Brier Score	0.1020	0.0204	0.0816	0.0816	0.0204
PR AUC	0.4601	0.4832	0.4511	0.4511	0.4832
ROC AUC	0.4583	0.9895	0.9583	0.9583	0.9895

From the analysis shown in Table 69, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is seven (7).

Table 70: 5-fold cross-validation ANN model summary when applied to the Western Europe mean imputation data set.

Number of nodes	6	7	8	9	10
In sample	1	1	1	1	1
Out-of-sample	0.682	0.8412	0.8253	0.8095	0.8253
Sensitivity	0.74	1	1	0.98	1
Specificity	0.4615	0.2307	0.1538	0.1538	0.1538
Precision	0.8409	0.8333	0.8196	0.8166	0.8196
Recall	0.74	1	1	0.98	1
Brier Score	0.3174	0.1587	0.1746	0.1904	0.1746
PR AUC	0.4594	0.6136	0.6313	0.6136	0.6313
ROC AUC	0.6007	0.6153	0.5769	0.5669	0.5769

From the analysis shown in Table 70, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is seven (7).

B.4.3 5-Fold Cross-Validation MICE ANN Models

Table 71: 5-fold cross-validation ANN model summary when applied to the global MICE data set.

Number of nodes	6	7	8	9	10
In sample	0.9574	0.9595	0.9554	0.9585	0.9746
Out-of-sample	0.9248	0.9172	0.9172	0.9172	0.8834
Sensitivity	0.976	0.976	0.972	0.96	0.936
Specificity	0.125	0	0.0625	0.25	0.0625
Precision	0.9457	0.9384	0.9418	0.9523	0.9397
Recall	0.976	0.976	0.972	0.96	0.936
Brier Score	0.0751	0.0827	0.0827	0.0827	0.1165
PR AUC	0.5266	0.5350	0.5266	0.5058	0.4972
ROC AUC	0.5505	0.488	0.5172	0.605	0.4992

From the analysis shown in Table 71, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is six (6).

Table 72: 5-fold cross-validation ANN model summary when applied to the South and East Asia MICE data set.

Number of nodes	6	7	8	9	10
In sample	0.9692	0.9923	1	1	0.9923
Out-of-sample	0.8684	0.8026	0.9868	0.9868	0.9868
Sensitivity	0.8571	0.7857	1	1	0.9857
Specificity	1	1	0.8333	0.8333	1
Precision	1	1	0.9859	0.9859	1
Recall	0.8571	0.7857	1	1	0.9857
Brier Score	0.1315	0.1973	0.0131	0.0132	0.0131
PR AUC	0.4374	0.4172	0.5095	0.5095	0.4913
ROC AUC	0.9285	0.8928	0.9166	0.9167	0.9928

From the analysis shown in Table 72, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is ten (10).

Table 73: 5-fold cross-validation ANN model summary when applied to the Latin America MICE data set.

Number of nodes	6	7	8	9	10
In sample	0.9871	0.9775	0.9871	0.9903	1
Out-of-sample	0.9167	0.9405	0.8690	0.8809	0.8452
Sensitivity	0.9146	0.9390	0.8780	0.9024	0.8536
Specificity	1	1	0.5	0	0.5
Precision	1	1	0.9863	0.9736	0.9859
Recall	0.9146	0.9390	0.8780	0.9024	0.8536
Brier Score	0.0833	0.0595	0.1309	0.1190	0.1547
PR AUC	0.4507	0.4612	0.4414	0.4557	0.4332
ROC AUC	0.9573	0.9695	0.6890	0.4512	0.6768

From the analysis shown in Table 73, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is seven (7).

Table 74: 5-fold cross-validation ANN model summary when applied to the Middle East and Africa MICE data set.

Number of nodes	6	7	8	9	10
In sample	1	1	1	1	1
Out-of-sample	0.9591	0.9183	0.9183	0.9591	0.9591
Sensitivity	0.9791	0.9375	0.9375	0.9791	0.9791
Specificity	0	0	0	0	0
Precision	0.9791	0.9782	0.9782	0.9791	0.9791
Recall	0.9791	0.9375	0.9375	0.9791	0.9791
Brier Score	0.0408	0.0816	0.0816	0.0408	0.0408
PR AUC	0.5	0.4705	0.4705	0.5	0.5
ROC AUC	0.4895	0.4687	0.4687	0.4895	0.4895

From the analysis shown in Table 74, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is six (6) and ten (10).

Table 75: 5-fold cross-validation ANN model summary when applied to the Western Europe mean imputation data set.

Number of nodes	6	7	8	9	10
In sample	1	1	1	1	1
Out-of-sample	0.8095	0.7778	0.7778	0.8412	0.8253
Sensitivity	0.98	0.9	0.88	0.94	1
Specificity	0.1538	0.3076	0.3846	0.4615	0.1538
Precision	0.8167	0.8333	0.8461	0.8703	0.8196
Recall	0.98	0.9	0.88	0.94	1
Brier Score	0.1904	0.2222	0.2222	0.1587	0.1746
PR AUC	0.6136	0.5357	0.5167	0.5357	0.6314
ROC AUC	0.5669	0.6038	0.6323	0.7007	0.5769

From the analysis shown in Table 61, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is nine (9).

B.4.4 ANN Best Model Outputs Per Neuron

A selection was made for the best performing ANN model per data set, since there we 5 models per data set, the best model was selected based on producing the best overall performance per metrics used.

Table 76: ANN regional model comparison for the case-wise deletion data.

	Global	South and East Asia model	Latin America	The Middle East and Africa	Western Europe
Nodes	10	8	7	8	Nan
In sample	0.9929	0.9812	1	1	Nan
Out-of-sample	0.9090	1	0.8679	1	Nan
Sensitivity	0.9732	1	0.8823	1	Nan
Specificity	0.1111	1	0.5	1	Nan
Precision	0.9316	1	0.9782	1	Nan
Recall	0.9732	1	0.8823	1	Nan
Brier Score	0.0909	1	0.1321	1	Nan
PR AUC	0.5354	0.5	0.4474	0.5	Nan
ROC AUC	0.5421	1	0.6911	1	Nan

From the regional and global model comparison in Table 76, what can be concluded is the regional models performed better than the global models in almost all metrics. One thing to note is that the The Middle East and Africa model built a perfect model.

Table 77: ANN regional model comparison for the mean imputation data.

	Global	South and East Asia model	Latin America	The Middle East and Africa	Western Europe
Nodes	10	8	10	7	7
In sample	0.9696	0.9923	1	1	1
Out-of-sample	0.8947	1	0.9167	0.9795	0.8412
Sensitivity	0.936	1	0.9146	0.9791	1
Specificity	0.25	1	1	1	0.2307
Precision	0.9512	1	1	1	0.8333
Recall	0.936	1	0.9146	0.9791	1
Brier Score	0.1052	0	0.0833	0.0204	0.1587
PR AUC	0.4895	0.5	0.4507	0.4832	0.6136
ROC AUC	0.593	1	0.9573	0.9895	0.6153

From the regional and global model comparison in Table 77, what can be concluded is the regional models with the exception of the Western European model performed better than the global models in almost all metrics. One thing to note is that the The Middle East and Africa model along with the South and East Asia model built a perfect model.

Table 78: ANN regional model comparison for the MICE data.

	Global	South and East Asia model	Latin America	The Middle East and Africa	Western Europe
Nodes	6	10	7	9	9
In sample	0.9574	0.9923	0.9775	1	1
Out-of-sample	0.9248	0.9868	0.9405	0.9591	0.8412
Sensitivity	0.976	0.9857	0.9390	0.9791	0.94
Specificity	0.125	1	1	0	0.4615
Precision	0.9457	1	1	0.9791	0.8703
Recall	0.976	0.9857	0.9390	0.9791	0.94
Brier Score	0.0751	0.0131	0.0595	0.0408	0.1587
PR AUC	0.5266	0.4913	0.4612	0.5	0.5357
ROC AUC	0.5505	0.9928	0.9695	0.4895	0.7007

From the regional and global model comparison in Table 78, what can be concluded is the regional models with the exception of the Western European model performed better than the global models in almost all metrics.

B.5 5-Fold Cross-Validation Model Comparison of All Techniques by Data Set

The following are the results of which 5-fold cross validation statistical technique, built the best overall model. Conclusions to which model performed best will be determined by looking at the ROC AUC score.

B.5.1 Global Data

Table 79: Model comparison summary for the case-wise deletion global date set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.8769	0.8769	0.9648	0.9472
Out-of-sample	0.9008	0.9091	0.9091	0.9008
Sensitivity	0.9732	0.9821	0.9821	0.9732
Specificity	0	0	0	0
Precision	0.9237	0.9243	0.9243	0.9237
Recall	0.9732	0.9821	0.9821	0.9732
Brier Score	0.0991	0.0910	0.0910	0.0991
PR AUC	0.5446	0.5551	0.5551	0.5446
ROC AUC	0.4866	0.4911	0.4910	0.4866

The first thing to notice from the summary in Table 79 is that all classifiers had a specificity of zero (0), implying that the models were eager to find the result of a default, even though they were not present. Even so, the models predicted rather well, and the competition was close. The probit classifier scored the highest among the other classifiers, making it the best classifier in this case. The BART classifier came in second, and the logit and ANN classifiers tied for third.

Table 80: Model comparison summary for the mean imputation global date set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.8684	0.8238	0.9352	0.9180
Out-of-sample	0.9436	0.9398	0.9210	0.9360
Sensitivity	0.996	1	0.976	0.988
Specificity	0.125	0	0.0625	0.125
Precision	0.9467	Nan	0.9420	0.9463
Recall	0.996	Nan	0.976	0.988
Brier Score	0.0563	0.0601	0.0789	0.0639
PR AUC	0.5498	Nan	0.5307	0.5396
ROC AUC	0.5605	0.5	0.5192	0.5565

When looking at Table 80, the following can be concluded; the first is that the probit model has missing values due to the lack of identifying the difference between default and non-default states of sovereigns. The logit scored the highest among the other classifiers, making it the best classifier in this case, putting ANN in second, the BART in third, and the probit in fourth.

Table 81: Model comparison summary for the MICE global date set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.8684	0.8684	0.9402	0.9119
Out-of-sample	0.9436	0.9436	0.9361	0.9323
Sensitivity	0.996	0.996	0.992	0.988
Specificity	0.125	0.125	0.0625	0.0625
Precision	0.9467	0.9467	0.9429	0.9427
Recall	0.996	0.996	0.992	0.988
Brier Score	0.0563	0.0563	0.0639	0.0676
PR AUC	0.5498	0.5498	0.5498	0.5445
ROC AUC	0.5605	0.5605	0.5272	0.5252

Table 81 shows that the probit and logit scored the highest among other classifiers, making them the best classifier in this case, the BART classifier came in second, and the ANN came in fourth.

B.5.2 South and East Asia

Table 82: Model comparison summary for the case-wise deletion South and East Asia date set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.925	0.925	1	0.9125
Out-of-sample	0.9412	0.9412	1	0.9705
Sensitivity	1	1	1	1
Specificity	0.6667	0.6667	1	0.8333
Precision	0.9333	0.9333	1	0.9655
Recall	1	1	1	1
Brier Score	0.0588	0.0588	0	0.0294
PR AUC	0.5354	0.5354	0.5	0.5164
ROC AUC	0.8333	0.8333	1	0.9166

From observing Table 82, the first standout observation is that the BART model performed exceptionally well in distinguishing the difference between default and non-default times. The BART scored the highest among the other classifiers, making it the best classifier in this case, with the ANN classifier taking second and both the probit and logit classifiers coming in third.

Table 83: Model comparison summary for the mean imputation South and East Asia data set

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	1	1	0.1480	0.9307
Out-of-sample	1	1	0.9692	1
Sensitivity	1	1	1	1
Specificity	1	1	0.8571	1
Precision	1	1	0.9859	1
Recall	1	1	1	1
Brier Score	0	0	0.0129	0
PR AUC	0.5	0.5	0.5089	0.5
ROC AUC	1	1	0.9285	1

The results from Table 83 show that the logit, probit, and ANN classifiers all had perfect scores, making any of them the best classifier in this case and also placing the BART classifier in second place to all of them.

Table 84: Model comparison summary for the MICE South and East Asia date set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.8923	0.8923	0.9692	0.9346
Out-of-sample	0.9605	0.9605	0.9736	0.9868
Sensitivity	1	1	1	1
Specificity	0.5	0.5	0.6667	0.8333
Precision	0.9589	0.9589	0.9722	0.9859
Recall	1	1	1	1
Brier Score	0.0394	0.0394	0.0263	0.0132
PR AUC	0.5325	0.5325	0.5203	0.5096
ROC AUC	0.75	0.75	0.8333	0.9167

The analysis summary in Table 84 shows that the ANN scored the highest amongst other classifiers, making it the best classifier in this case, with the BART coming in at second and the probit and logit securing third.

B.5.3 Latin America

Table 85: Model comparison summary for the case-wise deletion Latin America date set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.7944	0.8222	0.9166	0.9611
Out-of-sample	0.9434	0.9434	0.9622	0.9245
Sensitivity	0.9804	0.9803	1	0.9411
Specificity	0	0	0	0.5
Precision	0.9615	0.9615	Nan	0.9795
Recall	0.9803	0.9803	Nan	0.9411
Brier Score	0.0566	0.0566	0.0377	0.0754
PR AUC	0.5187	0.5187	Nan	0.4740
ROC AUC	0.4902	0.4902	0.5	0.7205

The following can be seen when looking at the results from Table 85, the following can be seen: The BART model failed to distinguish between default and non-default times for sovereigns. The ANN scored the highest among the other classifiers, making it the best classifier in this case; the BART is second, and the probit and logit are third.

Table 86: Model comparison summary for the mean imputation Latin America.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.7433	0.7724	0.875	0.9583
Out-of-sample	0.9643	0.9643	0.9761	0.9881
Sensitivity	0.9878	0.9878	1	0.9878
Specificity	0	0	0	1
Precision	0.9759	0.9759	Nan	1
Recall	0.9878	0.9878	Nan	0.9878
Brier Score	0.0357	0.0357	0.0238	0.0119
PR AUC	0.5131	0.5131	Nan	0.4897
ROC AUC	0.4939	0.4939	0.5	0.9939

The following can be seen when looking at the results from Table 86, the following can be seen; The BART model failed to distinguish between default and non-default times for sovereigns. The ANN scored the highest among the other classifiers, making it the best classifier in this case; the BART came in second and the logit and probit in third.

Table 87: Model comparison summary for the MICE Latin America date set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.7722	0.7722	0.8878	0.9551
Out-of-sample	0.7980	0.7980	0.9761	0.9404
Sensitivity	1	1	1	0.9390
Specificity	0	0	0	1
Precision	Nan	Nan	Nan	1
Recall	Nan	Nan	Nan	0.9390
Brier Score	0.0238	0.0238	0.0238	0.0595
PR AUC	Nan	Nan	Nan	0.4612
ROC AUC	0.5	0.5	0.5	0.9695

From the Table 87, the only model that could differentiate between default and non-default is the ANN. All other models failed, putting the ANN at the top and the BART, probit, and logit all in second place.

B.5.4 The Middle East and Africa

Table 88: Model comparison summary for the case-wise deletion the Middle East and Africa date set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.9770	0.9770	0.9655	0.9770
Out-of-sample	0.9697	0.9697	0.9697	1
Sensitivity	1	1	1	1
Specificity	0.9166	0.9166	0.9166	1
Precision	0.9545	0.9545	0.9545	1
Recall	1	1	1	1
Brier Score	0.0303	0.0303	0.0303	0
PR AUC	0.5124	0.5124	0.5124	0.5
ROC AUC	0.9583	0.9583	0.9583	1

From observing the results in Table 88, the following can be concluded: the ANN scored the highest amongst the other classifiers, making it the best classifier in this case. The BART, logit, and probit classifiers are all in second place.

Table 89: Model comparison summary for the mean imputation the Middle East and Africa data set

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.9231	0.9065	0.9231	0.9670
Out-of-sample	0.8776	0.8776	0.8775	1
Sensitivity	0.875	0.875	0.875	1
Specificity	1	1	1	1
Precision	1	1	1	1
Recall	0.875	0.875	0.875	1
Brier Score	0.1224	0.1224	0.1224	0
PR AUC	0.4359	0.4359	0.4359	0.5
ROC AUC	0.9375	0.9375	0.9375	1

Table 5.57 shows that the ANN scored the highest amongst other classifiers, making it the best classifier in this case, with the BART, logit, and probit all tied for second place.

Table 90: Summary of model comparisons for the MICE Middle East and Africa date set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.9395	0.9395	0.9505	0.9945
Out-of-sample	0.9591	0.9591	0.8979	0.9795
Sensitivity	0.9791	0.9791	0.8958	1
Specificity	0	0	1	0
Precision	0.9791	0.9791	1	Nan
Recall	0.9791	0.9791	0.8958	Nan
Brier Score	0.0408	0.0408	0.1020	0.0204
PR AUC	0.5	0.5	0.4431	Nan
ROC AUC	0.4895	0.4895	0.9479	0.5

From the analysis summary in Table 90, it is clear that the BART scored the highest among the other classifiers, making it the best classifier in this case; the ANN came in second, and both the probit and logit came in third.

B.5.5 Western Europe

Table 91: Model comparison summary for the case-wise deletion Western Europe date set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	0.6315	0.6315	Nan	1
Out-of-sample	1	1	Nan	0.6316
Sensitivity	1	1	Nan	1
Specificity	0	0	Nan	0
Precision	Nan	Nan	Nan	Nan
Recall	Nan	Nan	Nan	Nan
Brier Score	0.3684	0.3684	Nan	0.3684
PR AUC	Nan	Nan	Nan	Nan
ROC AUC	Nan	Nan	Nan	0.5

From the results in the Table91, the models could not distinguish between default and non-defaults states of sovereign since no default information was available.

Table 92: Model comparison summary for the mean imputation Western Europe data set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	1	1	1	0.9829
Out-of-sample	0.7619	0.7619	0.8095	0.7936
Sensitivity	0.9	0.9	0.98	1
Specificity	0.2307	0.2307	0.1538	0
Precision	0.8182	0.8182	0.8166	Nan
Recall	0.9	0.9	0.98	Nan
Brier Score	0.2381	0.2381	0.1904	0.2063
PR AUC	0.5462	0.5462	0.6136	Nan
ROC AUC	0.5653	0.5653	0.5669	0.5

From the analysis in Table 92 it is clear that the BART model performed best, putting the logit and probit in second place and the ANN in fourth. The ANN also failed to distinguish between default and non-default times.

Table 93: Model comparison summary for the MICE Western Europe date set.

	5-fold logit	5-fold probit	5-fold BART	5-fold ANN
In sample	1	1	Nan	0.9957
Out-of-sample	0.7301	0.7301	Nan	0.7778
Sensitivity	0.88	0.88	Nan	0.98
Specificity	0.1538	0.1538	Nan	0
Precision	0.8	0.8	Nan	0.7903
Recall	0.88	0.88	Nan	0.98
Brier Score	0.2698	0.2698	Nan	0.2222
PR AUC	0.5462	0.5462	Nan	0.6511
ROC AUC	0.5169	0.5169	Nan	0.49

From the analysis summary in Table 93, the BART and ANN models failed to generate a model, therefore putting them at last place, coming first to the log and probit.

Appendix C: BART Analysis

This part of the appendix is a detailed illustration of the analysis done using the BART classifier on all data sets. The summary and comparison of the outputs can be found in chapter 5.

C.1 Case Wise Deletion BART Models

C.1.1 Global Data

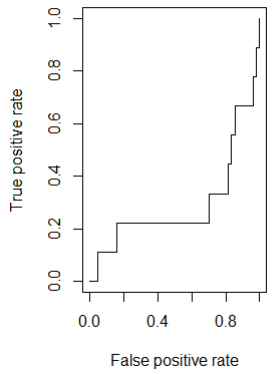

Figure 58: ROC curve applied to the case-wise deletion global data.

From the observation of Figure 58 and calculation in Table 94, it can be concluded that the classifier built using the case wise deletion global data set produced a non-skilled classifier since the since ROC AUC value is 50% and the curve arches below the diagonal line centre.

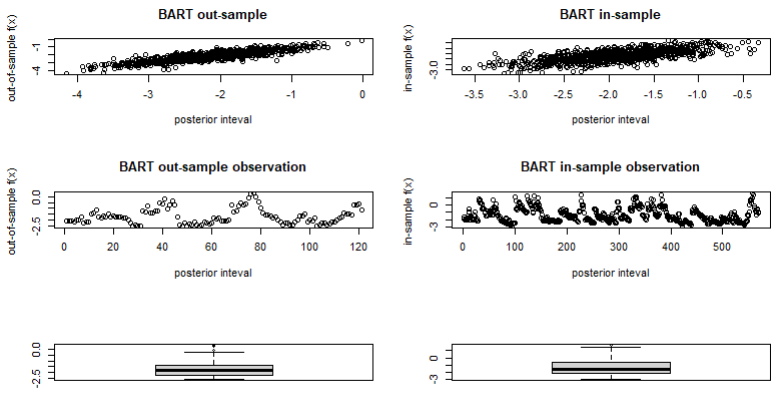

Figure 59: Plot and box plot of the model output results for training and testing case-wise deletion global data.

From Figure 59 the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults. The in and out of sample plots show that the model generated distributions are non-default times only, suggesting that the developed classifier failed to predict defaults. The box plot illustrates that their classifier failed to pick up the default.

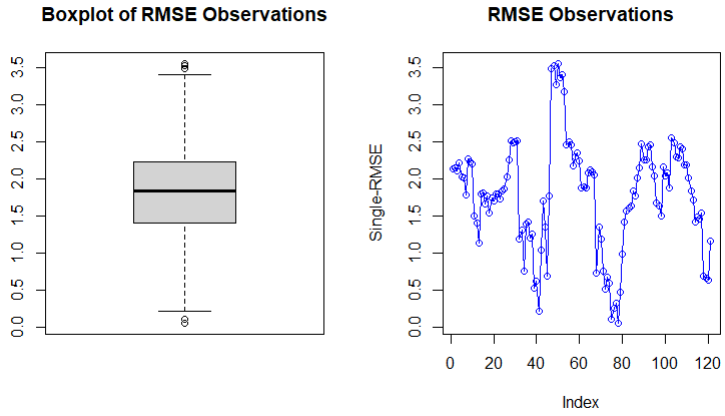

Figure 60: RMSE box plot and line plot for case-wise deletion global data.

The following can be deduced from Figure 60: The majority of RMSE observations are found between the range of 1 and 2.5. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default. As such, values close to zero (0) are seen as outliers.

Table 94: BART model summary of the global case-wise deletion data set.

RMSE	1.9537
RRMSE	0.1031
ROC AUC	0.5
Out of sample correctness	0.9256
In sample correctness	0.8717
RMSE(0.5,0.75)	(1.8377, 2.2202)
Sensitivity	0.9739
Specificity	0.5
Precision	0.9256
Recall	0.9739
PR AUC	0.5295
Brier Score	0.0902

From the observations in Table 94, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 92.56% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is 50%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 97.39% at predicting non-default times and having a specificity score of 50%, which shows that it did a descent job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 92.56% of non-default states.

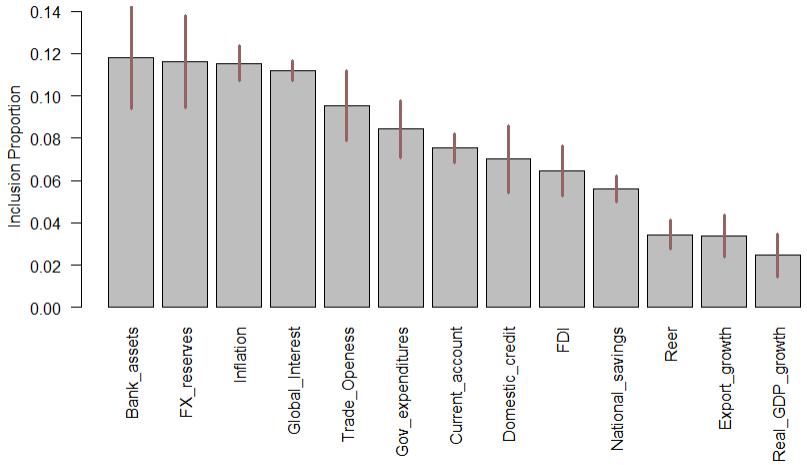

Figure 61: BART variable importance for the global case wise deletion data set.

From Figure 61 it can be concluded that when considering case wise deletion for the global data set, the top five variables in terms of inclusion proportion are bank assets, FX reserves, inflation, global interest, and trade openess.

C.1.2 South and East Asia Case Wise Deletion

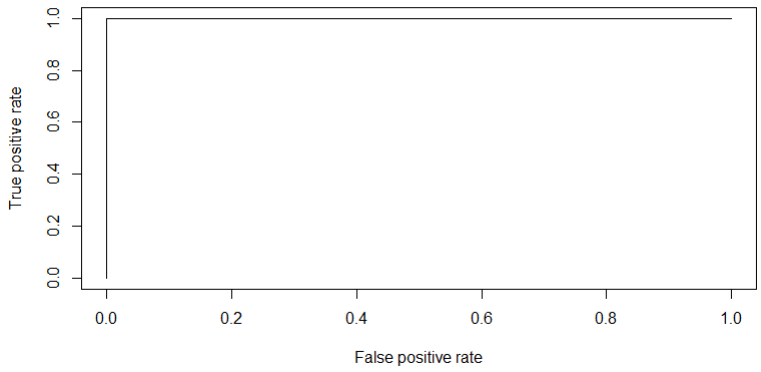

Figure 62: ROC curve applied to the case-wise deletion South and East Asia data.

From the observation of Figure 62 and calculation in Table 95, it can be concluded that the classifier built using the case wise deletion South and East Asia data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

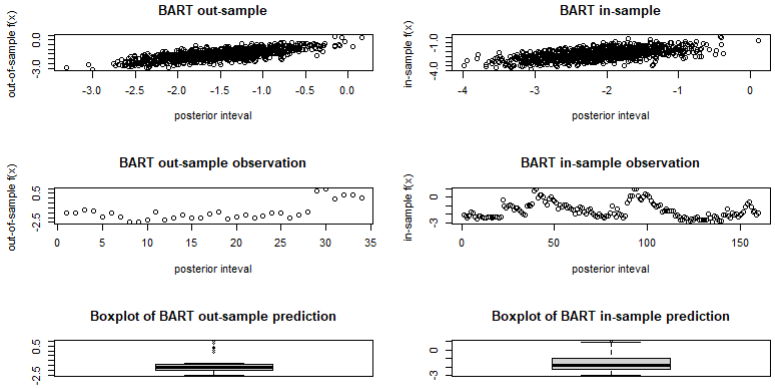

Figure 63: Plot and box plot of the model output results for training and testing case-wise deletion South and East Asia data.

From Figure 63, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults. The in and out of sample plots show that the model generated distributions are mostly non-default times, suggesting that the developed classifier failed to predict defaults. The box plot illustrates that the classifier predicted some defaults.

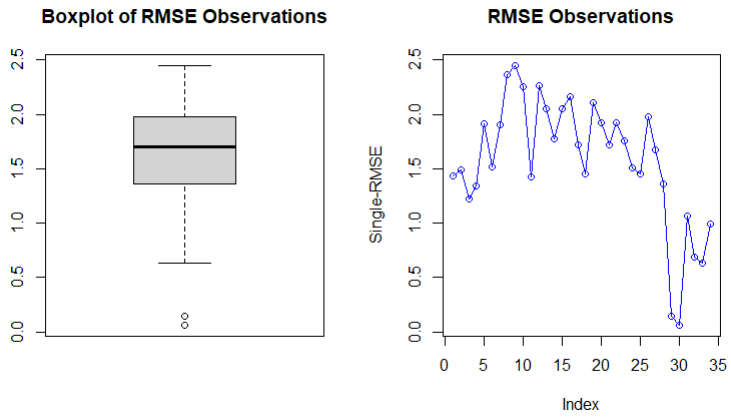

Figure 64: RMSE box plot and line plot for case-wise deletion South and East Asia data.

The following can be deduced from Figure 64: The majority of RMSE observations are found between the range of 1 and 2.5. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default. As such, values close to zero (0) are seen as outliers.

Table 95: BART model summary of the South and East Asia case-wise deletion data set.

RMSE	1.6915
RRMSE	0.2040
ROC AUC	0.6667
Out of sample correctness	0.8823
In sample correctness	0.875
RMSE(0.5,0.75)	(1.697, 1.9678)
Sensitivity	0.875
Specificity	0.7142
Precision	0.875
Recall	0.875
PR AUC	0.5
Brier Score	0.1739

From the observations in Table 95, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 88.23% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 66.67%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 87.5% at predicting non-default times and having a specificity score of 71.42%, which shows that it did a good job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 87.5% of non-default states.

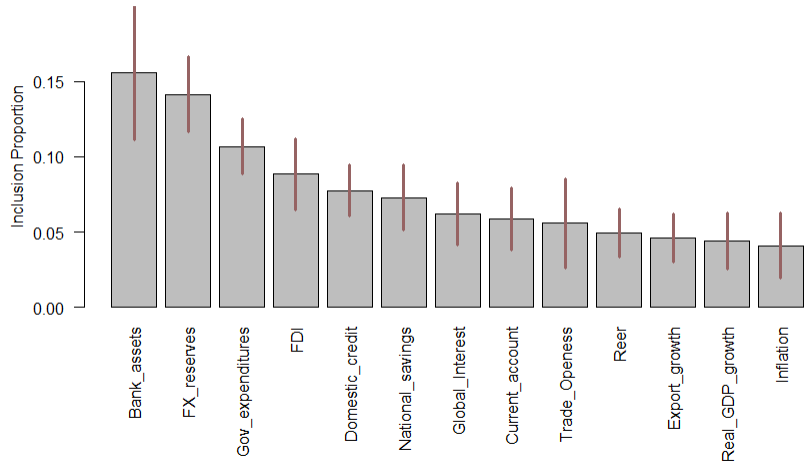

Figure 65: BART variable importance for the South and East Asia case wise deletion data set.

From Figure 65, it can be concluded that when considering case wise deletion for the South and East Asia data set, the top five variables in terms of inclusion proportion are bank assets, FX reserves, government expenditure, FDI, and domestic credit.

C.1.3 Latin America Case Wise Deletion

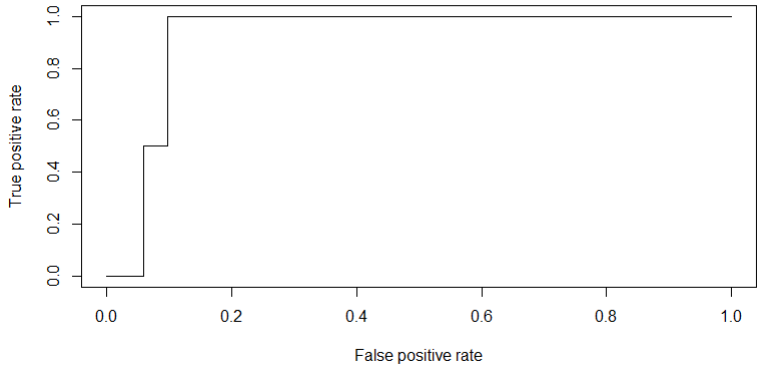

Figure 66: ROC curve applied to the case-wise deletion Latin America data.

From the observation of Figure 66 and calculation in Table 96, it can be concluded that the classifier built using the case wise deletion Latin America data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

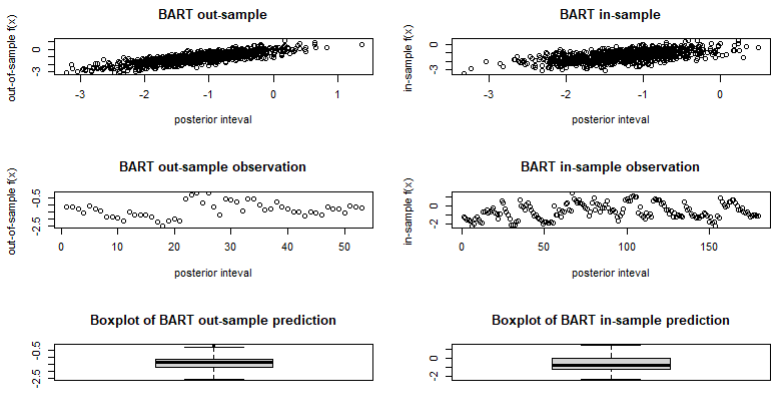

Figure 67: Plot and box plot of the model output results for training and testing case-wise deletion Latin America data.

From Figure 67, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults. The in and out of sample plots show that the model generated distributions are mostly non-default times since the observations are below the benchmark. The box plot depicting in and out of sample shows that defaults are uncommon.

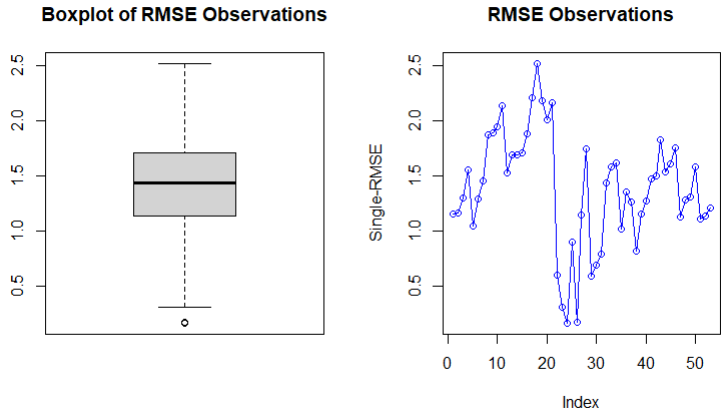

Figure 68: RMSE box plot and line plot for case-wise deletion Latin America data.

The following can be deduced from Figure 68: The majority of RMSE observations are found between the range of 1 and 2.5. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default. As such, values close to zero (0) are seen as outliers.

Table 96: BART model summary of the Latin America case-wise deletion data set.

RMSE	1.4787
RRMSE	0.1506
ROC AUC	0.9215
Out of sample correctness	0.9622
In sample correctness	0.7944
RMSE(0.5,0.75)	(1.4359, 1.7069)
Sensitivity	0.9444
Specificity	0.8181
Precision	0.9622
Recall	0.9444
PR AUC	0.4920
Brier Score	0.0769

From the observations in Table 96, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 92.15% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 92.15%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 94.44% at predicting non-default times and having a specificity score of 81.81%, which shows that it did a good job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 96.22% of non-default states.

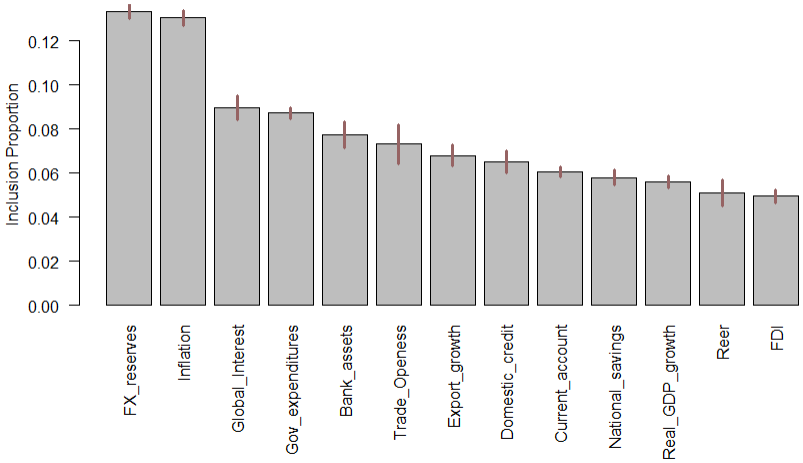

Figure 69: BART variable importance for the Latin America case wise deletion data set.

From Figure 69 it can be concluded that when considering case wise deletion for the Latin America data set, the top five variables in terms of inclusion proportion are FX reserves, inflation, global interest, government expenditure, and bank assets.

C.1.4 The Middle East and Africa Case Wise Deletion

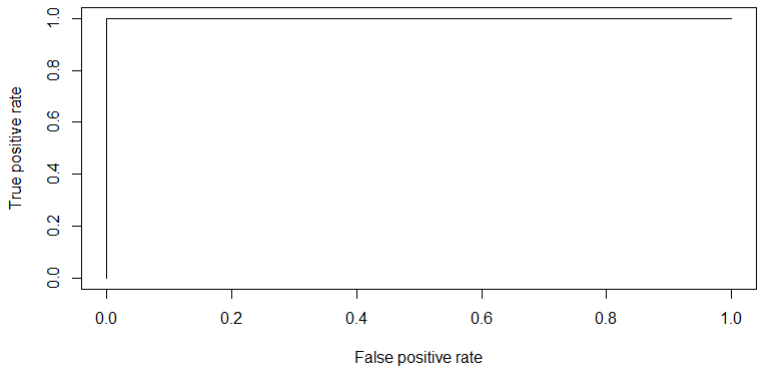

Figure 70: ROC curve applied to the case-wise deletion Middle East and Africa data.

From the observation of Figure 70 and calculation in Table 97, it can be concluded that the classifier built using the case wise deletion Middle East and Africa data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

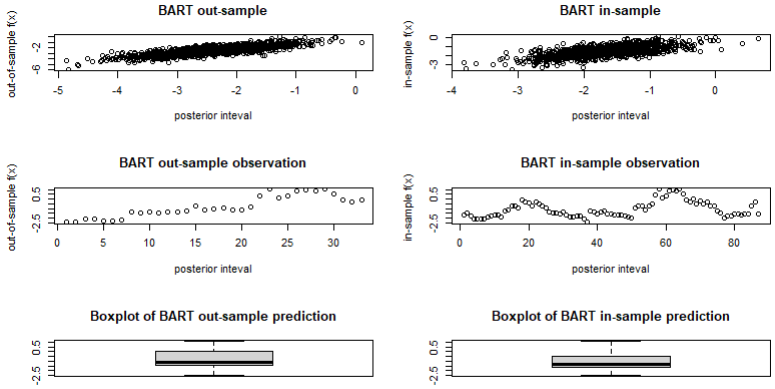

Figure 71: Plot and box plot of the model output results for training and testing case-wise deletion The Middle East and Africa data.

From Figure 71, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults. The in and out of sample plots show that the model generated distributions are mostly non-default times since the observations are below the benchmark. The box plot depicting in and out of sample shows that defaults are uncommon.

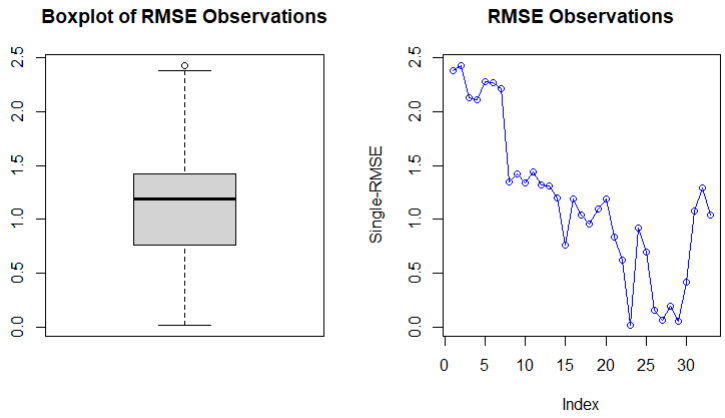

Figure 72: RMSE box plot and line plot for case-wise deletion The Middle East and Africa data.

The following can be deduced from Figure 72: The majority of RMSE observations are found between the range of 0.5 and 1.5. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default. As such, values close to zero (0) are seen as outliers.

Table 97: BART model summary of the Middle East and Africa case-wise deletion data set.

RMSE	1.366
RRMSE	0.2871
AUC	1
Out of sample correctness	0.8181
In sample correctness	0.8505
RMSE(0.5,0.75)	$(1.18869, 1.4222)$
Sensitivity	1
Specificity	0.5
Precision	0.7778
Recall	1
PR AUC	0.5918
Brier Score	0.1818

From the observations in Table 97, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 81.81% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 100%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 100% at predicting non-default times and having a specificity score of 50%, which shows that it did a descent job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 77.78% of non-default states.

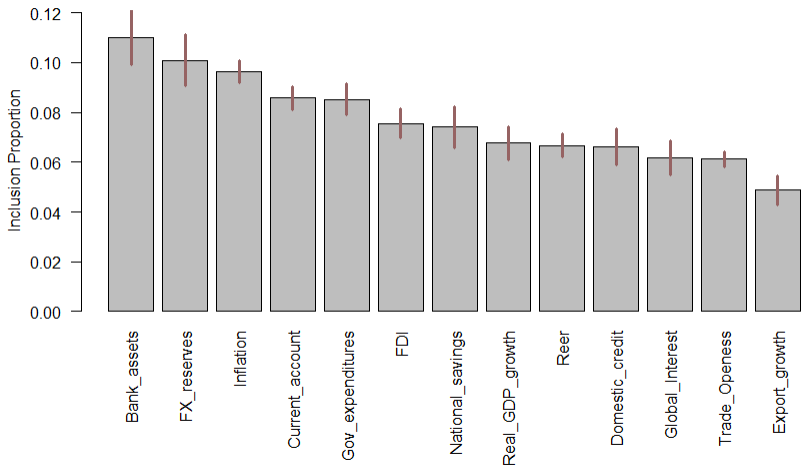

Figure 73: BART variable importance for the Middle East and Africa case wise deletion data set.

From Figure 73, it can be concluded that when considering case wise deletion for the Middle East and Africa data set, the top five variables in terms of inclusion proportion are bank assets, FX reserves, inflation, current account, and government expenditure.

C.2 Mean Imputation BART Models

C.2.1 Global Data

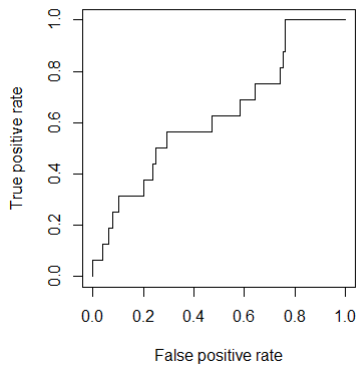

Figure 74: ROC curve applied to the mean imputated global data.

From the observation of Figure 74 and calculation in Table 98, it can be concluded that the classifier built using the mean imputed global data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

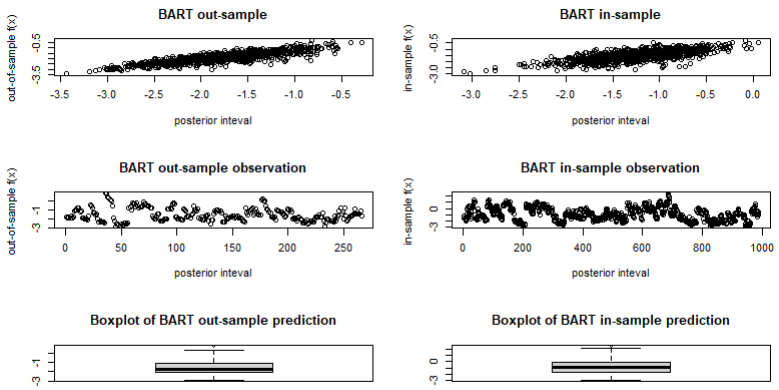

Figure 75: Plot and box plot of the model output results for training and testing mean imputation global data.

From Figure 75, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults. The in and out of sample plots show that the model generated distributions are mostly non-default times since the observations are below the benchmark. The box plot illustrates that defaults are seen as outliers.

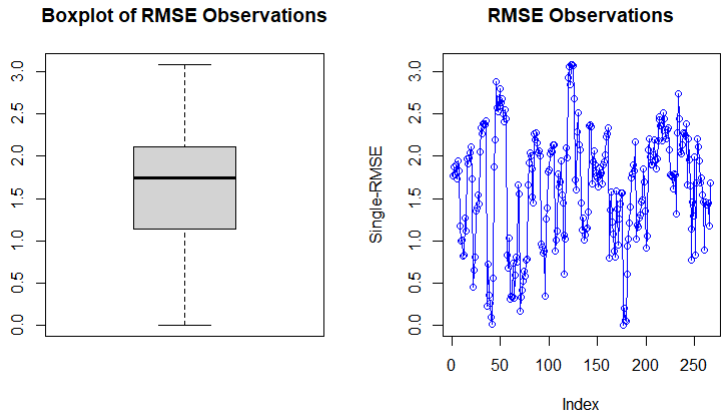

Figure 76: RMSE box plot and line plot for mean imputation global data.

The following can be deduced from Figure 76: The majority of RMSE observations are found between the range of 1 and 2.5. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default.

Table 98: BART model summary of the global mean imputation data set.

RMSE	1.7632
RRMSE	0.0694
ROC AUC	0.6265
Out of sample correctness	0.9398
In sample correctness	0.8117
RMSE(0.5,0.75)	(1.7446, 2.1128)
Sensitivity	0.996
Specificity	0.4
Precision	0.9431
Recall	0.996
PR AUC	0.5554
Brier Score	0.0601

From the observations in Table 98, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 93.98% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 62.65%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 99.6% at predicting non-default times and having a specificity score of 40%, which shows that it did a bad job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 94.31% of non-default states.

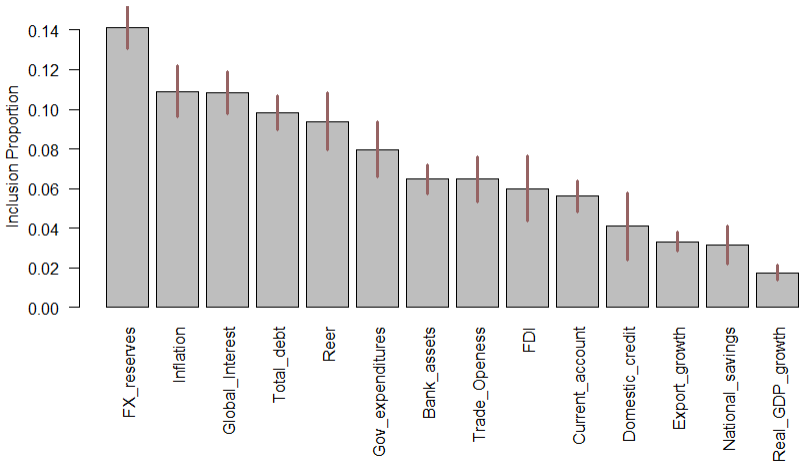

Figure 77: BART variable importance for the global mean by region data set.

From Figure 77 it can be concluded that when considering mean imputation for the global data set, the top five variables in terms of inclusion proportion are FX reserves, inflation, global interest, total debt, and REER.

C.2.2 South and East Asia

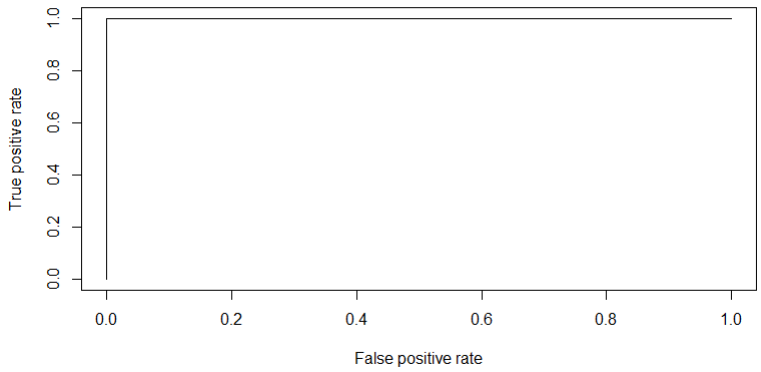

Figure 78: ROC curve applied to the mean imputated South and East Asia data.

From the observation of Figure 78 and calculation in Table 99, it can be concluded that the classifier built using the mean imputed South and East Asia data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

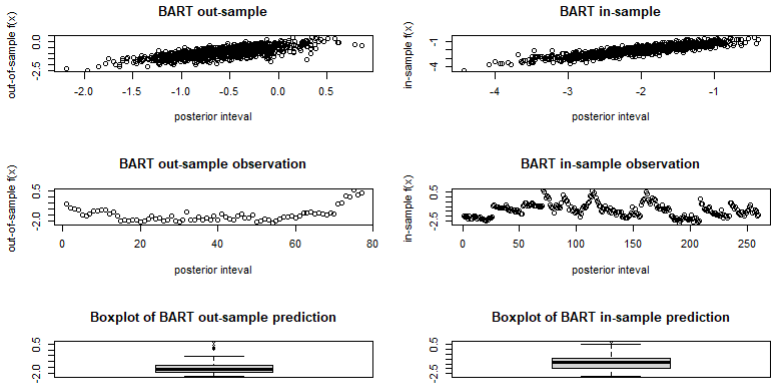

Figure 79: Plot and box plot of the model output results for training and testing mean imputation South and East Asia data.

From Figure 79, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults. The in and out of sample plots show that the model generated distributions are mostly non-default times since the observations are below the benchmark. The box plot illustrates that defaults are seen as outliers.

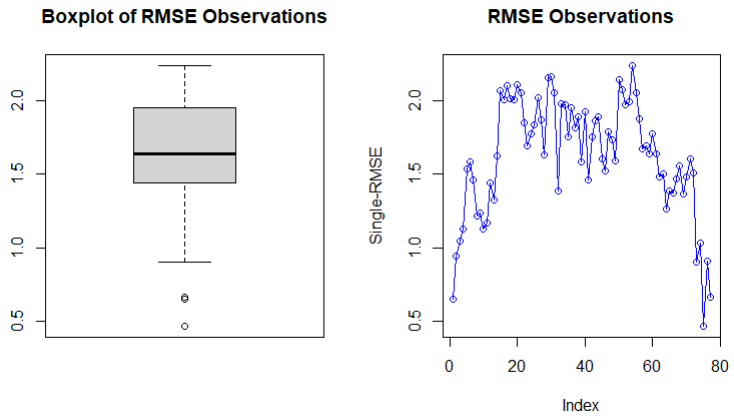

Figure 80: RMSE box plot and line plot for mean imputation South and East Asia data.

The following can be deduced from Figure 80: The majority of RMSE observations are found between the range of 1 and 2. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default. As such, values close to zero (0) are seen as outliers.

Table 99: BART model summary of the South and East Asia mean imputation data set.

RMSE	1.6692
RRMSE	0.1246
AUC	1
Out of sample correctness	0.9220
In sample correctness	0.8576
RMSE(0.5,0.75)	$(1.6356, 1.9535)$
Sensitivity	1
Specificity	0.9589
Precision	0.9210
Recall	1
PR AUC	0.5736
Brier Score	0.0779

From the observations in Table 99, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 92.20% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 100%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 100% at predicting non-default times and having a specificity score of 95.89%, which shows that it did a good job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 92.10% of non-default states.

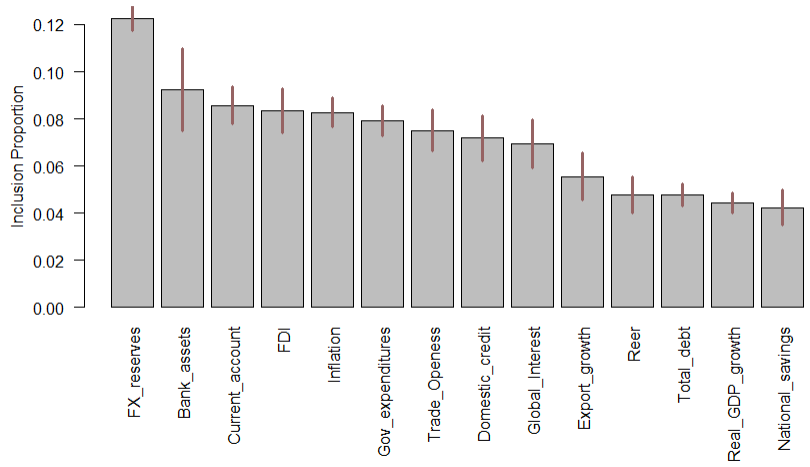

Figure 81: BART variable importance for the South and East Asia mean by region data set.

From Figure 81 it can be concluded that when considering mean imputation for the South and East Asia data set, the top five variables in terms of inclusion proportion are FX reserves, bank assets, current account, FDI, and inflation.

C.2.3 Latin America

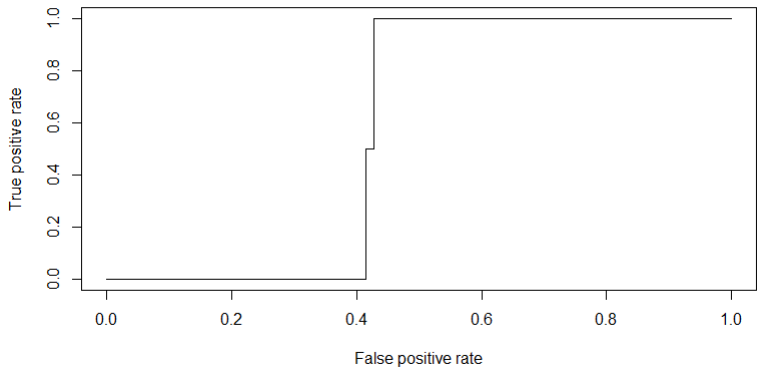

Figure 82: ROC curve applied to the mean imputated Latin America data.

From the observation of Figure 82 and calculation in Table 100, it can be concluded that the classifier built using the mean imputed South and East Asia data set produced a non-skilled classifier since the since ROC AUC value is above 50% and the curve arches below the diagonal line centre.

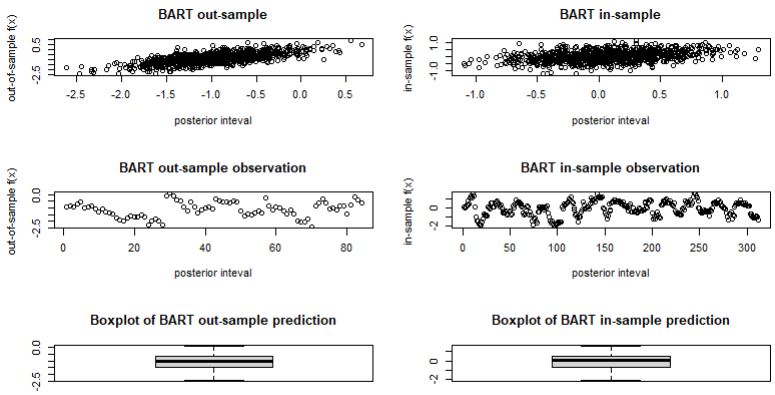

Figure 83: Plot and box plot of the model output results for training and testing mean imputation Latin America data.

From Figure 83, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults. The in and out of sample plots show that the model generated distributions are mostly non-default times since the observations are below the 0.5 benchmark. The box plot depicting in and out of samples shows that defaults are uncommon.

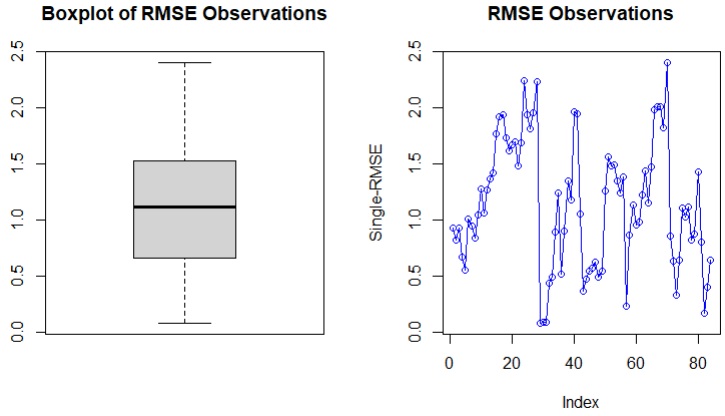

Figure 84: RMSE box plot and line plot for mean imputation Latin America data.

The following can be deduced from Figure 84: The majority of RMSE observations are found between the range of 0.5 and 2. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default.

Table 100: BART model summary of the Latin America mean imputation data set.

RMSE	1.2742
RRMSE	0.1250
ROC AUC	0.8914
Out of sample correctness	0.9761
In sample correctness	0.75
RMSE(0.5,0.75)	(1.1108, 1.5061)
Sensitivity	0.9647
Specificity	0.8181
Precision	0.9761
Recall	0.9647
PR AUC	0.4938
Brier Score	0.0520

From the observations in Table 100, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 97.61% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 89.14%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 96.47% at predicting non-default times and having a specificity score of 81.81%, which shows that it did a great job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 97.61% of non-default states.

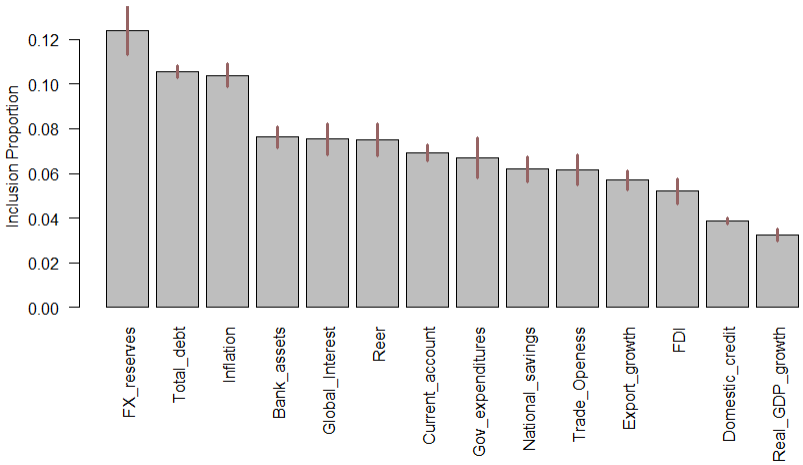

Figure 85: BART variable importance for the Latin America mean by region data set.

From Figure 85 it can be concluded that when considering mean imputation for the Latin America data set, the top five variables in terms of inclusion proportion are FX reserves, total debt, inflation, bank assets, and global interest.

C.2.4 The Middle East and Africa

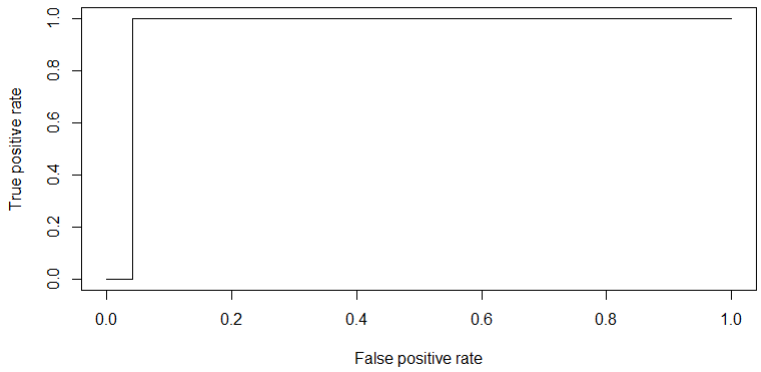

Figure 86: ROC curve applied to the mean imputated Middle East and Africa data.

From the observation of Figure 86 and calculation in Table 101, it can be concluded that the classifier built using the mean imputed South and East Asia data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

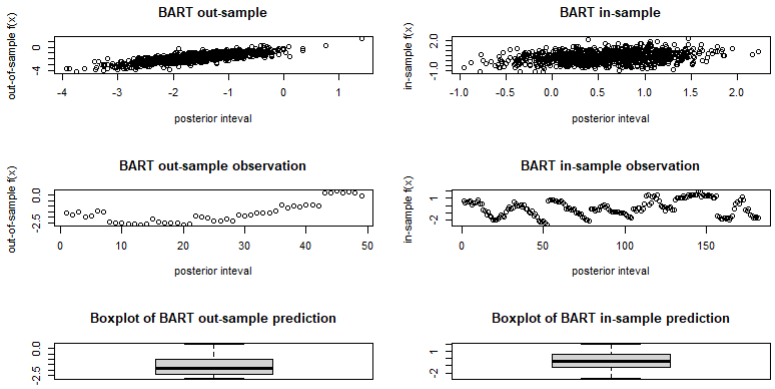

Figure 87: Plot and box plot of the model output results for training and testing mean imputation The Middle East and Africa data.

From Figure 87, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults, though unlike the other data sets, this one has quite a few observations on the positive side. The in and out of sample plots show that the model generated distributions are mostly non-default times since the observations are below the 0.5 benchmark. In addition, the model also classified a number of defaults in the in-sample data set. The box plots for both in and out of sample observations also show that the majority of outputs are non-default states of sovereigns.

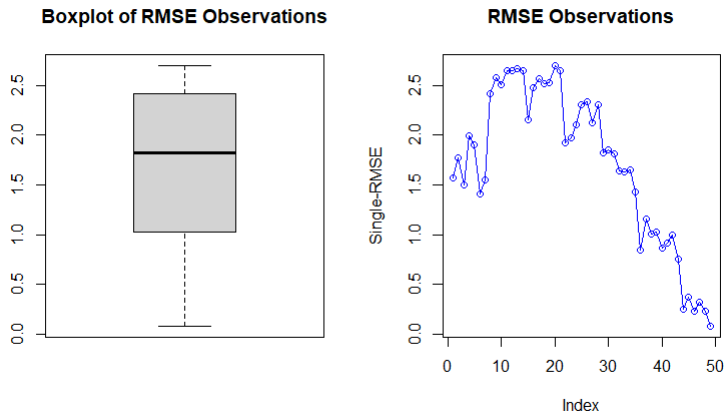

Figure 88: RMSE box plot and line plot for mean imputation the Middle East and Africa data.

The following can be deduced from Figure 88: The majority of RMSE observations are found between the range of 1 and 2.5. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default. As such, values close to zero (0) seem to strecth the lower tail, showing that there is an uneven spread of correctly classified RME values.

Table 101: BART model summary of the Middle East and Africa mean imputation data set.

RMSE	1.8706
RRMSE	0.1654
AUC	0.9583
Out of sample correctness	0.9795
In sample correctness	0.8186
RMSE(0.5,0.75)	$(1.8245, 2.4161)$
Sensitivity	0.9411
Specificity	0.9
Precision	0.9795
Recall	0.9411
PR AUC	0.4834
Brier Score	0.0655

From the observations in Table 101, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 97.95% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 95.83%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 94.11% at predicting non-default times and having a specificity score of 90%, which shows that it did a good job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 94.11% of non-default states.

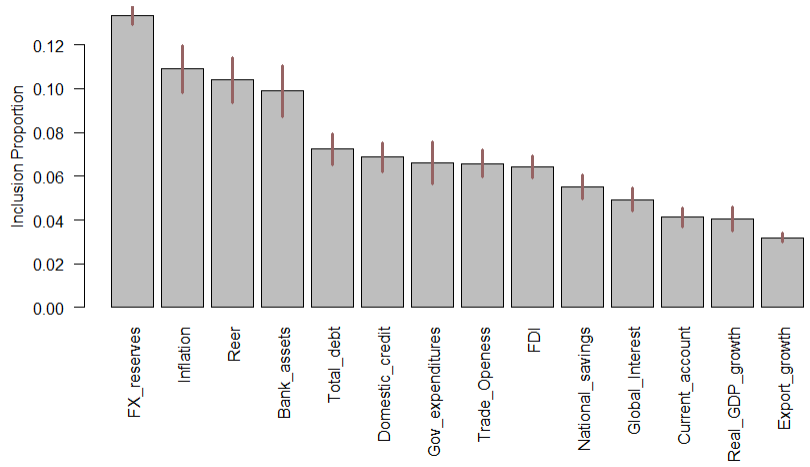

Figure 89: BART variable importance for the Middle East and Africa mean by region data set.

From Figure 89 it can be concluded that when considering mean imputation for the Middle East and Africa data set, the top five variables in terms of inclusion proportion are FX reserves, inflation, REER, bank assets, and total debt.

C.2.5 Western Europe

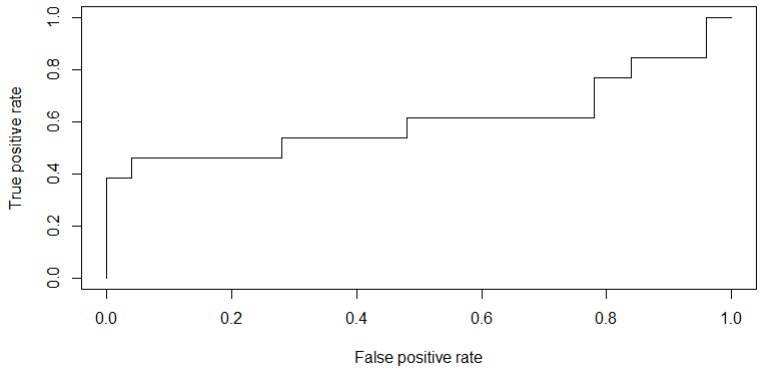

Figure 90: ROC curve applied to the mean imputated Western Europe data.

From the observation of Figure 90 and calculation in Table 102, it can be concluded that the classifier built using the mean imputed Western European data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

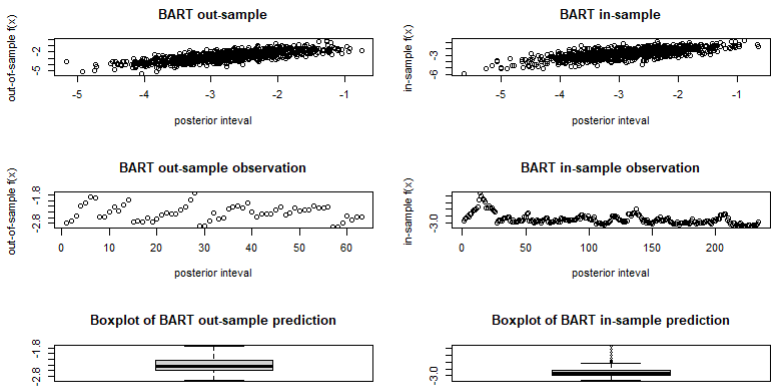

Figure 91: Plot and box plot of the model output results for training and testing mean imputation Western Europe data.

From Figure 91, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults. The in and out of sample plots show that the model generated distributions are all non-default times, suggesting that the developed classifier failed to predict defaults. The box plot illustrates that the classifier failed to develop an adequate model to predict defaults.

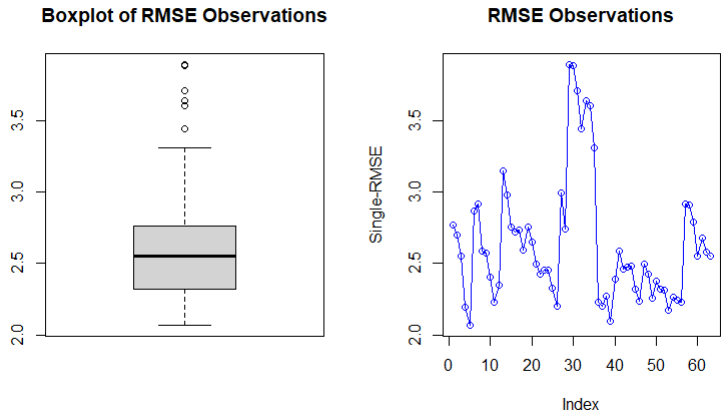

Figure 92: RMSE box plot and line plot for mean imputation Western Europe data.

The following can be deduced from Figure 92: The majority of RMSE observations are found between the range of 2 and 3. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default. In this case the classifier was failing to classify the different states of sovereigns, since not enough information was available to develop an appropriate EWS.

Table 102: BART model summary of the Western Europe mean imputation data set.

RMSE	2.6688
RRMSE	0.1385
AUC	0.6061
Out of sample correctness	0.7936
In sample correctness	0.9829
RMSE(0.5,0.75)	(2.5505, 2.7630)
Sensitivity	0
Specificity	1
Precision	*Na*
Recall	*Na*
PR AUC	*Na*
Brier Score	0.2933

From the observations in Table 102, the following can be concluded: The classifier outputs can not be summarised since they did not have enough defaults to conclude adequate statistics.

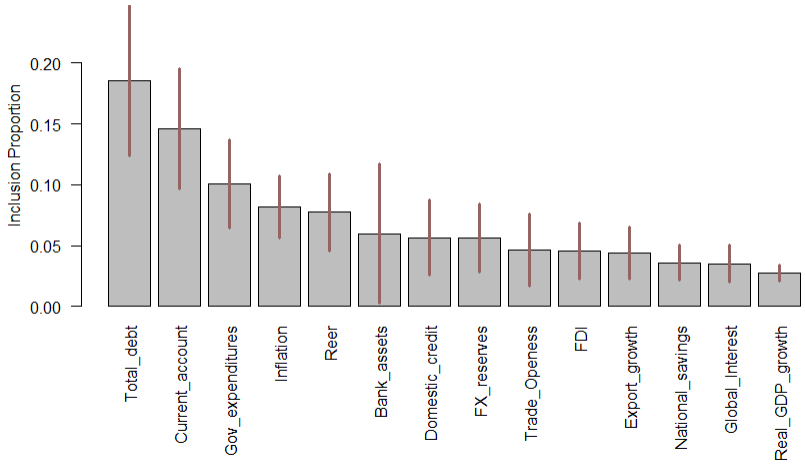

Figure 93: BART variable importance for the Western European mean by region data set.

From Figure 93 it can be concluded that when considering mean imputation for the Western European data set, the top five variables in terms of inclusion proportion are total debt, current account, government expenditure, inflation, and REER.

C.3 MICE BART Models

C.3.1 Global Data

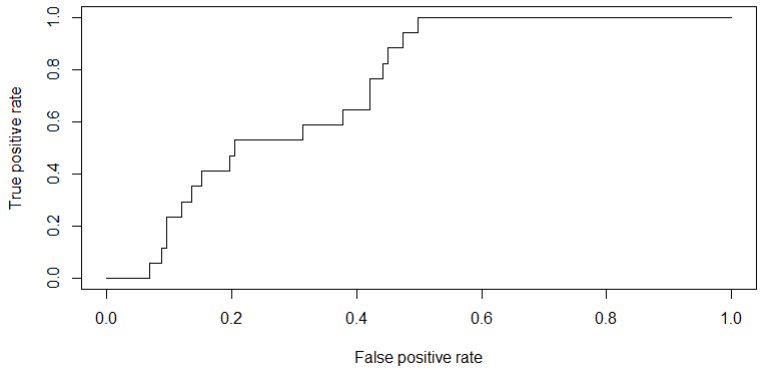

Figure 94: ROC curve applied to the MICE global data.

From the observation of Figure 94 and calculation in Table 103, it can be concluded that the classifier built using the MICE global data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

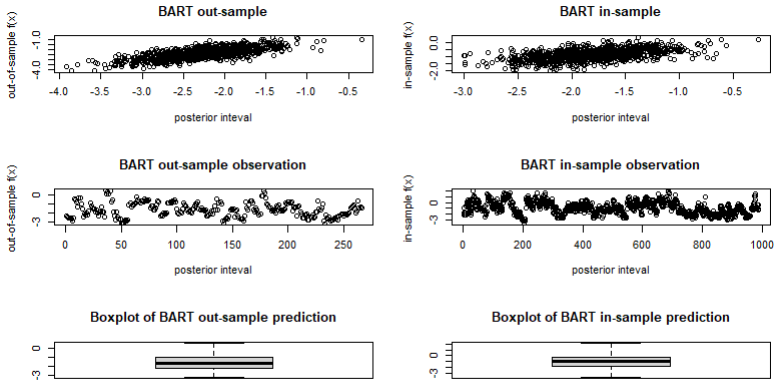

Figure 95: Plot and box plot of the model output results for training and testing MICE global data.

From Figure 95, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults. The in and out of sample plots show that the model generated distributions are mostly non-default times since the observations are below the 0.5 benchmark. The box plots for both in and out of sample observations also show that the majority of outputs are non-default states of sovereigns.

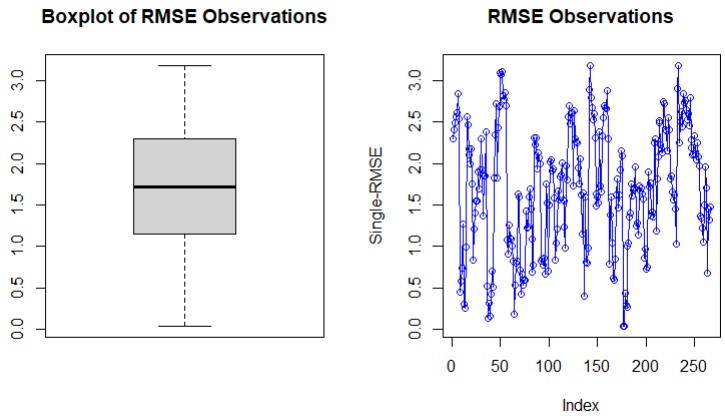

Figure 96: RMSE box plot and line plot for MICE global data.

The following can be deduced from Figure 96: The majority of RMSE observations are found between the range of 1 and 2.5. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default.

Table 103: BART model summary of the global MICE data set.

RMSE	1.8593
RRMSE	0.0701
AUC	0.7295
Out of sample correctness	0.9323
In sample correctness	0.8147
RMSE(0.5,0.75)	(1.7225, 2.2970)
Sensitivity	0.9762
Specificity	0.4
Precision	0.9427
Recall	0.9762
PR AUC	0.5445
Brier Score	0.0676

From the observations in Table 103, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 93.23% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 72.95%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 97.62% at predicting non-default times and having a specificity score of 40%, which shows that it did a poor job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 94.28% of non-default states.

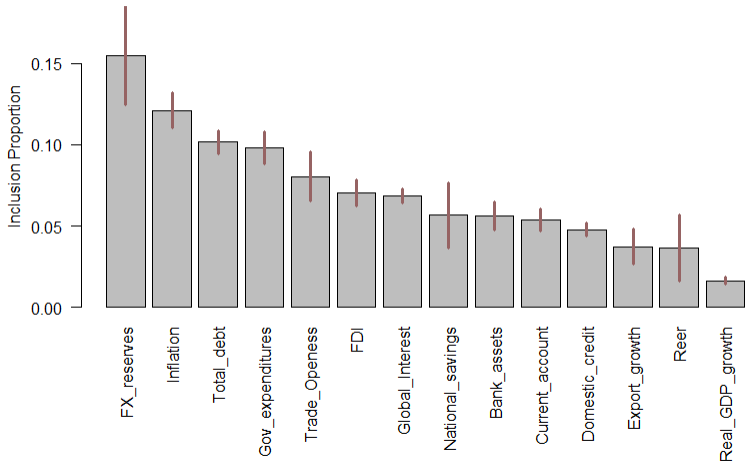

Figure 97: BART variable importance for the global MICE data set.

From Figure 97 it can be concluded that when considering MICE for the global data set, the top five variables in terms of inclusion proportion are FX reserves, inflation, total debt, government expenditure, and trade openess.

C.3.2 South and East Asia

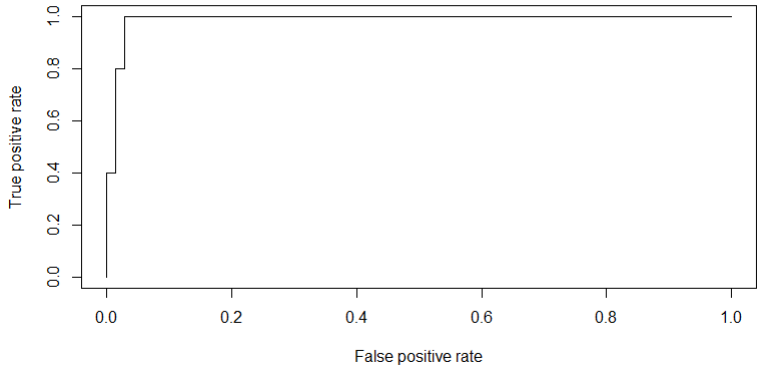

Figure 98: ROC curve applied to the MICE South and East Asia data.

From the observation of Figure 98 and calculation in Table 104, it can be concluded that the classifier built using the MICE South and East Asia data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

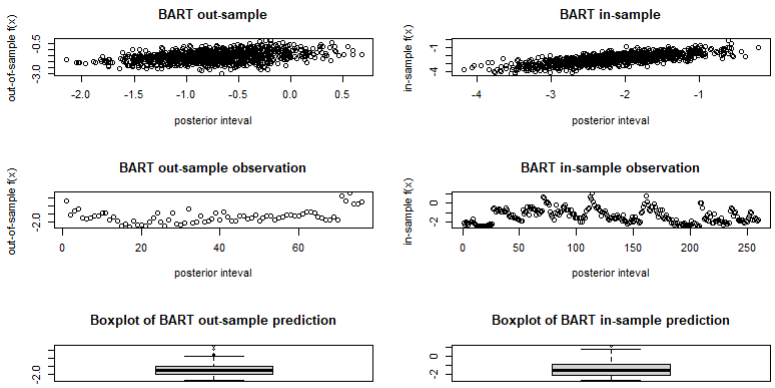

Figure 99: Plot and box plot of the model output results for training and testing MICE South and East Asia data.

From Figure 99, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults, though unlike the other data sets, this one has quite a few observations on the positive side. The in and out of sample plots show that the model generated distributions are mostly non-default times since the observations are below the 0.5 benchmark. The box plots for both in and out of sample observations also show that the majority of outputs are non-default states of sovereigns.

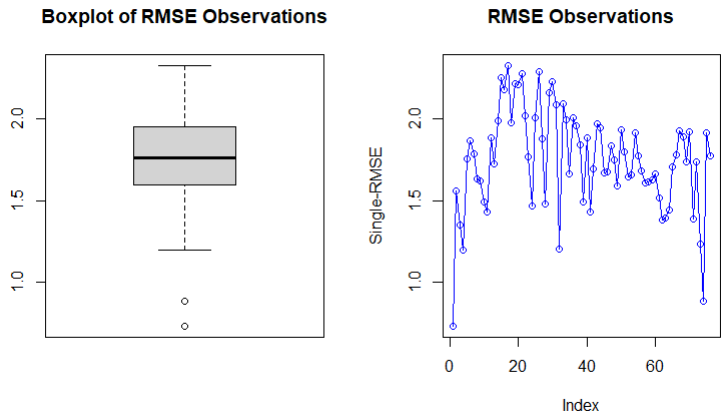

Figure 100: RMSE box plot and line plot for MICE South and East Asia data.

The following can be deduced from Figure 100: The majority of RMSE observations are found between the range of 1.5 and 2. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default. As such, values close to zero (0) are seen as outliers.

Table 104: BART model summary of the South and East Asia MICE data set.

RMSE	1.7787
RRMSE	0.1210
AUC	0.9887
Out of sample correctness	0.9342
In sample correctness	0.8692
RMSE(0.5,0.75)	(1.7624, 1.9499)
Sensitivity	0.9589
Specificity	0.6
Precision	0.9210
Recall	0.9589
PR AUC	0.5196
Brier Score	0.1022

From the observations in Table 104, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 93.42% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 98.87%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 95.89% at predicting non-default times and having a specificity score of 60%, which shows that it did a descent job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 92.10% of non-default states.

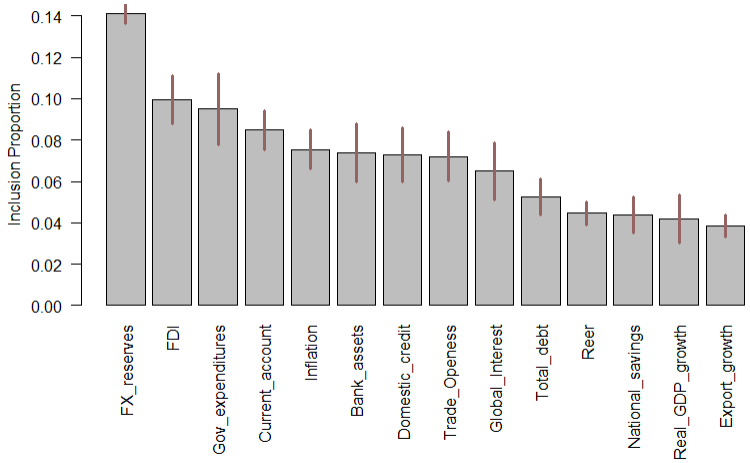

Figure 101: BART variable importance for the South and East Asia MICE data set.

From Figure 101 it can be concluded that when considering MICE for the South and East Asia data set, the top five variables in terms of inclusion proportion are FX reserves, FDI, government expenditure, current account, and inflation.

C.3.3 Latin America

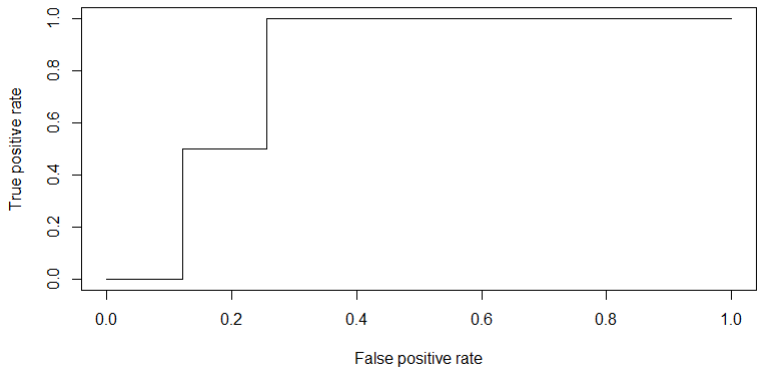

Figure 102: ROC curve applied to the MICE Latin America data.

From the observation of Figure 102 and calculation in Table 105, it can be concluded that the classifier built using the MICE Latin America data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

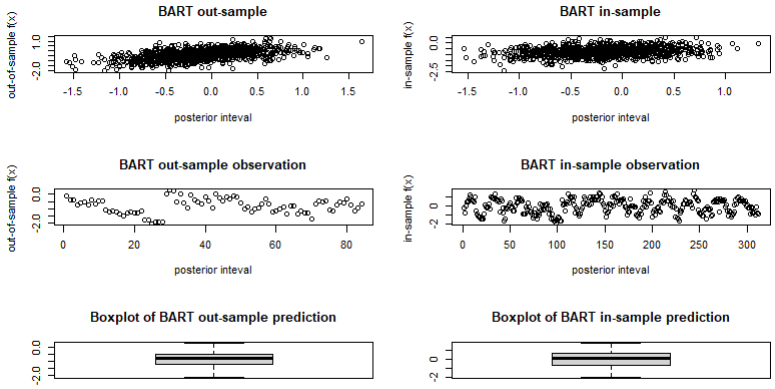

Figure 103: Plot and box plot of the model output results for training and testing MICE Latin America data.

From Figure 103, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults, though unlike the other data sets, this one has quite a few observations on the positive side. The in and out of sample plots show that the model generated distributions are mostly non-default times since the observations are below the 0.5 benchmark. The box plots for both in and out of sample observations also show that the majority of outputs are non-default states of sovereigns.

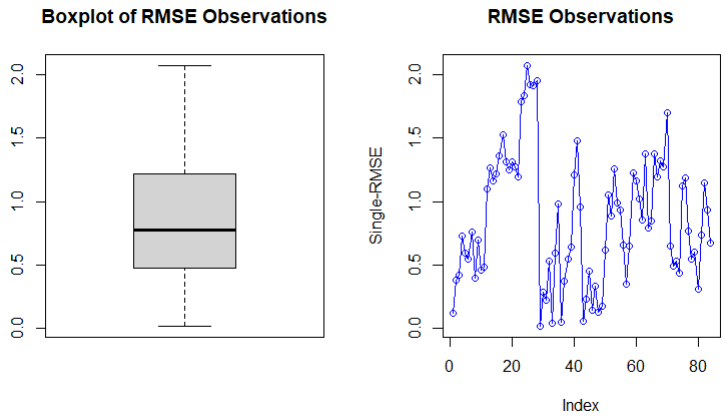

Figure 104: RMSE box plot and line plot for MICE Latin America data.

The following can be deduced from Figure 104: The majority of RMSE observations are found between the range of 1 and 2.5. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default.

Table 105: BART model summary of the Latin America MICE data set.

RMSE	0.9956
RRMSE	0.1322
AUC	0.81097
Out of sample correctness	0.9761
In sample correctness	0.7980
RMSE(0.5,0.75)	(0.77761.2196)
Sensitivity	0.9529
Specificity	0.8182
Precision	0.9759
Recall	0.9529
PR AUC	0.4879
Brier Score	0.0625

From the observations in Table 105, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 97.61% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 81.09%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 95.29% at predicting non-default times and having a specificity score of 81.82%, which shows that it did a descent job of predicting defaults. The precision score also shows that the model performed well since it correctly identified 97.59% of non-default states.

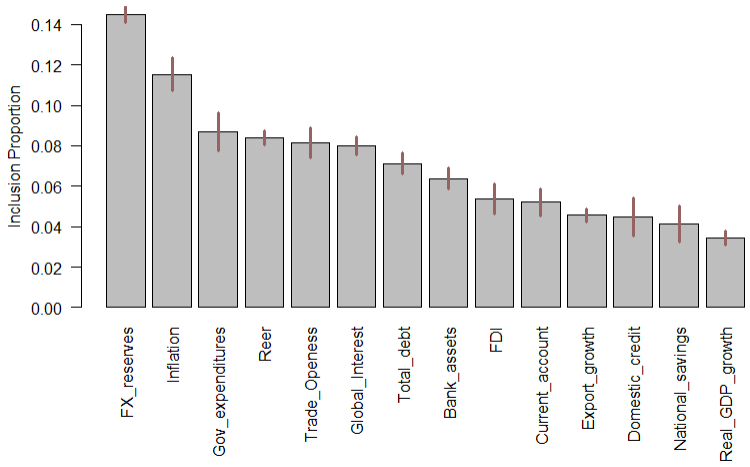

Figure 105: BART variable importance for the Latin America MICE data set.

From Figure 105 it can be concluded that when considering MICE for the Latin America data set, the top five variables in terms of inclusion proportion are FX inflation, government expenditure, REER, and trade openess.

C.3.4 The Middle East and Africa

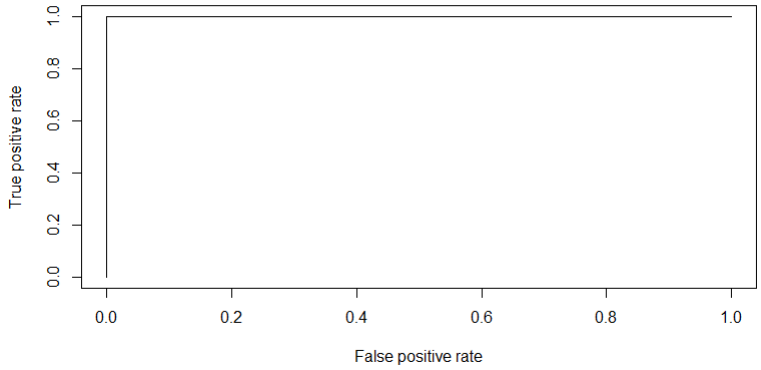

Figure 106: ROC curve applied to the MICE the Middle East and Africa data.

From the observation of Figure 106 and calculation in Table 106, it can be concluded that the classifier built using the MICE Middle East and Africa data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

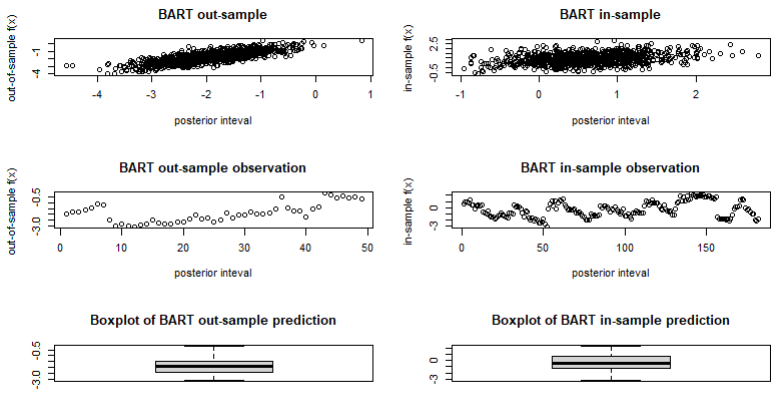

Figure 107: Plot and box plot of the model output results for training and testing MICE The Middle East and Africa data.

From Figure 103, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults, though unlike the other data sets, this one has quite a few observations on the positive side. The in and out of sample plots show that the model generated distributions are mostly non-default times since the observations are below the 0.5 benchmark. The box plots for both in and out of sample observations also show that the majority of outputs are non-default states of sovereigns.

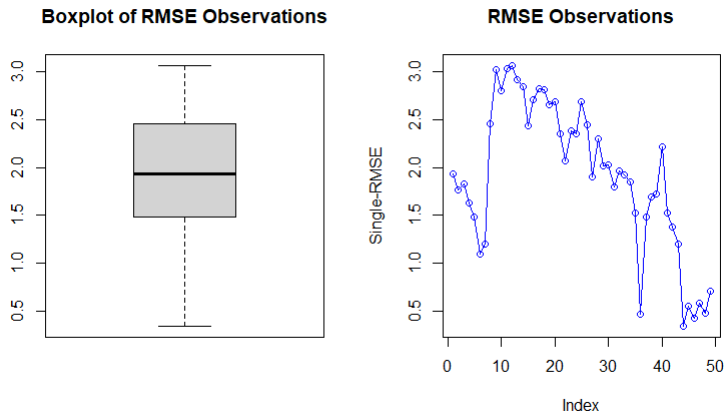

Figure 108: RMSE box plot and line plot for MICE the Middle East and Africa data.

The following can be deduced from Figure 108: The majority of RMSE observations are found between the range of 1 and 2.5. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default.

Table 106: BART model summary of the Middle East and Africa MICE data set.

RMSE	2.0580
RRMSE	0.1556
AUC	1
Out of sample correctness	0.9795
In sample correctness	0.8461
RMSE(0.5,0.75)	1.9307, 2.4549
Sensitivity	0.9411
Specificity	0.9
Precision	0.9795
Recall	0.9411
PR AUC	0.4834
Brier Score	0.0655

From the observations in Table 106, the following can be concluded: The out-of-sample-correctness values show that the classifier managed to predict about 97.95% of the observations. The ROC AUC value shows that the classifier is good as an EWS since the value is above 50%, being 100%. The sensitivity and recall scores illustrate that the classifier did a good job by scoring 94.11% at predicting non-default times and having a specificity score of 90%, which shows that it did a descent job of predicting defaults. The precision score also shows that the model performed great, since it correctly identified 97.94% of non-default states.

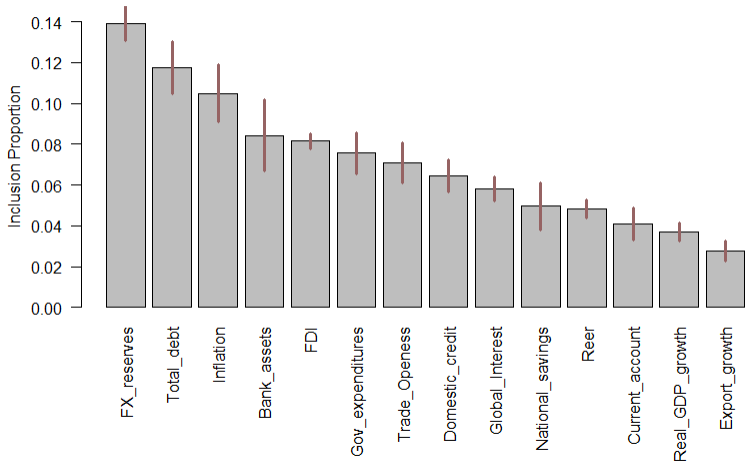

Figure 109: BART variable importance for the Middle East and Africa MICE data set.

From Figure 109, it can be concluded that when considering MICE for the Middle East and Africa data set, the top five variables in terms of inclusion proportion are FX inflation, total debt, inflation, bank assets, and FDI.

C.3.5 Western Europe

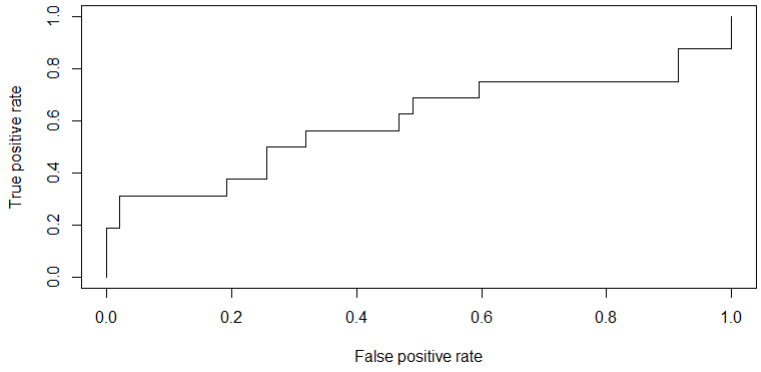

Figure 110: ROC curve the applied to MICE Western Europe data.

From the observation of Figure 110 and calculation in Table 107, it can be concluded that the classifier built using the MICE Wesern European data set produced a skilled classifier since the since ROC AUC value is above 50% and the curve arches above the diagonal line centre.

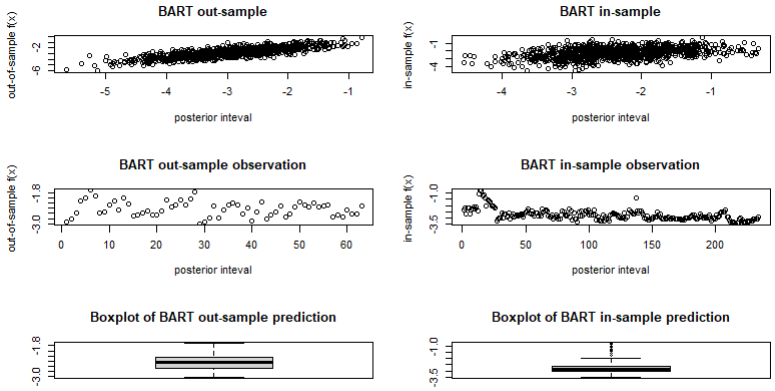

Figure 111: Plot and box plot of the model output results for training and testing MICE Western Europe data.

From Figure 111, the following is shown: the in and out of sample plots suggest that the classifier outputs are normally distributed along the negative axis, which suggests that their outputs are mostly non-defaults. The in and out of sample plots show that the model generated distributions are mostly non-default times since the observations are below the 0.5 benchmark. The box plots for both in and out of sample observations also show that the majority of outputs are non-default states of sovereigns.

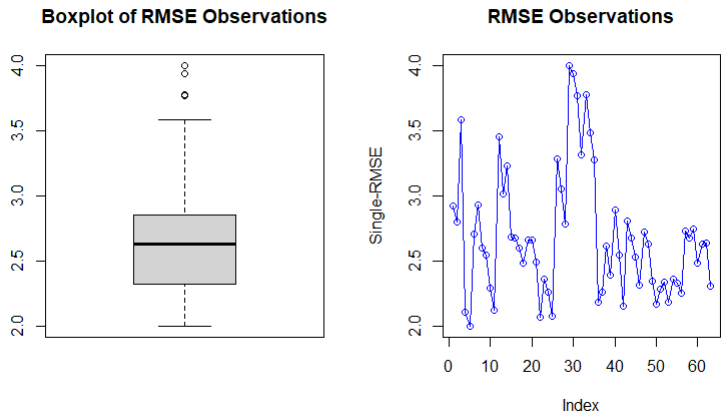

Figure 112: RMSE box plot and line plot for MICE Western Europe data.

The following can be deduced from Figure 112: The majority of RMSE observations are found between the range of 2 and 3. This is a result of the model either being aggressive, meaning the outputs are being exegerated (i.e. a 0 classification can be -2 or a 1 classification can be 3) or the classifier being conservative and failing to differentiate between default and non-default. In this case the classifier was failing to classify the different states of sovereigns, since not enough information was available to develop an appropriate EWS.

Table 107: BART model summary of the Western Europe MICE data set.

RMSE	2.7303
RRMSE	0.1384
AUC	0.5970
Out of sample correctness	0.7460
In sample correctness	0.9829
RMSE(0.5,0.75)	$(2.6330, 2.8524)$
Sensitivity	0
Specificity	1
Precision	*Na*
Recall	*Na*
PR AUC	*Na*
Brier Score	0.2933

From the observations in Table 107, the following can be concluded: The classifier outputs can not be summarised since they did not have enough defaults to conclude adequate statistics.

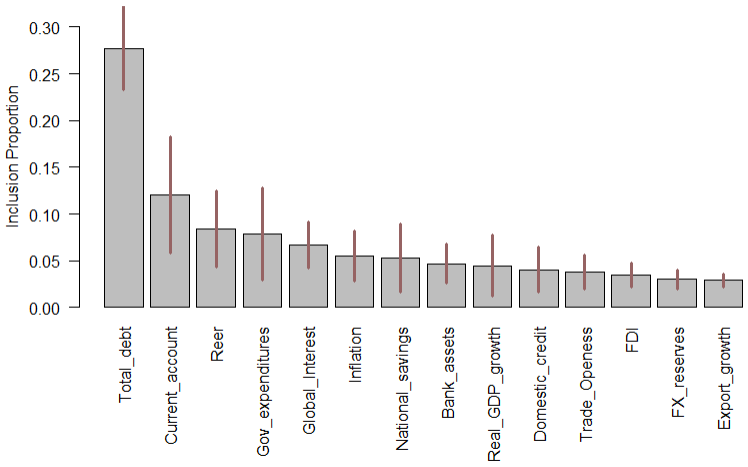

Figure 113: BART variable importance for the Western European MICE data set.

From Figure 113, it can be concluded that when considering MICE for the Western European data set, the top five variables in terms of inclusion proportion are total debt, current account, REER, government expenditure, and global interest.

Appendix D: Neural Networks Analysis

This part of the appendix is a detailed illustration of the analysis done using the ANN classifier on all data sets. The summary and comparison of the outputs can be found in chapter 5.

D.1 Case-wise Deletion ANN Models

Table 108: ANN model summary applied to the global case-wise deletion data set.

Number of nodes	6	7	8	9	10
In sample	0.2636	0.6994	0.6995	0.5887	0.1810
Out-of-sample	0.4462	0.7024	0.7024	0.5372	0.1404
Sensitivity	0.4017	0.75	0.75	0.5714	0.0714
Specificity	1	0.1111	0.111	0.1111	1
Precision	1	0.9130	0.9130	0.8889	1
Recall	0.4017	0.75	0.75	0.5714	0.0714
Brier Score	0.5455	0.2780	0.2781	0.4626	0.8595
PR AUC	0.3496	0.4262	0.4261	0.3859	0.3153
ROC AUC	0.5347	0.6447	0.4739	0.5691	0.5347

From the analysis shown in Table 108, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is six (6).

Table 109: ANN model summary applied to the South and East Asia case-wise deletion data set.

Number of nodes	6	7	8	9	10
In sample	0.7647	0.8235	0.7941	0.6764	0.1764
Out-of-sample	0.8529	0.8235	0.5882	0.8529	0.1764
Sensitivity	0.8571	1	0.5357	1	0.8437
Specificity	0.8333	0.8125	0.8333	0.1667	0.5714
Precision	0.96	0.8125	0.9375	0.8484	0.8181
Recall	0.8571	1	0.5357	1	0.8437
Brier Score	0.1400	0.1716	0.2097	0.1470	0.2609
PR AUC	0.4604	0.4523	0.3874	0.6148	0.5915
ROC AUC	0.7232	0.2767	0.7813	0.7075	0.7075

From the analysis shown in Table 109, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is eight (8).

Table 110: ANN model summary applied to the Latin America case-wise deletion data set.

Number of nodes	6	7	8	9	10
In sample	0.0566	0.3773	0.1132	0.2452	0.03773
Out-of-sample	0.4150	0.5660	0.2264	0.5094	0.0377
Sensitivity	0.5740	0.5490	0.1961	0.4902	0.1851
Specificity	0.1818	1	1	0.1818	1
Precision	0.775	1	1	0.7567	1
Recall	0.5740	0.5490	0.1961	0.4902	0.1851
Brier Score	0.4863	0.4265	0.7736	0.4909	0.9622
PR AUC	0.4191	0.3669	0.3247	0.3584	0.3359
ROC AUC	0.3779	0.3779	0.5925	0.3501	0.5925

From the analysis shown in Table 110, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is eight (8).

Table 111: ANN model summary applied to the Middle East and Africa case-wise deletion data set.

Number of nodes	6	7	8	9	10
In sample	0.3939	0.6222	0.3333	0.5152	0.3636
Out-of-sample	0.9091	0.8	0.4242	0.7879	0.3636
Sensitivity	0.8571	1	0.0952	0.9523	0.7916
Specificity	0.5714	0.5714	1	0.2857	0.4761
Precision	0.7	0.7273	1	0.6052	0.6333
Recall	0.8571	1	0.0952	0.9523	0.7916
Brier Score	0.0916	0.2052	0.5758	0.2056	0.3500
PR AUC	0.4672	0.5882	0.3593	0.6585	0.5545
ROC AUC	0.7232	0.7857	0.5416	0.6220	0.3174

From the analysis shown in Table 111, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is seven (7).

D.2 Mean Imputation ANN Models

Table 112: ANN model summary applied to the global mean imputation data set.

Number of nodes	6	7	8	9	10
In sample	0.1127	0.1278	0.1654	0.1954	0.1316
Out-of-sample	0.3872	0.5075	0.3797	0.5225	0.2293
Sensitivity	0.3840	0.512	0.34	0.54	0.188
Specificity	0.4375	0.4375	1	0.25	0.875
Precision	0.9142	0.9343	1	0.9183	0.9592
Recall	0.3840	0.512	0.34	0.54	0.188
Brier Score	0.5644	0.2494	0.5908	0.2928	0.7244
PR AUC	0.3507	0.3685	0.3411	0.3749	0.3259
ROC AUC	0.5902	0.5902	0.5902	0.5267	0.5528

From the analysis shown in Table 112, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is eight (8).

Table 113: ANN model summary applied to the South and East Asia mean imputation data set.

Number of nodes	6	7	8	9	10
In sample	0.2208	0.5325	0.3636	0.1429	0.9480
Out-of-sample	0.6104	0.8571	0.8442	0.4026	0.8701
Sensitivity	0.57714	0.8429	0.8286	0.3429	0.9286
Specificity	1	1	1	1	0.2857
Precision	1	1	1	1	0.9286
Recall	0.5714	0.8429	0.8286	0.3429	0.9286
Brier Score	0.3854	0.1429	0.1553	0.5974	0.1323
PR AUC	0.3755	0.4346	0.4304	0.3442	0.5
ROC AUC	0.7739	0.6434	0.8972	0.66438	0.7889

From the analysis shown in Table 113, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is eight (8).

Table 114: ANN model summary applied to the Latin America mean imputation data set.

Number of nodes	6	7	8	9	10
In sample	0.0595	0.0833	0.059	0.036	0.0714
Out-of-sample	0.1667	0.3333	0.071	0.167	0.1788
Sensitivity	0.1463	0.3171	0.0487	0.1463	0.1585
Specificity	1	1	1	1	1
Precision	1	1	1	1	1
Recall	0.1463	0.3171	0.0487	0.1463	0.1585
Brier Score	0.5006	0.6726	0.9243	0.8259	0.6707
PR AUC	0.3196	0.3366	0.3111	0.3196	0.3207
ROC AUC	0.5705	0.6529	0.5235	0.6823	0.6412

From the analysis shown in Table 114, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is eight (8).

Table 115: ANN model summary applied to the Middle East and Africa mean imputation data set.

Number of nodes	6	7	8	9	10
In sample	0.9796	0.2449	0.1837	0.4286	0.1632
Out-of-sample	0.9796	0.8367	0.7143	0.9796	0.7755
Sensitivity	0.9411	0.8431	0.6667	1	0.7708
Specificity	0.9	0.1	1	0	0.1
Precision	0.9795	0.8269	1	*Na*	0.8163
Recall	0.9411	0.8431	0.6667	*Na*	0.7708
Brier Score	0.0650	0.1334	0.1453	0.1618	0.2032
PR AUC	0.4834	0.5091	0.4027	*Na*	0.4070
ROC AUC	0.9205	0.4715	0.8333	0.5	0.4421

From the analysis shown in Table 115, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is eight (8).

Table 116: ANN model summary applied to the Western Europe mean imputation data set.

Number of nodes	6	7	8	9	10
In sample	0.7936	0.7936	0.7936	0.7936	0.7936
Out-of-sample	0.7936	0.7936	0.7936	0.7936	0.7778
Sensitivity	1	1	0.9433	1	0.98
Specificity	0	0	0.4091	0	0
Precision	*Na*	*Na*	0.7976	*Na*	0.7903
Recall	*Na*	*Na*	0.9433	*Na*	0.98
Brier Score	0.1914	0.1976	0.2047	0.28097	0.2056
PR AUC	*Na*	*Na*	0.5711	*Na*	0.6510
ROC AUC	0.5	0.5	0.6762	0.5	0.4905

From the analysis shown in Table 116, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is eight (8).

D.3 MICE ANN Models

Table 117: ANN model summary applied to the global MICE data set.

Number of nodes	6	7	8	9	10
In sample	0.4097	0.3834	0.6691	0.4060	0.5714
Out-of-sample	0.7857	0.8571	0.9323	0.8646	0.8721
Sensitivity	0.8353	0.8915	0.9959	0.9156	0.7707
Specificity	0.0588	0.3529	0	0.1176	0.56
Precision	0.9285	0.9527	0.9358	0.9382	0.9466
Recall	0.8353	0.8915	0.9959	0.9156	0.7707
Brier Score	0.1849	0.0893	0.0618	0.1551	0.2445
PR AUC	0.4500	0.4647	0.5648	0.4849	0.4262
ROC AUC	0.6486	0.6352	0.7110	0.6577	0.6653

From the analysis shown in Table 117, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is eight (8).

Table 118: ANN model summary applied to the South and East Asia MICE data set.

Number of nodes	6	7	8	9	10
In sample	0.4736	0.9078	0.3815	0.1711	0.4342
Out-of-sample	0.8552	0.9868	0.8421	0.5921	0.6315
Sensitivity	0.9154	1	0.8309	0.5633	0.6056
Specificity	0	0.8	1	1	1
Precision	0.9285	0.9861	1	1	1
Recall	0.9154	1	0.8309	0.5633	0.6056
Brier Score	0.1611	0.0115	0.1562	0.3415	0.6458
PR AUC	0.4909	0.5102	0.4277	0.3715	0.3784
ROC AUC	0.4671	0.8589	0.9315	0.3917	0.3506

From the analysis shown in Table 118, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is eight (8).

Table 119: ANN model summary applied to the Latin America MICE data set.

Number of nodes	6	7	8	9	10
In sample	0.1309	0.2619	0.3214	0.1667	0.0714
Out-of-sample	0.1309	0.1785	0.4285	0.1904	0.3214
Sensitivity	0.1585	0.5731	0.4146	0.1707	0.3048
Specificity	1	1	1	1	1
Precision	1	1	1	1	1
Recall	0.1585	0.5731	0.4146	0.1707	0.3048
Brier Score	0.7465	0.4407	0.4187	0.6961	0.5446
PR AUC	0.3207	0.4197	0.3479	0.3218	0.3353
ROC AUC	0.2909	0.8882	0.5470	0.6058	0.5235

From the analysis shown in Table 119, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is seven (7).

Table 120: ANN model summary applied to the Middle East and Africa MICE data set.

Number of nodes	6	7	8	9	10
In sample	0.8571	0.0408	0.3877	0.6122	0.1428
Out-of-sample	0.8775	0.5102	0.8979	0.9183	0.5714
Sensitivity	0.875	0.5	0.8958	0.9167	0.5625
Specificity	1	1	1	1	1
Precision	1	1	1	1	1
Recall	0.875	0.5	0.8958	0.9167	0.5625
Brier Score	0.1221	0.3589	0.0939	0.0616	0.4351
PR AUC	0.4359	0.4897	0.4431	0.4511	0.3681
ROC AUC	0.4029	0.4813	0.7058	0.7450	0.9607

From the analysis shown in Table 120, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is ten (10).

Table 121: ANN model summary applied to the Western Europe MICE data set.

Number of nodes	6	7	8	9	10
In sample	0.7936	0.7936	0.7936	0.7936	0.7936
Out-of-sample	0.7936	0.7936	0.7936	0.7936	0.7936
Sensitivity	1	1	1	1	0.9433
Specificity	0	0	0	0	0.4091
Precision	*Na*	*Na*	*Na*	*Na*	0.7936
Recall	*Na*	*Na*	*Na*	*Na*	0.9433
Brier Score	0.1864	0.1996	0.1915	0.2042	0.2076
PR AUC	*Na*	*Na*	*Na*	*Na*	0.5711
ROC AUC	0.5	0.5	0.5	0.5	0.6762

From the analysis shown in Table 121, the following is concluded: The number of nodes required to produce an adequate classifier when looking at ROC AUC is ten (10).

Appendix E: R-Code.

E.1 Summary Statistics

```
#Loading required packages
library(missForest)
library(mice)
library(VIM)

#Appointing new name for the data set
Data_set.mis <- Data_set

#Obtaining summary of the variables
summary(Data_set.mis)

#Generating subset of variable in order to view important variables
   only
Data_set.mis <- subset(Data_set.mis,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Generting missing data plots
md.pattern(Data_set.mis)
mice_plot <- aggr(Data_set.mis,
                  col=c('lightblue','red')
                  ,numbers=TRUE
                  ,sortVars=TRUE
                  ,labels=names(Data_set.mis)
                  , cex.axis=.7
                  ,gap=3
                  , ylab=c("Missing data","Pattern"))



#Generating subsets of he data based on default and non-default
   times
crises<- filter(Data_set, Data_set$Dedt_Crises==1)
```

```
non_crises <- filter(Data_set, Data_set$Dedt_Crises==0)

#Generating Welsch t-test values
t.test(crises$Total_debt,non_crises$Total_debt)
t.test(crises$Global_Interest,non_crises$Global_Interest)
t.test(crises$FX_reserves,non_crises$FX_reserves)
t.test(crises$Trade_Openess,non_crises$Trade_Openess)
t.test(crises$Export_growth,non_crises$Export_growth)
t.test(crises$Current_account,non_crises$Current_account)
t.test(crises$FDI,non_crises$FDI)
t.test(crises$Real_GDP_growth,non_crises$Real_GDP_growth)
t.test(crises$Inflation,non_crises$Inflation)
t.test(crises$Reer,non_crises$Reer)
t.test(crises$Gov_expenditures,non_crises$Gov_expenditures)
t.test(crises$National_savings,non_crises$National_savings)
t.test(crises$Domestic_credit,non_crises$Domestic_credit)
t.test(crises$Bank_assets,non_crises$Bank_assets)

#Generating variance hypothesis testing scores
var.test(crises$Total_debt,non_crises$Total_debt)
var.test(crises$Global_Interest,non_crises$Global_Interest)
var.test(crises$FX_reserves,non_crises$FX_reserves)
var.test(crises$Trade_Openess,non_crises$Trade_Openess)
var.test(crises$Export_growth,non_crises$Export_growth)
var.test(crises$Current_account,non_crises$Current_account)
var.test(crises$FDI,non_crises$FDI)
var.test(crises$Real_GDP_growth,non_crises$Real_GDP_growth)
var.test(crises$Inflation,non_crises$Inflation)
var.test(crises$Reer,non_crises$Reer)
var.test(crises$Gov_expenditures,non_crises$Gov_expenditures)
var.test(crises$National_savings,non_crises$National_savings)
var.test(crises$Domestic_credit,non_crises$Domestic_credit)
var.test(crises$Bank_assets,non_crises$Bank_assets)

#Generating variances
var(crises$Dedt_Crises)
var(non_crises$Dedt_Crises)

var(crises$Total_debt, na.rm=TRUE)
var(non_crises$Total_debt, na.rm=TRUE)

var(crises$Global_Interest, na.rm=TRUE)
var(non_crises$Global_Interest, na.rm=TRUE)

var(crises$FX_reserves, na.rm=TRUE)
```

```
var(non_crises$FX_reserves, na.rm=TRUE)

var(crises$Trade_Openess, na.rm=TRUE)
var(non_crises$Trade_Openess, na.rm=TRUE)

var(crises$Export_growth, na.rm=TRUE)
var(non_crises$Export_growth, na.rm=TRUE)

var(crises$Current_account, na.rm=TRUE)
var(non_crises$Current_account, na.rm=TRUE)

var(crises$FDI, na.rm=TRUE)
var(non_crises$FDI, na.rm=TRUE)

var(crises$Real_GDP_growth, na.rm=TRUE)
var(non_crises$Real_GDP_growth, na.rm=TRUE)

var(crises$Inflation, na.rm=TRUE)
var(non_crises$Inflation, na.rm=TRUE)

var(crises$Reer, na.rm=TRUE)
var(non_crises$Reer, na.rm=TRUE)

var(crises$Gov_expenditures, na.rm=TRUE)
var(non_crises$Gov_expenditures, na.rm=TRUE)

var(crises$National_savings, na.rm=TRUE)
var(non_crises$National_savings, na.rm=TRUE)

var(crises$Domestic_credit, na.rm=TRUE)
var(non_crises$Domestic_credit, na.rm=TRUE)

var(crises$Bank_assets, na.rm=TRUE)
var(non_crises$Bank_assets, na.rm=TRUE)

#Generating boxplots
boxplot(Data_set$Total_debt, na.rm=TRUE)
boxplot(Data_set$Global_Interest, na.rm=TRUE)
boxplot(Data_set$FX_reserves, na.rm=TRUE)
boxplot(Data_set$Trade_Openess, na.rm=TRUE)
boxplot(Data_set$Export_growth, na.rm=TRUE)
boxplot(Data_set$Current_account, na.rm=TRUE)
boxplot(Data_set$FDI, na.rm=TRUE)
boxplot(Data_set$Real_GDP_growth, na.rm=TRUE)
boxplot(Data_set$Inflation, na.rm=TRUE)
```

```
boxplot(Data_set$Reer, na.rm=TRUE)
boxplot(Data_set$Gov_expenditures, na.rm=TRUE)
boxplot(Data_set$National_savings, na.rm=TRUE)
boxplot(Data_set$Domestic_credit, na.rm=TRUE)
boxplot(Data_set$Bank_assets, na.rm=TRUE)
```

151C__Users_Thendo_Desktop_Thesis_Paper_RCode_Summary_statistics.R

E.2 MICE

```
#Load required packages
library(missForest)
library(mice)
library(VIM)

##World Data

#Rename the data set
Data_test.mis <-Test_set_mean
Data_train.mis <- Data_set_mean

#Get a summary of the data sets
summary(Data_test.mis)
summary(Data_train.mis)

#Subset data set by only keeping important variables
Data_train.mis <- subset(Data_train.mis,select=c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_test.mis <- subset(Data_test.mis,select=c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Observe missing data pattern
md.pattern(Data_train.mis)
md.pattern(Data_test.mis)

#Generate layout of missing values
mice_plot.asia <- aggr(Data_train.mis,
                       col=c('lightblue','red')
                       ,numbers=TRUE
                       ,sortVars=TRUE
                       ,labels=names(Data_train.mis)
                       , cex.axis=.7
                       ,gap=3
```

```
                        , ylab=c("Missing data","Pattern"))
mice_plot.asia <- aggr(Data_test.mis,
                        col=c('lightblue','red')
                        ,numbers=TRUE
                        ,sortVars=TRUE
                        ,labels=names(Data_test.mis)
                        , cex.axis=.7
                        ,gap=3
                        , ylab=c("Missing data","Pattern"))

#Use MICE to generate missing values
Data_train.mis <- mice(Data_train.mis, m=5, maxit = 50, method = '
    pmm', seed = 500)
Data_test.mis <- mice(Data_test.mis, m=5, maxit = 50, method = 'pmm'
    , seed = 500)

#Select data set needed for the analysis
completeTest <- complete(Data_train.mis,1)
completeTest <- complete(Data_test.mis,1)

##South and East Asia Data

#Rename the data set
Data_test.mis <-Asia_test
Data_train.mis <- Asia_train

#Get a summary of the data sets
summary(Data_test.mis)
summary(Data_train.mis)

#Subset data set by only keeping important variables
Data_train.mis <- subset(Data_train.mis,select=c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_test.mis <- subset(Data_test.mis,select=c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Observe missing data pattern
md.pattern(Data_train.mis)
md.pattern(Data_test.mis)

#Generate layout of missing values
mice_plot.asia <- aggr(Data_train.mis,
                        col=c('lightblue','red')
                        ,numbers=TRUE
                        ,sortVars=TRUE
```

```
                        ,labels=names(Data_train.mis)
                        , cex.axis=.7
                        ,gap=3
                        , ylab=c("Missing data","Pattern"))
mice_plot.asia <- aggr(Data_test.mis,
                        col=c('lightblue','red')
                        ,numbers=TRUE
                        ,sortVars=TRUE
                        ,labels=names(Data_test.mis)
                        , cex.axis=.7
                        ,gap=3
                        , ylab=c("Missing data","Pattern"))

#Use MICE to generate missing values
Data_train.mis <- mice(Data_train.mis, m=5, maxit = 50, method = '
    pmm', seed = 500)
Data_test.mis <- mice(Data_test.mis, m=5, maxit = 50, method = 'pmm'
    , seed = 500)

#Select data set needed for the analysis
completeTest <- complete(Data_train.mis,1)
completeTest <- complete(Data_test.mis,1)

##Latin America Data

#Rename the data set
Data_test.mis <-Latin_test
Data_train.mis <- Latin_train

#Get a summary of the data sets
summary(Data_test.mis)
summary(Data_train.mis)

#Subset data set by only keeping important variables
Data_train.mis <- subset(Data_train.mis,select=c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_test.mis <- subset(Data_test.mis,select=c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Observe missing data pattern
md.pattern(Data_train.mis)
md.pattern(Data_test.mis)

#Generate layout of missing values
mice_plot.asia <- aggr(Data_train.mis,
```

```
                        col=c('lightblue','red')
                        ,numbers=TRUE
                        ,sortVars=TRUE
                        ,labels=names(Data_train.mis)
                        , cex.axis=.7
                        ,gap=3
                        , ylab=c("Missing data","Pattern"))
mice_plot.asia <- aggr(Data_test.mis,
                        col=c('lightblue','red')
                        ,numbers=TRUE
                        ,sortVars=TRUE
                        ,labels=names(Data_test.mis)
                        , cex.axis=.7
                        ,gap=3
                        , ylab=c("Missing data","Pattern"))

#Use MICE to generate missing values
Data_train.mis <- mice(Data_train.mis, m=5, maxit = 50, method = '
    pmm', seed = 500)
Data_test.mis <- mice(Data_test.mis, m=5, maxit = 50, method = 'pmm'
    , seed = 500)

#Select data set needed for the analysis
completeTest <- complete(Data_train.mis,1)
completeTest <- complete(Data_test.mis,1)

##The Middle East and Africa Data

#Rename the data set
Data_test.mis <-Africa _test
Data_train.mis <- Africa_train

#Get a summary of the data sets
summary(Data_test.mis)
summary(Data_train.mis)

#Subset data set by only keeping important variables
Data_train.mis <- subset(Data_train.mis,select=c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_test.mis <- subset(Data_test.mis,select=c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Observe missing data pattern
md.pattern(Data_train.mis)
md.pattern(Data_test.mis)
```

```
#Generate layout of missing values
mice_plot.asia <- aggr(Data_train.mis,
                       col=c('lightblue','red')
                       ,numbers=TRUE
                       ,sortVars=TRUE
                       ,labels=names(Data_train.mis)
                       , cex.axis=.7
                       ,gap=3
                       , ylab=c("Missing data","Pattern"))
mice_plot.asia <- aggr(Data_test.mis,
                       col=c('lightblue','red')
                       ,numbers=TRUE
                       ,sortVars=TRUE
                       ,labels=names(Data_test.mis)
                       , cex.axis=.7
                       ,gap=3
                       , ylab=c("Missing data","Pattern"))

#Use MICE to generate missing values
Data_train.mis <- mice(Data_train.mis, m=5, maxit = 50, method = '
    pmm', seed = 500)
Data_test.mis <- mice(Data_test.mis, m=5, maxit = 50, method = 'pmm'
    , seed = 500)

#Select data set needed for the analysis
completeTest <- complete(Data_train.mis,1)
completeTest <- complete(Data_test.mis,1)


##Western Europe Data

#Rename the data set
Data_test.mis <-Europe_test
Data_train.mis <- Europe_train

#Get a summary of the data sets
summary(Data_test.mis)
summary(Data_train.mis)

#Subset data set by only keeping important variables
Data_train.mis <- subset(Data_train.mis,select=c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_test.mis <- subset(Data_test.mis,select=c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
```

```
#Observe missing data pattern
md.pattern(Data_train.mis)
md.pattern(Data_test.mis)

#Generate layout of missing values
mice_plot.asia <- aggr(Data_train.mis,
                       col=c('lightblue','red')
                       ,numbers=TRUE
                       ,sortVars=TRUE
                       ,labels=names(Data_train.mis)
                       , cex.axis=.7
                       ,gap=3
                       , ylab=c("Missing data","Pattern"))
mice_plot.asia <- aggr(Data_test.mis,
                       col=c('lightblue','red')
                       ,numbers=TRUE
                       ,sortVars=TRUE
                       ,labels=names(Data_test.mis)
                       , cex.axis=.7
                       ,gap=3
                       , ylab=c("Missing data","Pattern"))

#Use MICE to generate missing values
Data_train.mis <- mice(Data_train.mis, m=5, maxit = 50, method = '
    pmm', seed = 500)
Data_test.mis <- mice(Data_test.mis, m=5, maxit = 50, method = 'pmm'
    , seed = 500)

#Select data set needed for the analysis
completeTest <- complete(Data_train.mis,1)
completeTest <- complete(Data_test.mis,1)
```

152C__Users_Thendo_Desktop_Thesis_Paper_RCode_MICE_building_code.R

E.3 Logistic Classifier

```
#The list of packages used
library(ROCR)
library(caret)
library(DescTools)
library(PRROC)
library(pscl)
```

```
###Case-wise deletion Analysis
##World Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(CWD_Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Test_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
testinsample <- subset(CWD_Data_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_y <- CWD_Test_set$Debt_Crises
y_inner<- CWD_Data_set$Dedt_Crises
y <- CWD_Test_set$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
```

```
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(CWD_Asia,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Asia_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
testinsample <- subset(CWD_Asia,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_y <- CWD_Asia_test$Debt_Crises
y_inner<- CWD_Asia$Dedt_Crises
y <- CWD_Asia_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance
```

```
#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(prob,Test_y)
recall(prob,Test_y)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Latin America Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(CWD_Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Latin_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
testinsample <- subset(CWD_Latin_America,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_y <- CWD_Latin_test$Debt_Crises
```

```
y_inner<- CWD_Latin_America$Dedt_Crises
y <- CWD_Latin_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
    logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
    variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
    generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)
```

```
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##The middile East and Africa Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(CWD_Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Africa_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
testinsample <- subset(CWD_Africa,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_y <- CWD_Africa_test$Debt_Crises
y_inner<- CWD_Africa$Dedt_Crises
y <- CWD_Africa_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))
```

```
#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Western European Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(CWD_Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Europe_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
testinsample <- subset(CWD_Europe,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_y <- CWD_Europe_test$Debt_Crises
y_inner<- CWD_Europe$Dedt_Crises
y <- CWD_Europe_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)
```

```
#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)
```

153C__Users_Thendo_Desktop_Thesis_Paper_RCode_CWD_logit_test.R

```
#The list of packages used
library(ROCR)
library(caret)
library(DescTools)
library(PRROC)
library(pscl)
```

```
###Mean imputation Analysis
##World Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Data_set_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Test_set_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Data_set_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Test_set_mean$Debt_Crises
y_inner<- Data_set_mean$Dedt_Crises
y <- Test_set_mean$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
```

```
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Asia_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Asia_mean_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Asia_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Asia_test_mean$Debt_Crises
y_inner<- Asia_mean$Dedt_Crises
y <- Asia_test_mean$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance
```

```
#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(prob,Test_y)
recall(prob,Test_y)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Latin America Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Latin_America_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Latin_test_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Latin_America_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Latin_test_mean$Debt_Crises
```

```
y_inner<- Latin_America_mean$Dedt_Crises
y <- Latin_test_mean$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
    logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
    variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
    generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)
```

```
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##The middile East and Africa Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Africa_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Africa_test_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Africa_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Africa_test_mean$Debt_Crises
y_inner<- Africa_mean$Dedt_Crises
y <- Africa_test_mean$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))
```

```
#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Western European Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Europe_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Europe_test_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Europe_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Europe_test_mean$Debt_Crises
y_inner<- Europe_mean$Dedt_Crises
y <- Europe_test_mean$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)
```

```
#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)
```

154C__Users_Thendo_Desktop_Thesis_Paper_RCode_Mean_logit_test.R

```
#The list of packages used
library(ROCR)
library(caret)
library(DescTools)
library(PRROC)
library(pscl)
```

```
###Mean imputation Analysis
##World Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Test_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Data_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Test_set$Debt_Crises
y_inner<- Data_set$Dedt_Crises
y <- Test_set$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
```

```
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Asia,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Asia_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Asia,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Asia_test$Debt_Crises
y_inner<- Asia$Dedt_Crises
y <- Asia_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance
```

```
#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(prob,Test_y)
recall(prob,Test_y)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Latin America Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Latin_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Latin_America,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Latin_test$Debt_Crises
```

```
y_inner<- Latin_America$Dedt_Crises
y <- Latin_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
    logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
    variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
    generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)
```

```
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##The middile East and Africa Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Africa_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Africa,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Africa_test$Debt_Crises
y_inner<- Africa$Dedt_Crises
y <- Africa_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))
```

```
#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Western European Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Europe_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Europe,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Europe_test$Debt_Crises
y_inner<- Europe$Dedt_Crises
y <- Europe_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   logit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)
```

```
#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)
```

155C__Users_Thendo_Desktop_Thesis_Paper_RCode_MICE_logit_test.R

E.4 Probit Classifier

```
#The list of packages used
```

```
library(ROCR)
library(caret)
library(DescTools)
library(PRROC)
library(pscl)


###Case-wise deletion Analysis
##World Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(CWD_Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Test_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
testinsample <- subset(CWD_Data_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_y <- CWD_Test_set$Debt_Crises
y_inner<- CWD_Data_set$Dedt_Crises
y <- CWD_Test_set$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
```

```
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(CWD_Asia,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Asia_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
testinsample <- subset(CWD_Asia,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_y <- CWD_Asia_test$Debt_Crises
y_inner<- CWD_Asia$Dedt_Crises
y <- CWD_Asia_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
```

```
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(prob,Test_y)
recall(prob,Test_y)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Latin America Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(CWD_Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
```

```
Test_df <- subset(CWD_Latin_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
testinsample <- subset(CWD_Latin_America,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_y <- CWD_Latin_test$Debt_Crises
y_inner<- CWD_Latin_America$Dedt_Crises
y <- CWD_Latin_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
```

```
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##The middile East and Africa Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(CWD_Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Africa_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
testinsample <- subset(CWD_Africa,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_y <- CWD_Africa_test$Debt_Crises
y_inner<- CWD_Africa$Dedt_Crises
y <- CWD_Africa_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))
```

```
#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Western European Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(CWD_Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Europe_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
testinsample <- subset(CWD_Europe,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_y <- CWD_Europe_test$Debt_Crises
y_inner<- CWD_Europe$Dedt_Crises
y <- CWD_Europe_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
```

```
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)
```



```
#The list of packages used
```

```
library(ROCR)
library(caret)
library(DescTools)
library(PRROC)
library(pscl)


###Mean imputation Analysis
##World Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Data_set_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Test_set_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Data_set_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Test_set_mean$Debt_Crises
y_inner<- Data_set_mean$Dedt_Crises
y <- Test_set_mean$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
```

```
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Asia_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Asia_mean_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Asia_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Asia_test_mean$Debt_Crises
y_inner<- Asia_mean$Dedt_Crises
y <- Asia_test_mean$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
```

```
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(prob,Test_y)
recall(prob,Test_y)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Latin America Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Latin_America_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
```

```
Test_df <- subset(Latin_test_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Latin_America_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Latin_test_mean$Debt_Crises
y_inner<- Latin_America_mean$Dedt_Crises
y <- Latin_test_mean$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
```

```
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##The middile East and Africa Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Africa_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Africa_test_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Africa_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Africa_test_mean$Debt_Crises
y_inner<- Africa_mean$Dedt_Crises
y <- Africa_test_mean$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))
```

```
#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Western European Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Europe_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Europe_test_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Europe_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Europe_test_mean$Debt_Crises
y_inner<- Europe_mean$Dedt_Crises
y <- Europe_test_mean$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
```

```
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)
```

157C__Users_Thendo_Desktop_Thesis_Paper_RCode_Mean_probit_test.R

```
#The list of packages used
```

```
library(ROCR)
library(caret)
library(DescTools)
library(PRROC)
library(pscl)


###Mean imputation Analysis
##World Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Test_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Data_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Test_set$Debt_Crises
y_inner<- Data_set$Dedt_Crises
y <- Test_set$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
```

```
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Asia,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Asia_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Asia,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Asia_test$Debt_Crises
y_inner<- Asia$Dedt_Crises
y <- Asia_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
```

```
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(prob,Test_y)
recall(prob,Test_y)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Latin America Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
```

```
Test_df <- subset(Latin_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Latin_America,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Latin_test$Debt_Crises
y_inner<- Latin_America$Dedt_Crises
y <- Latin_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
```

```
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##The middile East and Africa Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Africa_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Africa,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Africa_test$Debt_Crises
y_inner<- Africa$Dedt_Crises
y <- Africa_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))
```

```
#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)

##Western European Model

#Subset the variables, in order to only have variables to be used
Data_df <- subset(Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Europe_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
testinsample <- subset(Europe,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_y <- Europe_test$Debt_Crises
y_inner<- Europe$Dedt_Crises
y <- Europe_test$Debt_Crises


#Running regression model
model.data_set <- glm(Data_df$Dedt_Crises ~.,family=binomial(link='
   probit'),data=Data_df)
summary(model.data_set)
```

```
round(varImp(model.data_set),4)

#Generating the pseudo R-squared value
pR2(model.data_set)

#Generating deviance
model.data_set$deviance

#Using test and training  set on the model without dependent
   variable to get model generated results.
predict<- predict(model.data_set,Test_df)
predict_inner <- predict(model.data_set, testinsample)

#Setting results to zero or one based on the below criteria
prob <- (ifelse(predict>0.5, 1, 0))
prob_inner <- (ifelse(predict_inner>0.5, 1, 0))

#Generating Accuracy results
confusionMatrix(as.factor(prob_inner),as.factor(y_inner))
confusionMatrix(as.factor(prob),as.factor(y))

#Turning variable to a factor so that is can be used in metrics
   generating
prob <- factor(prob)
Test_y<- factor(Test_y)

#Generating test sensitivity and precision
sensitivity(prob, Test_y)
specificity(prob, Test_y)

#Generating test precision and recall value
precision(Test_y,prob)
recall(Test_y,prob)

#Generating test precision and recall AUC
pr.curve(Test_y,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generating test set Brier score
BrierScore(Y, prob_brier)
```

E.5 Bayesian Additive Regression Trees Classifier

```
#Load the appropriate package
library(BayesTree)
library(Metrics)
library(caret)
library(DescTools)
library(PRROC)
library(ROCR)

###Case-wise deletion
#World Model

#Extract the desirable variables
train_df <- subset(CWD_Data_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(CWD_Data_set, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Test_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Data_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Test_set, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= CWD_Data_set$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= CWD_Data_set$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), CWD_Test_set$Debt_
   Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), CWD_Test_set$
   Debt_Crises)
```

```
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), CWD_Test_set$Debt_
    Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), CWD_
    Data_set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(CWD_Test_set$Debt_Crises,y_test)
accuracy(CWD_Test_set$Debt_Crises,y_test)
accuracy(CWD_Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-CWD_Test_set$Debt_
    Crises)^2
single.rmse <- as.data.frame(sqrt(difference))
```

```
#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
   Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(CWD_Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

#South and East Asia Model

#Extract the desirable variables
train_df <- subset(CWD_Asia, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(CWD_Asia, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Asia_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Asia, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Asia_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
```

```
                    y.train= CWD_Asia$Dedt_Crises,
                    x.test= test_df,
                    binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= CWD_Asia$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), CWD_Test_set$Debt_
    Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), CWD_Test_set$
    Debt_Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), CWD_Test_set$Debt_
    Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), CWD_
    Data_set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(CWD_Test_set$Debt_Crises,y_test)
accuracy(CWD_Test_set$Debt_Crises,y_test)
accuracy(CWD_Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model
```

```
plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-CWD_Test_set$Debt_
    Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(CWD_Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
```

```
BrierScore(Y, prob_brier)

#Latin America Model

#Extract the desirable variables
train_df <- subset(CWD_Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(CWD_Latin_America, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Latin_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Latin_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= CWD_Latin_America$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train=CWD_Latin_America$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), CWD_Test_set$Debt_
   Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), CWD_Test_set$
   Debt_Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), CWD_Test_set$Debt_
   Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), CWD_
   Data_set$Dedt_Crises)

#Plot the prediction
```

```
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(CWD_Test_set$Debt_Crises,y_test)
accuracy(CWD_Test_set$Debt_Crises,y_test)
accuracy(CWD_Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-CWD_Test_set$Debt_
    Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
```

```
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(CWD_Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

#The Middle East and Africa Model

#Extract the desirable variables
train_df <- subset(CWD_Africa, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(CWD_Africa, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Africa_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Africa, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Africa_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= CWD_Africa$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= CWD_Africa$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)
```

```
#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), CWD_Test_set$Debt_
    Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), CWD_Test_set$
    Debt_Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), CWD_Test_set$Debt_
    Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), CWD_
    Data_set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(CWD_Test_set$Debt_Crises,y_test)
accuracy(CWD_Test_set$Debt_Crises,y_test)
accuracy(CWD_Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")
```

```
boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-CWD_Test_set$Debt_
   Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
   Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(CWD_Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

#Western Europe Model

#Extract the desirable variables
train_df <- subset(CWD_Europe, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(CWD_Europe, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
```

```
test_df <- subset(CWD_Europe_test, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Europe, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Europe_test, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= CWD_Europe$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= CWD_Europe$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), CWD_Test_set$Debt_
    Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), CWD_Test_set$
    Debt_Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), CWD_Test_set$Debt_
    Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), CWD_
    Data_set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))
```

```
#Generate AUC along with accuracy score
auc(CWD_Test_set$Debt_Crises,y_test)
accuracy(CWD_Test_set$Debt_Crises,y_test)
accuracy(CWD_Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-CWD_Test_set$Debt_
    Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(CWD_Test_set$Debt_Crises)


#Generating performance metrics score
```

```
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)
```

159C__Users_Thendo_Desktop_Thesis_Paper_RCode_CWD_BART_test.R

```
#Load the appropriate package
library(BayesTree)
library(Metrics)
library(caret)
library(DescTools)
library(PRROC)
library(ROCR)

###Mean Imputation
#World Model

#Extract the desirable variables
train_df <- subset(Data_set_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltrain_df <- subset(Data_set_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df <- subset(Test_set_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df_inner <- subset(Data_set_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltest_df <- subset(Test_set_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= Data_set_mean$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= Data_set_mean$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)
```

```
#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), Data_
    set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(Test_set$Debt_Crises,y_test)
accuracy(Test_set$Debt_Crises,y_test)
accuracy(Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
```

```
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-Test_set$Debt_Crises)
    ^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

#South and East Asia Model

#Extract the desirable variables
train_df <- subset(Asia_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltrain_df <- subset(Asia_mean, select = c
```

```
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df <- subset(Asia_test_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df_inner <- subset(Asia_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltest_df <- subset(Asia_test_mean, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= Asia_mean$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= Asia_mean$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), Data_
    set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))
```

```
#Generate AUC along with accuracy score
auc(Test_set$Debt_Crises,y_test)
accuracy(Test_set$Debt_Crises,y_test)
accuracy(Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-Test_set$Debt_Crises)
    ^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(Test_set$Debt_Crises)
```

```
#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

#Latin America Model

#Extract the desirable variables
train_df <- subset(Latin_America_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltrain_df <- subset(Latin_America_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df <- subset(Latin_test_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df_inner <- subset(Latin_America_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltest_df <- subset(Latin_test_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= Latin_America_mean$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train=Latin_America_mean$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), Test_set$Debt_
```

```
    Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), Data_
    set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(Test_set$Debt_Crises,y_test)
accuracy(Test_set$Debt_Crises,y_test)
accuracy(Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-Test_set$Debt_Crises)
    ^2
```

```
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
   Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

#The Middle East and Africa Model

#Extract the desirable variables
train_df <- subset(Africa_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltrain_df <- subset(Africa_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df <- subset(Africa_test_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df_inner <- subset(Africa_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltest_df <- subset(Africa_test_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Generate model for predicting both in and out of sample data
```

```
model.bart<- bart(x.train=train_df,
                  y.train= Africa_mean$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= Africa_mean$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), Data_
    set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(Test_set$Debt_Crises,y_test)
accuracy(Test_set$Debt_Crises,y_test)
accuracy(Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model
```

```
plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-Test_set$Debt_Crises)
    ^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
```

```
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

#Western Europe Model

#Extract the desirable variables
train_df <- subset(Europe_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltrain_df <- subset(Europe_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df <- subset(Europe_test_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df_inner <- subset(Europe_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltest_df <- subset(Europe_test_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= Europe_mean$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= Europe_mean$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), Data_
   set$Dedt_Crises)
```

```
#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(Test_set$Debt_Crises,y_test)
accuracy(Test_set$Debt_Crises,y_test)
accuracy(Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-Test_set$Debt_Crises)
    ^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
```

```
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)
```

160C__Users_Thendo_Desktop_Thesis_Paper_RCode_Mean_BART_test.R

```
#Load the appropriate package
library(BayesTree)
library(Metrics)
library(caret)
library(DescTools)
library(PRROC)
library(ROCR)

###MICE
#World Model

#Extract the desirable variables
train_df <- subset(Data_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltrain_df <- subset(Data_set, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df <- subset(Test_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df_inner <- subset(Data_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltest_df <- subset(Test_set, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
```

```
#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= Data_set$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= Data_set$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), Data_
   set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
   >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
   .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(Test_set$Debt_Crises,y_test)
accuracy(Test_set$Debt_Crises,y_test)
accuracy(Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
```

```
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-Test_set$Debt_Crises)
    ^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
      main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
```

```
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

#South and East Asia Model

#Extract the desirable variables
train_df <- subset(Asia, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltrain_df <- subset(Asia, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df <- subset(Asia_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df_inner <- subset(Asia, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltest_df <- subset(Asia_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= Asia$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= Asia$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), Data_
   set$Dedt_Crises)
```

```
#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(Test_set$Debt_Crises,y_test)
accuracy(Test_set$Debt_Crises,y_test)
accuracy(Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-Test_set$Debt_Crises)
    ^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
```

```
    Single-RMSE",
      main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

#Latin America Model

#Extract the desirable variables
train_df <- subset(Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltrain_df <- subset(Latin_America, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df <- subset(Latin_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df_inner <- subset(Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltest_df <- subset(Latin_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= Latin_America$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train=Latin_America$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)
```

```
#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), Data_
    set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(Test_set$Debt_Crises,y_test)
accuracy(Test_set$Debt_Crises,y_test)
accuracy(Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
```

```
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-Test_set$Debt_Crises)
    ^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

#The Middle East and Africa Model

#Extract the desirable variables
train_df <- subset(Africa, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
```

```
alltrain_df <- subset(Africa, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df <- subset(Africa_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df_inner <- subset(Africa, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltest_df <- subset(Africa_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= Africa$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= Africa$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
bart.rmse

#Generate RMSE values for the training set
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), Data_
   set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
   >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
```

```
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(Test_set$Debt_Crises,y_test)
accuracy(Test_set$Debt_Crises,y_test)
accuracy(Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-Test_set$Debt_Crises)
    ^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(Test_set$Debt_Crises)
```

```
#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

#Western Europe Model

#Extract the desirable variables
train_df <- subset(Europe, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltrain_df <- subset(Europe, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df <- subset(Europe_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
test_df_inner <- subset(Europe, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
alltest_df <- subset(Europe_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

#Generate model for predicting both in and out of sample data
model.bart<- bart(x.train=train_df,
                  y.train= Europe$Dedt_Crises,
                  x.test= test_df,
                  binaryOffset=0)
model.bart.inner<- bart(x.train=train_df,
                        y.train= Europe$Dedt_Crises,
                        x.test= test_df_inner,
                        binaryOffset=0)



#Generate RMSE values for the test set
bart.rmse <- rmse(colMeans(model.bart$yhat.test), Test_set$Debt_
   Crises)
bart.rmse

#Generate RMSE values for the training set
```

```
bart.rrmse <- rrmse(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
bart.rrmse

#Generate prediction from the model
pr <- prediction(colMeans(model.bart$yhat.test), Test_set$Debt_
    Crises)
pr_inner <- prediction(colMeans(model.bart.inner$yhat.test), Data_
    set$Dedt_Crises)

#Plot the prediction
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)

#Converting outcomes to binary
y_test <- as.data.frame( ifelse(colMeans(model.bart$yhat.test)
    >0.5,1,0))
y_test_inner <- as.data.frame( ifelse(colMeans(model.bart.inner$yhat
    .test)>0.5,1,0))

#Generate AUC along with accuracy score
auc(Test_set$Debt_Crises,y_test)
accuracy(Test_set$Debt_Crises,y_test)
accuracy(Data_set$Dedt_Crises,y_test_inner)


par(mfrow=c(3,2))
#Plotting the desired graphs based on the outputs of the model

plot(model.bart$yhat.test, xlab="posterior inteval",ylab="out-of-
    sample f(x) ",main="BART out-sample")
plot(model.bart$yhat.train, xlab="posterior inteval",ylab="in-sample
     f(x) ",main="BART in-sample")

plot(colMeans(model.bart$yhat.test), xlab="posterior inteval",ylab="
    out-of-sample f(x) ",main="BART out-sample observation")
plot(colMeans(model.bart$yhat.train), xlab="posterior inteval",ylab=
    "in-sample f(x) ",main="BART in-sample observation")

boxplot(colMeans(model.bart$yhat.test))
boxplot(colMeans(model.bart$yhat.train))


par(mfrow=c(1,2))
difference <- (colMeans(model.bart$yhat.test)-Test_set$Debt_Crises)
```

```
    ^2
single.rmse <- as.data.frame(sqrt(difference))

#Boxplot of RMSE values
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
#Line gra
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)


#Converting variable to binary
prob <- factor(ifelse(colMeans(model.bart$yhat.test) > 0.5, 1, 0))
Test_Y1<- factor(Test_set$Debt_Crises)


#Generating performance metrics score
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)

#Converting results so to be generated by Brierscore function
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)
```

161C__Users_Thendo_Desktop_Thesis_Paper_RCode_MICE_BART_test.R

E.6 Artificial Neural Network Classifier

```
#Load the desired packages
library(neuralnet)
library(ROCR)
library(Metrics)
library(caret)
library(DescTools)
library(PRROC)
```

```
##Case-wise deletion
#World Model

#Extract desired variables
Data_df <- subset(CWD_Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Data_df_t <- subset(CWD_Data_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Data_Y <-  subset(CWD_Data_set,select=c(3))
Test_df <- subset(CWD_Test_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_Y <- subset(CWD_Test_set,select=c(3))
y <-  CWD_Test_set$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
                   hidden=6,
                   act.fct = "logistic",
                   linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
```

```
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)

#South and EaST Asia Model

#Extract desired variables
Data_df <- subset(CWD_Asia,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Data_df_t <- subset(CWD_Asia,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Data_Y <-  subset(CWD_Asia,select=c(3))
Test_df <- subset(CWD_Asia_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_Y <- subset(CWD_Asia_test,select=c(3))
y <-  CWD_Asia_test$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
```

```
                        Trade_Openess +
                        Export_growth +
                        Current_account +
                        FDI +
                        Real_GDP_growth +
                        Inflation +
                        Reer +
                        Gov_expenditures +
                        National_savings +
                        Domestic_credit +
                        Bank_assets,
                      data=Data_df,
                      hidden=6,
                      act.fct = "logistic",
                      linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)
```

```
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)


#Latin America Model

#Extract desired variables
Data_df <- subset(CWD_Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Data_df_t <- subset(CWD_Latin_America,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Data_Y <-  subset(CWD_Latin_America,select=c(3))
Test_df <- subset(CWD_Latin_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_Y <- subset(CWD_Latin_test,select=c(3))
y <-  CWD_Latin_test$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
                   hidden=6,
                   act.fct = "logistic",
                   linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)
```

```
prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)


#The Middle East and Africa Model

#Extract desired variables
Data_df <- subset(CWD_Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Data_df_t <- subset(CWD_Africa,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Data_Y <-  subset(CWD_Africa,select=c(3))
Test_df <- subset(CWD_Africa_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_Y <- subset(CWD_Africa_test,select=c(3))
y <-  CWD_Africa_test$Debt_Crises
```

```
#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
                   hidden=6,
                   act.fct = "logistic",
                   linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
```

```
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)


#Western Europe Model

#Extract desired variables
Data_df <- subset(CWD_Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Data_df_t <- subset(CWD_Europe,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Data_Y <-  subset(CWD_Europe,select=c(3))
Test_df <- subset(CWD_Europe_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_Y <- subset(CWD_Europe_test,select=c(3))
y <-  CWD_Europe_test$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
```

```
                    hidden=6,
                    act.fct = "logistic",
                    linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)
```

162C__Users_Thendo_Desktop_Thesis_Paper_RCode_CWD_ANN_test.R

```
#Load the desired packages
library(neuralnet)
library(ROCR)
library(Metrics)
library(caret)
```

```
library(DescTools)
library(PRROC)

##Mean imputation
#World Model

#Extract desired variables
Data_df <- subset(Data_set_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_df_t <- subset(Data_set_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_Y <-  subset(Data_set_mean,select=c(3))
Test_df <- subset(Test_set_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_Y <- subset(Test_set_mean,select=c(3))
y <-  Test_set_mean$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
                   hidden=6,
                   act.fct = "logistic",
                   linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
```

```
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)

#South and EaST Asia Model

#Extract desired variables
Data_df <- subset(Asia_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_df_t <- subset(Asia_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_Y <-  subset(Asia_mean,select=c(3))
Test_df <- subset(Asia_test_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_Y <- subset(Asia_test_mean,select=c(3))
y <-  Asia_test_mean$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
```

```
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
                   hidden=6,
                   act.fct = "logistic",
                   linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
```

```
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)


#Latin America Model

#Extract desired variables
Data_df <- subset(Latin_America_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_df_t <- subset(Latin_America_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_Y <-  subset(Latin_America_mean,select=c(3))
Test_df <- subset(Latin_test_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_Y <- subset(Latin_test_mean,select=c(3))
y <-  Latin_test_mean$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
                   hidden=6,
                   act.fct = "logistic",
                   linear.output = FALSE)
```

```
#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)


#The Middle East and Africa Model

#Extract desired variables
Data_df <- subset(Africa_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_df_t <- subset(Africa_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_Y <-  subset(Africa_mean,select=c(3))
Test_df <- subset(Africa_test_mean,select=c
```

```
    (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_Y <- subset(Africa_test_mean,select=c(3))
y <-  Africa_test_mean$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
                   hidden=6,
                   act.fct = "logistic",
                   linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))
```

```
#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)


#Western Europe Model

#Extract desired variables
Data_df <- subset(Europe_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_df_t <- subset(Europe_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_Y <-  subset(Europe_mean,select=c(3))
Test_df <- subset(Europe_test_mean,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_Y <- subset(Europe_test_mean,select=c(3))
y <-  Europe_test_mean$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
```

```
                        Domestic_credit +
                        Bank_assets,
                      data=Data_df,
                      hidden=6,
                      act.fct = "logistic",
                      linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)
```

163C__Users_Thendo_Desktop_Thesis_Paper_RCode_Mean_NN_test.R

```
#Load the desired packages
library(neuralnet)
```

```
library(ROCR)
library(Metrics)
library(caret)
library(DescTools)
library(PRROC)

##Mean imputation
#World Model

#Extract desired variables
Data_df <- subset(Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_df_t <- subset(Data_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_Y <-  subset(Data_set,select=c(3))
Test_df <- subset(Test_set,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_Y <- subset(Test_set,select=c(3))
y <-  Test_set$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
                   hidden=6,
                   act.fct = "logistic",
                   linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)
```

```
prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)

#South and EaST Asia Model

#Extract desired variables
Data_df <- subset(Asia,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_df_t <- subset(Asia,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_Y <-  subset(Asia,select=c(3))
Test_df <- subset(Asia_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_Y <- subset(Asia_test,select=c(3))
y <-  Asia_test$Debt_Crises
```

```
#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
                   hidden=6,
                   act.fct = "logistic",
                   linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
```

```
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)


#Latin America Model

#Extract desired variables
Data_df <- subset(Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_df_t <- subset(Latin_America,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_Y <-  subset(Latin_America,select=c(3))
Test_df <- subset(Latin_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_Y <- subset(Latin_test,select=c(3))
y <-  Latin_test$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
                   hidden=6,
```

```
                    act.fct = "logistic",
                    linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)


#The Middle East and Africa Model

#Extract desired variables
Data_df <- subset(Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_df_t <- subset(Africa,select=c
```

```
    (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_Y <-  subset(Africa,select=c(3))
Test_df <- subset(Africa_test,select=c
    (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_Y <- subset(Africa_test,select=c(3))
y <-  Africa_test$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
    in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
                     Reer +
                     Gov_expenditures +
                     National_savings +
                     Domestic_credit +
                     Bank_assets,
                   data=Data_df,
                   hidden=6,
                   act.fct = "logistic",
                   linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)
```

```
prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
BrierScore(Y, prob_brier)


#Western Europe Model

#Extract desired variables
Data_df <- subset(Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_df_t <- subset(Europe,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Data_Y <-  subset(Europe,select=c(3))
Test_df <- subset(Europe_test,select=c
   (5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_Y <- subset(Europe_test,select=c(3))
y <-  Europe_test$Debt_Crises

#Build neural network model using training set data
##hidden is set to 6,7,8,9 and 10 for different neurons considered
   in the model
model <- neuralnet(Dedt_Crises ~
                     Global_Interest +
                     FX_reserves +
                     Trade_Openess +
                     Export_growth +
                     Current_account +
                     FDI +
                     Real_GDP_growth +
                     Inflation +
```

```
                        Reer +
                        Gov_expenditures +
                        National_savings +
                        Domestic_credit +
                        Bank_assets,
                      data=Data_df,
                      hidden=6,
                      act.fct = "logistic",
                      linear.output = FALSE)

#Predict outcomes from new inputs using the model
predict<- compute(model,Test_df)
predict_inner <- compute(model,Data_df_t)

prob <- predict$net.result
prob_brier <- predict$net.result
prob_inner <- predict_inner$net.result

#convert results to binary
prob <- (ifelse(prob>0.5, 1, 0))
prob_inner <- (ifelse(prob_inner>0.5, 1, 0))

#Generate accuracy results
accuracy(prob_inner,Data_Y)
accuracy(prob,Test_Y)


prob <- factor(prob)
Test_Y1<- factor(ifelse(Test_Y > 0.5, 1, 0))

#Generate performance metrics scores
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob, Test_Y1)
recall(prob, Test_Y1)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))

#Generate brierscore
```

```
BrierScore(Y, prob_brier)
```

164C__Users_Thendo_Desktop_Thesis_Paper_RCode_MICE_NN_test.R

E.7 Regression Analysis

```
###MICE
#World
#Rename data set to make it more generic to use
completeData <- Data_set

#Subset based on observations for default and non-default
crises<- subset(completeData, completeData$Dedt_Crises==1)
non_crises <- subset(completeData, completeData$Dedt_Crises==0)


#Run the t-test on the different crisis cases also obtaining the
   summary statistics
t.test(crises$Global_Interest,non_crises$Global_Interest)
t.test(crises$FX_reserves,non_crises$FX_reserves)
t.test(crises$Trade_Openess,non_crises$Trade_Openess)
t.test(crises$Export_growth,non_crises$Export_growth)
t.test(crises$Current_account,non_crises$Current_account)
t.test(crises$FDI,non_crises$FDI)
t.test(crises$Real_GDP_growth,non_crises$Real_GDP_growth)
t.test(crises$Inflation,non_crises$Inflation)
t.test(crises$Reer,non_crises$Reer)
t.test(crises$Gov_expenditures,non_crises$Gov_expenditures)
t.test(crises$National_savings,non_crises$National_savings)
t.test(crises$Domestic_credit,non_crises$Domestic_credit)
t.test(crises$Bank_assets,non_crises$Bank_assets)

#Obtain the varianve of the difference variables per crisis
   situation
var(crises$Total_debt)
var(non_crises$Total_debt)
var(crises$Global_Interest)
var(non_crises$Global_Interest)
var(crises$FX_reserves)
var(non_crises$FX_reserves)
var(crises$Trade_Openess)
var(non_crises$Trade_Openess)
var(crises$Export_growth)
```

```
var(non_crises$Export_growth)
var(crises$Current_account)
var(non_crises$Current_account)
var(crises$FDI)
var(non_crises$FDI)
var(crises$Real_GDP_growth)
var(non_crises$Real_GDP_growth)
var(crises$Inflation)
var(non_crises$Inflation)
var(crises$Reer)
var(non_crises$Reer)
var(crises$Gov_expenditures)
var(non_crises$Gov_expenditures)
var(crises$National_savings)
var(non_crises$National_savings)
var(crises$Domestic_credit)
var(non_crises$Domestic_credit)
var(crises$Bank_assets)
var(non_crises$Bank_assets)

test <- glm(formula=completeData$Dedt_Crises~
              Global_Interest +
              FX_reserves+
              Trade_Openess+
              Real_GDP_growth+
              Reer+
              Inflation+
              National_savings+
              Gov_expenditures+
              Domestic_credit+
              Bank_assets+
              Export_growth+
              Current_account ,completeData, family = "binomial")
par(mfrow=c(2,2))
plot(test)
#South and East Asia
#Rename data set to make it more generic to use
completeData <- Asia

#Subset based on observations for default and non-default
crises<- subset(completeData, completeData$Dedt_Crises==1)
non_crises <- subset(completeData, completeData$Dedt_Crises==0)
```

```
#Run the t-test on the different crisis cases also obtaining the
    summary statistics
t.test(crises$Global_Interest,non_crises$Global_Interest)
t.test(crises$FX_reserves,non_crises$FX_reserves)
t.test(crises$Trade_Openess,non_crises$Trade_Openess)
t.test(crises$Export_growth,non_crises$Export_growth)
t.test(crises$Current_account,non_crises$Current_account)
t.test(crises$FDI,non_crises$FDI)
t.test(crises$Real_GDP_growth,non_crises$Real_GDP_growth)
t.test(crises$Inflation,non_crises$Inflation)
t.test(crises$Reer,non_crises$Reer)
t.test(crises$Gov_expenditures,non_crises$Gov_expenditures)
t.test(crises$National_savings,non_crises$National_savings)
t.test(crises$Domestic_credit,non_crises$Domestic_credit)
t.test(crises$Bank_assets,non_crises$Bank_assets)

#Obtain the varianve of the difference variables per crisis
    situation
var(crises$Total_debt)
var(non_crises$Total_debt)
var(crises$Global_Interest)
var(non_crises$Global_Interest)
var(crises$FX_reserves)
var(non_crises$FX_reserves)
var(crises$Trade_Openess)
var(non_crises$Trade_Openess)
var(crises$Export_growth)
var(non_crises$Export_growth)
var(crises$Current_account)
var(non_crises$Current_account)
var(crises$FDI)
var(non_crises$FDI)
var(crises$Real_GDP_growth)
var(non_crises$Real_GDP_growth)
var(crises$Inflation)
var(non_crises$Inflation)
var(crises$Reer)
var(non_crises$Reer)
var(crises$Gov_expenditures)
var(non_crises$Gov_expenditures)
var(crises$National_savings)
var(non_crises$National_savings)
var(crises$Domestic_credit)
var(non_crises$Domestic_credit)
var(crises$Bank_assets)
```

```
var(non_crises$Bank_assets)

test <- glm(formula=completeData$Dedt_Crises~
              Global_Interest +
              FX_reserves+
              Trade_Openess+
              Real_GDP_growth+
              Reer+
              Inflation+
              National_savings+
              Gov_expenditures+
              Domestic_credit+
              Bank_assets+
              Export_growth+
              Current_account ,completeData, family = "binomial")
par(mfrow=c(2,2))
plot(test)

#Latin America
#Rename data set to make it more generic to use
completeData <- Latin_America

#Subset based on observations for default and non-default
crises<- subset(completeData, completeData$Dedt_Crises==1)
non_crises <- subset(completeData, completeData$Dedt_Crises==0)


#Run the t-test on the different crisis cases also obtaining the
   summary statistics
t.test(crises$Global_Interest,non_crises$Global_Interest)
t.test(crises$FX_reserves,non_crises$FX_reserves)
t.test(crises$Trade_Openess,non_crises$Trade_Openess)
t.test(crises$Export_growth,non_crises$Export_growth)
t.test(crises$Current_account,non_crises$Current_account)
t.test(crises$FDI,non_crises$FDI)
t.test(crises$Real_GDP_growth,non_crises$Real_GDP_growth)
t.test(crises$Inflation,non_crises$Inflation)
t.test(crises$Reer,non_crises$Reer)
t.test(crises$Gov_expenditures,non_crises$Gov_expenditures)
t.test(crises$National_savings,non_crises$National_savings)
t.test(crises$Domestic_credit,non_crises$Domestic_credit)
t.test(crises$Bank_assets,non_crises$Bank_assets)

#Obtain the varianve of the difference variables per crisis
   situation
```

```
var(crises$Total_debt)
var(non_crises$Total_debt)
var(crises$Global_Interest)
var(non_crises$Global_Interest)
var(crises$FX_reserves)
var(non_crises$FX_reserves)
var(crises$Trade_Openess)
var(non_crises$Trade_Openess)
var(crises$Export_growth)
var(non_crises$Export_growth)
var(crises$Current_account)
var(non_crises$Current_account)
var(crises$FDI)
var(non_crises$FDI)
var(crises$Real_GDP_growth)
var(non_crises$Real_GDP_growth)
var(crises$Inflation)
var(non_crises$Inflation)
var(crises$Reer)
var(non_crises$Reer)
var(crises$Gov_expenditures)
var(non_crises$Gov_expenditures)
var(crises$National_savings)
var(non_crises$National_savings)
var(crises$Domestic_credit)
var(non_crises$Domestic_credit)
var(crises$Bank_assets)
var(non_crises$Bank_assets)

test <- glm(formula=completeData$Dedt_Crises~
              Global_Interest +
              FX_reserves+
              Trade_Openess+
              Real_GDP_growth+
              Reer+
              Inflation+
              National_savings+
              Gov_expenditures+
              Domestic_credit+
              Bank_assets+
              Export_growth+
              Current_account ,completeData, family = "binomial")
par(mfrow=c(2,2))
plot(test)
```

```
#The Middle East and Africa
#Rename data set to make it more generic to use
completeData <- Africa

#Subset based on observations for default and non-default
crises<- subset(completeData, completeData$Dedt_Crises==1)
non_crises <- subset(completeData, completeData$Dedt_Crises==0)


#Run the t-test on the different crisis cases also obtaining the
   summary statistics
t.test(crises$Global_Interest,non_crises$Global_Interest)
t.test(crises$FX_reserves,non_crises$FX_reserves)
t.test(crises$Trade_Openess,non_crises$Trade_Openess)
t.test(crises$Export_growth,non_crises$Export_growth)
t.test(crises$Current_account,non_crises$Current_account)
t.test(crises$FDI,non_crises$FDI)
t.test(crises$Real_GDP_growth,non_crises$Real_GDP_growth)
t.test(crises$Inflation,non_crises$Inflation)
t.test(crises$Reer,non_crises$Reer)
t.test(crises$Gov_expenditures,non_crises$Gov_expenditures)
t.test(crises$National_savings,non_crises$National_savings)
t.test(crises$Domestic_credit,non_crises$Domestic_credit)
t.test(crises$Bank_assets,non_crises$Bank_assets)

#Obtain the varianve of the difference variables per crisis
   situation
var(crises$Total_debt)
var(non_crises$Total_debt)
var(crises$Global_Interest)
var(non_crises$Global_Interest)
var(crises$FX_reserves)
var(non_crises$FX_reserves)
var(crises$Trade_Openess)
var(non_crises$Trade_Openess)
var(crises$Export_growth)
var(non_crises$Export_growth)
var(crises$Current_account)
var(non_crises$Current_account)
var(crises$FDI)
var(non_crises$FDI)
var(crises$Real_GDP_growth)
var(non_crises$Real_GDP_growth)
var(crises$Inflation)
var(non_crises$Inflation)
```

```
var(crises$Reer)
var(non_crises$Reer)
var(crises$Gov_expenditures)
var(non_crises$Gov_expenditures)
var(crises$National_savings)
var(non_crises$National_savings)
var(crises$Domestic_credit)
var(non_crises$Domestic_credit)
var(crises$Bank_assets)
var(non_crises$Bank_assets)

test <- glm(formula=completeData$Dedt_Crises~
              Global_Interest +
              FX_reserves+
              Trade_Openess+
              Real_GDP_growth+
              Reer+
              Inflation+
              National_savings+
              Gov_expenditures+
              Domestic_credit+
              Bank_assets+
              Export_growth+
              Current_account ,completeData, family = "binomial")
par(mfrow=c(2,2))
plot(test)

#Europe
#Rename data set to make it more generic to use
completeData <- Europe

#Subset based on observations for default and non-default
crises<- subset(completeData, completeData$Dedt_Crises==1)
non_crises <- subset(completeData, completeData$Dedt_Crises==0)


#Run the t-test on the different crisis cases also obtaining the
   summary statistics
t.test(crises$Global_Interest,non_crises$Global_Interest)
t.test(crises$FX_reserves,non_crises$FX_reserves)
t.test(crises$Trade_Openess,non_crises$Trade_Openess)
t.test(crises$Export_growth,non_crises$Export_growth)
t.test(crises$Current_account,non_crises$Current_account)
t.test(crises$FDI,non_crises$FDI)
t.test(crises$Real_GDP_growth,non_crises$Real_GDP_growth)
```

```
t.test(crises$Inflation,non_crises$Inflation)
t.test(crises$Reer,non_crises$Reer)
t.test(crises$Gov_expenditures,non_crises$Gov_expenditures)
t.test(crises$National_savings,non_crises$National_savings)
t.test(crises$Domestic_credit,non_crises$Domestic_credit)
t.test(crises$Bank_assets,non_crises$Bank_assets)

#Obtain the varianve of the difference variables per crisis
    situation
var(crises$Total_debt)
var(non_crises$Total_debt)
var(crises$Global_Interest)
var(non_crises$Global_Interest)
var(crises$FX_reserves)
var(non_crises$FX_reserves)
var(crises$Trade_Openess)
var(non_crises$Trade_Openess)
var(crises$Export_growth)
var(non_crises$Export_growth)
var(crises$Current_account)
var(non_crises$Current_account)
var(crises$FDI)
var(non_crises$FDI)
var(crises$Real_GDP_growth)
var(non_crises$Real_GDP_growth)
var(crises$Inflation)
var(non_crises$Inflation)
var(crises$Reer)
var(non_crises$Reer)
var(crises$Gov_expenditures)
var(non_crises$Gov_expenditures)
var(crises$National_savings)
var(non_crises$National_savings)
var(crises$Domestic_credit)
var(non_crises$Domestic_credit)
var(crises$Bank_assets)
var(non_crises$Bank_assets)

test <- glm(formula=completeData$Dedt_Crises~
                Global_Interest +
                FX_reserves+
                Trade_Openess+
                Real_GDP_growth+
                Reer+
                Inflation+
```

```
                National_savings+
                Gov_expenditures+
                Domestic_credit+
                Bank_assets+
                Export_growth+
                Current_account ,completeData, family = "binomial")
par(mfrow=c(2,2))
plot(test)
```

165C__Users_Thendo_Desktop_Thesis_Paper_RCode_MICE_regression_test.R

E.8 5-Fold Binary Logit Classifier

```
#Load required packages
library(readr)
library(ROCR)
library(Metrics)
library(pscl)
library(caret)
library(DescTools)
library(PRROC)

###Case-wise deletion
##World Model

#Subset important variable
Data_df <- subset(CWD_Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Test_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y_inner<- CWD_Data_set$Dedt_Crises
y <- CWD_Test_set$Debt_Crises
Test_y <- factor(CWD_Test_set$Debt_Crises)
df <- subset(CWD_Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"
```

```
#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~ Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "logit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))
```

```
#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset important variable
Data_df <- subset(CWD_Asia,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Asia_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y_inner<- CWD_Asia$Dedt_Crises
y <- CWD_Asia_test$Debt_Crises
Test_y <- factor(CWD_Asia_test$Debt_Crises)
df <- subset(CWD_Asia,select=c(3,5,6,7,8,9,10,11,12,13,14,15,16,17))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~ Global_Interest +
```

```
                    FX_reserves +
                    Trade_Openess +
                    Export_growth +
                    Current_account +
                    FDI +
                    Real_GDP_growth +
                    Inflation +
                    Reer +
                    Gov_expenditures +
                    National_savings +
                    Domestic_credit +
                    Bank_assets,
                  data=Data_df,
                  method = "glm",
                  family=binomial(link = "logit"),
                  trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
```

```
BrierScore(Y, prob_brier)

##Latin America Model

#Subset important variable
Data_df <- subset(CWD_Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Latin_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y_inner<- CWD_Latin_America$Dedt_Crises
y <- CWD_Latin_test$Debt_Crises
Test_y <- factor(CWD_Latin_test$Debt_Crises)
df <- subset(CWD_Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~ Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
```

```
                    Bank_assets,
                  data=Data_df,
                  method = "glm",
                  family=binomial(link = "logit"),
                  trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##The Middle East and Africa Model

#Subset important variable
Data_df <- subset(CWD_Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Africa_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y_inner<- CWD_Africa$Dedt_Crises
y <- CWD_Africa_test$Debt_Crises
```

```
Test_y <- factor(CWD_Africa_test$Debt_Crises)
df <- subset(CWD_Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~ Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "logit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)
```

```
#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##World Model

#Subset important variable
Data_df <- subset(CWD_Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Europe_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y_inner<- CWD_Europe$Dedt_Crises
y <- CWD_Europe_test$Debt_Crises
Test_y <- factor(CWD_Europe_test$Debt_Crises)
df <- subset(CWD_Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"
```

```
#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~ Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "logit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))
```

```
#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)
```

166C__Users_Thendo_Desktop_Thesis_Paper_RCode_CWD_5–fold_logit_test.R

```
#Load required packages
library(readr)
library(ROCR)
library(Metrics)
library(pscl)
library(caret)
library(DescTools)
library(PRROC)

###Case-wise deletion
##World Model

#Subset important variable
Data_df <- subset(Data_set_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Test_set_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Data_set_mean$Dedt_Crises
y <- Test_set_mean$Debt_Crises
Test_y <- factor(Test_set_mean$Debt_Crises)
df <- subset(Data_set_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"
```

```
#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "logit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
```

```
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset important variable
Data_df <- subset(Asia_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Asia_test_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Asia_mean$Dedt_Crises
y <- Asia_test_mean$Debt_Crises
Test_y <- factor(Asia_test_mean$Debt_Crises)
df <- subset(Asia_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)
```

```
#Specify logistic regression model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "logit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
```

```
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##Latin America Model

#Subset important variable
Data_df <- subset(Latin_America_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Latin_test_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Latin_America_mean$Dedt_Crises
y <- Latin_test_mean$Debt_Crises
Test_y <- factor(Latin_test_mean$Debt_Crises)
df <- subset(Latin_America_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
```

```
                    FDI +
                    Real_GDP_growth +
                    Inflation +
                    Reer +
                    Gov_expenditures +
                    National_savings +
                    Domestic_credit +
                    Bank_assets,
                  data=Data_df,
                  method = "glm",
                  family=binomial(link = "logit"),
                  trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##The Middle East and Africa Model
```

```
#Subset important variable
Data_df <- subset(Africa_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Africa_test_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Africa_mean$Dedt_Crises
y <- Africa_test_mean$Debt_Crises
Test_y <- factor(Africa_test_mean$Debt_Crises)
df <- subset(Africa_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
```

```
                  method = "glm",
                  family=binomial(link = "logit"),
                  trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##World Model

#Subset important variable
Data_df <- subset(Europe_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Europe_test_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Europe_mean$Dedt_Crises
y <- Europe_test_mean$Debt_Crises
Test_y <- factor(Europe_test_mean$Debt_Crises)
df <- subset(Europe_mean,select=c
```

```
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "logit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)
```

```
#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)
```

167C__Users_Thendo_Desktop_Thesis_Paper_RCode_Mean_5–fold_logit_test.R

```
#Load required packages
library(readr)
library(ROCR)
library(Metrics)
library(pscl)
library(caret)
library(DescTools)
library(PRROC)

###Case-wise deletion
##World Model

#Subset important variable
Data_df <- subset(Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Test_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Data_set$Dedt_Crises
y <- Test_set$Debt_Crises
```

```
Test_y <- factor(Test_set$Debt_Crises)
df <- subset(Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "logit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
```

```
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset important variable
Data_df <- subset(Asia,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Asia_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Asia$Dedt_Crises
y <- Asia_test$Debt_Crises
Test_y <- factor(Asia_test$Debt_Crises)
df <- subset(Asia,select=c(3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
```

```
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "logit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
```

```
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##Latin America Model

#Subset important variable
Data_df <- subset(Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Latin_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Latin_America$Dedt_Crises
y <- Latin_test$Debt_Crises
Test_y <- factor(Latin_test$Debt_Crises)
df <- subset(Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
```

```
                            classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "logit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
```

```
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##The Middle East and Africa Model

#Subset important variable
Data_df <- subset(Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Africa_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Africa$Dedt_Crises
y <- Africa_test$Debt_Crises
Test_y <- factor(Africa_test$Debt_Crises)
df <- subset(Africa,select=c(3,5,6,7,8,9,10,11,12,13,14,15,16,17,18)
   )


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
```

```
                    Current_account +
                    FDI +
                    Real_GDP_growth +
                    Inflation +
                    Reer +
                    Gov_expenditures +
                    National_savings +
                    Domestic_credit +
                    Bank_assets,
                  data=Data_df,
                  method = "glm",
                  family=binomial(link = "logit"),
                  trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##World Model
```

```
#Subset important variable
Data_df <- subset(Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Europe_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Europe$Dedt_Crises
y <- Europe_test$Debt_Crises
Test_y <- factor(Europe_test$Debt_Crises)
df <- subset(Europe,select=c(3,5,6,7,8,9,10,11,12,13,14,15,16,17,18)
   )


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
```

```
                  data=Data_df,
                  method = "glm",
                  family=binomial(link = "logit"),
                  trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)
```

168C__Users_Thendo_Desktop_Thesis_Paper_RCode_MICE_5–fold_logit_test.R

E.9 5-Fold Probit Classifier

```
#Load required packages
library(readr)
library(ROCR)
library(Metrics)
```

```
library(pscl)
library(caret)
library(DescTools)
library(PRROC)

###Case-wise deletion
##World Model

#Subset important variable
Data_df <- subset(CWD_Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Test_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y_inner<- CWD_Data_set$Dedt_Crises
y <- CWD_Test_set$Debt_Crises
Test_y <- factor(CWD_Test_set$Debt_Crises)
df <- subset(CWD_Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~ Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
```

```
                    Reer +
                    Gov_expenditures +
                    National_savings +
                    Domestic_credit +
                    Bank_assets,
                  data=Data_df,
                  method = "glm",
                  family=binomial(link = "probit"),
                  trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset important variable
Data_df <- subset(CWD_Asia,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
```

```
Test_df <- subset(CWD_Asia_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y_inner<- CWD_Asia$Dedt_Crises
y <- CWD_Asia_test$Debt_Crises
Test_y <- factor(CWD_Asia_test$Debt_Crises)
df <- subset(CWD_Asia,select=c(3,5,6,7,8,9,10,11,12,13,14,15,16,17))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~ Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "probit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)
```

```
#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##Latin America Model

#Subset important variable
Data_df <- subset(CWD_Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Latin_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y_inner<- CWD_Latin_America$Dedt_Crises
y <- CWD_Latin_test$Debt_Crises
Test_y <- factor(CWD_Latin_test$Debt_Crises)
df <- subset(CWD_Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"
```

```
Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~ Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "probit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
```

```
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##The Middle East and Africa Model

#Subset important variable
Data_df <- subset(CWD_Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Africa_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y_inner<- CWD_Africa$Dedt_Crises
y <- CWD_Africa_test$Debt_Crises
Test_y <- factor(CWD_Africa_test$Debt_Crises)
df <- subset(CWD_Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
```

```
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~ Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "probit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
```

```
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##World Model

#Subset important variable
Data_df <- subset(CWD_Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
Test_df <- subset(CWD_Europe_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y_inner<- CWD_Europe$Dedt_Crises
y <- CWD_Europe_test$Debt_Crises
Test_y <- factor(CWD_Europe_test$Debt_Crises)
df <- subset(CWD_Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify logistic regression model
model1 <- train(Dedt_Crises ~ Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
```

```
                    Reer +
                    Gov_expenditures +
                    National_savings +
                    Domestic_credit +
                    Bank_assets,
                  data=Data_df,
                  method = "glm",
                  family=binomial(link = "probit"),
                  trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)
```

169C__Users_Thendo_Desktop_Thesis_Paper_RCode_CWD_5–fold_probit_test.R

```
#Load required packages
library(readr)
library(ROCR)
library(Metrics)
```

```
library(pscl)
library(caret)
library(DescTools)
library(PRROC)

###Case-wise deletion
##World Model

#Subset important variable
Data_df <- subset(Data_set_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Test_set_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Data_set_mean$Dedt_Crises
y <- Test_set_mean$Debt_Crises
Test_y <- factor(Test_set_mean$Debt_Crises)
df <- subset(Data_set_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify probit model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
```

```
                    Real_GDP_growth +
                    Inflation +
                    Reer +
                    Gov_expenditures +
                    National_savings +
                    Domestic_credit +
                    Bank_assets,
                  data=Data_df,
                  method = "glm",
                  family=binomial(link = "probit"),
                  trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset important variable
```

```
Data_df <- subset(Asia_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Asia_test_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Asia_mean$Dedt_Crises
y <- Asia_test_mean$Debt_Crises
Test_y <- factor(Asia_test_mean$Debt_Crises)
df <- subset(Asia_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify probit model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
```

```
                family=binomial(link = "probit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##Latin America Model

#Subset important variable
Data_df <- subset(Latin_America_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Latin_test_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Latin_America_mean$Dedt_Crises
y <- Latin_test_mean$Debt_Crises
Test_y <- factor(Latin_test_mean$Debt_Crises)
df <- subset(Latin_America_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
```

```
#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify probit model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "probit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)
```

```
#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##The Middle East and Africa Model

#Subset important variable
Data_df <- subset(Africa_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Africa_test_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Africa_mean$Dedt_Crises
y <- Africa_test_mean$Debt_Crises
Test_y <- factor(Africa_test_mean$Debt_Crises)
df <- subset(Africa_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"
```

```
#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify probit model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "probit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))
```

```
#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##World Model

#Subset important variable
Data_df <- subset(Europe_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Europe_test_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Europe_mean$Dedt_Crises
y <- Europe_test_mean$Debt_Crises
Test_y <- factor(Europe_test_mean$Debt_Crises)
df <- subset(Europe_mean,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)
```

```
#Specify probit model
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "probit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
```

```
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)
```

170C__Users_Thendo_Desktop_Thesis_Paper_RCode_Mean_5–fold_probit_test.R

```
#Load required packages
library(readr)
library(ROCR)
library(Metrics)
library(pscl)
library(caret)
library(DescTools)
library(PRROC)

###Case-wise deletion
##World Model

#Subset important variable
Data_df <- subset(Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Test_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Data_set$Dedt_Crises
y <- Test_set$Debt_Crises
Test_y <- factor(Test_set$Debt_Crises)
df <- subset(Data_set,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
```

```
                          classProbs = TRUE)

#Specify probitmodel
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "probit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
```

```
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##South and East Asia Model

#Subset important variable
Data_df <- subset(Asia,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Asia_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Asia$Dedt_Crises
y <- Asia_test$Debt_Crises
Test_y <- factor(Asia_test$Debt_Crises)
df <- subset(Asia,select=c(3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify probitmodel
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
```

```
                    FDI +
                    Real_GDP_growth +
                    Inflation +
                    Reer +
                    Gov_expenditures +
                    National_savings +
                    Domestic_credit +
                    Bank_assets,
                  data=Data_df,
                  method = "glm",
                  family=binomial(link = "probit"),
                  trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##Latin America Model
```

```
#Subset important variable
Data_df <- subset(Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Latin_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Latin_America$Dedt_Crises
y <- Latin_test$Debt_Crises
Test_y <- factor(Latin_test$Debt_Crises)
df <- subset(Latin_America,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify probitmodel
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
```

```
                  method = "glm",
                  family=binomial(link = "probit"),
                  trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##The Middle East and Africa Model

#Subset important variable
Data_df <- subset(Africa,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Africa_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Africa$Dedt_Crises
y <- Africa_test$Debt_Crises
Test_y <- factor(Africa_test$Debt_Crises)
df <- subset(Africa,select=c(3,5,6,7,8,9,10,11,12,13,14,15,16,17,18)
```

```
    )


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"

#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify probitmodel
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "probit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)
```

```
#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)

##World Model

#Subset important variable
Data_df <- subset(Europe,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
Test_df <- subset(Europe_test,select=c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17,18))
y_inner<- Europe$Dedt_Crises
y <- Europe_test$Debt_Crises
Test_y <- factor(Europe_test$Debt_Crises)
df <- subset(Europe,select=c(3,5,6,7,8,9,10,11,12,13,14,15,16,17,18)
   )


#re-label train df and Test_df (1=default, 0= non-default)
Data_df$Dedt_Crises[Data_df$Dedt_Crises==1]<- "Default"
Data_df$Dedt_Crises[Data_df$Dedt_Crises==0]<- "NonDefault"

Test_df$Debt_Crises[Test_df$Debt_Crises==1]<- "Default"
Test_df$Debt_Crises[Test_df$Debt_Crises==0]<- "NonDefault"
```

```
#convert outcome variable to type factor
Data_df$Dedt_Crises <- as.factor(Data_df$Dedt_Crises)
Test_df$Debt_Crises <- as.factor(Test_df$Debt_Crises)

#specify the type of training method used and # of folds
ctrlspecs <- trainControl(method = "cv", number= 5,
                          savePredictions = "all",
                          classProbs = TRUE)

#Specify probitmodel
model1 <- train(Dedt_Crises ~
                  Total_debt +
                  Global_Interest +
                  FX_reserves +
                  Trade_Openess +
                  Export_growth +
                  Current_account +
                  FDI +
                  Real_GDP_growth +
                  Inflation +
                  Reer +
                  Gov_expenditures +
                  National_savings +
                  Domestic_credit +
                  Bank_assets,
                data=Data_df,
                method = "glm",
                family=binomial(link = "probit"),
                trControl= ctrlspecs)
summary(model1)
varImp(model1)

#Apply model
#Predict outcome using model Data_df
prediction <- predict(model1, newdata = Test_df)


#create confusion matrix
confusionMatrix(data = prediction, Test_df$Debt_Crises)
predict_inner <- predict(model1, Data_df)


#Convert prediciton to factor
prob <- (ifelse(prediction=="Default", 1, 0))
```

```
prob_inner <- (ifelse(predict_inner=="Default", 1, 0))

#Predict the in-sample accuracy
accuracy(prob_inner, y_inner)

prob <- factor(prob)

#Generate performance metrics scores
precision(prob,Test_y)
recall(prob,Test_y)
pr.curve(Test_y,prob)
auc(Test_y,prob)
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(y))
BrierScore(Y, prob_brier)
```

171C__Users_Thendo_Desktop_Thesis_Paper_RCode_MICE_5–fold_probit_test.R

E.10 5-Fold Bayesian Additive Regression Trees Classifier

```
#Load required package
library(bartMachine)
library(ROCR)
library(caret)
library(PRROC)
library(Metrics)

###Case-wise deletion

##World Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(CWD_Data_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(CWD_Data_set, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Test_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Data_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Test_set, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
   folds
```

```
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
    folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
    data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
```

```
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

##South and East Asia Model
#Subset data set by selecting required varables
```

```
train_df <-as.data.frame(subset(CWD_Asia, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(CWD_Asia, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Asia_test, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Asia, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Asia_test, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
    folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
    folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
    data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)
```

```
y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
```

```
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)


##World Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(CWD_Latin_America, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(CWD_Latin_America, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Latin_test, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Latin_America, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Latin_test, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
    folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
    folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
    data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
```

```
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")
```

```
single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
      main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)


##The Middle East and Africa Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(CWD_Africa, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(CWD_Africa, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Africa_test, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Africa, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Africa_test, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
    folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
    folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
    data
```

```
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")
```

```
boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
      main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)


##Western Europe Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(CWD_Europe, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(CWD_Europe, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Europe_test, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Europe, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
```

```
alltest_df <- subset(CWD_Europe_test, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
    folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
    folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
    data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)
```

```
par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
```

```
BrierScore(Y, prob_brier)
```

172C__Users_Thendo_Desktop_Thesis_Paper_RCode_CWD_5–fold_BART_test.R

```
#Load required package
library(bartMachine)
library(ROCR)
library(caret)
library(PRROC)
library(Metrics)

###Case-wise deletion

##World Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(Data_set_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(Data_set_mean, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Test_set_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Data_set_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Test_set_mean, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
    folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
    folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
    data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse
```

```
#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
```

```
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

##South and East Asia Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(Asia_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(Asia_mean, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Asia_test_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Asia_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Asia_test_mean, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
    folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
    folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
```

```
    data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")
```

```
boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
      main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)


##World Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(Latin_America_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(Latin_America_mean, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Latin_test_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Latin_America_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
```

```
alltest_df <- subset(Latin_test_mean, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
    folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
    folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
    data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)
```

```
par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)
```

```
##The Middle East and Africa Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(Africa_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(Africa_mean, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Africa_test_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Africa_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Africa_test_mean, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
    folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
    folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
    data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
```

```
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)
```

```
#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)


##Western Europe Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(Europe_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(Europe_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Europe_test_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Europe_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Europe_test_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
   folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
   folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
   data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
```

```
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))
```

```
boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
   Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)
```

173C__Users_Thendo_Desktop_Thesis_Paper_RCode_Mean_5–fold_BART_test.R

```
#Load required package
library(bartMachine)
library(ROCR)
#Load required packages
library(bartMachine)
library(caret)
library(PRROC)
library(Metrics)

###Case-wise deletion

##World Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(Data_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(Data_set, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Test_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
```

```
test_df_inner <- subset(Data_set, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Test_set, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
    folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
    folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
    data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
```

```
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
```

```
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)

##South and East Asia Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(Asia, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(Asia, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Asia_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Asia, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Asia_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
   folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
   folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
   data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
```

```
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
```

```
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)


##World Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(Latin_America, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Latin_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Latin_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
   folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
   folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
   data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)
```

```
#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
```

```
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
   Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)


##The Middle East and Africa Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(Africa, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(Africa, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Africa_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Africa, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Africa_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
   folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
   folds = 5)
```

```
#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
    data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
```

```
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)

prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)


##Western Europe Model
#Subset data set by selecting required varables
train_df <-as.data.frame(subset(Europe, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17)))
alltrain_df <- subset(Europe, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
```

```
test_df <- subset(Europe_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Europe, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Europe_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))

#Build BART cross validation model with the specified number of
   folds
model <- bartMachineCV(X= train_df,y = alltrain_df$Dedt_Crises,k_
   folds = 5)

#Generate model summary
summary(model)

# Generate prediction by using both in and out of sample dependent
   data
y <- predict(model,test_df)
y_r <- predict(model,test_df)
y_inner <- predict(model,test_df_inner)
y_inner_r <-predict(model,test_df_inner)
y <- ifelse(y>0.5,1,0)
y_inner <- ifelse(y_inner>0.5,1,0)

#Generate RMSE scores for the out-of-ample data
bart.rmse <- rmse(y, alltest_df$Debt_Crises)
bart.rmse

#Generate RRMSE scores for the out-of-ample data
bart.rrmse <- rrmse(y, alltest_df$Debt_Crises)
bart.rrmse

pr <- prediction(y, alltest_df$Debt_Crises)
pr_inner <- prediction(y_inner, alltrain_df$Dedt_Crises)

par(mfrow=c(1,1))
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)



y_test <- as.data.frame(y)
y_test_inner <- as.data.frame(y_inner)

#Generate accuracy scores for both in and out of sample
```

```
auc(alltest_df$Debt_Crises,y)
accuracy(alltest_df$Debt_Crises,y)
accuracy( alltrain_df$Dedt_Crises,y_inner)

par(mfrow=c(3,2))

#Generate visualization plots
plot(model$residuals, xlab="X",ylab="Y",main="BART residuals")
plot(model$y_hat_train, xlab="posterior inteval",ylab="in-sample f(x
    ) ",main="BART in-sample y_hat_train")

plot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",main="
    BART out-sample observation")
plot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",main
    ="BART in-sample observation")

boxplot(y_r, xlab="posterior inteval",ylab="out-of-sample f(x) ",
    main="BART out-sample observation")
boxplot(y_inner_r, xlab="posterior inteval",ylab="in-sample f(x) ",
    main="BART in-sample observation")


par(mfrow=c(1,2))
difference <- (y_r-alltest_df$Debt_Crises)^2
single.rmse <- as.data.frame(sqrt(difference))

boxplot(single.rmse,
        main = "Boxplot of RMSE Observations")

single.rmse <- sqrt(difference)
plot(single.rmse,type = "o", col = "blue", xlab = "Index", ylab = "
    Single-RMSE",
     main = "RMSE Observations")
quantile(single.rmse)

prob <- factor(y)
Test_Y1<- factor(alltest_df$Debt_Crises)

#Generate model performance metrics
sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(Test_Y1,prob)
recall(Test_Y1,prob)
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)
```

```
prob_brier<-as.numeric(as.character(prob))
Y<-as.numeric(as.character(Test_Y1))
BrierScore(Y, prob_brier)
```

174C__Users_Thendo_Desktop_Thesis_Paper_RCode_MICE_5–fold_BART_test.R

E.11 5-Fold Artificial Neural Network Classifier

```
#Load required packages
library(Metrics)
library(ROCR)
library(caret)
library(PRROC)
library(plotrix)

###case-wise deletion

##World

#Subset data set by only getting required data sets
train_df <- subset(CWD_Data_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(CWD_Data_set, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Test_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Data_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Test_set, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)
```

```
predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))

##South and East Asia

#Subset data set by only getting required data sets
train_df <- subset(CWD_Asia, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(CWD_Asia, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Asia_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Asia, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Asia_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
```

```
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))
```

```
##Latin America

#Subset data set by only getting required data sets
train_df <- subset(CWD_Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(CWD_Latin_America, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Latin_America, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)
```

```
prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))


##The Middle East and Africa

#Subset data set by only getting required data sets
train_df <- subset(CWD_Africa, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(CWD_Africa, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Africa_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Africa, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Africa_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)
```

```
accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))


##Europe

#Subset data set by only getting required data sets
train_df <- subset(CWD_Europe, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(CWD_Europe, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(CWD_Europe_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(CWD_Europe, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(CWD_Europe_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model
```

```
model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))
```

175C__Users_Thendo_Desktop_Thesis_Paper_RCode_CWD_5–fold_NN_test.R

```
#Load required packages
library(Metrics)
library(ROCR)
library(caret)
library(PRROC)
library(plotrix)
```

```
###case-wise deletion

##World

#Subset data set by only getting required data sets
train_df <- subset(Data_set_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(Data_set_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Test_set_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Data_set_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Test_set_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
```

```
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))

##South and East Asia

#Subset data set by only getting required data sets
train_df <- subset(Asia_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(Asia_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Asia_test_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Asia_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Asia_test_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)
```

```
accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))


##Latin America

#Subset data set by only getting required data sets
train_df <- subset(Latin_America_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(Latin_America_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Latin_America_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Latin_America_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Latin_America_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
```

```
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))


##The Middle East and Africa

#Subset data set by only getting required data sets
train_df <- subset(Africa_mean, select = c
```

```
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(Africa_mean, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Africa_test_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Africa_mean, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Africa_test_mean, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)
```

```
pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))


##Europe

#Subset data set by only getting required data sets
train_df <- subset(Europe_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(Europe_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Europe_test_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Europe_mean, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Europe_test_mean, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)
```

```
prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))
```

176C__Users_Thendo_Desktop_Thesis_Paper_RCode_Mean_5–fold_NN_test.R

```
#Load required packages
library(Metrics)
library(ROCR)
library(caret)
library(PRROC)
library(plotrix)

###case-wise deletion

##World

#Subset data set by only getting required data sets
train_df <- subset(Data_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(Data_set, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Test_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Data_set, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Test_set, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)
```

```
#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))

##South and East Asia
```

```
#Subset data set by only getting required data sets
train_df <- subset(Asia, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(Asia, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Asia_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Asia, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Asia_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
```

```
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))


##Latin America

#Subset data set by only getting required data sets
train_df <- subset(Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(Latin_America, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Latin_America, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Latin_America, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)
```

```
prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))


##The Middle East and Africa

#Subset data set by only getting required data sets
train_df <- subset(Africa, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(Africa, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Africa_test, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Africa, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Africa_test, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
```

```
                trControl= control,
                preProcess=c("scale","center"),
                na.action = na.omit,
                tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)


summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))


##Europe

#Subset data set by only getting required data sets
train_df <- subset(Europe, select = c
   (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltrain_df <- subset(Europe, select = c
   (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df <- subset(Europe_test, select = c
```

```
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
test_df_inner <- subset(Europe, select = c
    (5,6,7,8,9,10,11,12,13,14,15,16,17))
alltest_df <- subset(Europe_test, select = c
    (3,5,6,7,8,9,10,11,12,13,14,15,16,17))
y <- as.factor(alltrain_df$Dedt_Crises)

#Specify the number of fold required in the study
control <- trainControl(method = "repeatedcv", number = 5)
# train the model

model <- train(train_df, y,
               method = "nnet",
               trControl= control,
               preProcess=c("scale","center"),
               na.action = na.omit,
               tuneGrid = data.frame(size=1,decay=0),skip=FALSE)



predict<- predict(model,test_df)
predict_inner <- predict(model,train_df)


accuracy(predict_inner,alltrain_df$Dedt_Crises)
accuracy(predict,alltest_df$Debt_Crises)


prob <- factor(predict)
Test_Y1<- factor(alltest_df$Debt_Crises)
confusionMatrix(prob,Test_Y1)

sensitivity(prob, Test_Y1)
specificity(prob, Test_Y1)
precision(prob,Test_Y1)
recall(prob,Test_Y1)


prob_brier<-as.numeric(as.character(predict))
Y<-as.numeric(as.character(alltest_df$Debt_Crises))
BrierScore(Y, prob_brier)

pr.curve(Test_Y1,prob)
auc(Test_Y1,prob)
```

```
summary(as.numeric(as.character(predict)))
std.error(as.numeric(as.character(predict)))
```

177C__Users_Thendo_Desktop_Thesis_Paper_RCode_MICE_5–fold_NN_test.R

E.12 Additional ANN analysis

```
#Load the required package
library(plotrix)

##World

##find the different metrics mean and standard errors per neurons
   for all regions
mean(World$‘6‘)
std.error(World$‘6‘)
mean(World$‘7‘)
std.error(World$‘7‘)
mean(World$‘8‘)
std.error(World$‘8‘)
mean(World$‘9‘)
std.error(World$‘9‘)
mean(World$‘10‘)
std.error(World$‘10‘)

##Africa
mean(Asia$‘6‘)
std.error(Asia$‘6‘)
mean(Asia$‘7‘)
std.error(Asia$‘7‘)
mean(Asia$‘8‘)
std.error(Asia$‘8‘)
mean(Asia$‘9‘)
std.error(Asia$‘9‘)
mean(Asia$‘10‘)
std.error(Asia$‘10‘)

##Africa
mean(Latin_America$‘6‘)
std.error(Latin_America$‘6‘)
mean(Latin_America$‘7‘)
std.error(Latin_America$‘7‘)
```

```
mean(Latin_America$`8`)
std.error(Latin_America$`8`)
mean(Latin_America$`9`)
std.error(Latin_America$`9`)
mean(Latin_America$`10`)
std.error(Latin_America$`10`)

##Africa
mean(Africa$`6`)
std.error(Africa$`6`)
mean(Africa$`7`)
std.error(Africa$`7`)
mean(Africa$`8`)
std.error(Africa$`8`)
mean(Africa$`9`)
std.error(Africa$`9`)
mean(Africa$`10`)
std.error(Africa$`10`)

##Europe
mean(Europe$`6`)
std.error(Europe$`6`)
mean(Europe$`7`)
std.error(Europe$`7`)
mean(Europe$`8`)
std.error(Europe$`8`)
mean(Europe$`9`)
std.error(Europe$`9`)
mean(Europe$`10`)
std.error(Europe$`10`)

#Plotting the mean and standard error line graphs for the different
    metrics
ggplot(StatP, aes(x = Neurons, y = `Accuracy`, colour=Region))+
  geom_errorbar(aes(ymin=`Accuracy`-SE, ymax=`Accuracy`+SE), width
     =.1) +
  geom_line() +
  geom_point()

ggplot(StatP, aes(x = Neurons, y = `Sensitivity`, colour=Region))+
  geom_errorbar(aes(ymin=`SSensitivity`-SE, ymax=`Sensitivity`+SE),
     width=.1) +
  geom_line() +
  geom_point()
```

```
ggplot(StatP, aes(x = Neurons, y = `Specificity`, colour=Region))+
  geom_errorbar(aes(ymin=`Specificity`-SE, ymax=`Specificity`+SE),
     width=.1) +
  geom_line() +
  geom_point()

ggplot(StatP, aes(x = Neurons, y = `ROC AUC`, colour=Region))+
  geom_errorbar(aes(ymin=`ROC AUC`-SE, ymax=`ROC AUC`+SE), width=.1)
      +
  geom_line() +
  geom_point()
```

178C__Users_Thendo_Desktop_Thesis_Paper_RCode_ANN_plot.R