Project Report

Jabir Mohammed

EXECUTIVE SUMMARY

The objective of this report is to analyse the credit information of customers of a bank, and predict whether a customer would fully pay their loan amount or not. This prediction is done to help various bank determine the risky customers who most likely would not be able to repay the loan. The objective of this report is achieved by utilizing various data mining and machine learning algorithms. We concluded that Naive Bayes Classifier would be the best model because it correctly predicted the highest number of potential risky customers, ie. highest count of true negatives.

INTRODUCTION

The current national debt in the US is 27 trillion USD. In the US, the student loan debt alone is 1.48 trillion USD. One of the biggest problems for banks are unpaid loans. Banks want to minize their risks and maximize their investments. They cannot just stop sanctioning loans. Hence it is important for banks to gauge the riskiness of a customer. For such risky customers the banks can either charge higher interest rates or reject loan requests. Prediction of risky customers can be done using predictive models.

DATASET DESCRIPTION

The dataset contains credit information of customers of a bank. It has 100514 rows and 19 columns. It contains features like Credit Score, Annual Income, Monthly Debt etc. It contains both Quantitative and Qualitative variables.

DATA PREPROCESSING

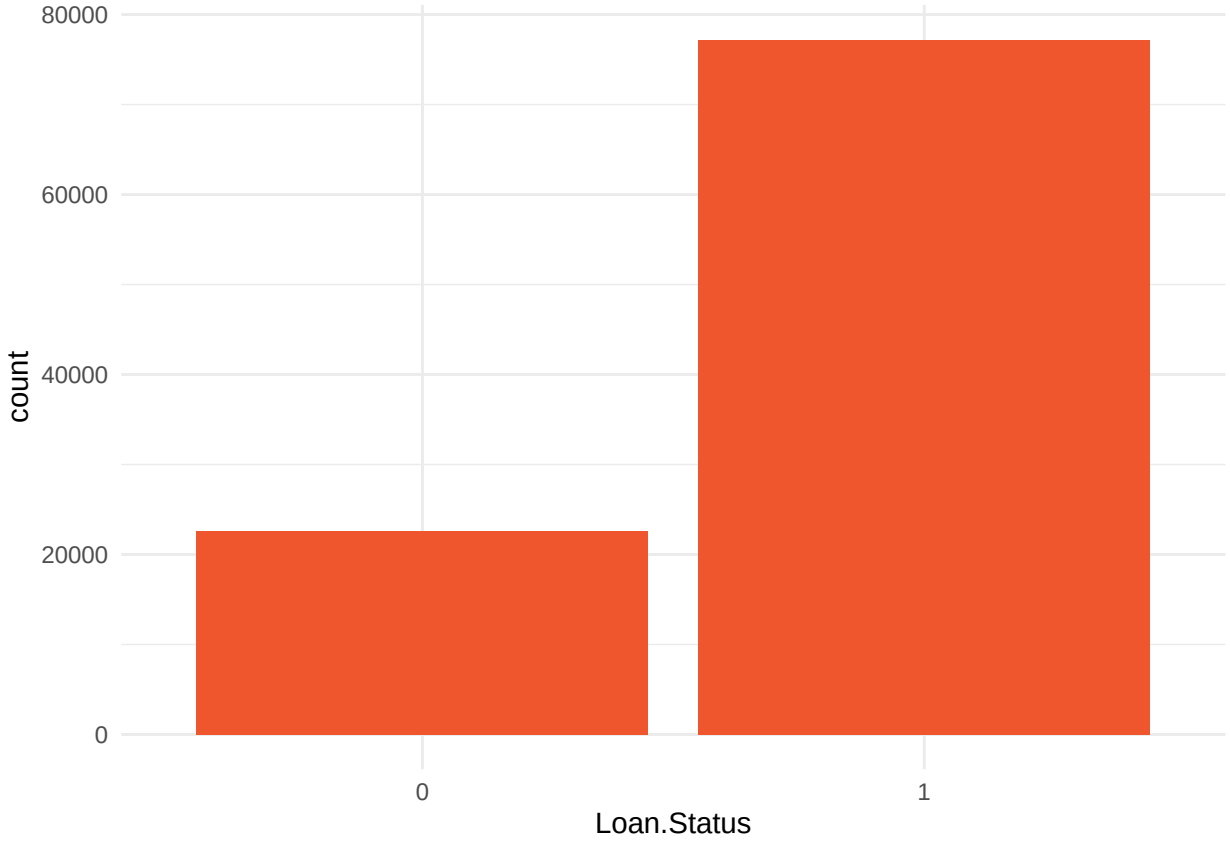
The data has large number of missing/null values. Columns LoanID and CustomerID seem irrelevant for prediction so they have been removed. 50% of data in the column Months since last delinquent is filled with null values, so it was removed as well.

The Current Loan Amount Column has median 312246 but max value as 99999999. Upon further investigation, there was only one values 99999999 which had multiple instances. So this value had to converted to null, then imputed the nulls using MICE package.

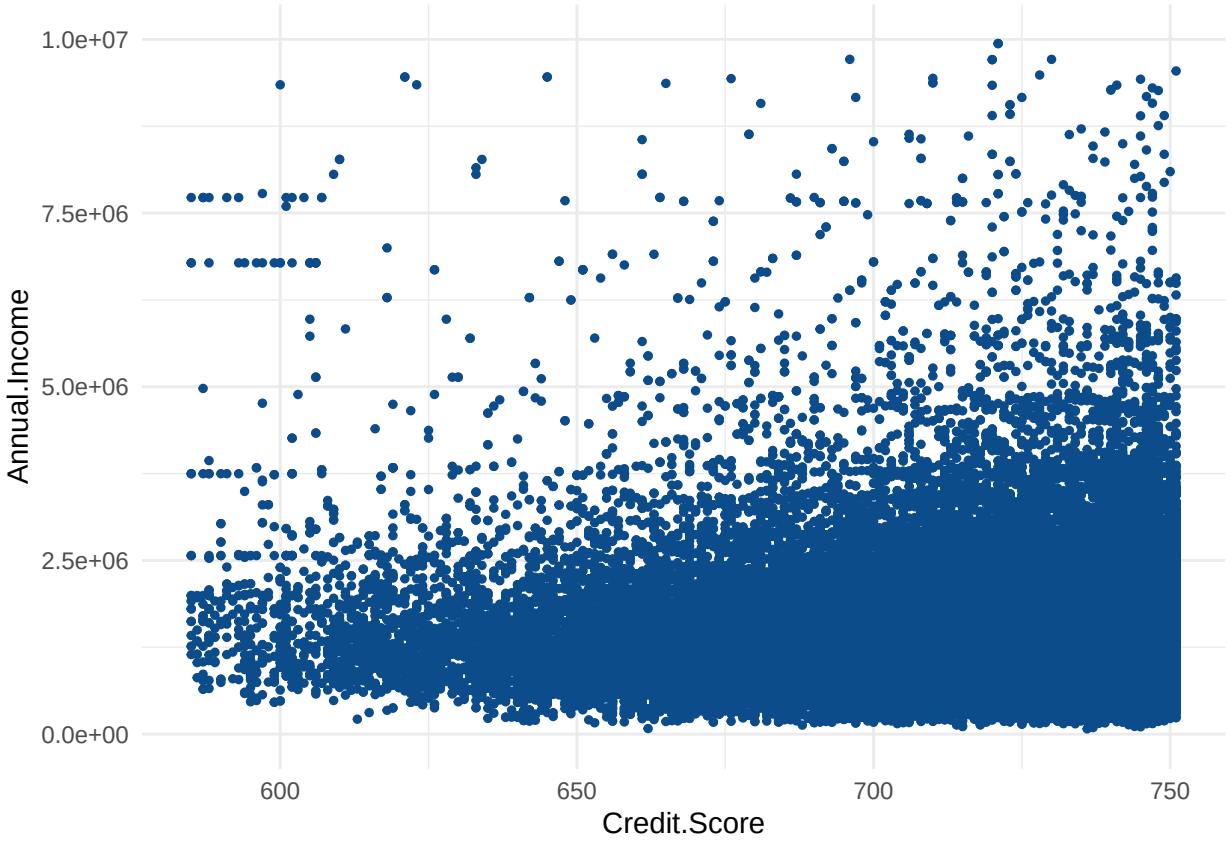
The values in credit score range from 585 to 7510, which is strange considering the credit score is within the range of 300-850. It was deduced that some values had been scaled up by 10. These values had to be scaled down by 10. Columns Credit score and Annual Income also had large number of null values. These nulls were imputed using the MICE package.

The column Years in current job also had null values. Since the values are categorical, the nulls were replaced with the highest accuring category. The categories in the predictor variable Loan Status were converted to '0' and '1'. Finally, most of the rows contained exactly 514 null values. These values were omitted.

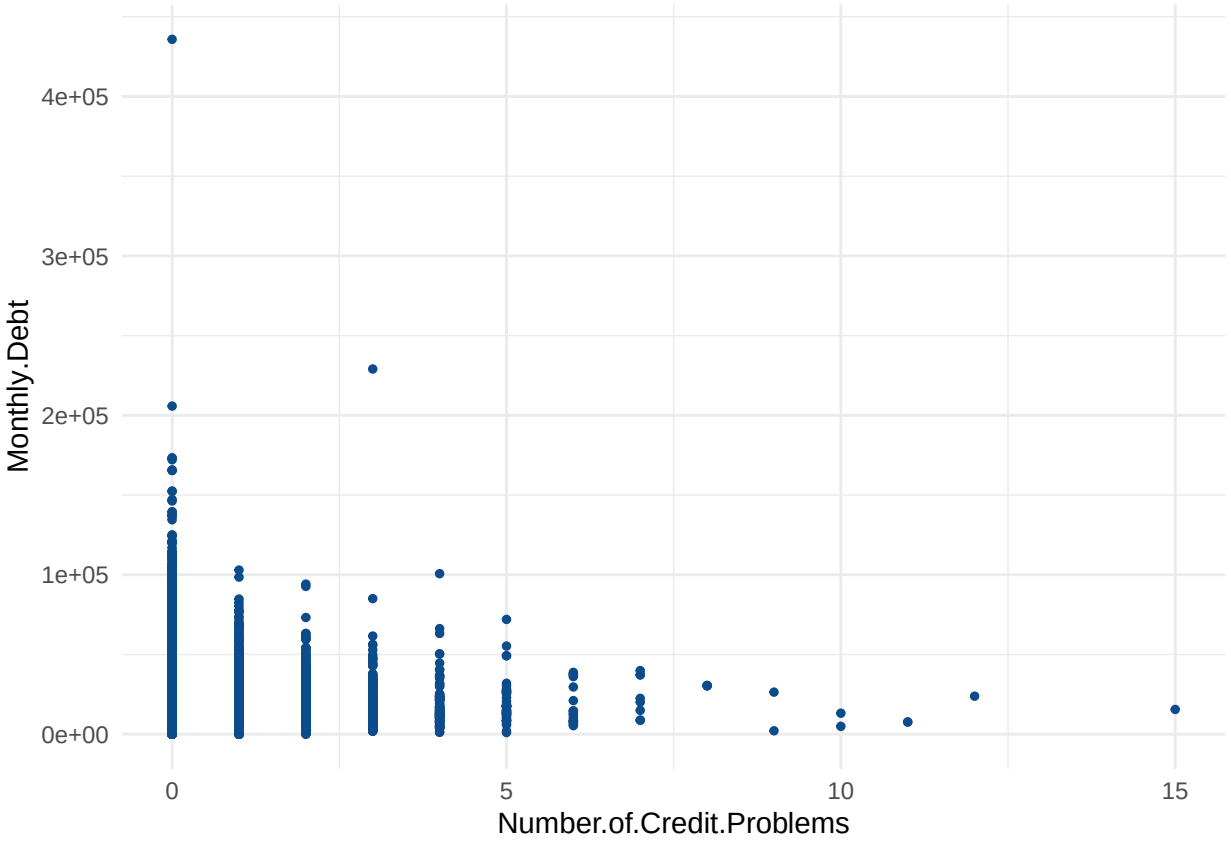
EXPLORATORY DATA ANALYSIS



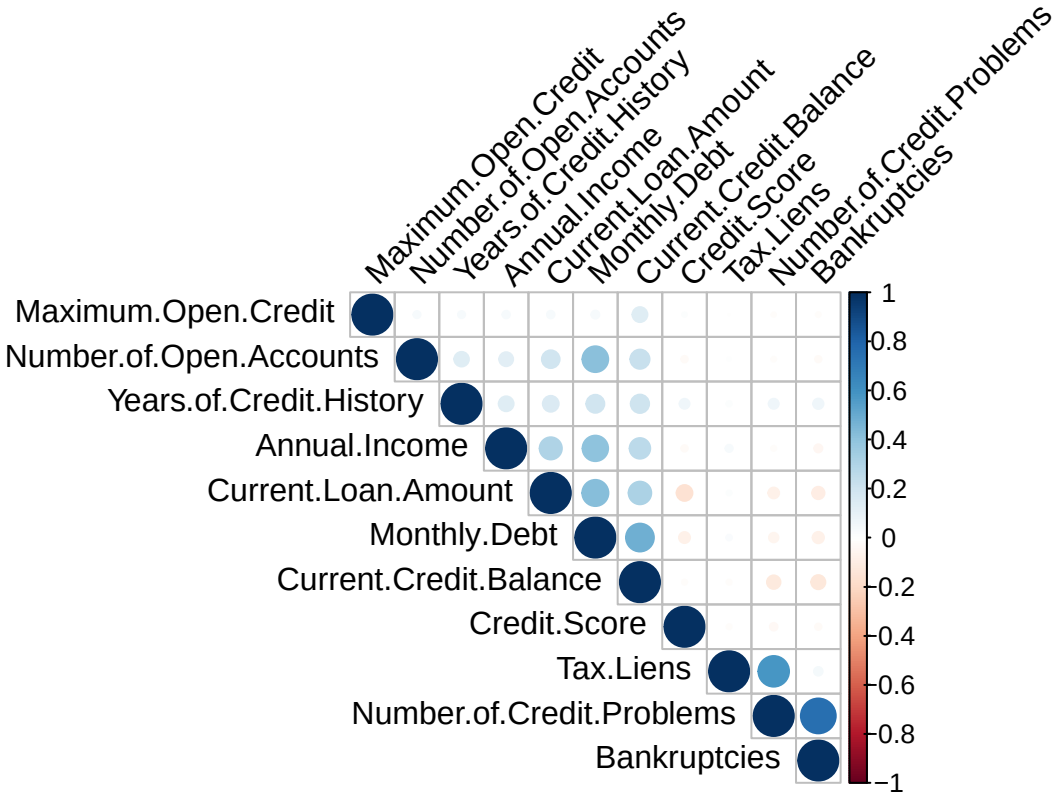
This is our predictor variable Loan Status. It has two categories, Fully Paid which '1' and Charged Off which is '0'. We can notice that count of Fully paid is at 80,000, whereas count of Charged off is only at 20,000. This could lead to our predictive model being biased towards fully paid.



This is a plot of Credit Score against Annual Income. We would expect that these two variables would have a strong relation, because generally, customers with high income would have high credit score. But this is not the case here. These two variables do not seem to have a strong relation between them.



This is a plot of Number of credit problems against Monthly Debt. We would expect that customers with high monthly debt would have more number of credit problems. But that is not the case here. These two variables do not display a strong relationship either.



The correlation between Number of credit problems and bankruptcies is very high at 0.75. This makes sense because people with a lot of credit problems eventually get bankrupt. The correlation between Number of credit problems and tax liens is also quite high at 0.58.This also makes sense because lien is imposed on failure of payment, and a person with lot of credit problems is most likely to not pay the loan amount. Bankruptcies and tax liens should not be added to our model because this could lead to multicollinearity.

SUPERVISED MACHINE LEARNING

1. Decision Tree
    - Using RPART

```
##           Actual
## Predicted     0     1
##         0     0     0
##         1  6776 23162
```

```
## [1] 0.774
```

Accuracy of the model is 77.4%. Even though the accuracy of the model is good, it has correctly predicted the Charged off class 0 times.

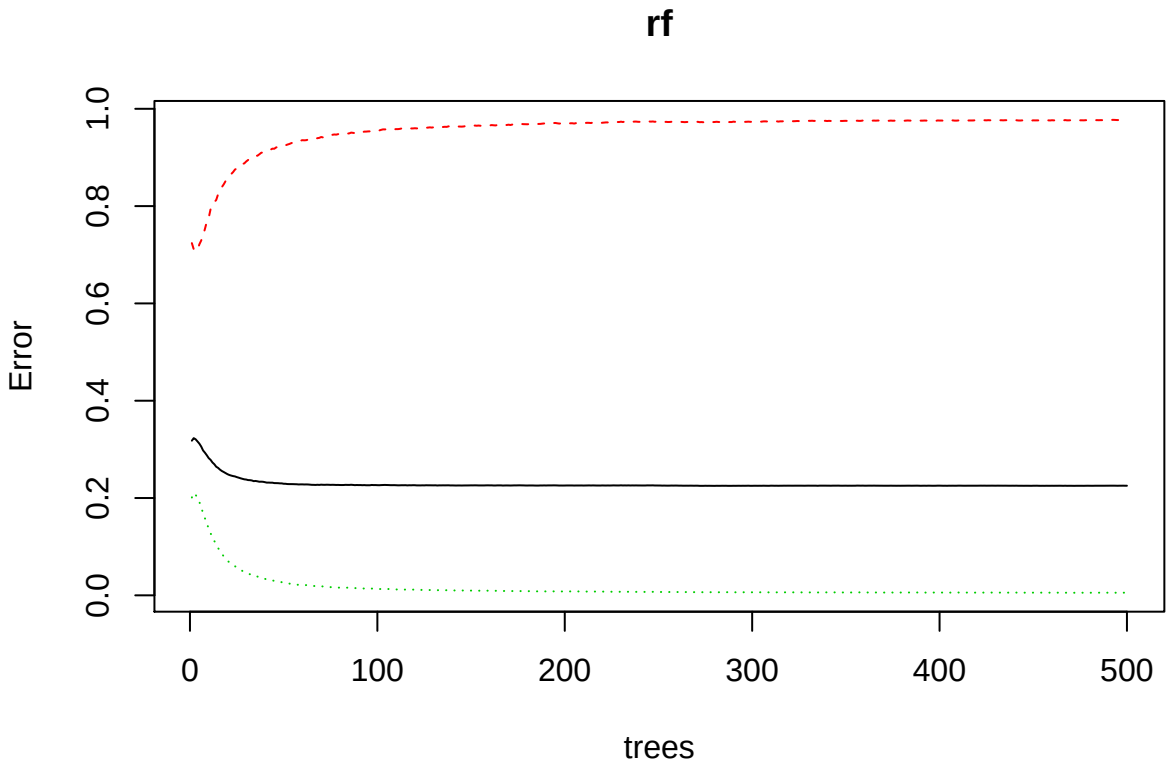
1. Decision Tree
    - Using CTREE

```
##          Actual
## Predicted     0     1
##         0    50    50
##         1  6726 23112
```

```
## [1] 0.774
```

Accuracy of the model is 77.4%. The model has correctly predicted the Charged off class 50 times.

2. Random Forest
    - Before Tuning

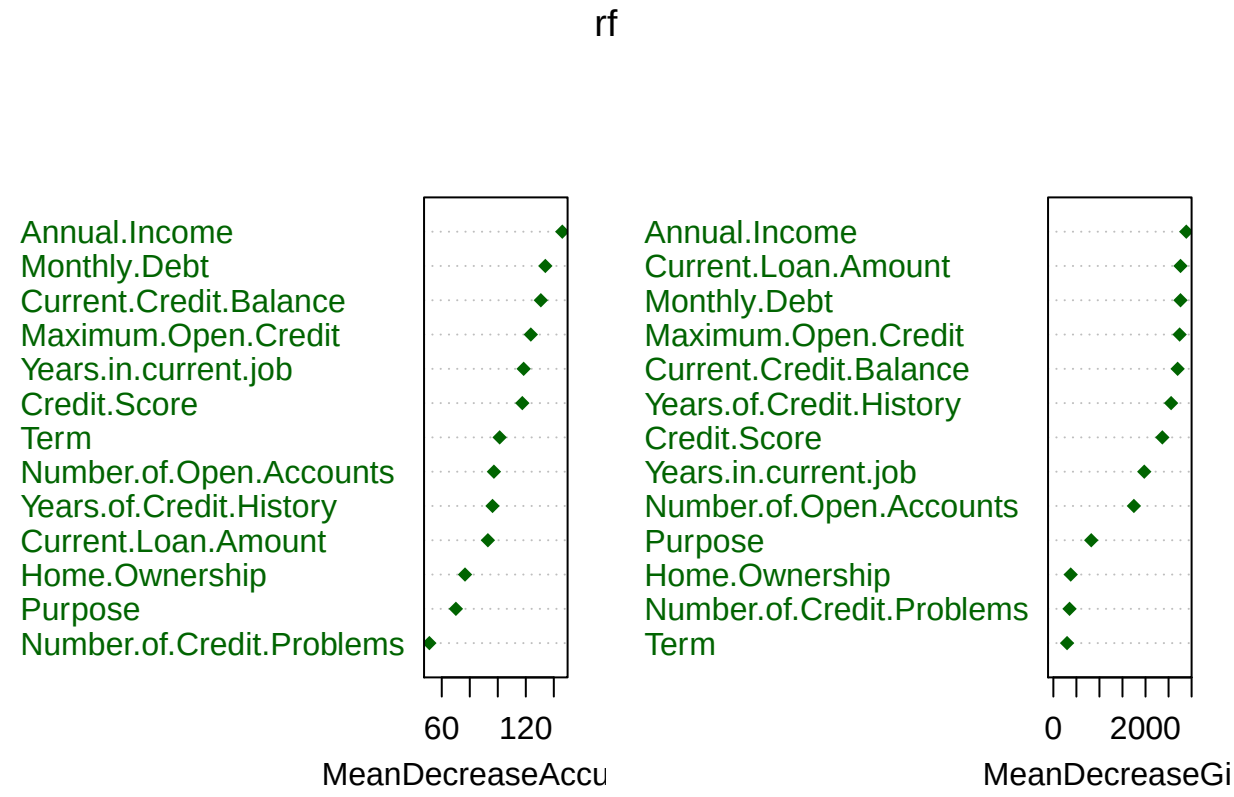


Here the black line is the Out Of Bag error, red is the error in prediciting '0' class, and green is error in prediciting '1' class. We can notice that at 50 trees the error is minimum.

```
##          Actual
## Predicted     0     1
##         0   142   104
##         1  6634 23058
```

```
## [1] 0.775
```

Accuracy of the model is 77.5%. The model has correctly predicted Charged off class 142 times.



From the variable importance plot, Annual Income, Monthly Debt, Maximum Open Credit, Current Credit Balance, and Credit Score seem like important variables for predicition.

2. Random Forest
    - After Tuning

```
##           Actual
## Predicted     0     1
##         0   331   480
##         1  6445 22682
```

```
## [1] 0.769
```

The accuracy of the model is 76.9%. Even though the accuracy of this model is lower than previous models, this model correctly predicted 331 instances as '0' or Charged Off, which is much greater than our previous models.

3. Naives Bayes Classifier

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction     0     1
##          0  1443  4151
```

```
##          1  5333 19011
##
##                Accuracy : 0.683
##                  95% CI : (0.678, 0.688)
##     No Information Rate : 0.774
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.036
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.2130
##             Specificity : 0.8208
##          Pos Pred Value : 0.2580
##          Neg Pred Value : 0.7809
##              Prevalence : 0.2263
##          Detection Rate : 0.0482
##    Detection Prevalence : 0.1869
##       Balanced Accuracy : 0.5169
##
##        'Positive' Class : 0
##
```

The accuracy of this model is 68.3 which by far the lowest. The model has correctly predicted the Charged off class 1443 times, which is an immense improvement from the previous models.

4. Neural Networks

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0     0     0
##          1  6776 23162
##
##                Accuracy : 0.774
##                  95% CI : (0.769, 0.778)
##     No Information Rate : 0.774
##     P-Value [Acc > NIR] : 0.503
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.000
##             Specificity : 1.000
##          Pos Pred Value :   NaN
##          Neg Pred Value : 0.774
##              Prevalence : 0.226
##          Detection Rate : 0.000
##    Detection Prevalence : 0.000
##       Balanced Accuracy : 0.500
##
```

```
##        'Positive' Class : 0
##
```

The accuracy of this model is 77.4%. The model has correctly predicted Charged off class 0 times, same as the Decision Tree model.

CONCLUSION

The selection of model is solely based on the task at hand. For our case, it is riskier for banks to give loans to customers who would not repay it rather than not give loans to customers who would repay it. So, it is important for banks to predict the customers whose loans were charged off rather than the customers that fully paid their loans. Based on this theory, the Naive Bayes Classification model would be the best model for the banks.

REFERENCES

https://www.usdebtclock.org/world-debt-clock.html

https://www.kaggle.com/zaurbegiev/my-dataset

https://cran.r-project.org/web/packages/naniar/vignettes/naniar-visualisation.html

https://cran.r-project.org/web/packages/mice/mice.pdf

https://www.tutorialspoint.com/r/r_decision_tree.htm

https://towardsdatascience.com/random-forest-in-r-f66adf80ec9

https://www.edureka.co/blog/naive-bayes-in-r/

https://towardsdatascience.com/build-your-own-neural-network-classifier-in-r-b7f1f183261d