BUSINESS REPORT- PREDICTING DEFAULT RISK

Section 1: Business Understanding

Business Situation
Our bank receives 200 loan applications per week, but due to a financial scandal that hit a competitor the credit risk unit of the bank will be processing 500 applications this week.The influx of new credit applications is a great opportunity the bank wants to immediately pursue.

The Complication
The bank will want to maintain there processing turnaround time while ensuring that the credit risk unit is able to effectively determine creditworthy applications, while reducing the risk of default by effectively determining non-creditworthy applications.

Key Decision that needs to be made
The Head of the credit risk department needs to decide if a loan should be approved for each of the 500 loan applications received this week.

Approach
This project is data rich; it has readily available information that can be used to predict creditworthiness of the 500 loan applications. The data will be acquired internally from already processed loan applications,'customers-to-score' and the data from the 500 loan applications yet to be reviewed,'customers-to-score'. The two sets of data include personal details about the customer, such as their age and how long they have been at their current job. It will also include details on the individual's banking and credit history, such as their account balance, number of credits at this bank, and their payment status of previous credit.

We will use the data set with already processed loan application to build a binary classification predictive model from 4 different algorithms to determine if a customer is creditworthy or non-creditworthy. We will then make some comparisons with some performance metric to determine the algorithm that will be used to predict the Creditworthiness of the 500 new loan applications.

Section 2: Data Structure & Quality

The data we used to train the model was an equivalent sum of 500 loan applications with 19 variables that includes the outcome variable. The 19 variables included in the data set are listed as follows:

```
##  [1] "Credit.Application.Result"      "Account.Balance"
##  [3] "Duration.of.Credit.Month"       "Payment.Status.of.Previous.Credit"
##  [5] "Purpose"                        "Credit.Amount"
##  [7] "Value.Savings.Stocks"           "Length.of.current.employment"
##  [9] "Instalment.per.cent"            "Guarantors"
## [11] "Duration.in.Current.address"    "Most.valuable.available.asset"
## [13] "Age.years"                      "Concurrent.Credits"
## [15] "Type.of.apartment"              "No.of.Credits.at.this.Bank"
## [17] "Occupation"                     "No.of.dependents"
## [19] "Telephone"                      "Foreign.Worker"
```

Table 2.1 List of variables

We checked the data structure and quality to check for missing values.

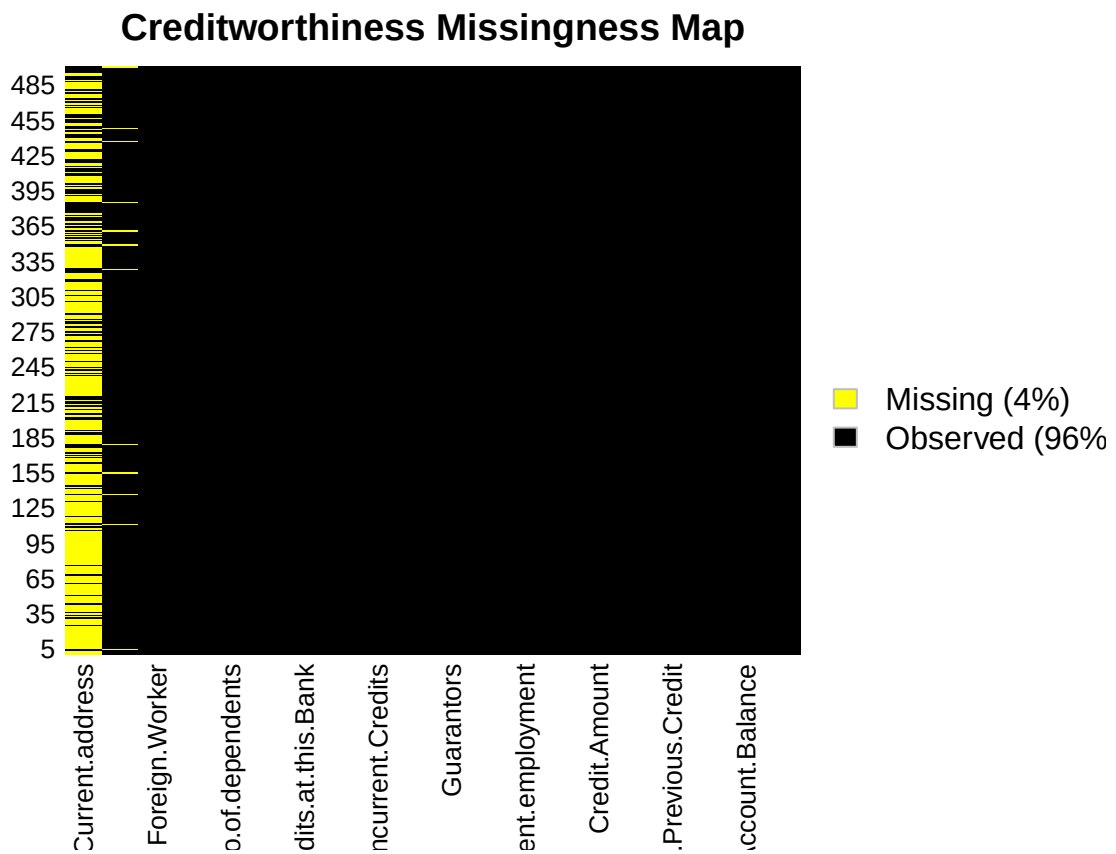


Fig 2.1 Missing Values

	NAs
Credit.Application.Result	0
Account.Balance	0
Duration.of.Credit.Month	0
Payment.Status.of.Previous.Credit	0
Purpose	0
Credit.Amount	0
Value.Savings.Stocks	0
Length.of.current.employment	0
Instalment.per.cent	0
Guarantors	0
Duration.in.Current.address	344
Most.valuable.available.asset	0
Age.years	12
Concurrent.Credits	0
Type.of.apartment	0
No.of.Credits.at.this.Bank	0
Occupation	0
No.of.dependents	0
Telephone	0
Foreign.Worker	0

Table 2.2 NA's per variable

The missing values plot showed that the variables *Duration.in.Current.address* had more than 50% of its

values missing and *Age.years* had less than 5% missing values. We dropped *Duration.in.Current.address* and imputed the missing *Age.years* with the median value.Character variables were also encoded as factors.

Section 3: Exploratory Data Analysis

Next we performed some exploratory data analysis to generate some insights from our internal data. We started with some distributions of categorical variables to get a sense of variables that have zero variance or near zro variance that can affect our modeling.

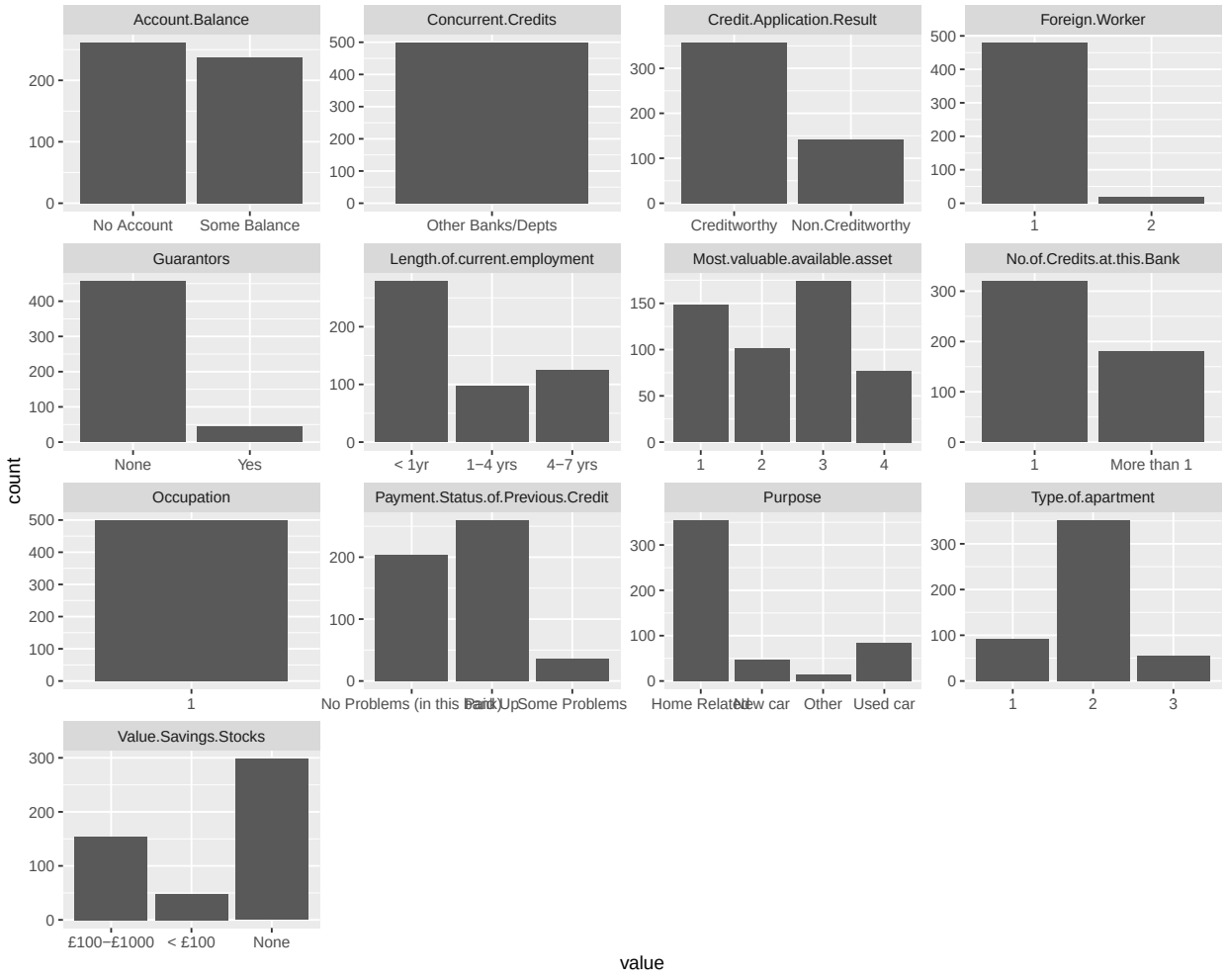


Fig_3.1 Bar charts of categorical variables

We can identify immediately 3 variables, *Concurrent.Credits*, *Foreign.Worker*, *Guarantors* and *Occupation*. To be sure of this we calculate the frequency ratio of the most occurrence over the second most occurrence value within the variables and also the percent unique to check their validity.

```
##                    freqRatio percentUnique zeroVar  nzv
## Concurrent.Credits   0.00000           0.2    TRUE TRUE
## Occupation           0.00000           0.2    TRUE TRUE
## Foreign.Worker      25.31579           0.4   FALSE TRUE
```

Table 3.1 Near Zero Variance predictor variables

We discovered that only three variables meet this criteria as shown in Table 3.1 above and these variables were dropped. We continue with our analysis by plotting the distribution of the continuous variables.

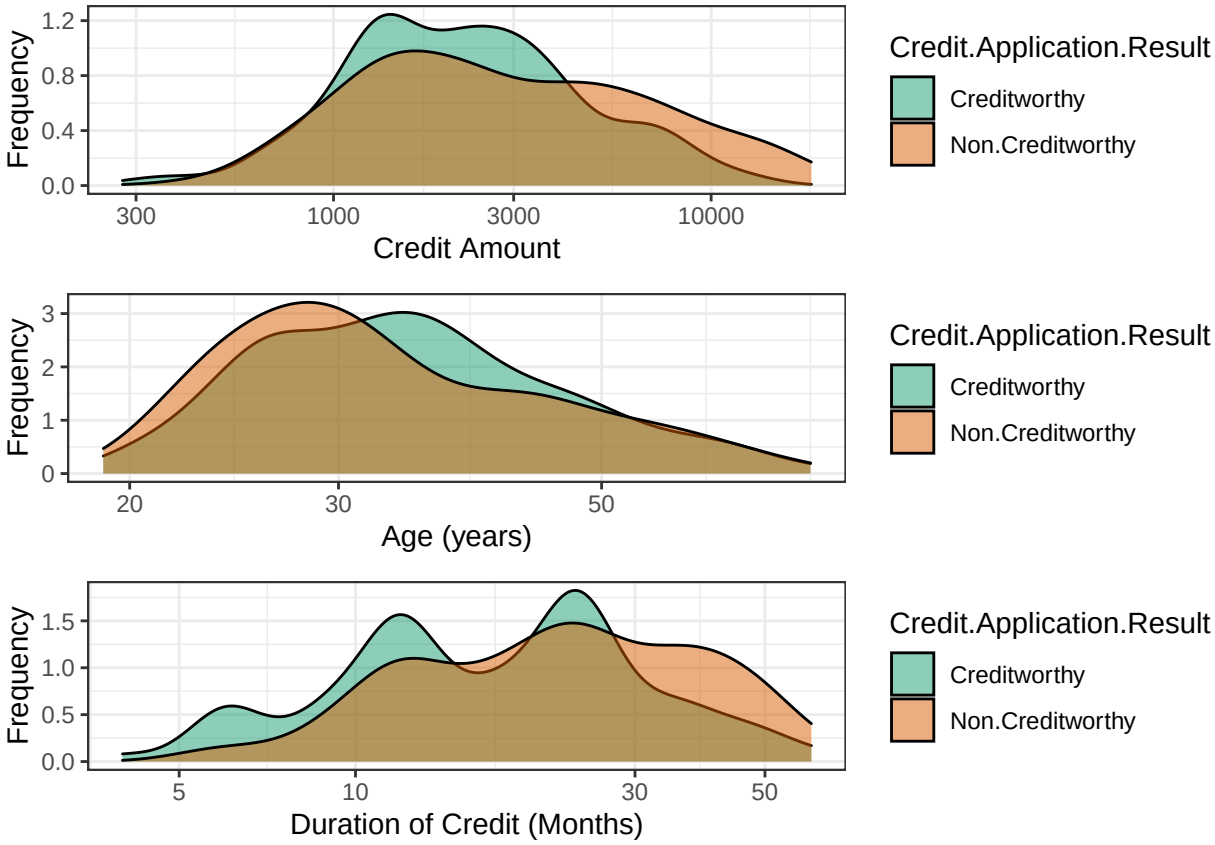


Fig 3.2 Density Plots

We can see that they are not normally distributed and it seems as if the central tendencies for each category in the three plots are different. We also see that the categories follow the same skew direction. The defined humps in Credit Amount and Duration of Credit were there are 3 suggests that there are about 2-3 defined groups in the loan applicants which will be verified with categorical plots subsequently. We also sense that these be strong predictors and this will be further verified with boxplots to visualize the difference in their means.

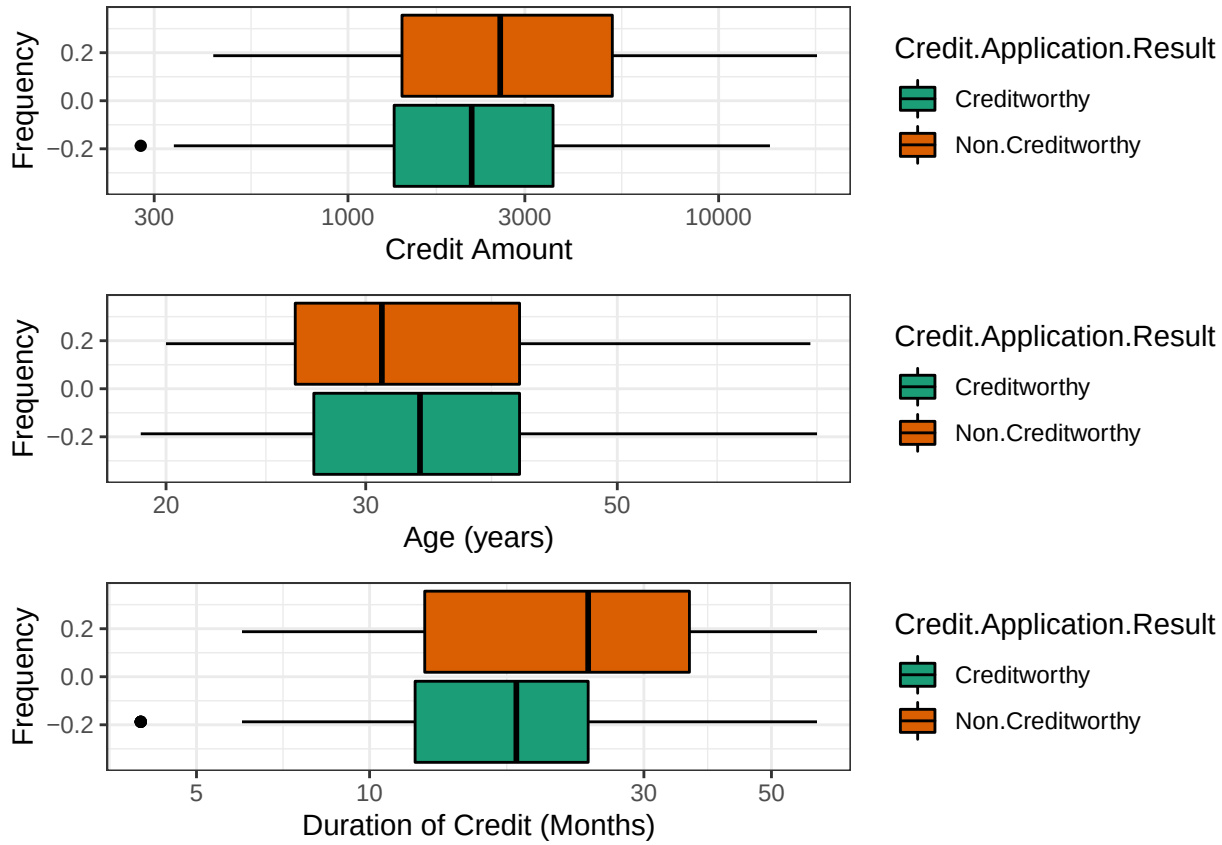


Fig 3.3 Box Plots

We discover that the difference is means for the two categories are evident for the 3 continuous predictor variables and Duration of Credit is the strongest of the three. This was tested using the t-test and their strength measured by the p-values is shown in the table below.

```
## # A tibble: 3 x 2
## # Groups:   var [3]
##   var                          p.value
##   <chr>                          <dbl>
## 1 Age.years                   0.247
## 2 Credit.Amount               0.000260
## 3 Duration.of.Credit.Month 0.0000207
```

Table 3.2 Strength of continuous predictor variables

Here we look for insights on some of the categorical variables by plotting some bar graphs.

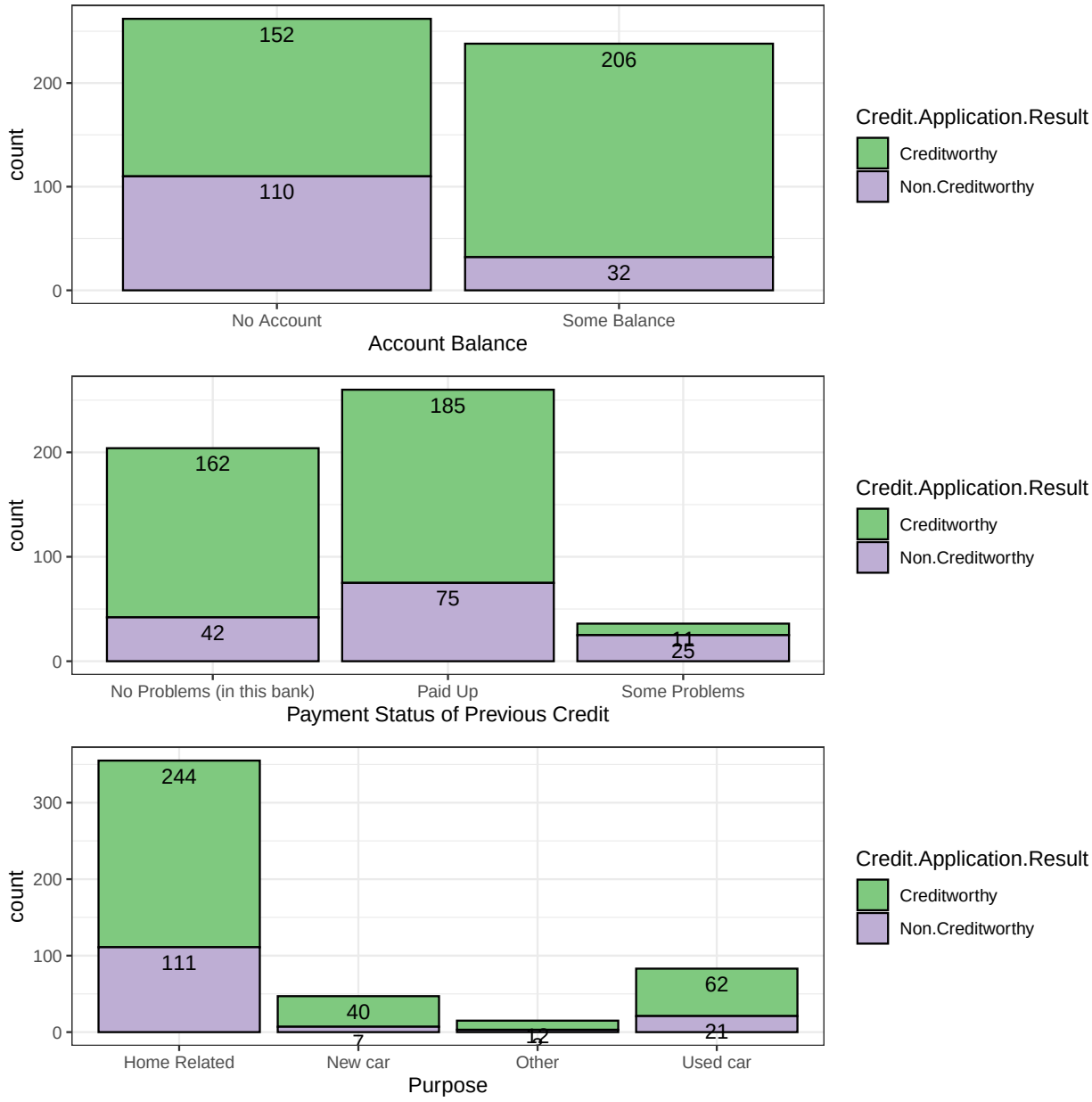


Fig 3.4 Barplots group 1

Some observations from Fig 3.4

Account Balance: there are more Creditworthy people with *Some Balance* than *No Account* and consequently less people Non.Creditworthy people with *Some Balance* than *No Account* . It suggests that having some amount in your account may determine creditworthiness.

Payment Status of previous Credit:This chart suggests that while there may be 25 accounts with *Some Problems* that are classified as Non.Creditworthy, having probably defaulted, there were 75 accounts that are now *Paid Up* but classified as Non.Creditworthy. The predictive model should be able to reduce this risk with an acceptable mis-classification rate.

Purpose: The bank has a bigger appetite for *Home Related* loans.

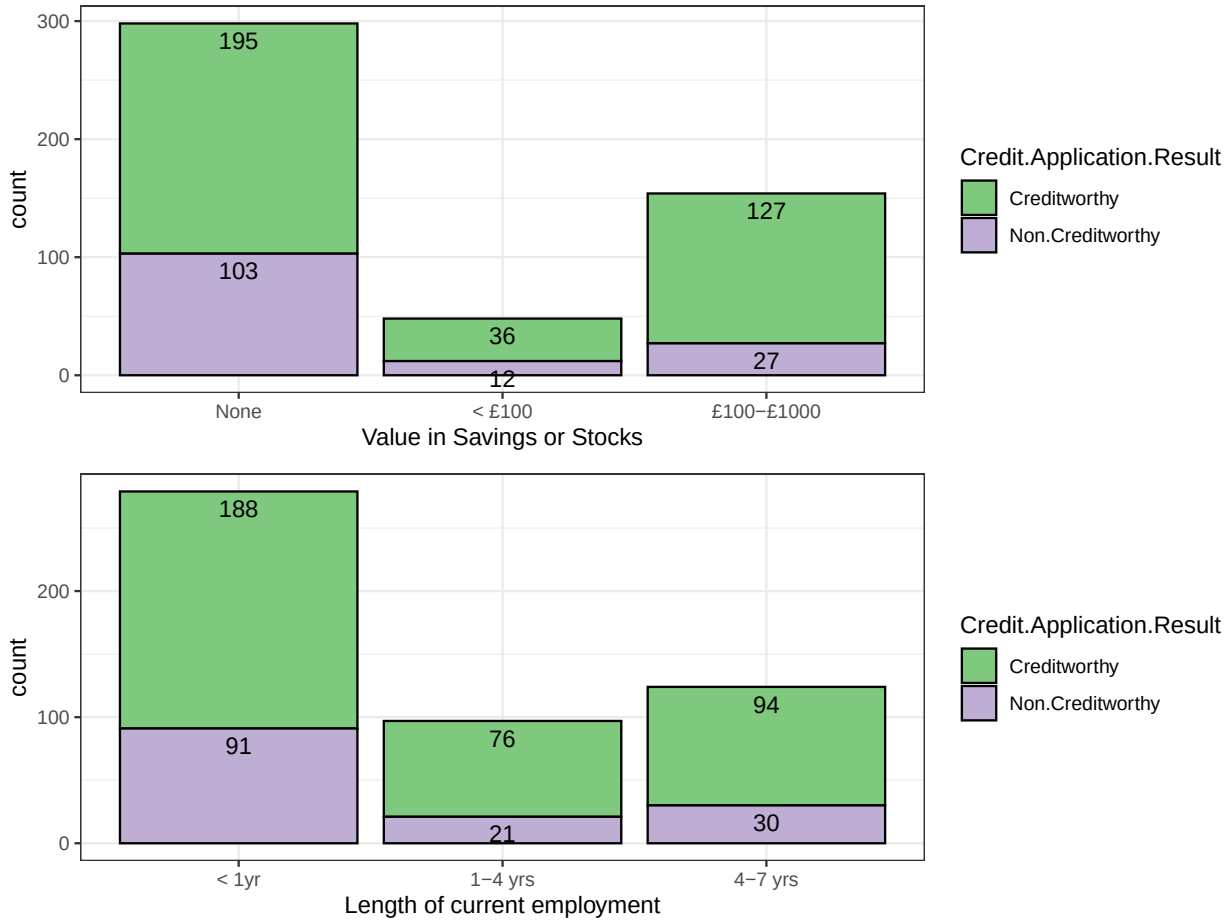


Fig 3.5 Bar plots group2

This group of charts seem to follow the same trend. It shows that people with less than a year in current may not have attained an acceptable credit score and without any savings further supports the suggestion that these may be new employees or those just starting out in their careers.

Section 4: Training the Model

We used 4 different algorithms to train the binary classification model and compared their ROC and Accuracy performance of metrics to determine the best performance for scoring the 500 new loan applications.The algorithms we used are:

1.Logistic Regression
2.Decision Tree
3.Random Forest
4.Boosted Tree

We did a train/test split of 70/30 for our external validation and set the hyper parameters to tune the different algorithms using repeated cross validation of 10 for to determine the accuracy of the models and class probability with two class summary to measure the ROC of the models.

1. Logistic Regression

The model gave an accuracy of 73% with a 10-fold repeated cross validation as can be see in the model output below.

```
## Generalized Linear Model
##
## 351 samples
##  15 predictor
##   2 classes: 'Creditworthy', 'Non.Creditworthy'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 316, 316, 316, 316, 316, 316, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7329683  0.2616023
```

Table 4.1 Logistic Regression model Accuracy measure output

We measured the performance of the Logistic Regression using the ROC metric with the output of 74.5% as seen below.

```
## Generalized Linear Model
##
## 351 samples
##  15 predictor
##   2 classes: 'Creditworthy', 'Non.Creditworthy'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 316, 316, 316, 316, 316, 316, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.7159923  0.8836462  0.347
```

Table 4.1b Logistic Regression with ROC measure output

Logistic Regression test prediction and confusion matrix

We applied the model to the hold-out sample and got an overall of 79.2% as seen in the Confusion Matrix and its statistics output below. This was an improvement on the accuracy of the model. However, the calculated Positive Predictive Value (PPV) of 81.5% and Negative Predictive Value (NPV) of 70.37% showed that the model is biased towards Creditworthy.

```
## Confusion Matrix and Statistics
##
##
## glm.pred           Creditworthy Non.Creditworthy
##   Creditworthy               99               23
##   Non.Creditworthy            8               19
##
##                Accuracy : 0.7919
##                  95% CI : (0.7179, 0.854)
##     No Information Rate : 0.7181
##     P-Value [Acc > NIR] : 0.02538
##
##                   Kappa : 0.4236
```

```
##
##  Mcnemar's Test P-Value : 0.01192
##
##             Sensitivity : 0.9252
##             Specificity : 0.4524
##          Pos Pred Value : 0.8115
##          Neg Pred Value : 0.7037
##              Prevalence : 0.7181
##          Detection Rate : 0.6644
##    Detection Prevalence : 0.8188
##       Balanced Accuracy : 0.6888
##
##        'Positive' Class : Creditworthy
##
```

Table 4.2 Confusion Matrix and Statistics output

2. Decision Tree

We used the Decision Tree algorithm to train the model and got an accuracy of 73.27% at a complexity parameter of 0.0375. This can be seen in the line plot of the tuning parameter in Fig 4.2 below.

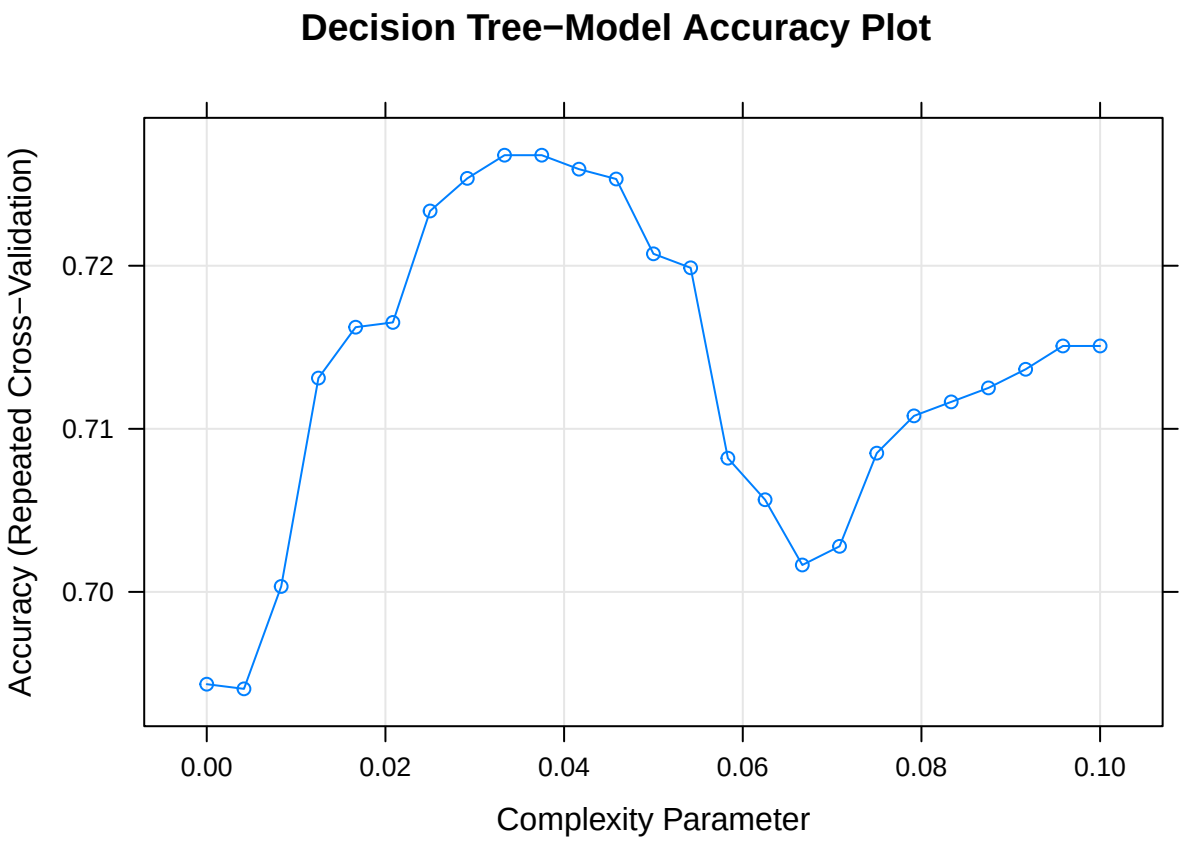


Fig 4.2 Decision Tree complexity parameter tuning plot

Decision Tree variable importance

We can also see from the variable importance plot that *Payment.Status.of Previous.Credit* is the most important followed by *Duration.Credit.Month* and *Account.Balance* compared to the Logistic Regression model that showed *Account.Balance* as the most significant predictor.

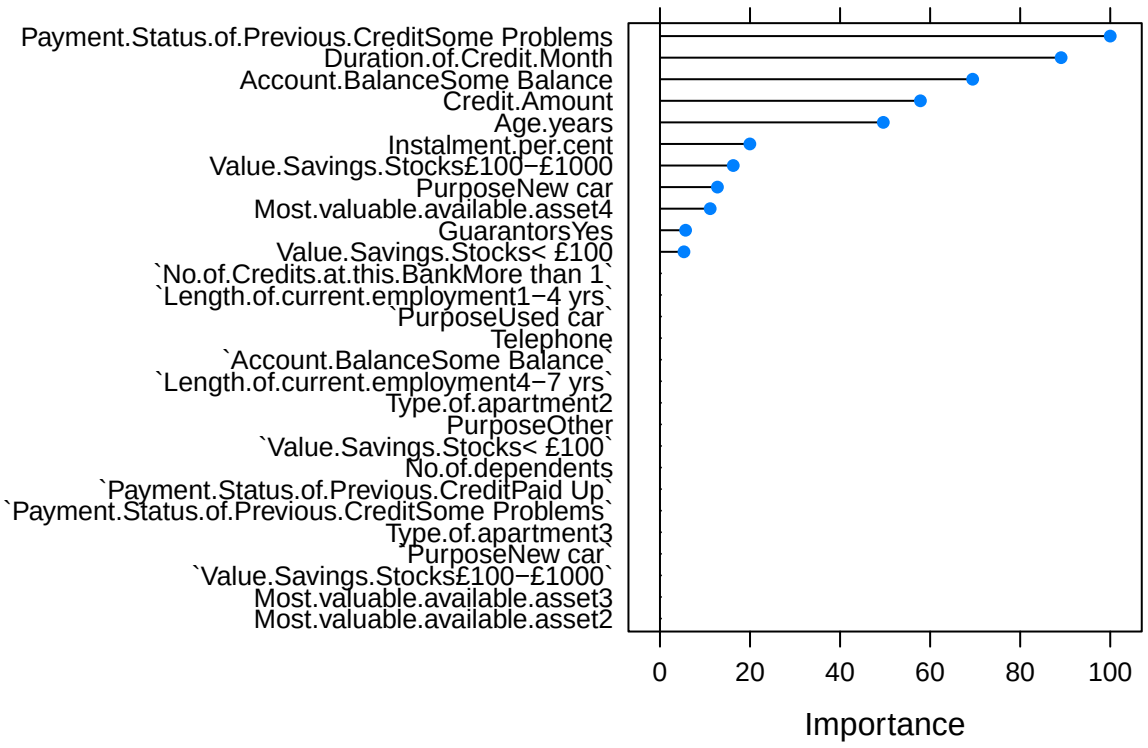


Fig 4.3 Decision Tree Variable importance plot

Decision Tree model with ROC as metric

Next we tuned the Decision Tree model using the ROC metric. This gave us the optimal ROC of 69.39% at a complexity parameter value of 0.03333333. See the ROC Vs CP plot in Fig 4.4 below.

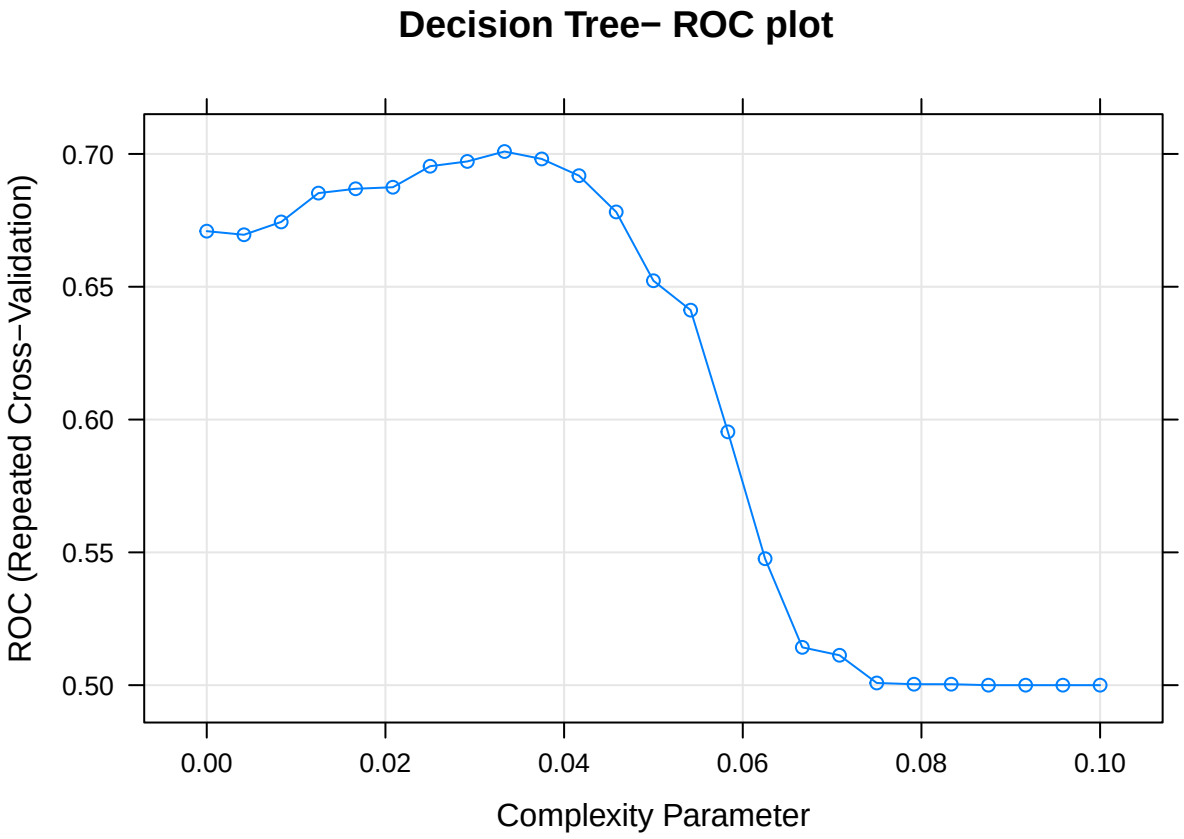


Fig 4.4 Decision Tree ROC plot

Decision Tree test prediction and confusion matrix

Applying the model with accuracy as metric of choice to the hold-out sample gave an overall accuracy of 78.52%. This mode did not perform better than the Logistic Regression.Similar to the Logistic Regression, it is biased towards the Creditworthy category from the values of the PPV and NPV.

```
## Confusion Matrix and Statistics
##
##
## tree.pred          Creditworthy Non.Creditworthy
##   Creditworthy              100               25
##   Non.Creditworthy            7               17
##
##                Accuracy : 0.7852
##                  95% CI : (0.7106, 0.8482)
##     No Information Rate : 0.7181
##     P-Value [Acc > NIR] : 0.039154
##
##                   Kappa : 0.3901
##
##  Mcnemar's Test P-Value : 0.002654
##
##             Sensitivity : 0.9346
##             Specificity : 0.4048
##          Pos Pred Value : 0.8000
##          Neg Pred Value : 0.7083
```

```
##              Prevalence : 0.7181
##          Detection Rate : 0.6711
##    Detection Prevalence : 0.8389
##       Balanced Accuracy : 0.6697
##
##        'Positive' Class : Creditworthy
##
```

Table 4.3 Decision Tree Confusion Matrix Output

3. Random Forest

The Random Forest model showed better accuracy than the first two models. The model gave an accuracy of 75.82% with 4 as the optimal no of randomly selected predictors. This can also be seen in the Accuracy Plot show in Fig 4.5

```
## Random Forest
##
## 351 samples
##  15 predictor
##   2 classes: 'Creditworthy', 'Non.Creditworthy'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 316, 316, 316, 316, 316, 316, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.7367460  0.1577397
##    4    0.7547143  0.2758024
##    6    0.7529921  0.2841400
##    9    0.7470238  0.2789810
##   11    0.7435952  0.2771340
##   13    0.7410159  0.2746097
##   16    0.7415952  0.2776069
##   18    0.7372937  0.2723203
##   20    0.7410079  0.2832446
##   23    0.7375873  0.2772160
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```

Table 4.4 Random Forest Model (metric=Accuracy)

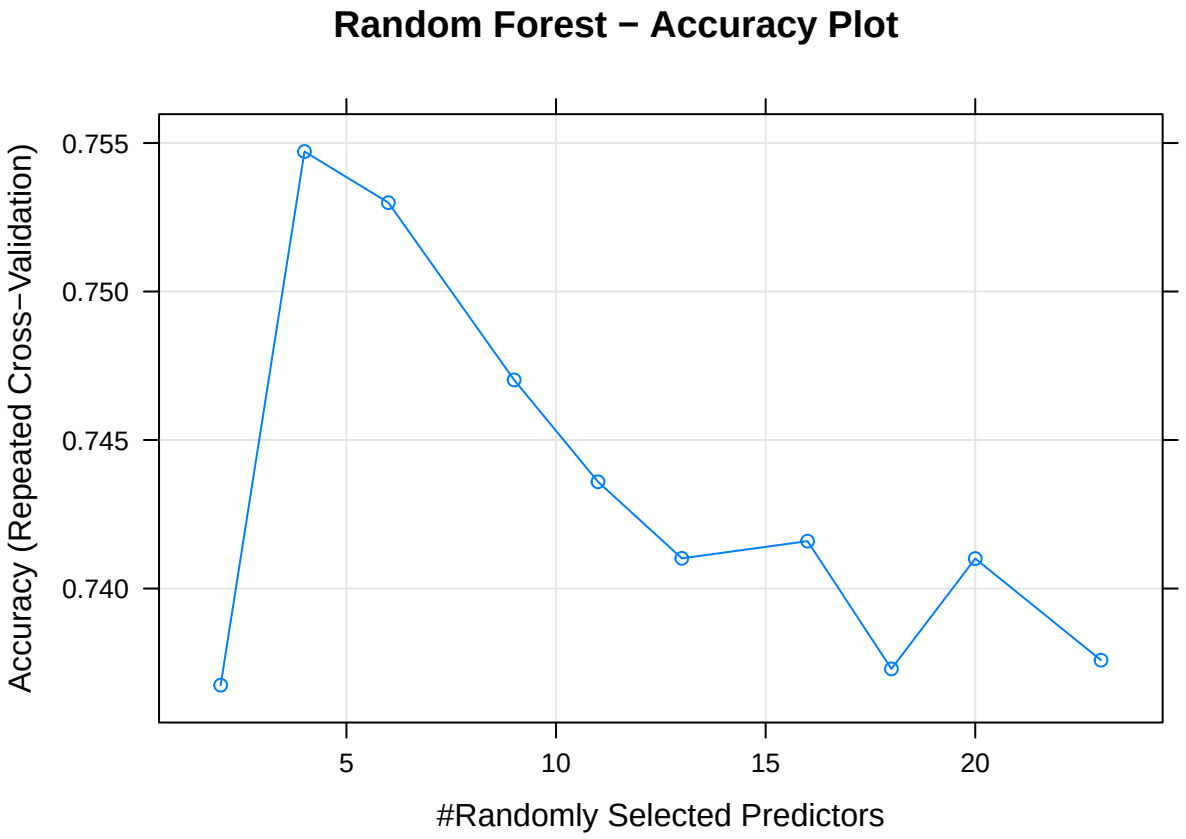


Fig 4.5 Random Forest- Accuracy Plot

Random Forest variable importance

Unlike the Decision Tree that showed the first 3 variables of importance as *Payment.Status.of Previous.Credit*, *Duration.of.Credit.Month*,and *Account Balance*; the Random Forest Model showed *Account.Balance*, *Payment.Status.of Previous.Credit* and *Duration.of.Credit.Month* as the most significant predictors.

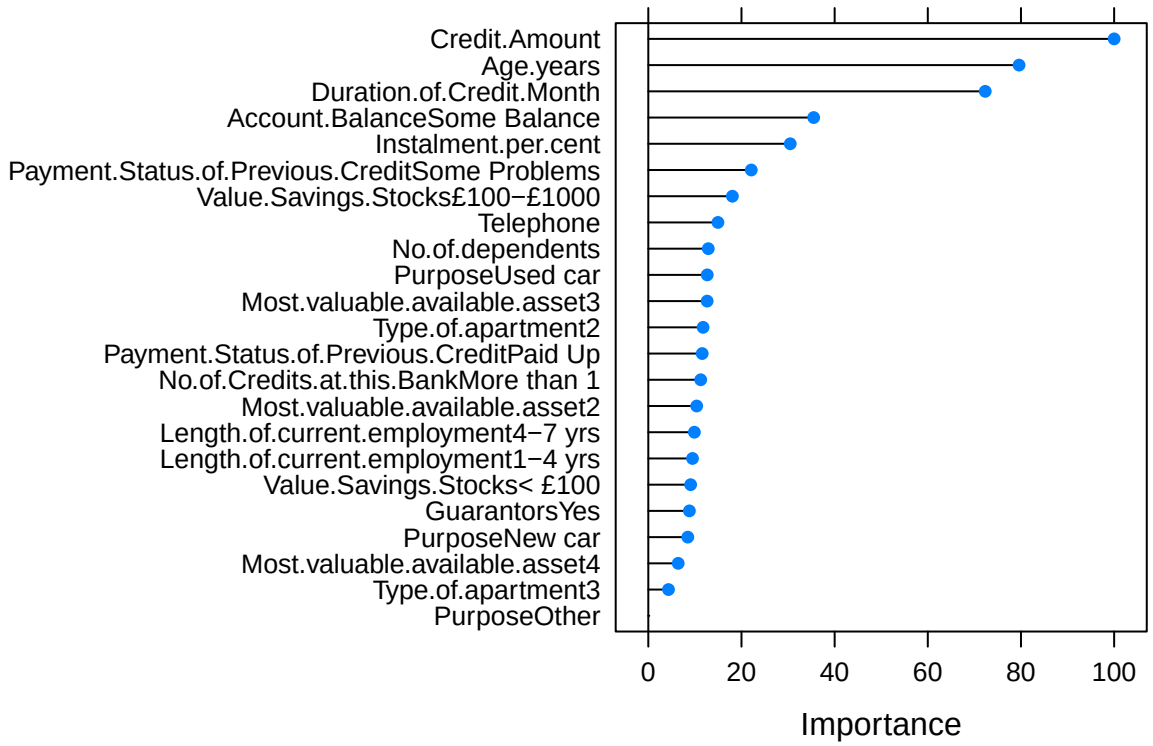


Fig 4.6 Random Forest Variable Importance Plot

Random Forest with ROC as metric

We used the ROC metric to select the optimal model for the Random Forest algorithm. This gave us the optimal ROC of 74.91% with 2 as the value of randomly selected predictors. We can see the plot of values for the ROC against 10 different randomly selected predictors in Fig 4.6 below.

```
## Random Forest
##
## 351 samples
##  15 predictor
##   2 classes: 'Creditworthy', 'Non.Creditworthy'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 316, 315, 316, 316, 316, 316, ...
## Resampling results across tuning parameters:
##
##   mtry  ROC        Sens       Spec
##    2    0.7532823  0.9685077  0.164
##    4    0.7516346  0.9394000  0.313
##    6    0.7452615  0.9254769  0.330
##    9    0.7404546  0.9087077  0.342
##   11    0.7339462  0.9011385  0.352
##   13    0.7336600  0.8948462  0.358
##   16    0.7293992  0.8832923  0.366
##   18    0.7285500  0.8857077  0.374
```

```
##    20    0.7286223  0.8817385  0.371
##    23    0.7278131  0.8805231  0.371
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

Table 4.5 Random Forest Model output (metric = Accuracy)

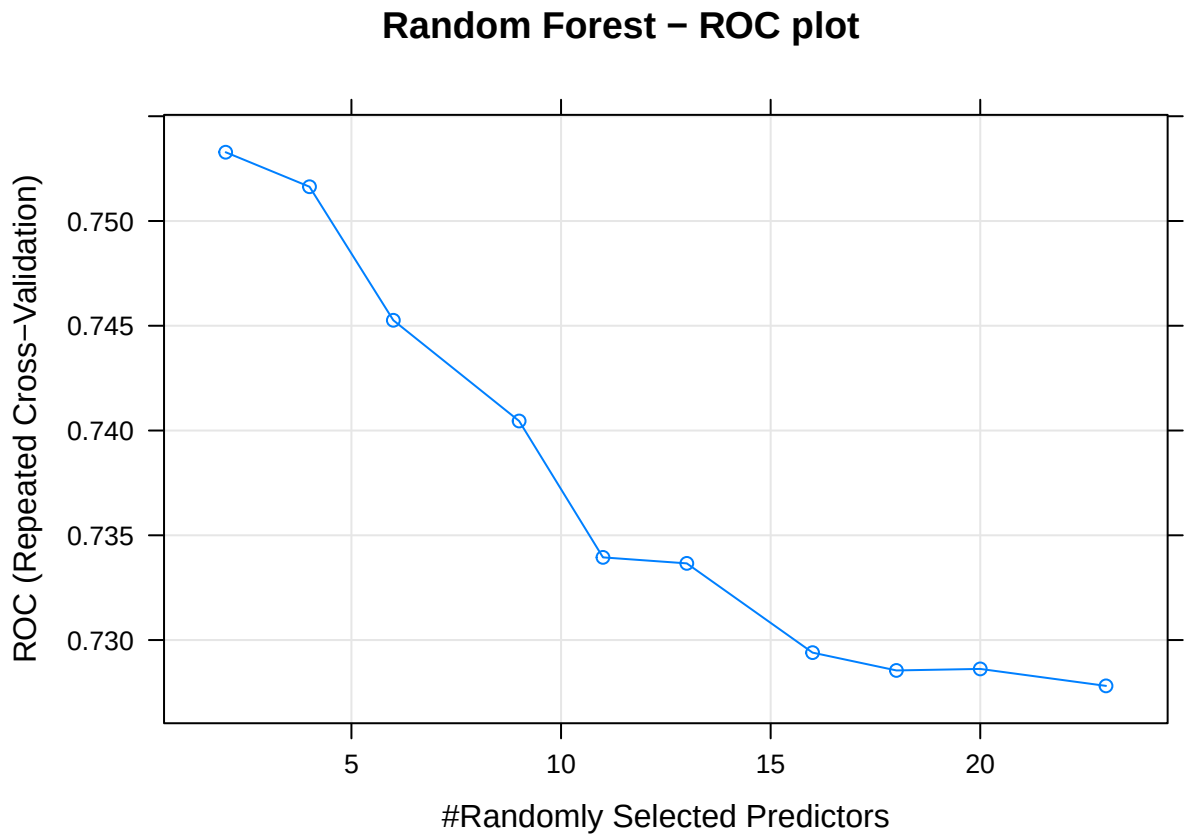


Fig 4.7 Random Forest - ROC plot

Random Forest Test set prediction and confusion matrix Next we applied the model with accuracy as metric of choice to the hold-out sample. This gave an overall accuracy of 78.52%. This model performed at par with the Decision Tree which also had an accuracy of 78.52%, but did not perform better than the Logistic Regression.However, we this model had a higher percentage for the NPV with a lesser difference from the PPV. This is particularly important as the bank needs to ensure they do not grant loans to Non-creditworthy customers, while ensuring that they adequately harness the opportunity not wrongly classifying Creditworthy customers in the process.Based on this we can conclude that there is no bias towards any of the classes.

```
## Confusion Matrix and Statistics
##
##
## rf.pred            Creditworthy Non.Creditworthy
##   Creditworthy              103               27
##   Non.Creditworthy            4               15
##
##                Accuracy : 0.7919
```

```
##                  95% CI : (0.7179, 0.854)
##     No Information Rate : 0.7181
##     P-Value [Acc > NIR] : 0.02538
##
##                   Kappa : 0.3836
##
##  Mcnemar's Test P-Value : 7.772e-05
##
##             Sensitivity : 0.9626
##             Specificity : 0.3571
##          Pos Pred Value : 0.7923
##          Neg Pred Value : 0.7895
##              Prevalence : 0.7181
##          Detection Rate : 0.6913
##    Detection Prevalence : 0.8725
##       Balanced Accuracy : 0.6599
##
##        'Positive' Class : Creditworthy
##
```

Table 4.6 Random Forest Confusion Matrix

4. Boosted Tree Model

This optimal model had an accuracy of 74.04.% with an n.trees of 134. See the details for the optimal parameter in table 4.7 below. The Accuracy plot is also seen in Fig 4.8.

```
##     n.trees interaction.depth shrinkage n.minobsinnode
## 105     134                 1       0.1             15
```

Table 4.7 Boosted Model optimal performance parameters

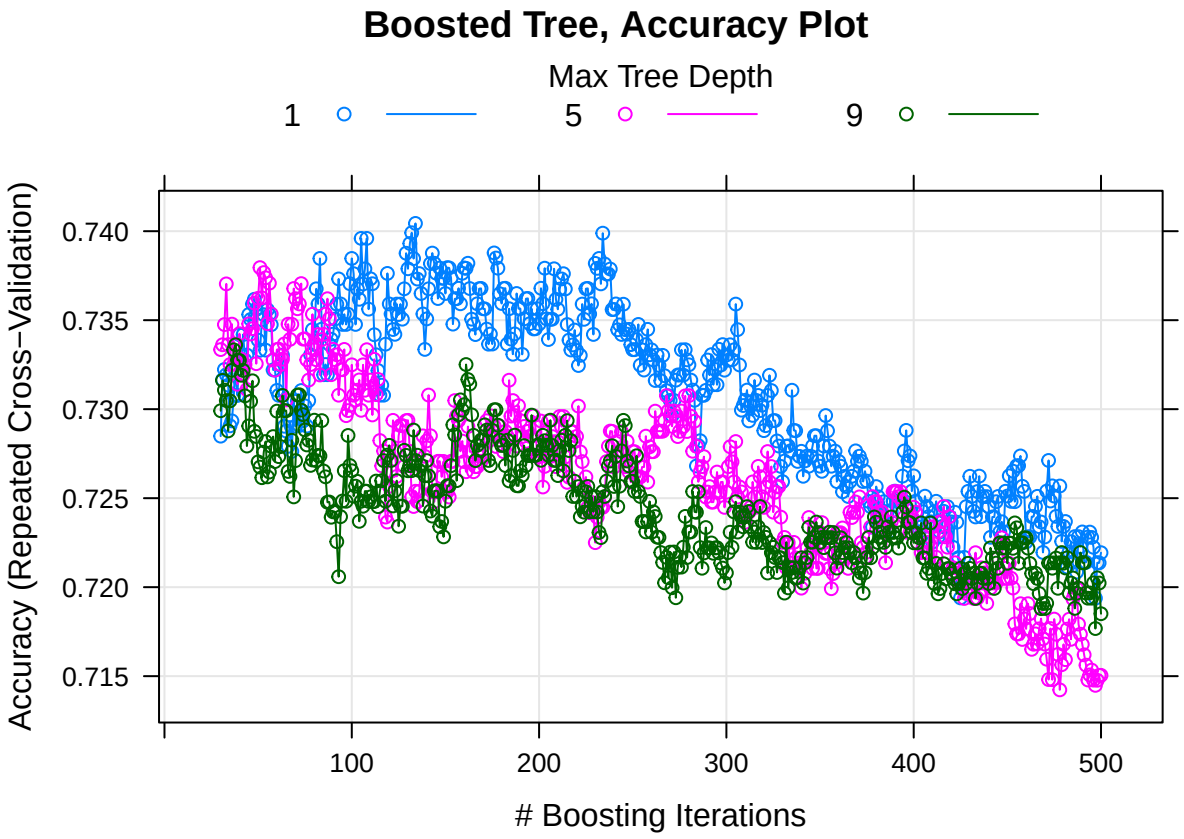


_Fig 4.8 Boosted Model Accuracy Plot

Boosted Model variable importance

This model introduced ranked the top three variables by importance as *Credit.Amount* , *Account.Balance* , and *Age.years*. This is a bit of a depature from the Decision Tree and Random Forest that had the same variables but in different order.

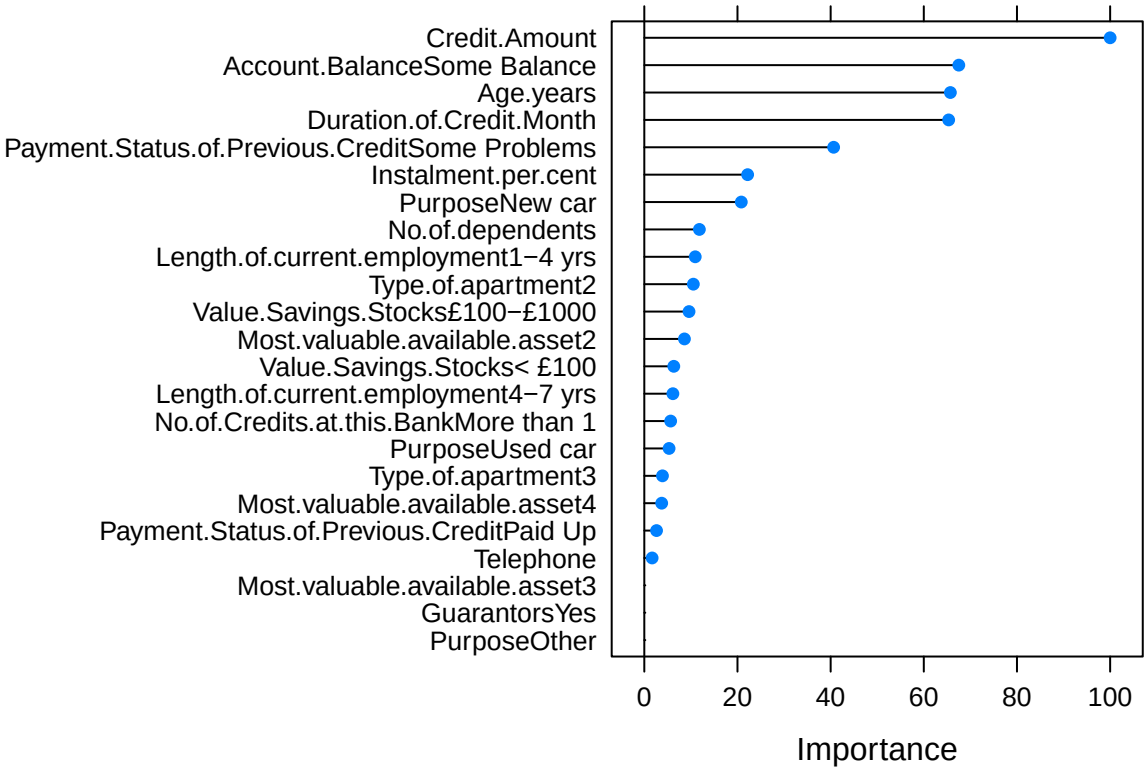


Fig 4.9 Boosted Model Variable Importance plot

Boosted Model with ROC as metric The ROC metric gave us an optimal performance with an ROC of 72% with n.tree value of 94. See

```
##    n.trees interaction.depth shrinkage n.minobsinnode
## 65      94                 1       0.1             15
```

Table 4.8 Boosted Model ROC optimal performance parameters

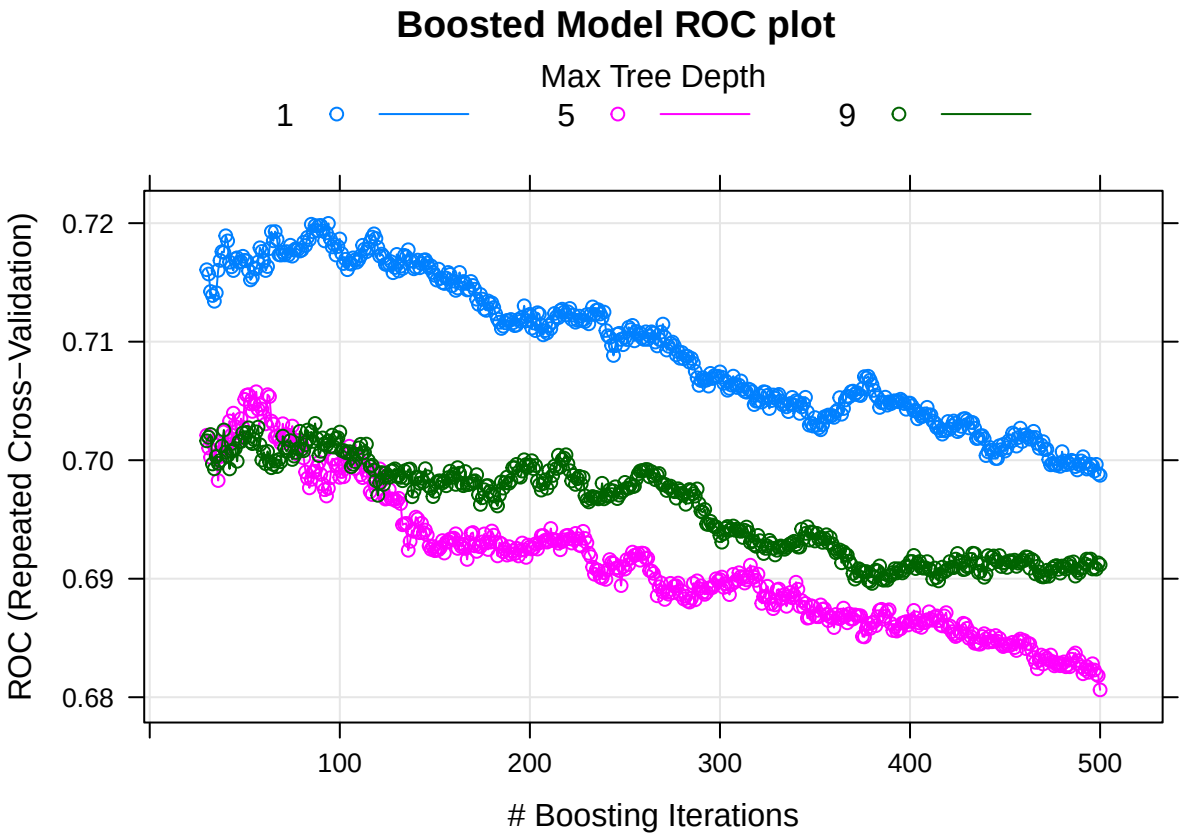


Fig 4.9 Boosted model ROC plot

Boosted Model test prediction and confusion matrix

The boosted model had an overall accuracy of 77.18% and the calculated values for the PPV and NPV revealed that it is biased towards the Creditworthy class.

```
## Confusion Matrix and Statistics
##
##
## gbm.pred           Creditworthy Non.Creditworthy
##   Creditworthy              101               28
##   Non.Creditworthy            6               14
##
##                Accuracy : 0.7718
##                  95% CI : (0.696, 0.8365)
##     No Information Rate : 0.7181
##     P-Value [Acc > NIR] : 0.0839845
##
##                   Kappa : 0.3297
##
##  Mcnemar's Test P-Value : 0.0003164
##
##             Sensitivity : 0.9439
##             Specificity : 0.3333
##          Pos Pred Value : 0.7829
##          Neg Pred Value : 0.7000
##              Prevalence : 0.7181
```

```
##          Detection Rate : 0.6779
##    Detection Prevalence : 0.8658
##       Balanced Accuracy : 0.6386
##
##        'Positive' Class : Creditworthy
##
```

Table 4.9 Boosted model confusion matrix output

Section 5: Model Performance Metrics Evaluation

Before we chose the model to use for prediction, we evaluated the performance of the 4 algorithms. We did a resampling of all the models, summarizing the results and plotting boxplots to visualize their performance metrics.

```
##
## Call:
## summary.resamples(object = fit.compare)
##
## Models: LogisticRegression, Tree, Forest, Boosted
## Number of resamples: 100
##
## Accuracy
##                          Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## LogisticRegression 0.6000000 0.6857143 0.7428571 0.7329683 0.7714286 0.8888889
## Tree               0.5714286 0.7142857 0.7428571 0.7267778 0.7714286 0.8571429
## Forest             0.6111111 0.7142857 0.7428571 0.7547143 0.7833333 0.8571429
## Boosted            0.6000000 0.6857143 0.7428571 0.7404365 0.7777778 0.8857143
##                    NA's
## LogisticRegression    0
## Tree                  0
## Forest                0
## Boosted               0
##
## Kappa
##                            Min.   1st Qu.    Median      Mean   3rd Qu.
## LogisticRegression -0.07058824 0.1463415 0.2588235 0.2616023 0.3676640
## Tree               -0.18867925 0.1538462 0.2588235 0.2447017 0.3537957
## Forest             -0.10526316 0.1592243 0.2631579 0.2758024 0.3683155
## Boosted            -0.19512195 0.1463415 0.3038462 0.2653916 0.3797468
##                         Max. NA's
## LogisticRegression 0.6842105    0
## Tree               0.6391753    0
## Forest             0.5882353    0
## Boosted            0.7021277    0
```

Table 5.1 Accuracy metric model evaluation output

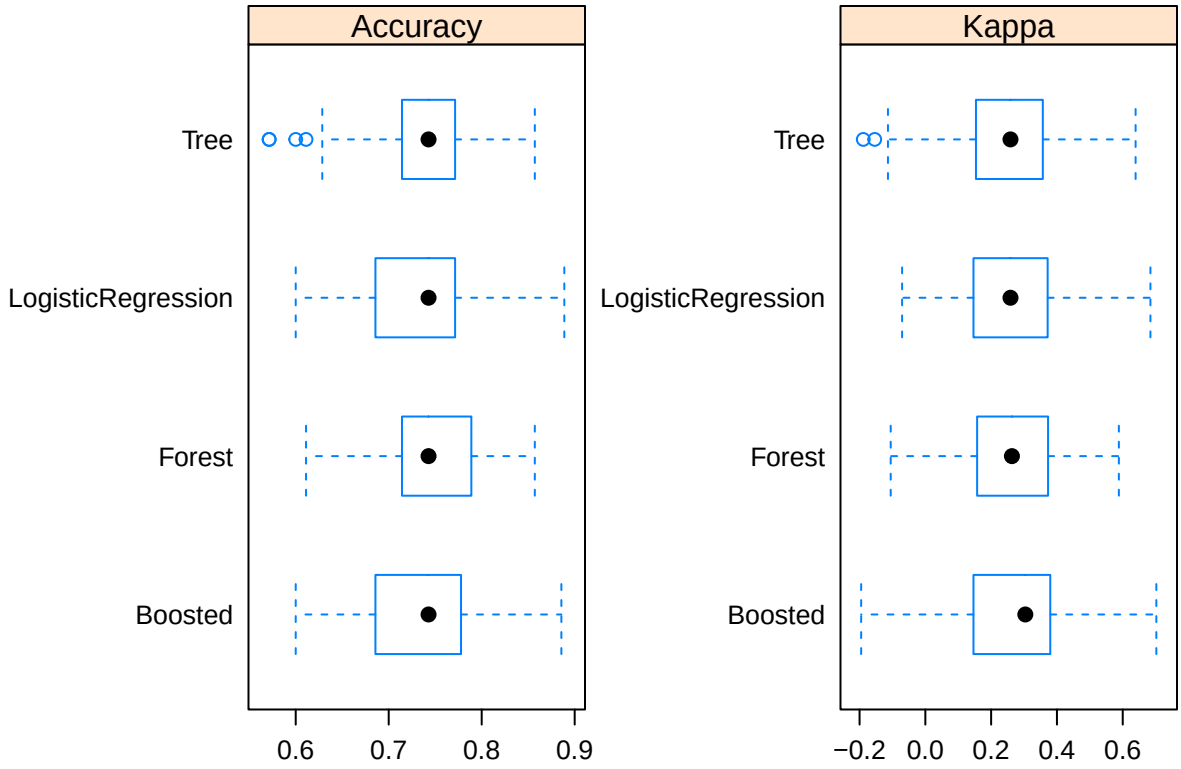


Fig 5.2 Accuracy metric evaluation boxplots

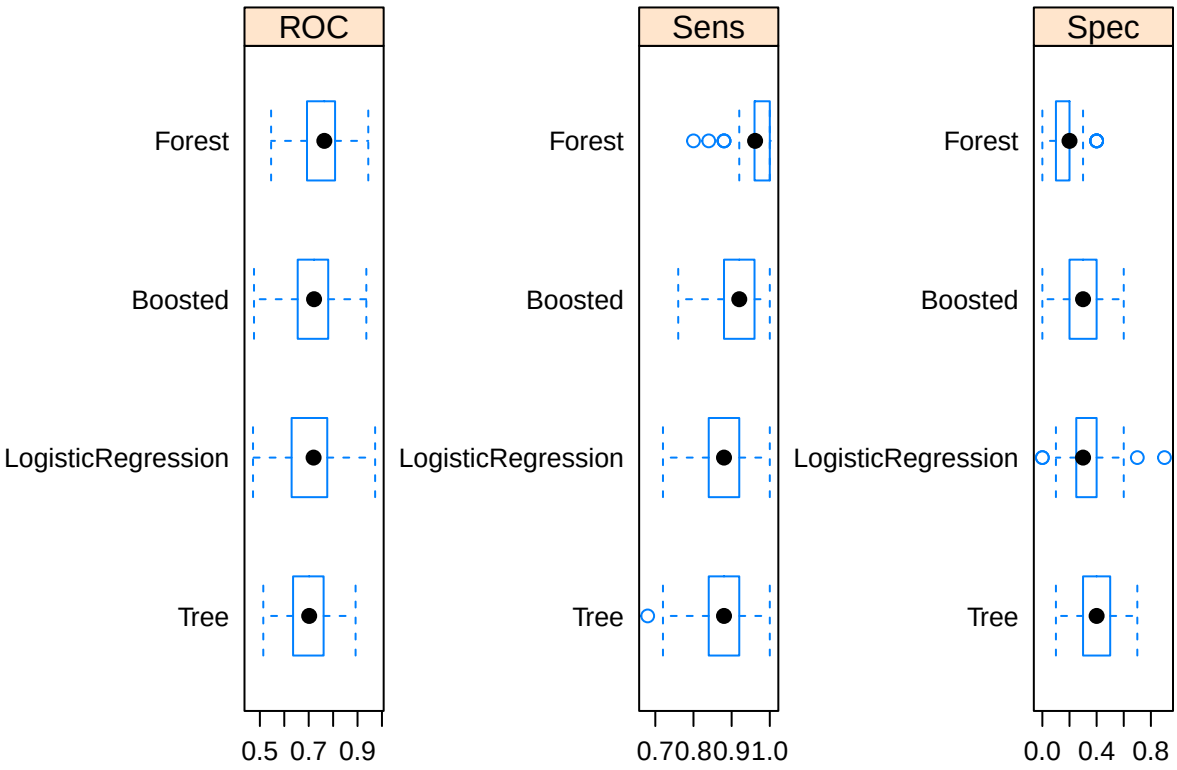
Analysis of Accuracy performance metric for the 4 algorithms
Earlier in section 4 the Logistic Regression had the highest model accuracy on test prediction.The summary of a 100 resamples in the output of our analysis in Table 5.1 above shows the Forest Model has the highest mean Accuracy performance of 75.47%.

```
##
## Call:
## summary.resamples(object = fit.compare2)
##
## Models: LogisticRegression, Tree, Forest, Boosted
## Number of resamples: 100
##
## ROC
##                      Min.   1st Qu. Median      Mean 3rd Qu.  Max. NA's
## LogisticRegression 0.472 0.6310000  0.720 0.7159923  0.7760 0.972    0
## Tree               0.514 0.6360000  0.702 0.7009185  0.7605 0.892    0
## Forest             0.546 0.6935000  0.764 0.7532823  0.8080 0.944    0
## Boosted            0.476 0.6554615  0.722 0.7199800  0.7800 0.936    0
##
## Sens
##                    Min. 1st Qu.    Median      Mean 3rd Qu. Max. NA's
## LogisticRegression 0.72    0.84 0.8800000 0.8836462    0.92    1    0
## Tree               0.68    0.84 0.8800000 0.8821538    0.92    1    0
## Forest             0.80    0.96 0.9615385 0.9685077    1.00    1    0
## Boosted            0.76    0.88 0.9200000 0.9171385    0.96    1    0
```

```
##
## Spec
##                      Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
## LogisticRegression  0.0   0.275    0.3 0.347     0.4  0.9    0
## Tree                0.1   0.300    0.4 0.367     0.5  0.7    0
## Forest              0.0   0.100    0.2 0.164     0.2  0.4    0
## Boosted             0.0   0.200    0.3 0.272     0.4  0.6    0
```

Table 5.2 ROC performance metric evaluation output

Analysis of ROC performance metric for the 4 algorithms
The summary output of the resamples shows that the Forest Model performed best with a mean ROC of 75.32% which is consistent with its position in section 4 ranking 1st with an optimal performance ROC of 74.91%. This can also be seen in the the boxplot in Fig 5.2.



Summary of Basis for Selecting Random Forest for Prediction
The comparison revealed that the **Forest Model** performed better than the other 3 models based on the foregoing:
* the model had the highest accuracy of 75.82% with 10 fold repeated cross validation
* the model the best optimal performance with an ROC metric of 74.91%
* the model has the highest mean accuracy of 75.47% over 100 samples
* the model has the highest mean ROC of 75.32% over a 100 samples and slightly better than the in-sample model
* the model has the ability to predict correctly more Non-creditworthy application with an NPV of 75%. This is particularly important as the bank needs to ensure they do not grant loans to Non-creditworthy customers, while ensuring that they adequately harness the opportunity of not wrongly classifying Creditworthy customers in the process.