CREDIT CARD DEFAULT PREDICTION

Group Project

Submitted for

BUSINESS ANALYTICS WITH R

Prof: Shujing Sun

By

Swapnil Deshpande (SXD220134)

NAVEEN JINDAL SCHOOL OF MANAGEMENT

THE UNIVERSITY OF TEXAS AT DALLAS

Richardson TX – 75080

APRIL 2023

Table of Contents

Introduction

Our team has chosen a dataset with over 1 million records that contain information on credit card approval. The dataset includes details on various socio-economic parameters such as income group, profession, and assets owned, which are responsible for approving or rejecting a credit card for an applicant. By assigning a credit score based on the information provided, the project classifies each applicant as a good or bad client based on their socio-economic profile. The risk profile and credit score of an individual are crucial in quantifying the magnitude of risk and the likelihood of timely payments using common risk control methods. The main goal of this project is to create a machine learning model using different analysis matrices to predict potential applicants. However, dealing with imbalanced and improper data may present challenges that can be addressed through data cleaning methodologies. The dataset will be visualized to understand how a specific parameter influences credit card approval and what weightage it should have to validate the machine learning model with the highest precision.

The credit score and risk profile of an individual are essential in assessing their creditworthiness and the probability of timely payments. This project aims to develop a machine learning model that predicts a customer's eligibility for a credit card based on their socio-economic profile and credit history.

Banks face various risks that could affect their returns, and credit risk is one of them. Efficient management of these risks is crucial for a bank's success, and one way to mitigate nonperforming assets is by monitoring customer payments regularly. To reduce the risk of nonperforming assets, banks must first assess the creditworthiness of an applicant before granting a loan. This study uses machine learning techniques to automate the process of assessing an applicant's creditworthiness and eligibility for a credit card, thereby eliminating the need for manual processes and paperwork. The study employs different supervised machine learning methods and validation techniques to develop accurate prediction models. The goal is to help credit card companies make smarter and more rational decisions by accelerating the credit approval process based on a customer's past credit history and economic profile. The study also aims to identify the relevant variables for the approval process and optimize decision-making using random forest classification.

Assumptions:

1. The model is developed solely based on the dataset provided and may not perform well on other datasets.
2. Customers who have defaulted on payments for 60 days or more are classified as bad customers.

Limitations:

1. The dataset is from pre-COVID times and does not consider post-COVID scenarios.
2. The reliability of the dataset is uncertain as it has been obtained from Kaggle without any confirmation of its authenticity.

Exploratory Data Analysis

The initial phase of developing the prediction model involves examining the data present in the credit card approval dataset, which was obtained from the Kaggle Open-Source Dataset Site. This dataset comprises 1,048,575 entries of individuals who applied for a credit card and includes various characteristics about them. The following table explains the features of the dataset. The dataset can be found in the link below :

https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction?resource=download

Feature name	Explanation	Data Type	Description/Remarks
ID	Client number	Numerical	
CODE_GENDER	Gender	Categorical	F,M
FLAG_OWN_CAR	Is there a car	Categorical	N,Y
FLAG_OWN_REALTY	Is there a property	Categorical	N.Y
CNT_CHILDREN	Number of children	Integer	
AMT_INCOME_TOTAL	Annual income	Float	
NAME_INCOME_TYPE	Income category	Categorical	Commercial associate Pensioner, State servant Student ,Working
NAME_EDUCATION_TYPE	Education level	Categorical	Academic degree, Higher education, Incomplete higher, Lower secondary, Secondary / secondary special
NAME_FAMILY_STATUS	Marital status	Categorical	Civil marriage, Married, Separated, Widow Single / not married,

NAME_HOUSING_TYPE	Way of living	Categorical	Co-op apartment, House / apartment, Municipal apartment, Office apartment Rented apartment, With parents
DAYS_BIRTH	Birthday	Integer	Count backwards from current day (0), -1 means yesterday
DAYS_EMPLOYED	Start date of employment	Integer	Count backwards from current day(0). If positive, it means the person currently unemployed.
FLAG_MOBIL	Is there a mobile phone	Integer	1,0
FLAG_WORK_PHONE	Is there a work phone	Integer	1,0
FLAG_PHONE	Is there a phone	Integer	1,0
FLAG_EMAIL	Is there an email	Integer	1,0
OCCUPATION_TYPE	Occupation	Categorical	Multiple Values
CNT_FAM_MEMBERS	Family size	Float	
ID	Client number	Integer	
MONTHS_BALANCE	Record month	Integer	The month of the extracted data is the starting point, backwards, 0 is the current
STATUS	Status	Categorical	0: 1-29 days past due 1: 3059 days past due 2: 60-89 days overdue 3: 90-119 days overdue 4:120-149 days overdue 5: Overdue or bad debts, write-offs for more than 150 days C: paid off that month X: No loan for the month

The chosen dataset comprises a total of 19 variables, which are categorized into 9 numerical and 9 categorical variables, as displayed in the tables above.

Missing Values:

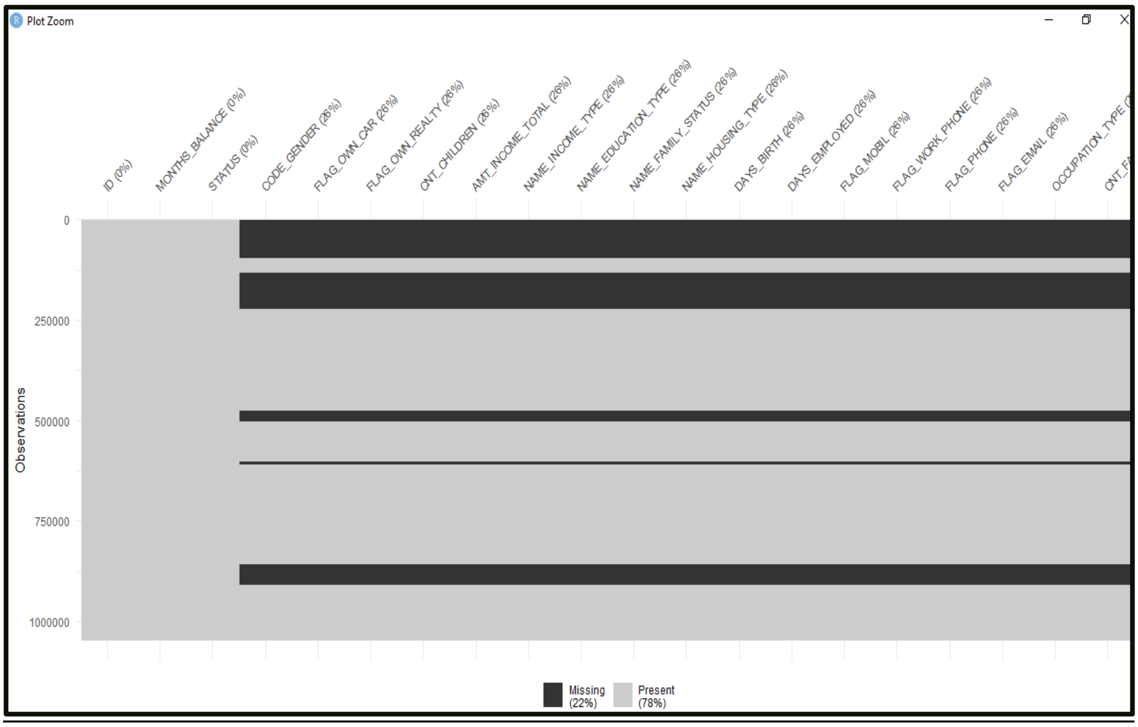


Figure 1 : Missing values

From the above graph we can infer that 22% of all the variables in the dataset have missing values. The reason could be regular or abnormal. We must ensure to consider the missing values to improve the accuracy of the model we are building. Therefore, we have substituted the missing values with the median value of the existing data points. This can be inferred from the graph below after treating the dataset.

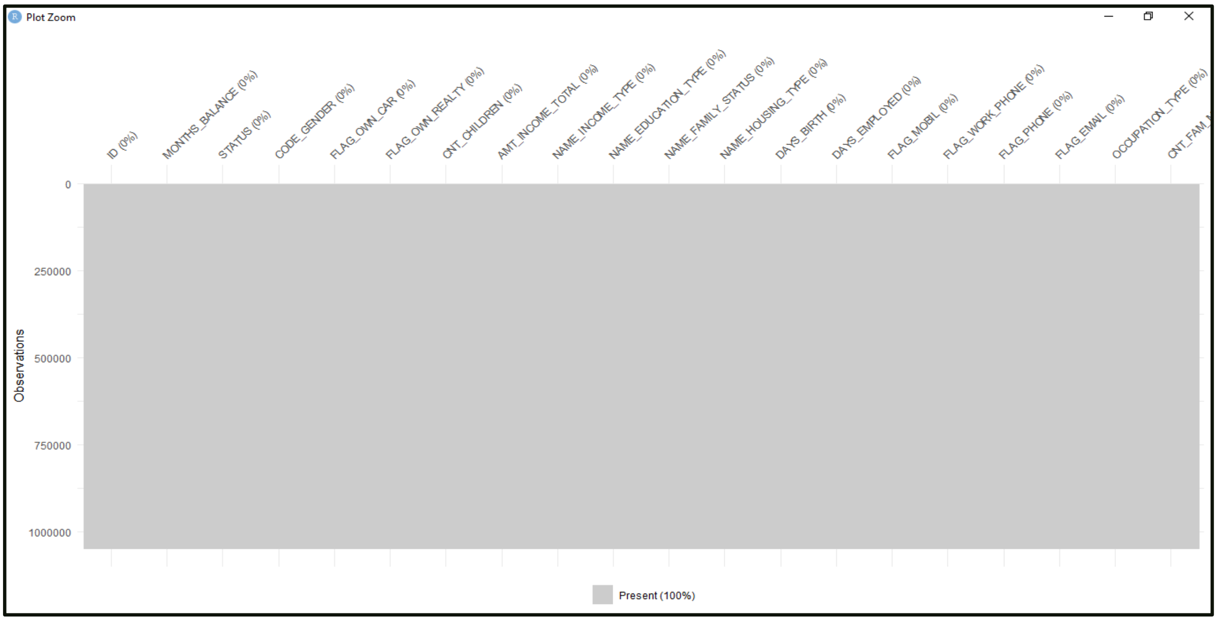
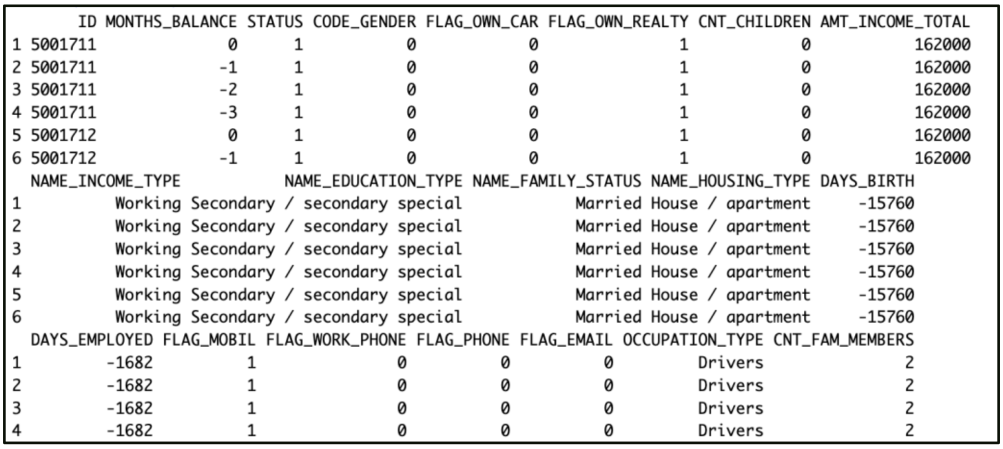


Figure 2 : No missing values

From the above screenshot we can infer now that there are no missing values.

One hot Encoding

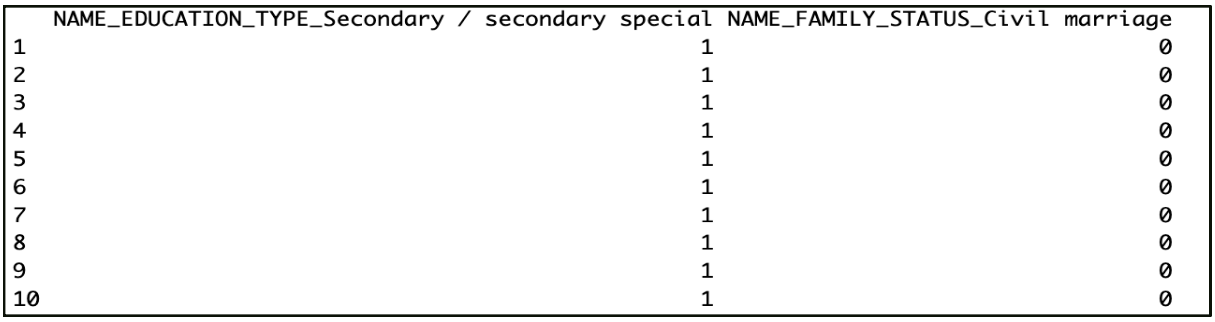
One hot encoding transforms categorical data into numerical - it transforms strings into numbers so that we can apply our Machine Learning algorithms without any problems.



```
       ID MONTHS_BALANCE STATUS CODE_GENDER FLAG_OWN_CAR FLAG_OWN_REALTY CNT_CHILDREN AMT_INCOME_TOTAL
1 5001711              0      1           0            0               1            0           162000
2 5001711             -1      1           0            0               1            0           162000
3 5001711             -2      1           0            0               1            0           162000
4 5001711             -3      1           0            0               1            0           162000
5 5001712              0      1           0            0               1            0           162000
6 5001712             -1      1           0            0               1            0           162000
  NAME_INCOME_TYPE           NAME_EDUCATION_TYPE NAME_FAMILY_STATUS NAME_HOUSING_TYPE DAYS_BIRTH
1          Working Secondary / secondary special            Married House / apartment     -15760
2          Working Secondary / secondary special            Married House / apartment     -15760
3          Working Secondary / secondary special            Married House / apartment     -15760
4          Working Secondary / secondary special            Married House / apartment     -15760
5          Working Secondary / secondary special            Married House / apartment     -15760
6          Working Secondary / secondary special            Married House / apartment     -15760
  DAYS_EMPLOYED FLAG_MOBIL FLAG_WORK_PHONE FLAG_PHONE FLAG_EMAIL OCCUPATION_TYPE CNT_FAM_MEMBERS
1         -1682          1               0          0          0         Drivers               2
2         -1682          1               0          0          0         Drivers               2
3         -1682          1               0          0          0         Drivers               2
4         -1682          1               0          0          0         Drivers               2
```

Figure 3: One hot encoding

Created dummy columns for NAME_INCOME_TYPE, NAME_EDUCATION_TYPE, NAME_FAMILY_STATUS, NAME_HOUSING_TYPE, OCCUPATION_TYPE using the fast Dummies package. This is done for columns with more than two categories. After the dummy columns are created, the original columns are deleted.



```
   NAME_EDUCATION_TYPE_Secondary / secondary special NAME_FAMILY_STATUS_Civil marriage
1                                                  1                                 0
2                                                  1                                 0
3                                                  1                                 0
4                                                  1                                 0
5                                                  1                                 0
6                                                  1                                 0
7                                                  1                                 0
8                                                  1                                 0
9                                                  1                                 0
10                                                 1                                 0
```

Figure 4 : Dummy columns

Data Visualization

Visualization 1:

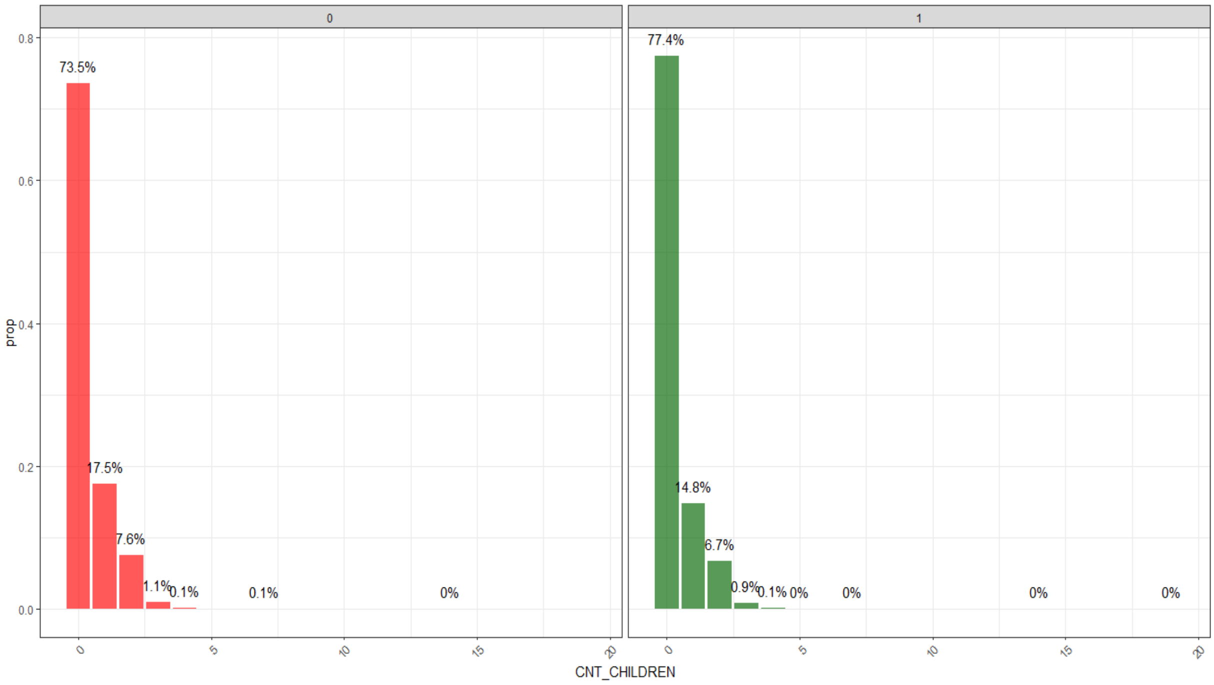


Figure 5 : Bar graph 1

The graph shows the percentage of defaulting and non-defaulting customers with respect to their number of children. Upon inspection, the data between the two graphs seems to be similarly distributed, with both showing a peak of individuals with no children.

However, as we explore the data further, we can observe that there is a slight increase in the number of defaulters as the number of children increases. This could be attributed to the added financial burden of having children, such as increased expenses for education, childcare, and healthcare.

It is important to consider the impact of family size when evaluating credit risk, as it is a factor that can greatly affect an individual's financial situation. By considering the number of children a person has, lenders can better assess their ability to pay back loans and avoid default.

Visualization 2:

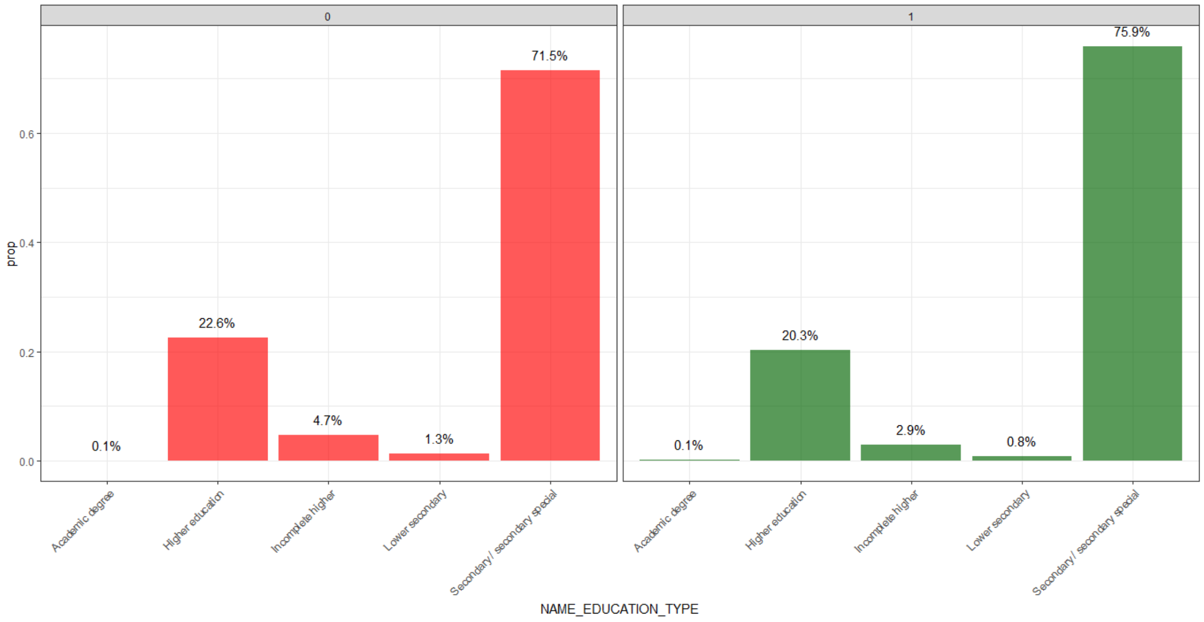


Figure 6 : Bar graph 2

The graph shows the percentage of defaulting and non-defaulting customers with respect to their level of education. It provides valuable insights into the credit default patterns for individuals based on their level of education.

Upon analysis of the graph, we can see that most people in the dataset have attained a secondary or Secondary special level of education, with higher education being the second most common level of education. The percentage of customers are also displayed for each education level in the defaulting and non-defaulting category.

On the other hand, individuals with only a basic or lower-secondary education level have the highest rate of default payments, which may be an indication of their lower income and financial stability.

By examining this graph, we can infer that education level has a significant impact on creditworthiness, and lenders should take this into account when assessing an individual's creditworthiness. This information can be used to design more accurate credit scoring models that take education level into account, which can improve credit risk assessment accuracy and help mitigate the risks associated with default payments.

Visualization 3:

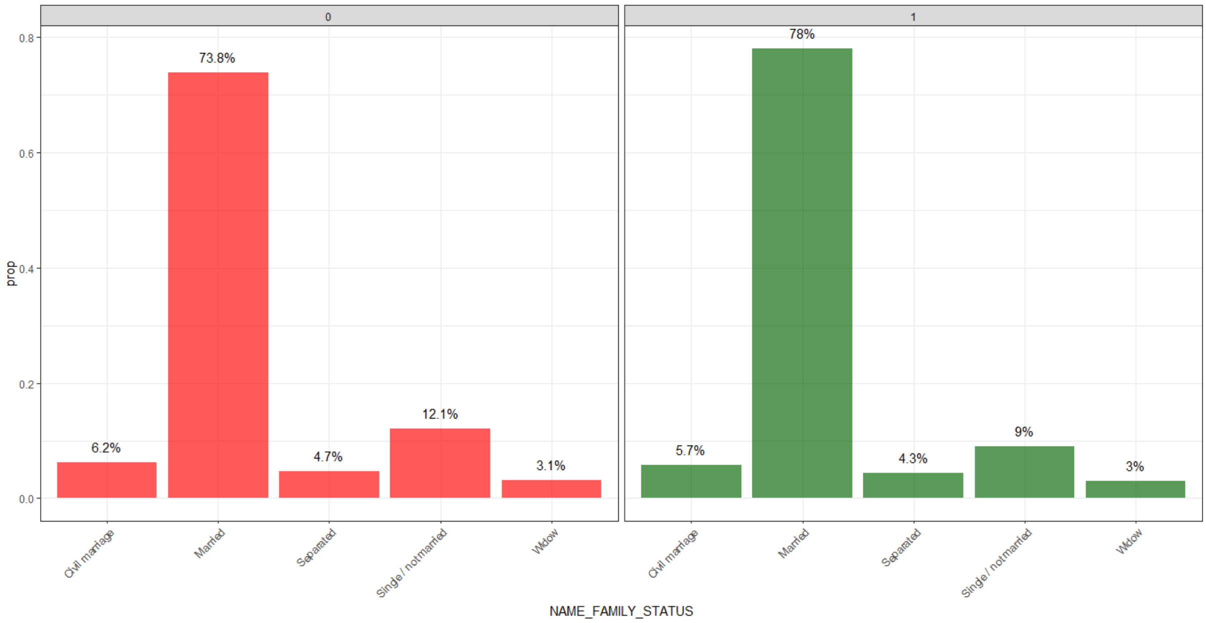


Figure 7: Bar graph 3

The analysis of the credit card default dataset can provide valuable insights into the factors that affect the timely payments of credit card holders. One of the key variables in this dataset is the family status of the credit card holder, which can have a significant impact on their payment behavior. The data has been visualized using a graph that displays the rate of default and timely payments with respect to family status.

Upon observation of the graph, we can infer that the highest number of people who own a credit card belong to the married family status category. This could be due to the fact that married individuals may have more stable sources of income and may be able to manage their finances better as compared to single individuals. However, it is important to note that this is just an inference and further analysis is required to validate this claim.

Another key observation from the graph is that people who are single have a slightly higher percentage of defaulting on their credit card payments as compared to those who are married. This could be attributed to the fact that single individuals may have fewer sources of income and may be unable to manage their expenses and credit card payments effectively. However, this is also just an inference and further analysis is required to understand the underlying factors that lead to this observation.

Overall, the family status variable appears to be an important factor that affects credit card payment behaviour. Further analysis could be carried out to explore the relationship between family status and other variables in the dataset to gain more insights into the credit card payment behaviour of different family status categories.

Visualization 4:

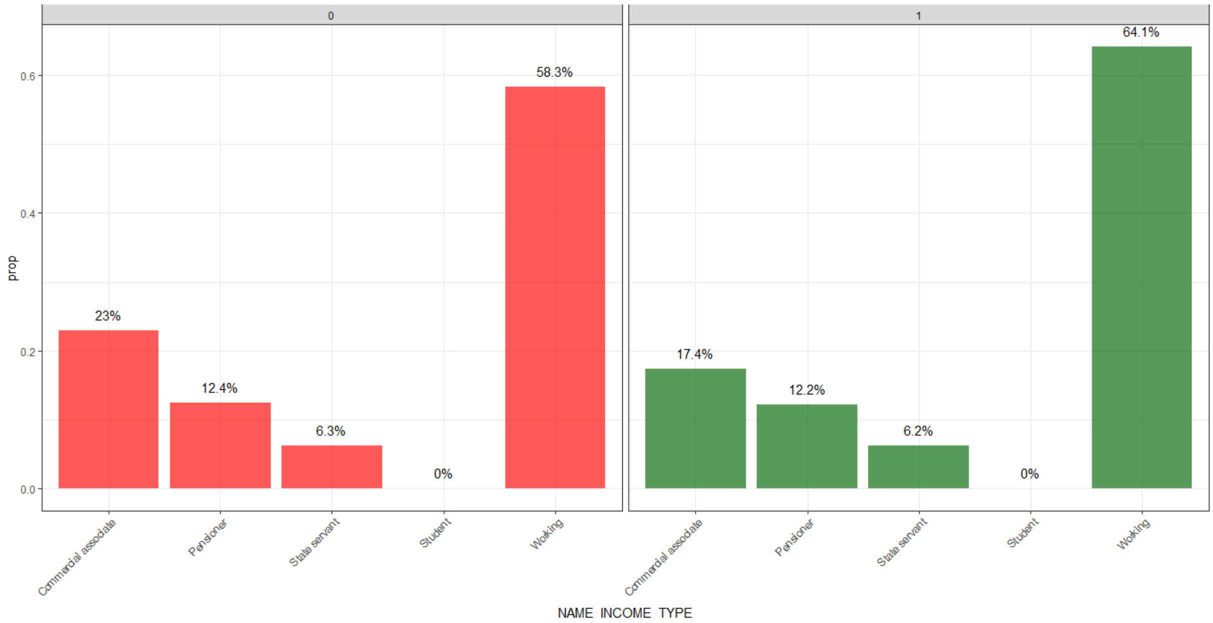


Figure 8 : Bar graph 4

The Income Status graph displays the distribution of credit card ownership and payment behaviour based on different income statuses. It reveals that the majority of credit card owners fall under the working category, followed by commercial associates. Interestingly, individuals who work appear to have a lower rate of defaults compared to commercial associates.

The graph's findings align with the traditional understanding that the working population typically has a more stable source of income than other categories. As a result, they may have a greater ability to make timely credit card payments. This data may be helpful to credit card companies when evaluating the risk associated with different income statuses.

The graph also indicates that commercial associates have a higher probability of defaulting on credit card payments. This trend may be a result of commercial associates facing more financial risk or instability compared to the working population. Additionally, the nature of their work may expose them to more credit-related risks.

Overall, the Income Status graph provides valuable insights into the relationship between income status and credit card payment behaviour. It highlights the importance of considering income status when assessing credit risk and may help credit card companies design more effective risk management strategies.

Visualization 5:

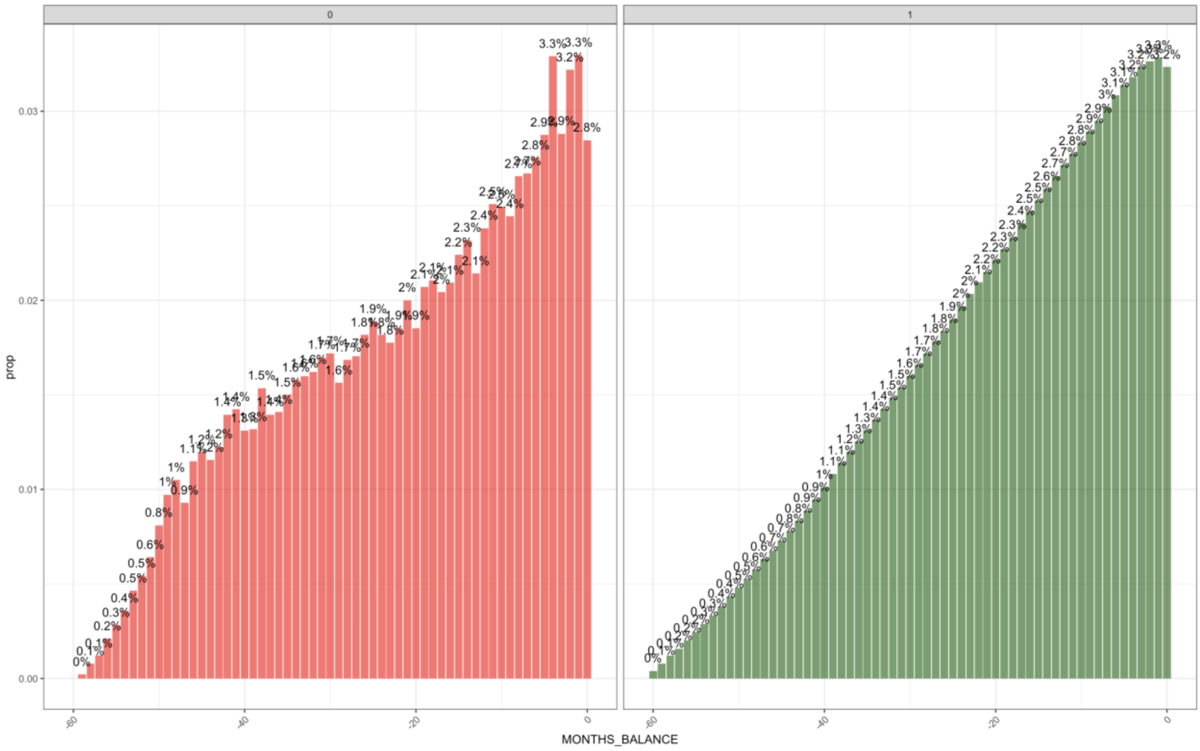


Figure 9 : Bar graph 5

The two graphs provided showcase an interesting insight into the distribution of credit card holders who defaulted and those who did not. The graph on the right illustrates a bell curve distribution, which is typically associated with normal distribution, whereas the graph on the left highlights the data distribution of credit card holders who defaulted.

Based on the above insights, it is essential to implement more sophisticated techniques that can handle non-linear relationships and provide a better prediction accuracy for credit card default. Therefore, we plan to utilize the random forest algorithm, a powerful and widely used ensemble-based classification algorithm that can handle complex, non-linear relationships between variables. With its ability to combine multiple decision trees and its capability to handle imbalanced data, we are confident that random forest will help us achieve better prediction accuracy.

Data Modeling

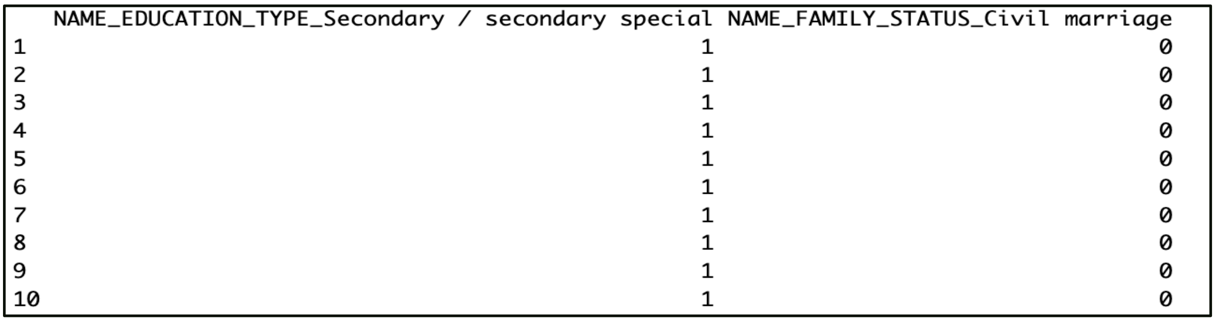
The objective was to predict credit card approval status based on the results obtained using the supervised learning models. To achieve this, we followed a series of steps.

Merging the dataset - We started with two CSV files, namely ApplicationRecord.csv and CreditCard.csv. These files were extracted into two separate data frames. Next, we merged the two data frames using the merge command or by using an inner join operation to form a single dataset. We considered all the columns present in the dataset while building the model. However, we used class weights to deal with class imbalance, which occurs when the number of observations in one class is significantly different from that in the other class. In such cases, we assign higher weights to the minority class during the training process to ensure that the model can learn to predict the minority class accurately.

Finding out the missing Values - To identify missing values in a large dataset, we utilized the Nanair package, which provides an effective solution for handling missing data. This package includes libraries that allow for graphical representation of the missing values, and can also indicate if the dataset is large. By applying the vis_miss method in the Nanair package, we were able to determine that 22% of the dataset contained missing values. This allowed us to proceed with the necessary steps to address and resolve any issues related to the missing data.

Imputation **-** To ensure that our model was not biased and could perform at its best, we needed to impute the missing values in the dataset. We decided to use the median imputation method, as it would be more robust to the presence of outliers, as opposed to mean or interquartile range (IQR) imputation methods. The median value is not affected by the extreme values in the dataset and is thus a better choice for imputation. This step allowed us to ensure that the dataset was complete and ready for use in training and evaluating the model.

Encoding The categorical Variables **–** Since our dataset mainly comprised non-ordinal data, we needed to create dummy variables to convert the categorical data into a format suitable for machine learning models. As the data had different levels, it was difficult to decide which level was higher or lower, and hence we opted to encode the non-ordinal data using the Fast Dummies package. We utilized the dummy_cols method in this package to create binary columns for each level of the categorical variables. However, this encoding process resulted in a dataset with almost 59 features, which significantly increased the risk of overfitting. To illustrate this, we have included an image below which shows how the Column NAME_INCOME_TYPE was broken down into factors with values of 0 or 1.



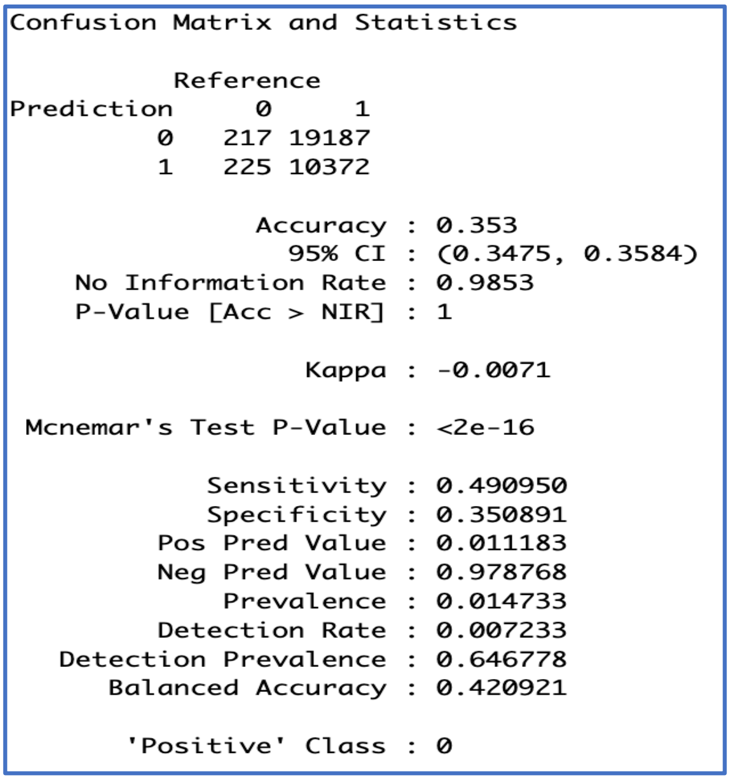
```
   NAME_EDUCATION_TYPE_Secondary / secondary special NAME_FAMILY_STATUS_Civil marriage
1                                                  1                                 0
2                                                  1                                 0
3                                                  1                                 0
4                                                  1                                 0
5                                                  1                                 0
6                                                  1                                 0
7                                                  1                                 0
8                                                  1                                 0
9                                                  1                                 0
10                                                 1                                 0
```

Figure 10: Encoding categorical variables

Sampling - As the data set was quite bulky, we chose a random sample of 100k records and created a training data set comprising of 70% observations and testing data of the remaining observations. We ran three supervised learning models namely Logistic Regression, Decision Tree and Random Forest Classifier on the training and testing data set.

Model 1 – Logistic Regression:

As per the obtained metrics, the accuracy of the model was only 0.353, indicating that the model was not performing well in predicting the dependent variable's outcome. However, since the dataset was highly imbalanced, our focus was not on accuracy. Instead, we applied class weights to the model to correct it and focused on the sensitivity metric, which was quite low at 0.49. The specificity metric was also low at 0.35. Hence, there was significant room for improvement in the model's performance. After running the logistic regression, we received a set of metrics, which are displayed in the attached screenshot below.

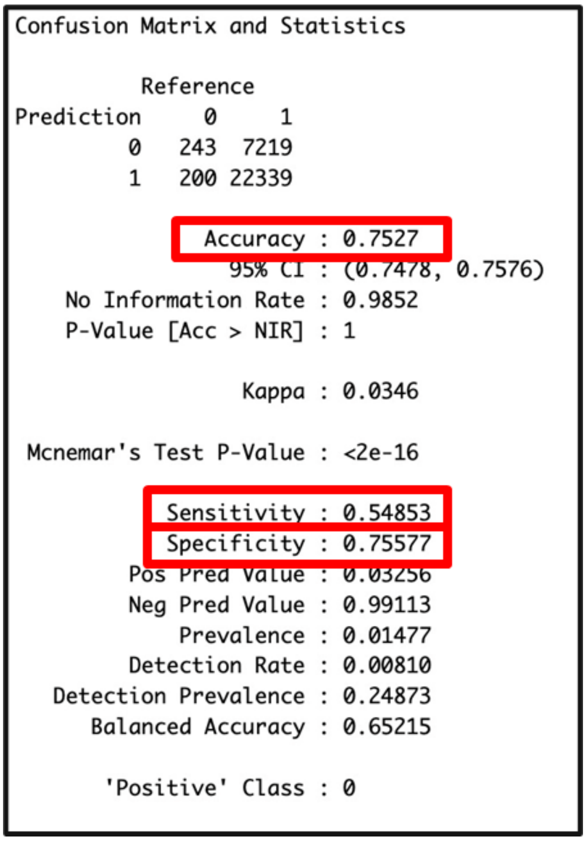


```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0   217 19187
         1   225 10372

               Accuracy : 0.353
                 95% CI : (0.3475, 0.3584)
    No Information Rate : 0.9853
    P-Value [Acc > NIR] : 1

                  Kappa : -0.0071

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.490950
            Specificity : 0.350891
         Pos Pred Value : 0.011183
         Neg Pred Value : 0.978768
             Prevalence : 0.014733
         Detection Rate : 0.007233
   Detection Prevalence : 0.646778
      Balanced Accuracy : 0.420921

       'Positive' Class : 0
```

Figure 11: logistic regression - confusion matrix

Model 2 – Decision Tree:

We then applied decision tree model to the same data set. We could infer that the accuracy improved from 0.353 to 0.75. However, the sensitivity was still quite low at 0.54, indicating that the model was not performing well in predicting the true positives. The specificity increased from 0.35 to 0.75, which was a positive outcome. After running the decision tree model, we received a set of metrics, which are displayed in the attached screenshot below.



```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0   243  7219
         1   200 22339

               Accuracy : 0.7527
                 95% CI : (0.7478, 0.7576)
    No Information Rate : 0.9852
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0346

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.54853
            Specificity : 0.75577
         Pos Pred Value : 0.03256
         Neg Pred Value : 0.99113
             Prevalence : 0.01477
         Detection Rate : 0.00810
   Detection Prevalence : 0.24873
      Balanced Accuracy : 0.65215

       'Positive' Class : 0
```

Figure 12: Decision Tree - confusion matrix

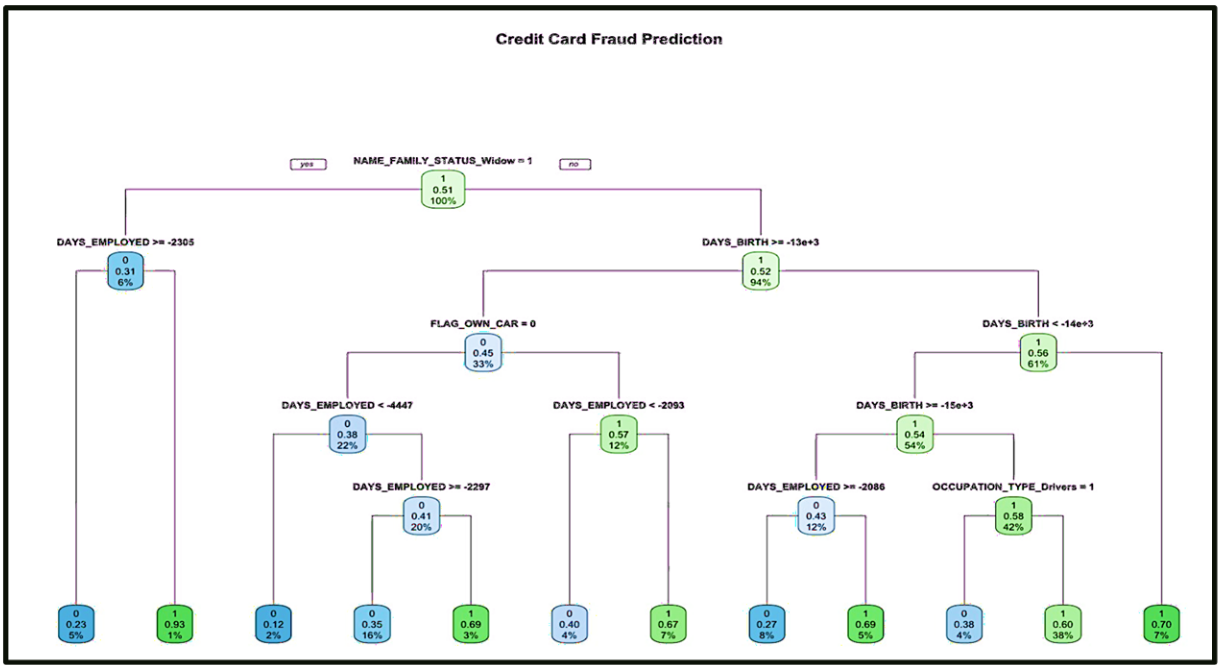


Figure 13: Decision Tree

Model 3 – Random Forest Classifier:

Identifying Significant Predictors – We implemented Random Forest Classifier to determine the number of significant predictors. Random forest algorithm provides a capability to visualize the number of significant predictors in the decreasing order of mean accuracy making the process easier. After implementing the algorithm, we got the following graph.

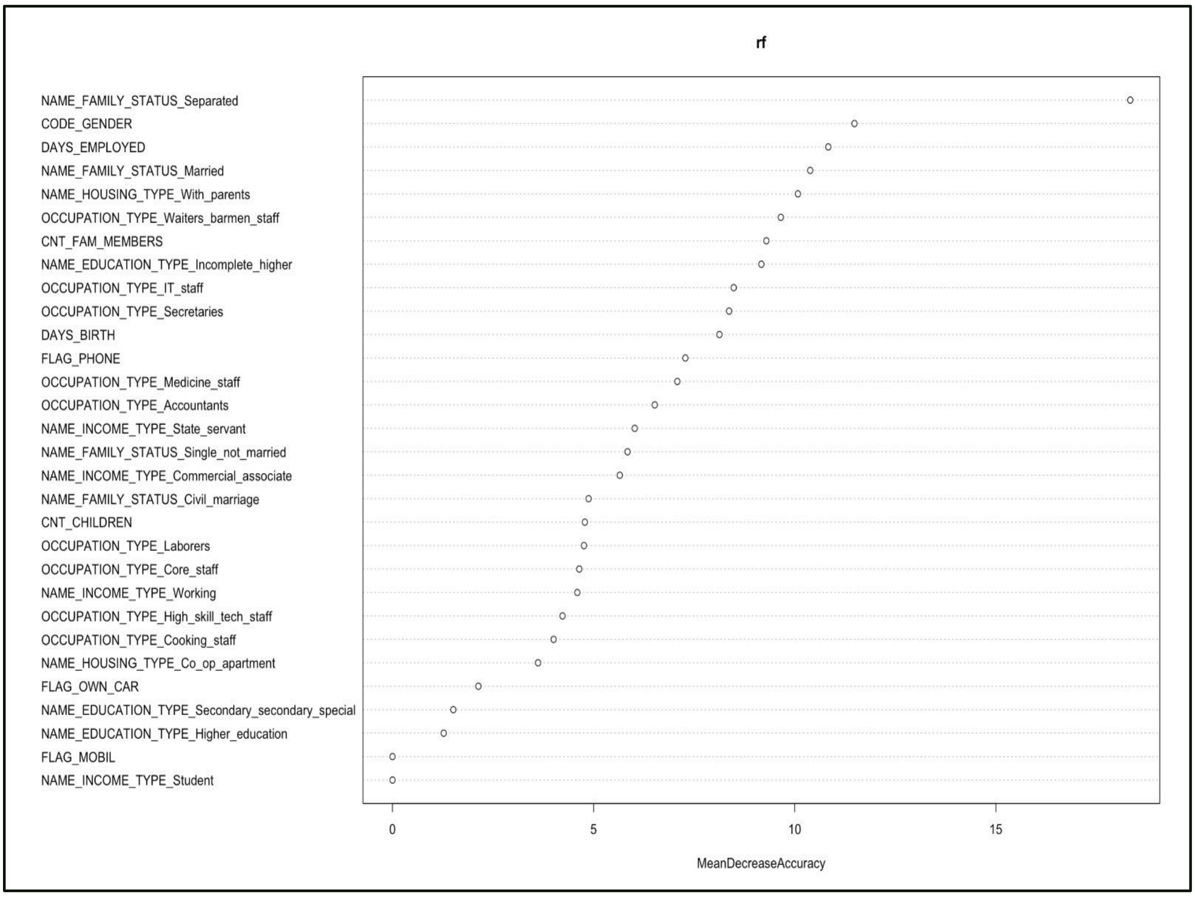
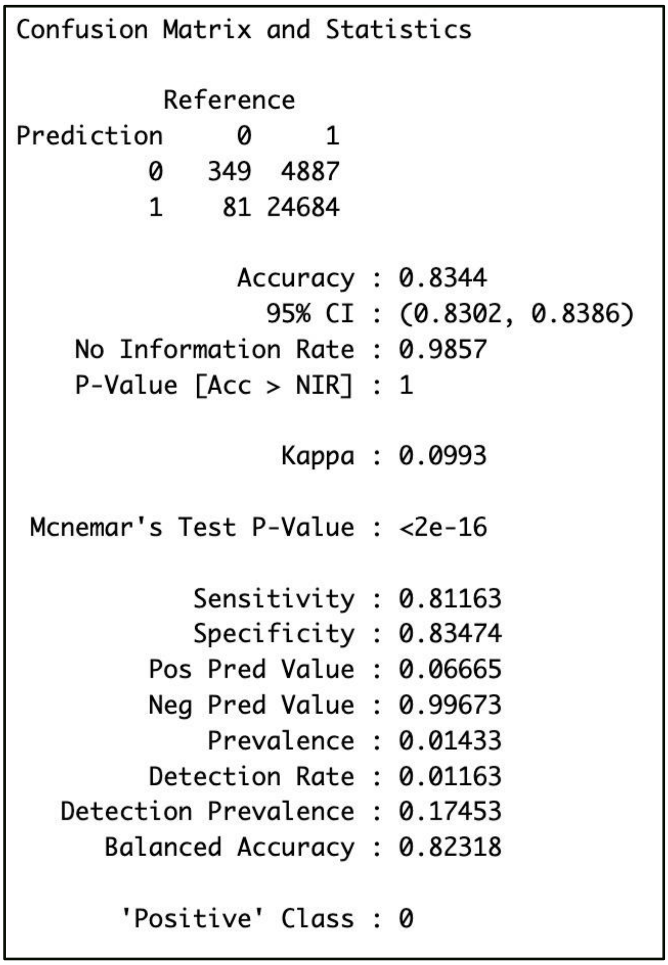


Figure 14: Random forest classifier - significant features

We ran the Random Forest Classifier as it is a more sophisticated non-linear bagging technique. Our main focus was on sensitivity, and we observed a significant improvement from 0.54 in the decision tree model to 0.81 in the Random Forest Classifier.

This improvement indicates that the Random Forest Classifier is better equipped to make accurate predictions. We also noted an improvement in specificity, from 0.75 in the decision tree to 0.83 in the Random Forest Classifier. Finally, the overall accuracy of the Random Forest Classifier improved to 0.83. These improvements demonstrate that the Random Forest Classifier is a superior model for predicting the outcome of the dependent variable. In the below screenshot all the metrics has been displayed



```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0   349  4887
         1    81 24684

               Accuracy : 0.8344
                 95% CI : (0.8302, 0.8386)
    No Information Rate : 0.9857
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0993

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.81163
            Specificity : 0.83474
         Pos Pred Value : 0.06665
         Neg Pred Value : 0.99673
             Prevalence : 0.01433
         Detection Rate : 0.01163
   Detection Prevalence : 0.17453
      Balanced Accuracy : 0.82318

       'Positive' Class : 0
```

Figure 15: random forest classifier - confusion matrix

Conclusion

In the process of building a credit card default prediction model to act as an insight to the credit card companies to approve or reject a customer, the following inferences were drawn. We selected a sample of 100K to train and validate various supervised learning models. The models chosen were Logistic regression, Decision Tree and Random Forest Classifier. The decision to choose the model was based on the highest sensitivity, which signifies the model's ability to predict higher defaulters vis a vis actual defaulters in the dataset.

After testing the various models, the Random Forest classifier was found to have the best sensitivity and accuracy. This model can be utilized to inform the issuer's decision on whom to give a credit card to and what credit limit to provide. Although models can aid in decision making to humans, but it cannot replace the human decisions in entirety.

Furthermore, the model has the potential to be enhanced with more data and computational resources, leading to even more accurate predictions. The development of such a model holds significant potential in reducing the risk of credit card defaults, thereby leading to more efficient and profitable business and minimizing bad loans.