Project: Creditworthiness

Complete each section. When you are ready, save your file as a PDF document and submit it here: https://classroom.udacity.com/nanodegrees/nd008/parts/11a7bf4c-2b69-47f3-9aec-108ce847f855/project

Step 1: Business and Data Understanding

Provide an explanation of the key decisions that need to be made. (250 word limit)

Key Decisions:

Answer these questions

- What decisions needs to be made?

  As a loan officer at a young and small bank (been in operations for two years) we need to come up with an efficient solution to classify new customers on whether they can be approved for a loan or not. Due to a financial scandal that hit a competitive bank last week, suddenly we have an influx of new people applying for loans at our bank instead of the other bank in your city. All of a sudden, we have nearly 500 loan applications to process this week! Historically our team used to get 200 loan applications per week and approve them by hand. This will be a tremendous task with the sudden influx of such overwhelming number of applications. We need an efficient way to do that by developing a classification model. The model will predict the creditworthiness of the customers who have applied for the loan.

- What data is needed to inform those decisions?

  We have the following information to work with:

  1. Data on all past applications.
     Credit Application Result, Account Balance, Duration of Credit Month, Payment Status of Previous Credit, Purpose, Credit Amount, Value Savings Stocks, Length of current employment, Instalment per cent, Most valuable available asset, Age years, Type of apartment, No of Credits at this Bank.
  2. The list of customers that need to be processed in the next few days
     Account Balance, Duration of Credit Month, Payment Status of Previous Credit, Purpose, Credit Amount, Value Savings Stocks, Length of current employment, Instalment per cent, Most valuable available asset, Age years, Type of apartment, No of Credits at this Bank

- What kind of model (Continuous, Binary, Non-Binary, Time-Series) do we need to use to help make these decisions?

As the answer to this problem is Binary (creditworthy / non- creditworthy), we need to build a model that best fit with it. In order to achieve it, I'll compare the following binary classification models and choose the one that performs best:

- Logistic Regression.
- Decision Tree.
- Random Forest.
- Boosted Model

We will compare the accuracy of the above models and would select the best model for prediction.

Step 2: Building the Training Set

Build your training set given the data provided to you. The data has been cleaned up for you already so you shouldn't ***need to convert any data fields to the appropriate data types.***

Here are some guidelines to help guide your data clean up:

- For numerical data fields, are there any fields that highly-correlate with each other? The correlation should be at least .70 to be considered "high".
- Are there any missing data for each of the data fields? Fields with a lot of missing data should be removed
- Are there only a few values in a subset of your data field? Does the data field look very uniform (there is only one value for the entire field?). This is called "low variability" and you should remove fields that have low variability. Refer to the "Tips" section to find examples of data fields with low-variability.
- Your clean data set should have 13 columns where the Average of **Age Years** should be 36 (rounded up)

Note: *For the sake of consistency in the data cleanup process, impute data using the median of the entire data field instead of removing a few data points. (100 word limit)*

Note: *For students using software other than Alteryx, please format each variable as:*

Variable	**Data Type**
Credit-Application-Result	String
Account-Balance	String
Duration-of-Credit-Month	Double
Payment-Status-of-Previous-	String

Credit	
Purpose	String
Credit-Amount	Double
Value-Savings-Stocks	String
Length-of-current-employment	String
Instalment-per-cent	Double
Guarantors	String
Duration-in-Current-address	Double
Most-valuable-available-asset	Double
Age-years	Double
Concurrent-Credits	String
Type-of-apartment	Double
No-of-Credits-at-this-Bank	String
Occupation	Double
No-of-dependents	Double
Telephone	Double
Foreign-Worker	Double

To achieve consistent results reviewers, expect.

Answer this question:

- In your clean-up process, which fields did you remove or impute? Please justify why you removed or imputed these fields. Visualizations are encouraged.

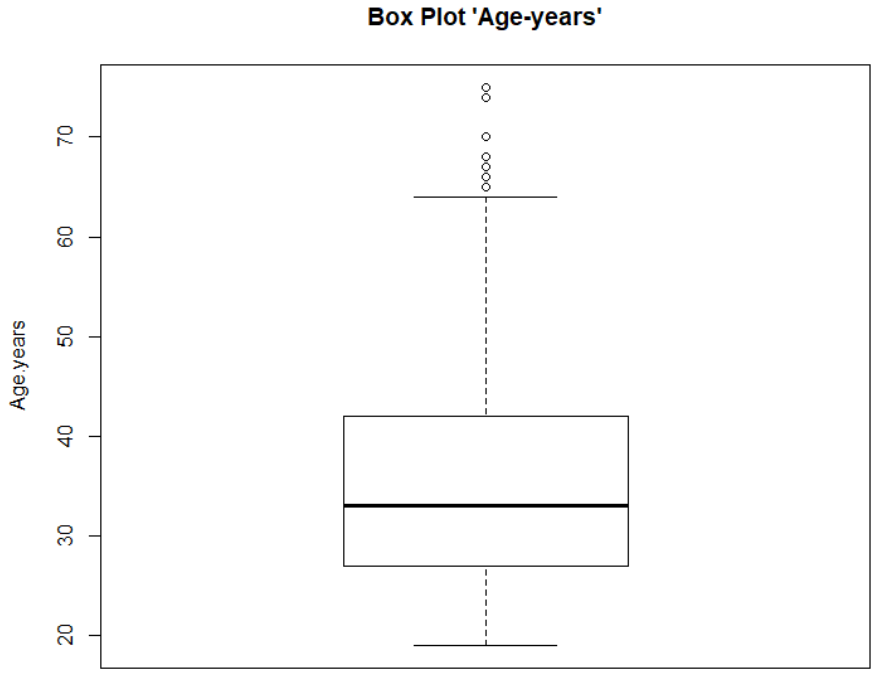
  I started with Data Exploration, exploring and visualizing the data distribution and identifying which fields could be removed because of its “Low Variability” and huge amount missing value.
  The summary of the full data set with ‘na_count’ (count of missing value) is as follows,

Variables	na_count
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

So as per the above table, we can see ‘Duration-in-Current-address’ & ‘Age-years’ are two variables with missing values. ‘Duration-in-Current-address’ has around 68.8 % values missing. We cannot accept this field in our model with such high number of missing values. Hence, we will drop this variable. ‘Age-years’ has only 2.4 % missing vales. Omitting the data associated with this small amount of missing value could be valuable. Hence, we would replace these missing values with their median value, 33.
Using the "average values" could insert a skewed graph in our model based on the age of the current clients, in this case we would be inputting an error/bias in our model and it could return a biased analysis, for example: if we train/teach our model that people with 50+ years are the most reliable/creditworthy, most of the new customers that have less than 50 years probably wouldn’t be accepted in our bank, but this concept of reliability was based in our current customers and this couldn’t be (and probably isn’t) an indisputable truth, especially when compared with a large amount of data.

When we’re using the median of age-years, we’re inputting the median value between the oldest and youngest, representing the age group that we attend in our bank. Even though this method input some errors in our dataset, at least our dataset isn’t biased, and these errors tend to be lower than the average method when applying our model with a large age group.
The box plot and the associated data is shown below.

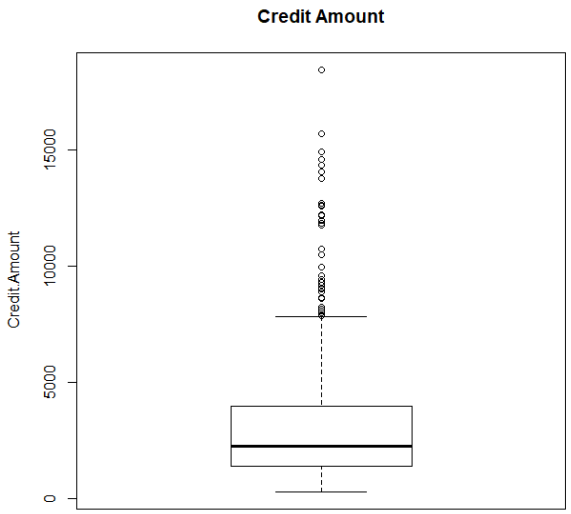


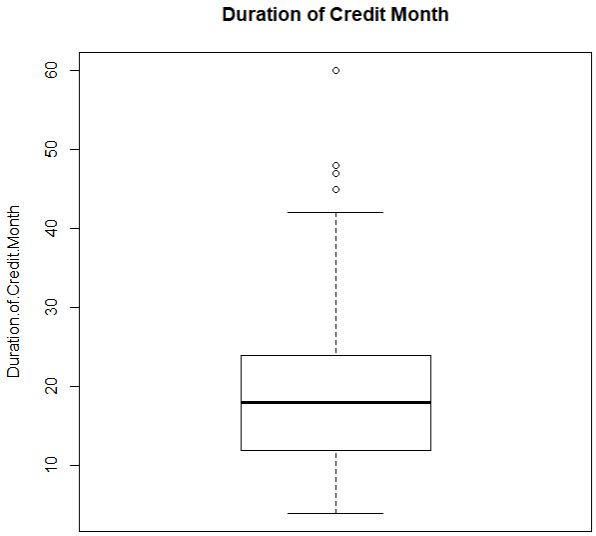

```
Min.    1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
19.00   27.00    33.00   35.64   42.00    75.00      12
```

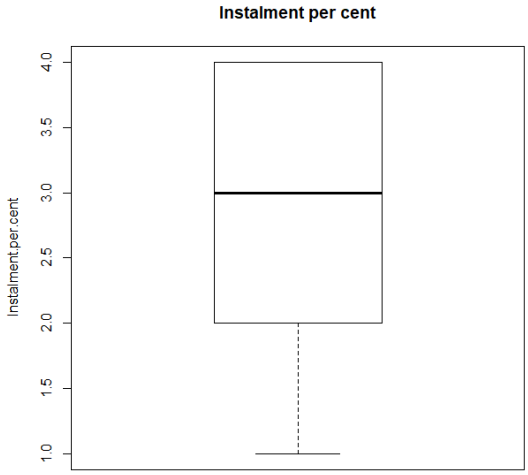
The statistical summary of the all the numeric variables are as follows,

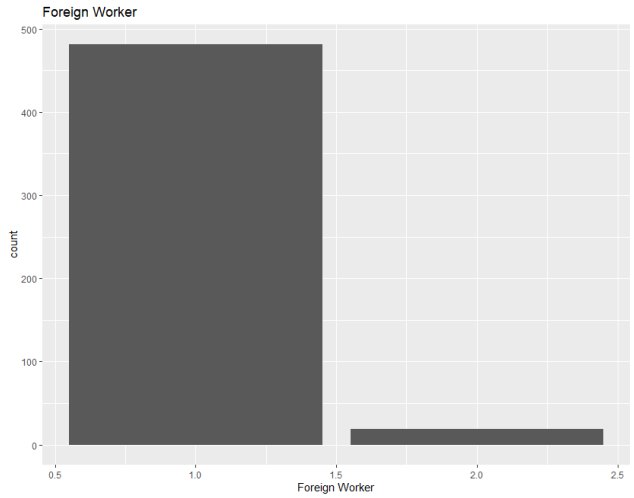
	mean	sd	IQR	0%	25%	50%	75%	100%	n	NA
Age.years	35.6373	11.5015222	15.00	19	27.00	33.0	42.0	75	488	12
Credit.Amount	3199.9800	2831.3868607	2584.25	276	1357.25	2236.5	3941.5	18424	500	0
Duration.of.Credit.Month	21.4340	12.3074201	12.00	4	12.00	18.0	24.0	60	500	0
Foreign.Worker	1.0380	0.1913877	0.00	1	1.00	1.0	1.0	2	500	0
Instalment.per.cent	3.0100	1.1137238	2.00	1	2.00	3.0	4.0	4	500	0
Most.valuable.available.asset	2.3600	1.0642675	2.00	1	1.00	3.0	3.0	4	500	0
No.of.dependents	1.1460	0.3534599	0.00	1	1.00	1.0	1.0	2	500	0
Occupation	1.0000	0.0000000	0.00	1	1.00	1.0	1.0	1	500	0
Telephone	1.4000	0.4903886	1.00	1	1.00	1.0	2.0	2	500	0
Type.of.apartment	1.9280	0.5398137	0.00	1	2.00	2.0	2.0	3	500	0

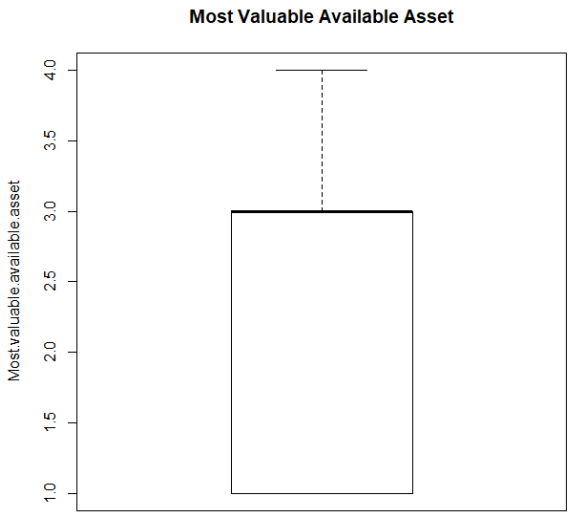
Some interesting results appear!! Let us visualize them.

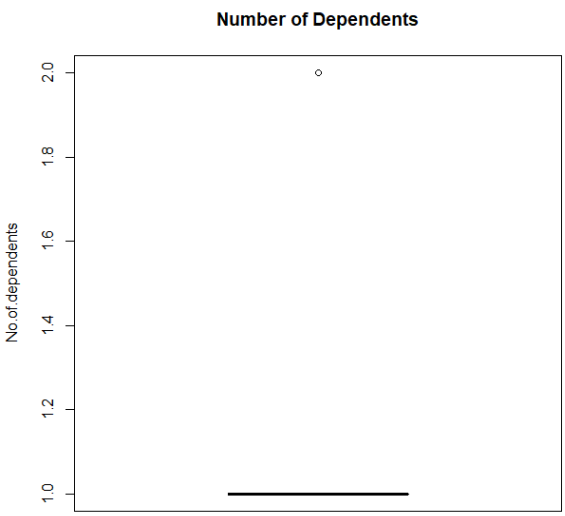

Credit Amount
Credit.Amount
0
5000
10000
15000

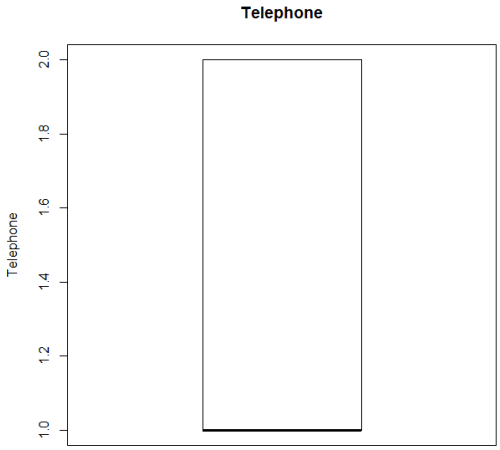

Duration of Credit Month
Duration.of.Credit.Month
10
20
30
40
50
60

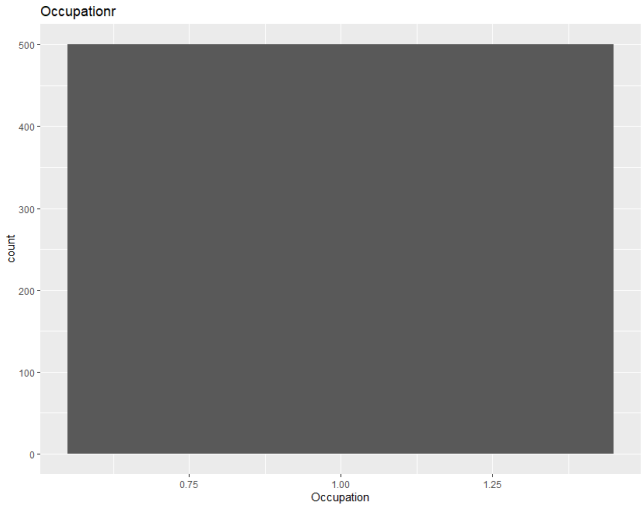

Instalment per cent
Instalment.per.cent
1.0
1.5
2.0
2.5
3.0
3.5
4.0

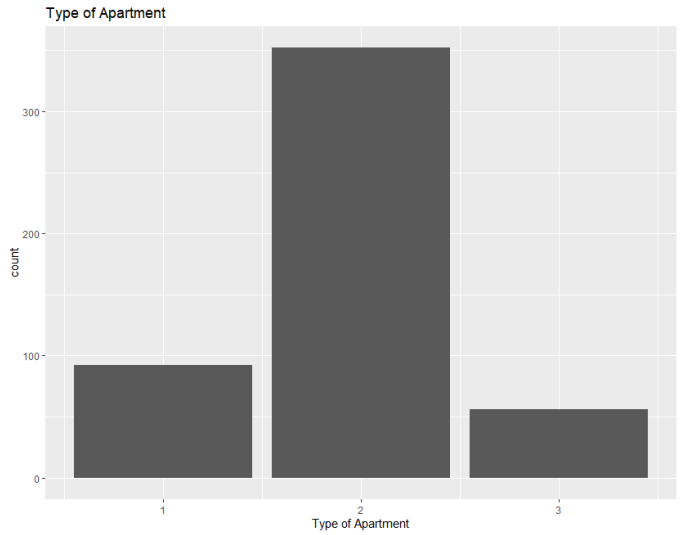

Foreign Worker
count
0
100
200
300
400
500
0.5
1.0
1.5
2.0
2.5
Foreign Worker


Most Valuable Available Asset
Most.valuable.available.asset
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Number of Dependents
No.of.dependents
1.0
1.2
1.4
1.6
1.8
2.0


Telephone
Telephone
1.0
1.2
1.4
1.6
1.8
2.0

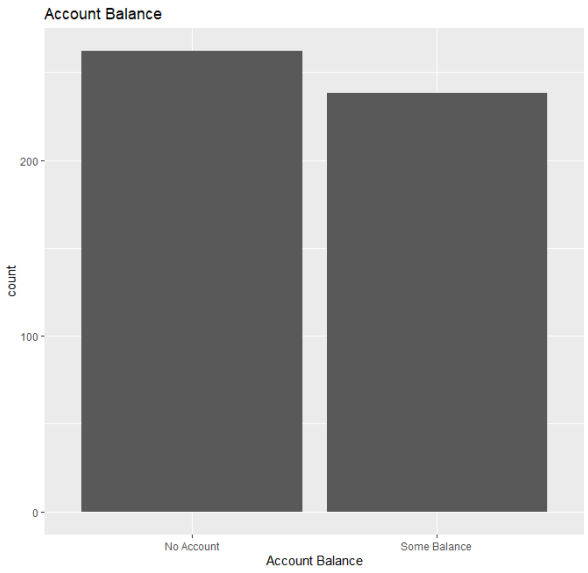

Occupationr
count
0
100
200
300
400
500
0.75
1.00
1.25
Occupation

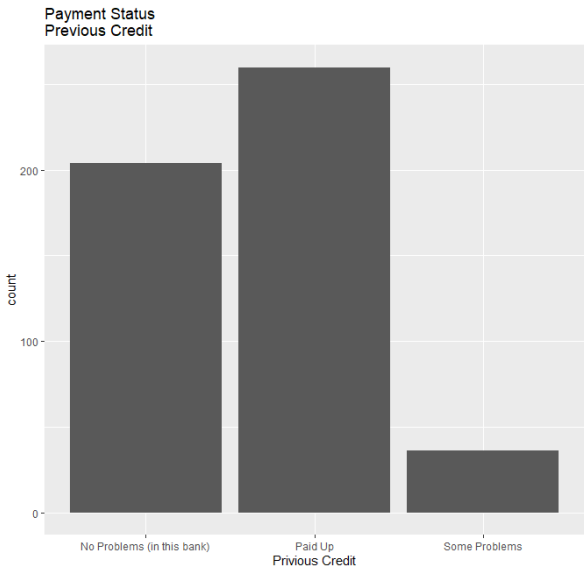

Type of Apartment
count
0
100
200
300
1
2
3
Type of Apartment

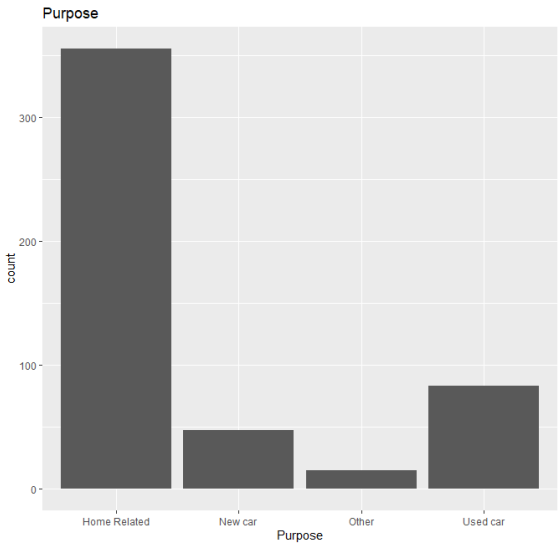
From the above boxplots, we can easily identify the “low variability” variables,

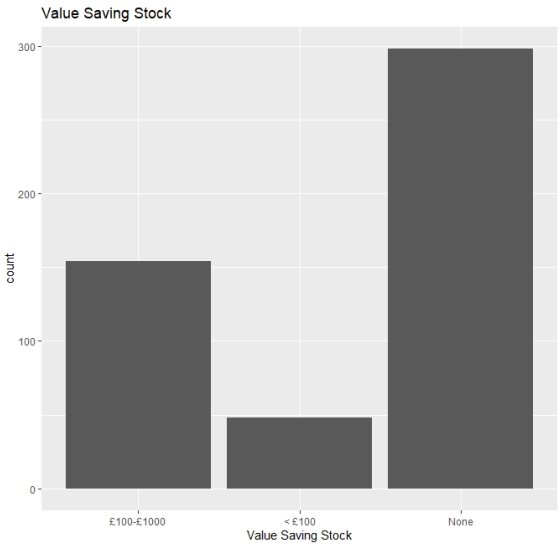
- Foreign Worker
- Number of dependents
- Occupation

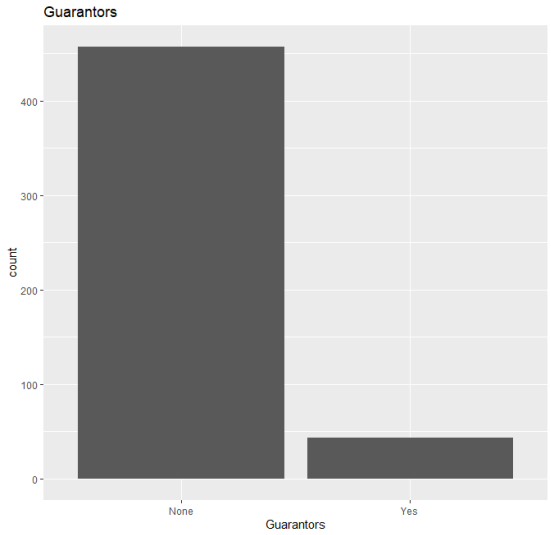
We would remove them from our model.
Now let us look at the non-numeric variables in data set,

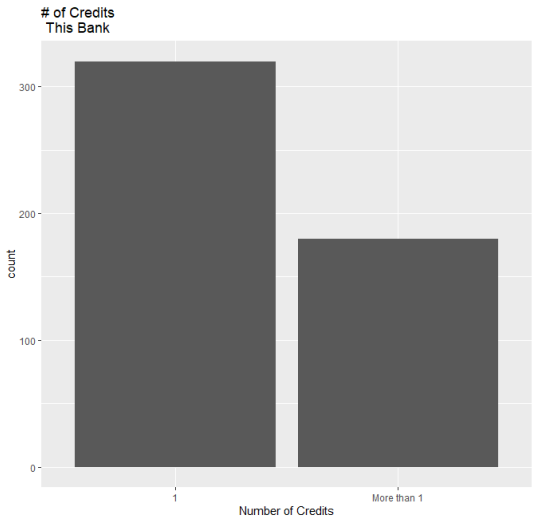


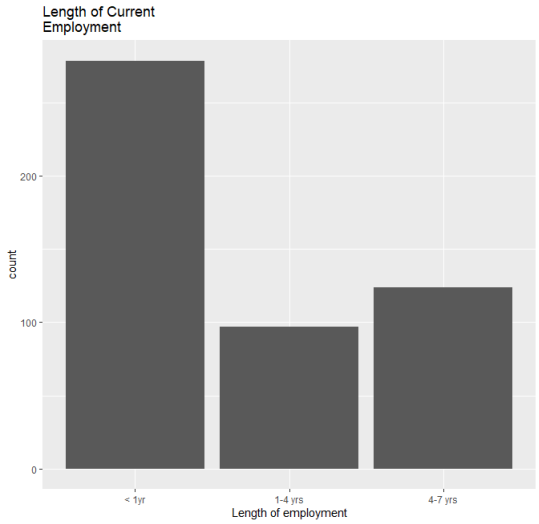


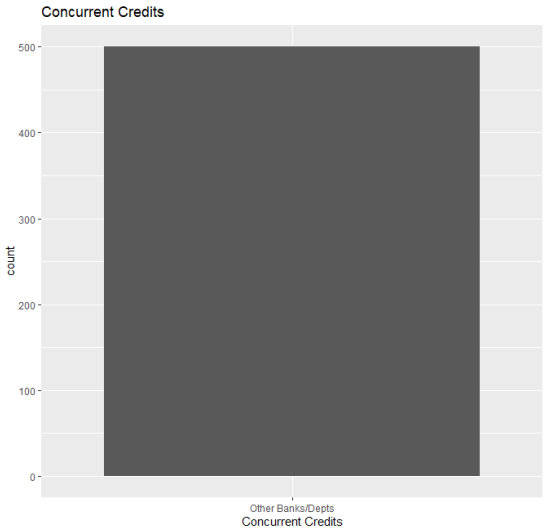












We can remove the variable ‘Concurrent Credit’ & ‘Guarantor’ due to low variability. So, the variables in the final cleaned data set is as follows,

Variable	Data Type
Credit-Application-Result	factor
Account-Balance	factor
Duration-of-Credit-Month	numeric
Payment-Status-of-Previous-Credit	factor
Purpose	factor
Credit-Amount	numeric
Value-Savings-Stocks	factor
Length-of-current-employment	factor
Instalment-per-cent	numeric
Most-valuable-available-asset	numeric

Age-years	numeric
No-of-Credits-at-this-Bank	factor
Telephone	numeric
Type of Apartment	factor

Step 3: Train your Classification Models

First, create your Estimation and Validation samples where 70% of your dataset should go to Estimation and 30% of your entire dataset should be reserved for Validation. Set the Random Seed to 1.

First task is to split the data set of 500 rows, into training set and the test set to validate the accuracy and acceptability of the model. So, we need a training set that will have 350 observations and a test set with 150 observations.

Randomly 350 observation were selected and the balanced one was selected for test set.

In order to test the uniformity of these two sets, the predicted variables percentage distribution was checked.

Set	Creditworthy (%)	Non-Creditworthy
Training Set	68.5%	31.5 %
Test Set	78.6 %	21.4 %

So, we can assume that both sets are relatively unbiased.

Create all of the following models: Logistic Regression, Decision Tree, Forest Model, Boosted Model

*Answer these questions for **each model** you created:*

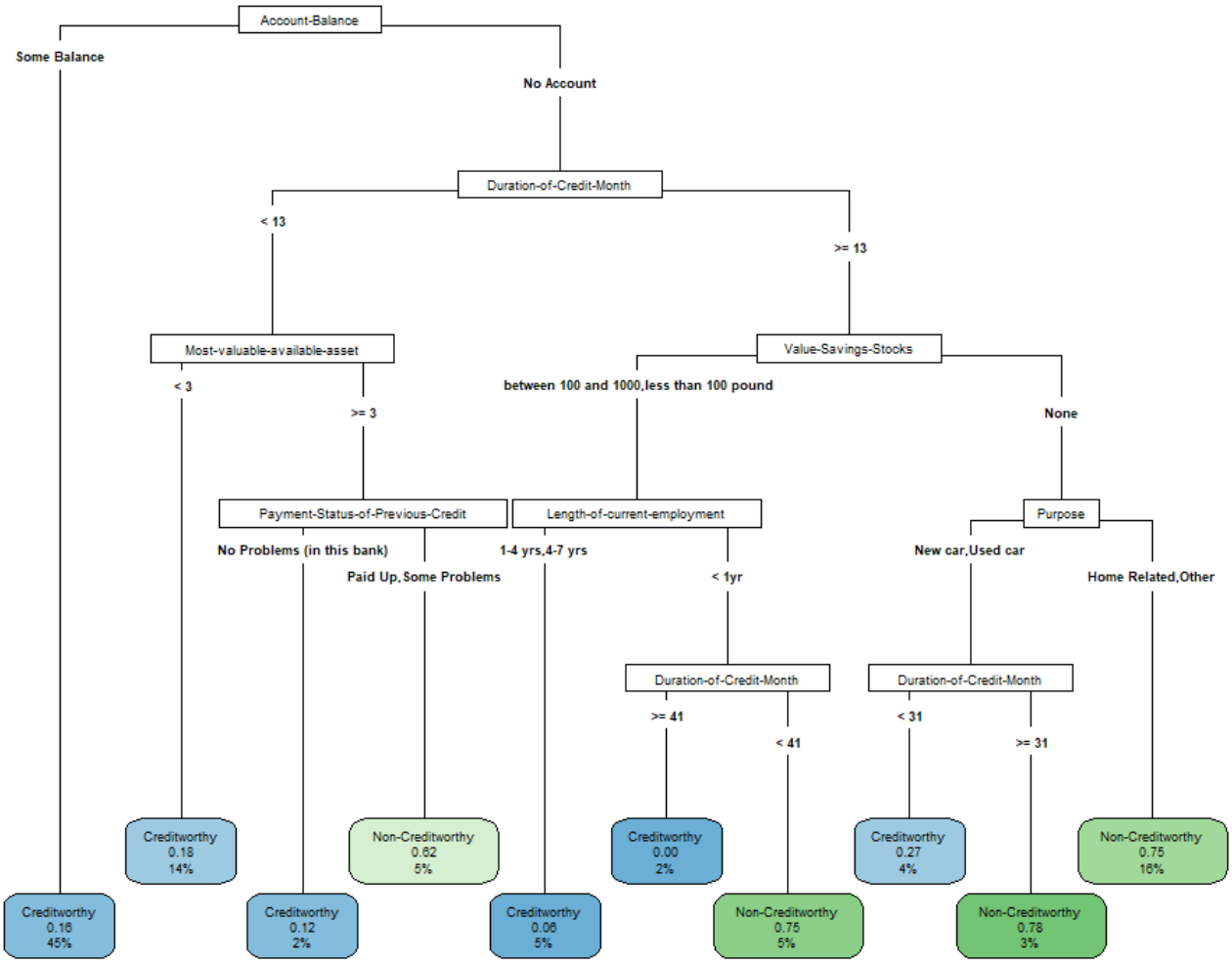
- Which predictor variables are significant or the most important? Please show the p-values or variable importance charts for all of your predictor variables.

- Validate your model against the Validation set. What was the overall percent accuracy? Show the confusion matrix. Are there any bias seen in the model's predictions?

You should have four sets of questions answered. (500 word limit)

1. **Model Name: Decision Tree**
   The first model that I have developed is a decision tree model.

The tree as per the training set are as follows,



The variable importance table are mention below,

Duration-of-Credit-Month	23
Account-Balance	19
Credit-Amount	14
Value-Savings-Stocks	12
Payment-Status-of-Previous-Credit	7
Length-of-current-employment	7
Purpose	6
Age-years	5
Most-valuable-available-asset	4

No-of-Credits-at-this-Bank	3
Instalment-per-cent	1

Account Balance, Value Savings Stocks and Duration of Credit Month are the top 3 most important variables.
The model summary report is as follow,

Now we used the model to validate the test data,
We found that the model predicted 116 application as “Creditworthy” and 34 “Non-Creditworthy”.
And the confusion matrix is as follows,

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  150
```

credit_test$`Credit-Application-Result`	credit_pred Creditworthy	Non-Creditworthy	Row Total
Creditworthy	99	19	118
	0.853	0.559	
	0.660	0.127	
Non-Creditworthy	17	15	32
	0.147	0.441	
	0.113	0.100	
Column Total	116	34	150
	0.773	0.227	

So, model accuracy (TP+TN / TP+TN+FP+FN) = 76 %.
Recall or Sensitivity (TP/ TP+FN) = 85%
Specificity (TN / FP +TN) = 44.1%
Precision (TP/ TP+FP) = 83.9 %
F-score (2*P*R/(R+P) = 0.8444

2. **Model Name: Logistic Regression**
   The second model was developed was logistics regression model.
   The full model was first developed and then a stepwise regression model was developed to find the best model.

   The full logistic model report is as follows,

```
Call:
glm(formula = `Credit-Application-Result` ~ ., family = binomial,
    data = credit_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0444  -0.7882  -0.5012   0.8949   2.5343

Coefficients:
                                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                                   -2.131e+00  8.908e-01  -2.392  0.01674 *
`Account-Balance`Some Balance                 -1.294e+00  2.941e-01  -4.401 1.08e-05 ***
`Duration-of-Credit-Month`                     1.247e-02  1.299e-02   0.960  0.33707
`Payment-Status-of-Previous-Credit`Paid Up      7.267e-01  3.542e-01   2.051  0.04023 *
`Payment-Status-of-Previous-Credit`Some Problems  1.528e+00  5.135e-01   2.976  0.00292 **
PurposeNew car                                -1.377e+00  5.835e-01  -2.360  0.01826 *
PurposeOther                                   7.841e-04  9.296e-01   0.001  0.99933
PurposeUsed car                               -4.672e-01  3.708e-01  -1.260  0.20769
`Credit-Amount`                                7.637e-05  6.587e-05   1.159  0.24627
`Value-Savings-Stocks`less than 100 pound     -1.892e-01  5.294e-01  -0.357  0.72081
`Value-Savings-Stocks`None                     6.282e-01  3.080e-01   2.040  0.04138 *
`Length-of-current-employment`1-4 yrs         -8.541e-01  3.808e-01  -2.243  0.02492 *
`Length-of-current-employment`4-7 yrs         -2.933e-01  3.616e-01  -0.811  0.41732
`Instalment-per-cent`                          1.164e-01  1.324e-01   0.879  0.37923
`Most-valuable-available-asset`                2.658e-01  1.579e-01   1.683  0.09233 .
`Age-years`                                   -3.930e-03  1.354e-02  -0.290  0.77160
`Type-of-apartment`2                          -1.272e-01  3.435e-01  -0.370  0.71113
`Type-of-apartment`3                          -3.546e-01  6.026e-01  -0.588  0.55622
`No-of-Credits-at-this-Bank`More than 1        2.486e-01  3.455e-01   0.719  0.47187
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 435.74  on 349  degrees of freedom
Residual deviance: 360.09  on 331  degrees of freedom
AIC: 398.09

Number of Fisher Scoring iterations: 4
```

The stepwise regression model report is as follows,

Stepwise Logistic
Model summary repo

The selected model based on lowest AIC value is as follows,

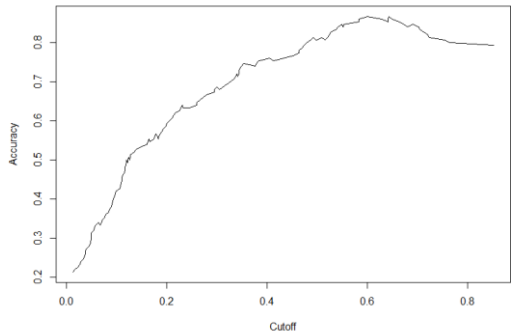
```
Step:  AIC=364.33
`Credit-Application-Result` ~ `Account-Balance` + `Payment-Status-of-Previous-Credit` +
    Purpose + `Credit-Amount` + `Length-of-current-employment` +
    `Instalment-per-cent`

                                    Df Deviance    AIC
<none>                                   342.33 364.33
- Purpose                           3   349.58 365.58
- `Instalment-per-cent`             1   350.60 370.60
- `Length-of-current-employment`    2   352.73 370.73
- `Credit-Amount`                   1   360.08 380.08
- `Account-Balance`                 1   361.54 381.54
- `Payment-Status-of-Previous-Credit`  2   364.34 382.34

Selected Model based on lowest AIC

Call:
glm(formula = `Credit-Application-Result` ~ `Account-Balance` +
    `Payment-Status-of-Previous-Credit` + Purpose + `Credit-Amount` +
    `Length-of-current-employment` + `Instalment-per-cent`, family = binomial,
    data = credit_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9040  -0.7693  -0.5109   0.6920   2.4325

Coefficients:
                                                   Estimate  Std. Error z value  Pr(>|z|)
(Intercept)                                      -2.21934574  0.59920615  -3.704  0.000212 ***
`Account-Balance`[T.Some Balance]                -1.22207032  0.28908601  -4.227 0.0000236 ***
`Payment-Status-of-Previous-Credit`[T.Paid Up]     0.44563640  0.29449600   1.513  0.130225
`Payment-Status-of-Previous-Credit`[T.Some Problems]  2.59917238  0.60853299   4.271 0.0000194 ***
Purpose[T.New car]                               -1.44802202  0.63510969  -2.280  0.022610 *
Purpose[T.Other]                                 -0.55904856  0.78788307  -0.710  0.477978
Purpose[T.Used car]                              -0.40408532  0.38737057  -1.043  0.296879
`Credit-Amount`                                   0.00021660  0.00005549   3.903 0.0000950 ***
`Length-of-current-employment`[T.1-4 yrs]        -1.27340076  0.42505403  -2.996  0.002737 **
`Length-of-current-employment`[T.4-7 yrs]        -0.33739070  0.32986883  -1.023  0.306401
`Instalment-per-cent`                             0.37961793  0.13652116   2.781  0.005425 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 424.16  on 349  degrees of freedom
Residual deviance: 342.33  on 339  degrees of freedom
AIC: 364.33

Number of Fisher Scoring iterations: 5
```

As per the logistics model, most significant variables with p-value of less than 0.05.are, Account balance, payment status of previous credit, Purpose, values saving stocks, length of current employment.

Now we used the model to validate the test data,
The confusion matrix is,

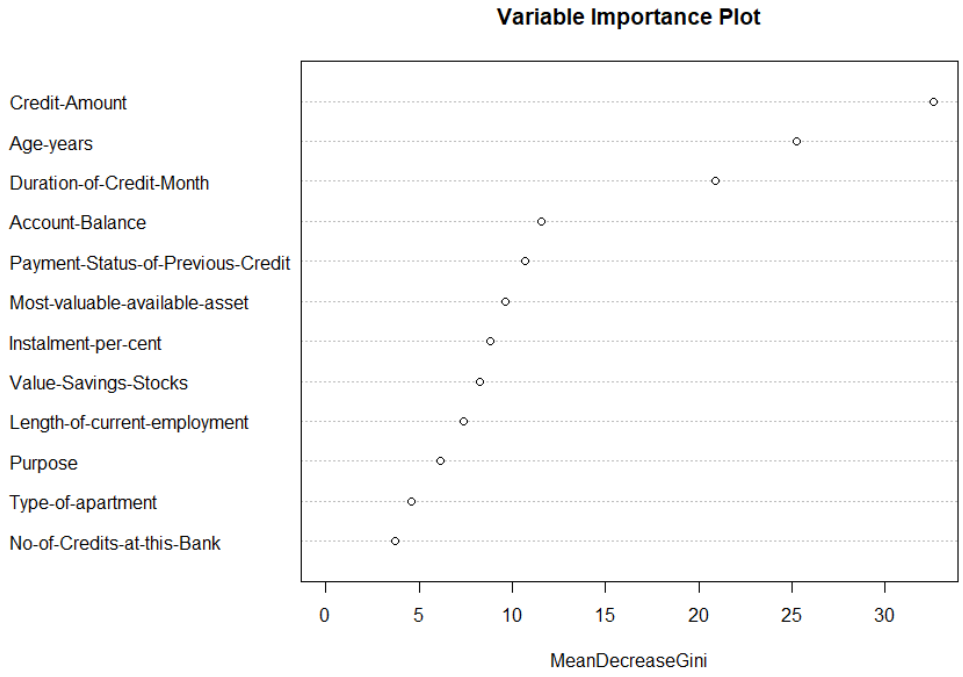
	Actual Creditworthy	Actual Non-Creditworthy
Predicted Creditworthy	97	14
Predicted Non-Creditworthy	20	19

ROC Curve,



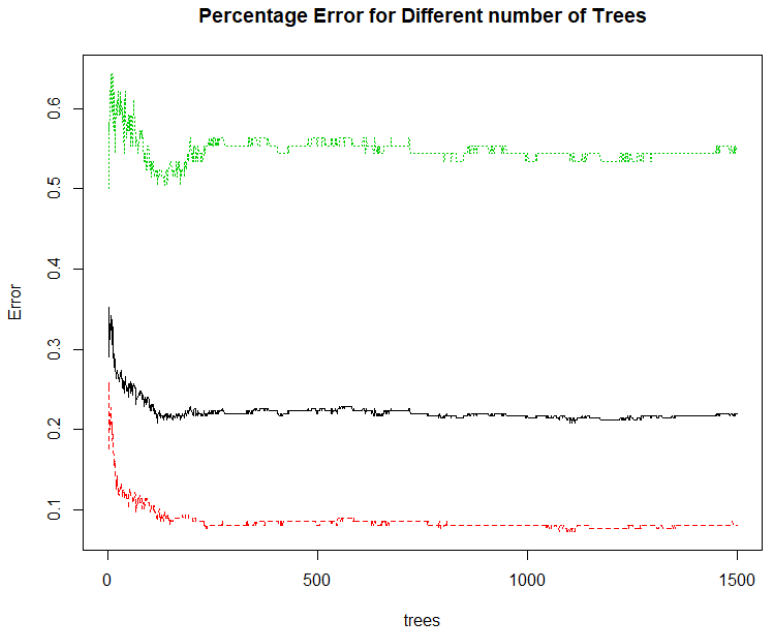
So, model accuracy (TP+TN / TP+TN+FP+FN) = 77 %.
Recall or Sensitivity (TP/ TP+FN) = 82%
Specificity (TN / FP +TN) = 57.5%
Precision (TP/TP+FP) = 87.4%
F-score (2*P*R/(R+P) = 0.8461

3. **Random Forest Model**
   The next model developed was random forest model.
   The training data was used to train the forest model with 1500 trees and
   Using Credit Application Result as the target variables, Credit Amount, Age Years and Duration of Credit Month are the 3 most important variables. The variable importance plot is as follows,

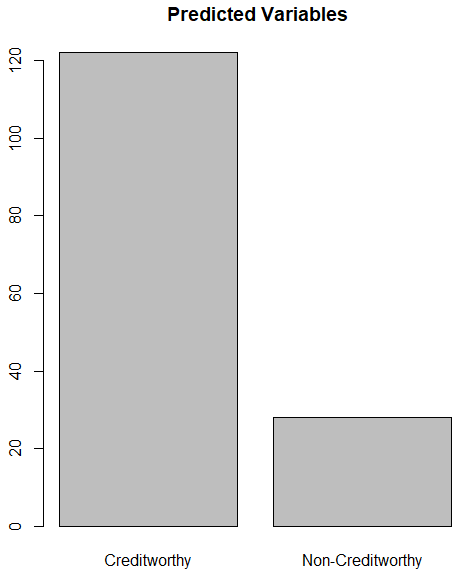



```
Call:
 randomForest(x = credit_train[2:13], y = credit_train$`Credit-Application-Result`,ntree = 1500, mtry = 5, data = credit_train)
               Type of random forest: classification
                     Number of trees: 1500
No. of variables tried at each split: 5

        OOB estimate of  error rate: 22%
Confusion matrix:
                Creditworthy Non-Creditworthy class.error
Creditworthy             227              20  0.08097166
Non-Creditworthy          57              46  0.55339806
```

The MSE error rate for the training model is as follows,



Once the Model is used to the test set, the predicted outcomes for the test set are,



The confusion matrix for the test set as follows,

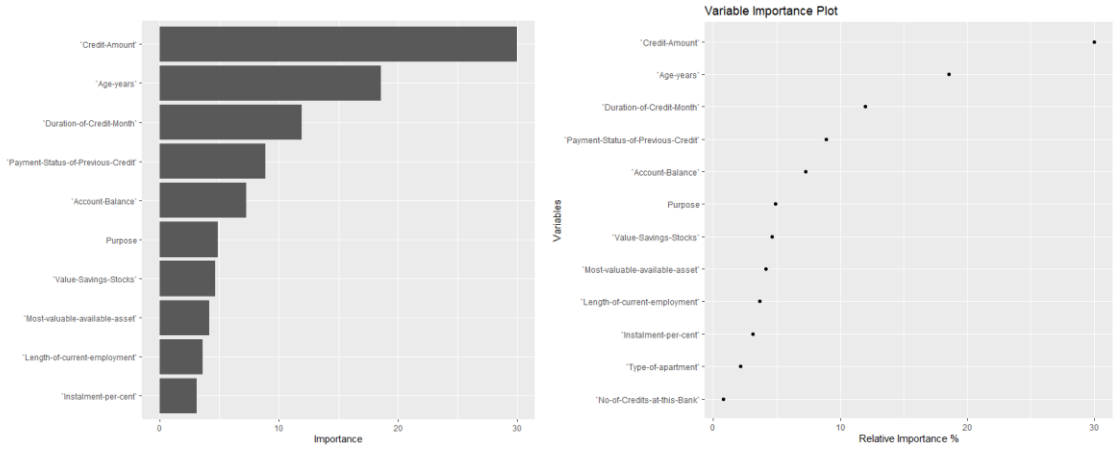
```
   Cell Contents
|-------------------------|
|                       N |
|           N / Col Total |
|-------------------------|

Total Observations in Table:  150
```

credit_test$`Credit-Application-Result`	forest_pred Creditworthy	Non-Creditworthy	Row Total
Creditworthy	107 0.877	11 0.393	118
Non-Creditworthy	15 0.123	17 0.607	32
Column Total	122 0.813	28 0.187	150

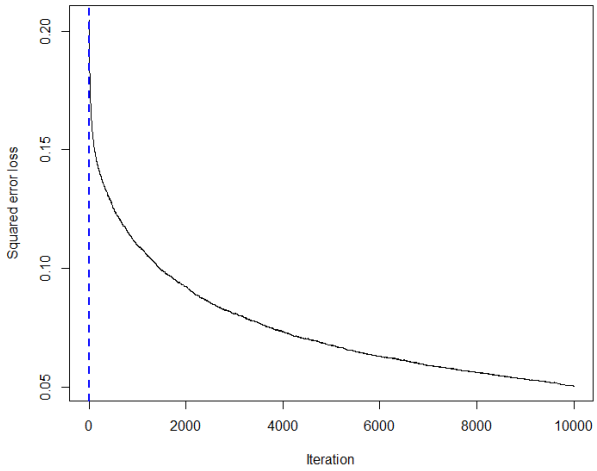
So, model accuracy (TP+TN / TP+TN+FP+FN) = 82.6 %.
Recall or Sensitivity (TP/ TP+FN) = 87.7%
Specificity (TN / FP +TN) = 60.7 %
Precision (TP/TP+FP) = 90.6%
F-score (2*P*R/(R+P) = 0.8922

4. **Boosted Model**

   The next model used was a boosted model.
   The variable importance plot is shown below,



The error loss plot of the model,



The confusion matrix is,

	Actual Creditworthy	Actual Non-Creditworthy
Predicted Creditworthy	100	27
Predicted Non-Creditworthy	5	18

So, model accuracy (TP+TN / TP+TN+FP+FN) = 78.6 %.
Recall or Sensitivity (TP/ TP+FN) = 95.2%
Specificity (TN / FP +TN) = 40 %
Precision (TP/TP+FP) = 78%
F-score (2*P*R/(R+P) = 0.8566

Step 4: Writeup

The purpose of this report was to create a classification model to systematically evaluate the creditworthiness of new loan applicants. To do so, four standard models were compared in Alteryx, namely, Logistic Regression, Decision Tree, Forest Model and Boosted Model.
All the examined models demonstrated high classification accuracy (76% – 82%). However, they also demonstrated a high bias to correctly predicting applicants who were creditworthy. Indeed, for all models, the accuracy of correctly classifying application who were not creditworthy was worse than guessing at random (37 – 49%) while correctly classifying applications as creditworthy was high (82% -96%). This is likely due to high class imbalance observed in the dataset and future reports should look to oversampling/under sampling methods.
The model Comparison with respect to the parameters calculated in step 3 are,

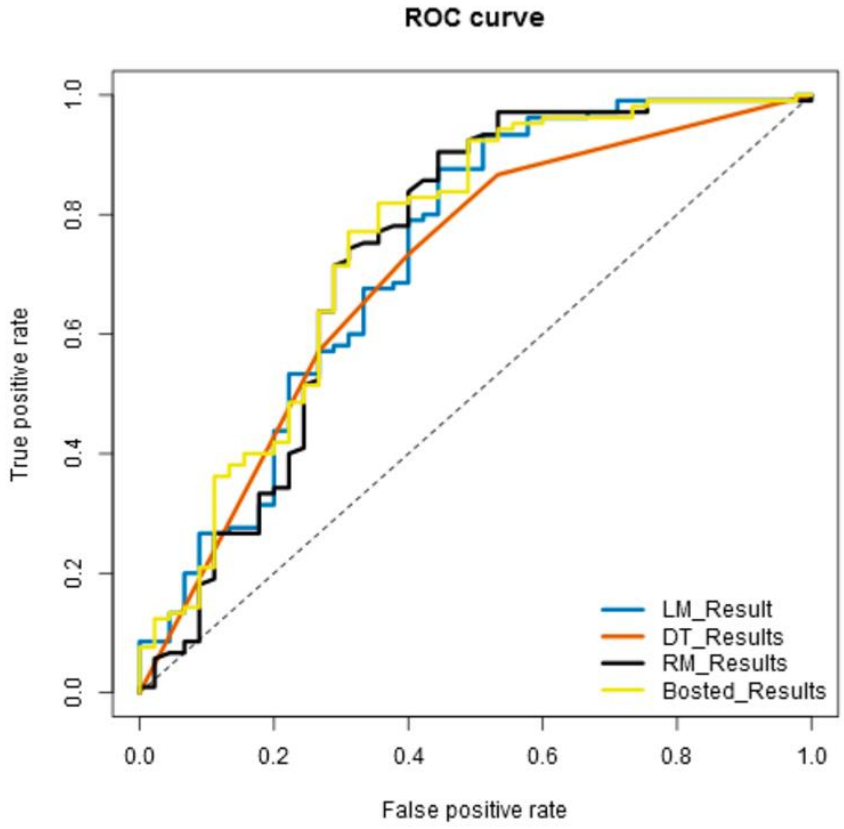
Model Name	Model Accuracy	Recall or Sensitivity	Specificity	Precision	F-score	
Decision Tree	76 %	85 %	44.1 %	83.9 %	0.8444	
Logistic	77 %	82 %	57.5 %	87.4 %	0.8566	
Random Forest	82.6 %	87.7%	60.7%	90.6%	0.8922	
Boosted Model	78.6%	95.2%	40%	78%	0.8566	

The above figure show, that the Random Forest model score is highest F-measure (F-score). Since F-score combine the precision and recall in a single parameter, it can be concluded in this case that the random forest model has least bias.

Final Model.

The final model used for prediction will be the Random Forest model due to its highest overall accuracy at 82% It has a high accuracy, also the highest F-score of 0.8922.

Below is the ROC chart for the models.



The ROC plots show the Random Forest model to be the second best with an AUC of 0.7380.

Applying the model to the new dataset, customers-to-score.xls and taking any applicant that has a greater Creditworthy accuracy score than non-creditworthy to mean the applicant should be granted a loan, the final count of individuals who are creditworthy are 411.