GROUP NAME: FIVE FINGER DATA PUNCH (GROUP 13)

MEMBERS: RUVINYA SIRIWARDENA MAHANAMA, ASHWIN MANOHAR DESAI, SHWETA NEGI, ROSMIN ANN RAJU, SEPHY SABU

Loan Defaulter Prediction Based on Consumer Behaviour

INTRODUCTION

A loan is the principal source of revenue for the lenders (a bank in our case) and also a substantial source of financial risk. The amount received from the rate of return makes a large number of bank assets. Even though giving loans is beneficial to both parties, there is a degree of risk associated with it as well. These risks pertain to a borrower's inability to repay a loan by the agreed due date. A default occurs when a debtor fails to pay the amount of money borrowed from the lender within a fixed term and thus violates the contract between the debtor and lender. Therefore, the lenders such as banks and state governments face a huge risk of default when lending out money and other services.

Financial institutions rely on traditional procedures to evaluate whether or not a borrower is qualified for a loan. When there were a significant number of loan applications, manual techniques were usually effective, but they were insufficient. Deciding whether to lend a loan to a customer would take a long time. As a result, a model for default loan prediction can be used to assess a customer's risk behaviour. The model is designed to be used as a reference tool by the bank to aid in the decision-making process when granting loans in order to reduce the risk of default and increase profit.

The lenders usually compensate for this default risk by increasing the interest rate on borrowing. Lenders of financial services conduct a background check of the individual or organisation to whom they lend the financial service and then assign a credit rating to each individual or organisation. Business organisations are normally assigned a credit rating by rating agencies such as Standard & Poor's (S&P), Moody's, or Fitch. These credit ratings are used to determine the borrowers' risk of default.

The source data used for the project was obtained from Kaggle which includes historic customer behaviour based on observation. It consists of 252000 customers' details which are used to identify any consumer behavioural patterns that may be causes for default risk. The risk flag specifies whether or not a default has occurred in the past. As a team, we have found out what are the possible indicators of default risk for the Consumer Loan Product of a Bank in India and have developed a machine learning model that can predict whether a consumer would be a defaulter or not so that when the bank acquires new customers they can predict who is riskier and who is not.

DATASET:
https://www.kaggle.com/subhamjain/loan-prediction-based-on-customer-behavior

The dataset is explained below:

Columns	Description
income	Income of the user
age	Age of the user
experience	Professional experience of the user in years
profession	Profession
married	Whether married or single
house_ownership	Owned or rented or neither
car_ownership	Does the person own a car
risk_flag	Defaulted on a loan
current*job*years	Years of experience in the current job
current*house*years	Number of years in the current residence
city	City of residence
state	State of residence

METHODS

1. Exploratory Data Analysis

Exploratory Data Analysis is the process of analyzing the dataset to find patterns, discover anomalies and test hypothesis by various summary statistics and graphical representations.

We load in the dataset and first check the shape of the dataset that is check how many rows and columns in the dataset.
Next, we derive a summary of the dataset to identify data type, range of values, presence of null values for each of the variables in the dataset.
Thirdly, we determine whether a particular variable is numeric or categorical. Based on the unique levels in each variable, we found out that only one of the variables is numeric (Income variable) and the rest are categorical variables.

Finally, using the ggplot library, we created graphs to find insights from the dataset and understand the difference between the number of defaulters and non-defaulters for each of the feature in the dataset. For example, we wanted to understand how many of the defaulters were married or single and same for the non-defaulters.
In summary, below are the insights we derived about defaulters from the EDA:

- Most number of defaulters are likely to be single than married
- 94% of the defaulters lived in rented houses, 3.7% of the defaulters lived in owned houses and the remaining 2.3% live in other type of houses.
- Most of the defaulters did not own a car
- The top 5 professions where defaulters are found are police officer, software developer, air traffic controller, surveyor and physician.
- When comparing the age distribution of the defaulters, the highest number of defaulters are in the ages of 22, 33, 54, 66 and 76
- When comparing the work experience of the defaulters, the highest number of defaulters are in the years of 1, 2, 15 and 16 in terms of work experience
- The top 5 states in India where defaulters are found are Uttar Pradesh, West Bengal, Andhra Pradesh, Maharashtra and Bihar

2. Pre-processing

- Handling Null Values

  Null values are the values missing in the dataset. It is important to handle null values as the machine learning algorithm fails if there are null values present in the dataset. We need to determine whether the value missing is completely at random or the reason for the missing value is dependent on other attributes. Based on that there are several ways to handle null values such as deletion of row consisting null value, deleting a feature if it consists of ~75% null values, imputing the null values using mean, median or mode and using algorithms that handle null values.

  In our dataset the age column had just 9 null values therefore, the method we have used to handle the null values in our data is to impute the missing values with the mean of that particular feature. The standard deviation measured before handling null values was 17.06364 and the standard deviation measured after handling null values using mean imputation method was 17.06334. To conclude mean imputation method to handle null values was ideal as there was not much variation in the age feature after handling null values.

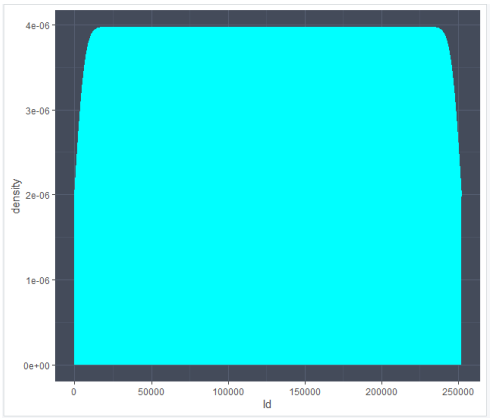
- Outlier Analysis

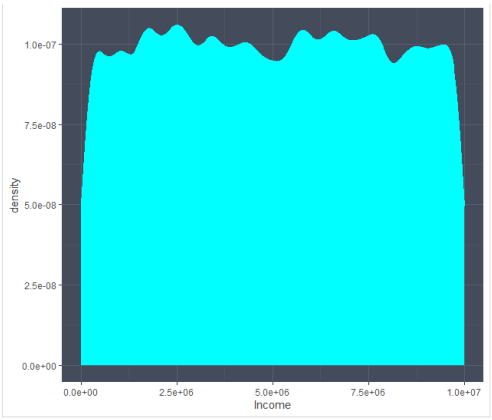
  In statistics, outliers are defined as observations which are significantly different or lie abnormally further away from other observations in the sample of data. It is important that outliers are handled so that the performance of machine learning algorithms or neural

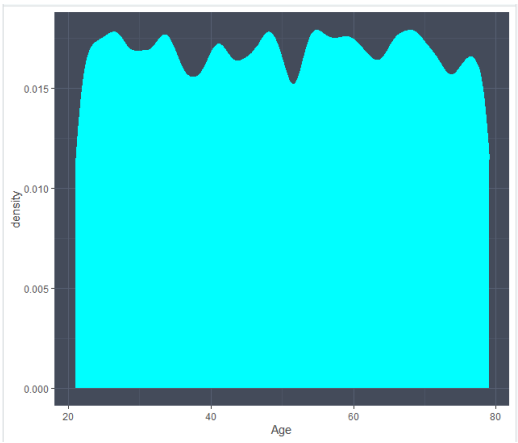
network is not impacted. Outliers slow the training process of algorithms particularly those involving gradient descent.

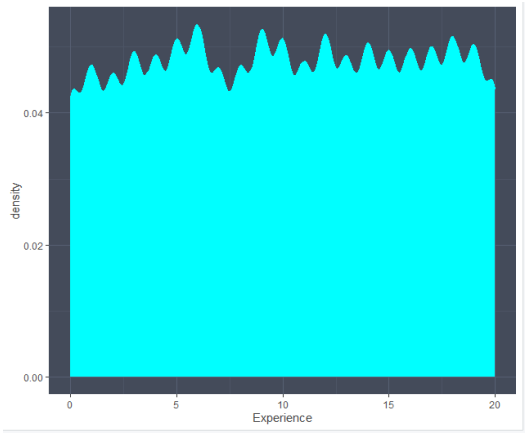
Using the ‘ggplot’ library in R, we have drawn density plots and boxplots of the numeric data to visualize the presence of outliers. Density plots tell us how the data is distributed and whether there is any skewness (left or right) in the dataset. Boxplots tell us how the data is distributed in terms of quartiles.

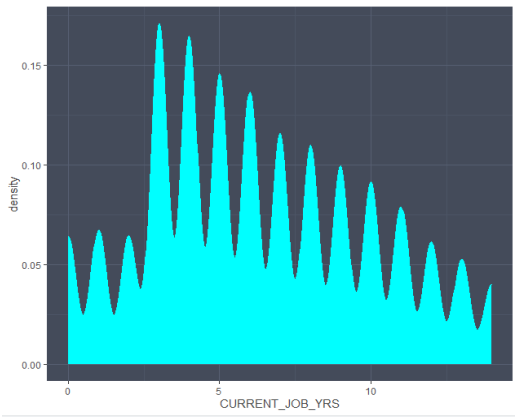
DENSITY GRAPHS

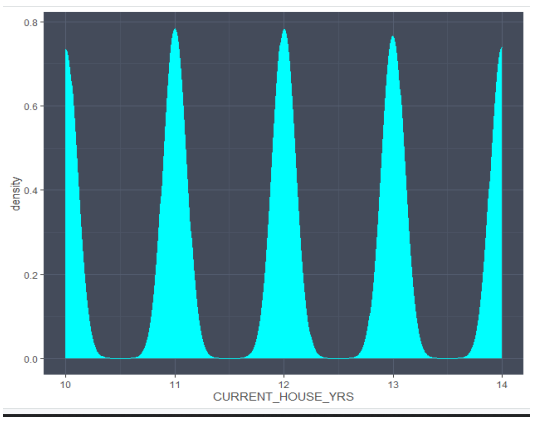




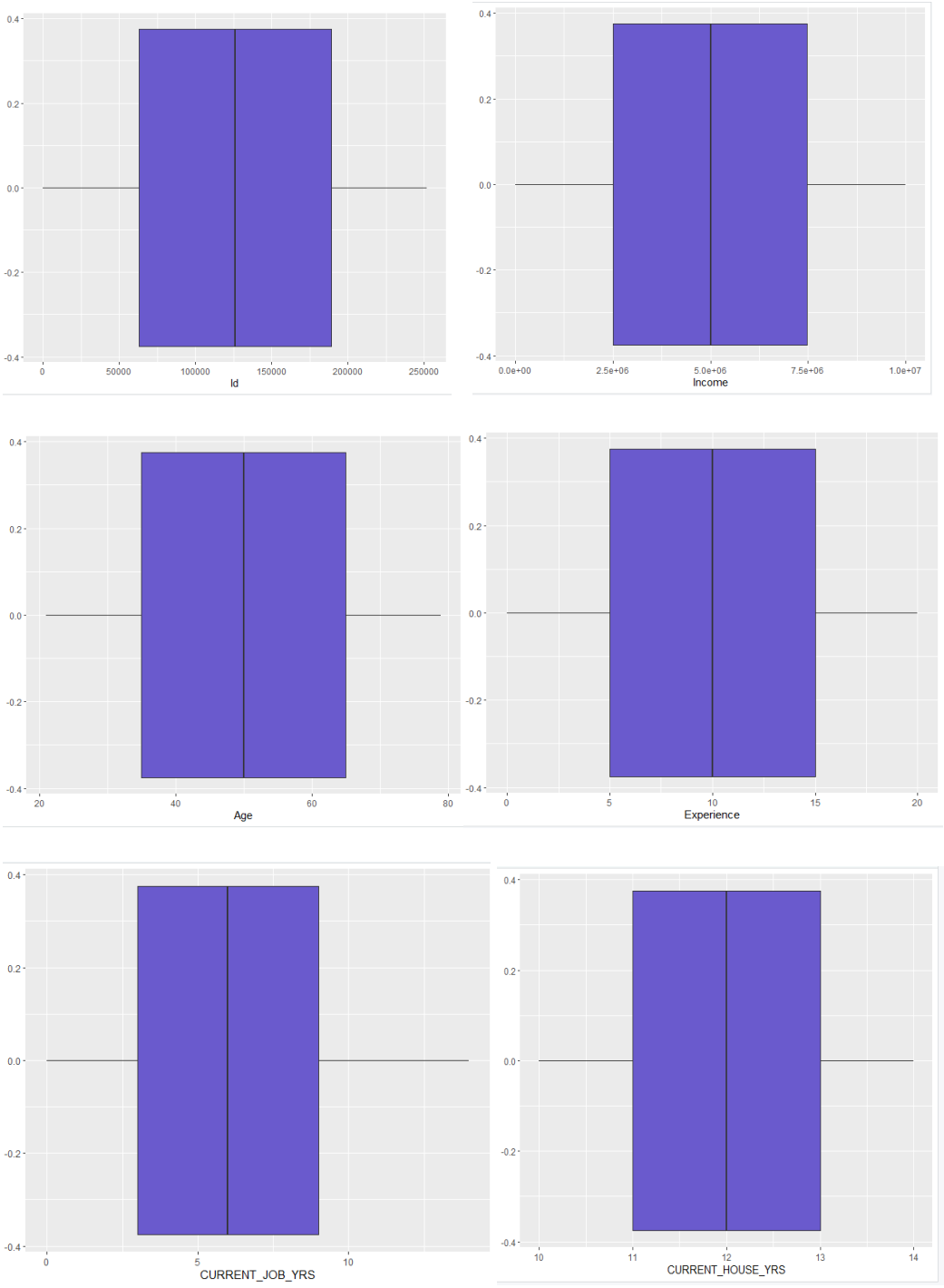








BOXPLOT GRAPHS



From the above density graphs we can conclude that there is no skewness in the dataset, all the features seem to follow almost a normal distribution and the boxplots tell us that the data is distributed in the interquartile range (Between 25th percentile to 75th percentile) and that there are no outliers present.

- <u>Encoding of Categorical Variables</u>

  Machine Learning and Neural Network algorithms expect the inputs to be numeric. Therefore, we need to encode the string attributes to be represented as numbers. There

are various encoding techniques such as one hot encoding, target encoding, mean encoding, label encoding to name a few.

The categorical attributes identified in our dataset are nominal categorical attributes, therefore we can make use of one-hot encoding and mean encoding to encode the attributes.

We have used one-hot encoding on “Married/Single”, “House Ownership” and “Car Ownership” attributes as the number of unique values in these attributes are less. Using one-hot encoding method, we create an additional feature for each unique value in the category.

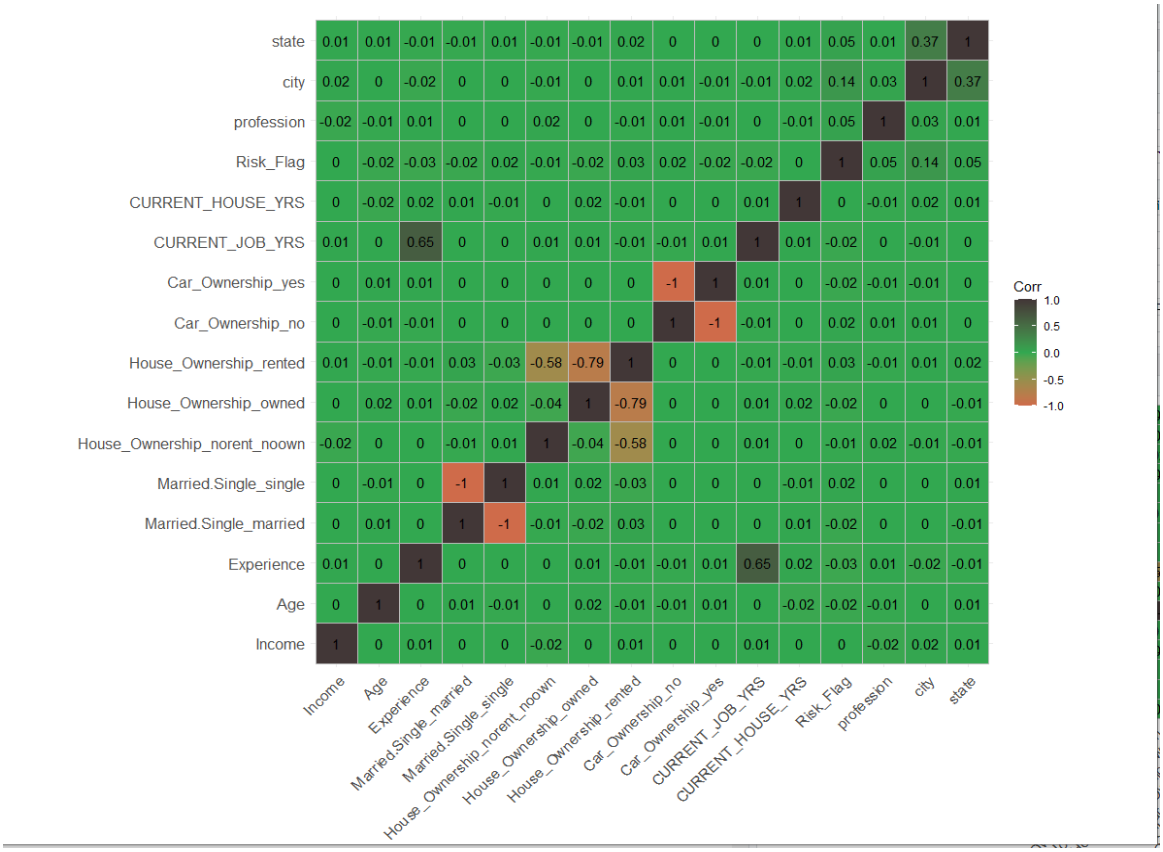
We have used mean encoding on “City”, “State” and “Profession” attributes as the number of unique values in these attributes is large and it would result in huge number of attributes if one hot encoding is used. Mean encoding specifies the probability of the target feature conditional on each value of the feature

- <u>Feature Selection</u>

  In feature selection, we remove any redundant variables from the dataset and select the most relevant features to predict the target variable.

  We have removed the ‘Id’ feature from the dataset as it had unique values and hence did not contribute to the variance in the target variable.

  We created a Correlation Matrix in order to determine the correlation between each of the variables in the dataset. Correlation measures the strength of the linear relationship between two variables. Using the Correlation Matrix, we remove variables that are highly correlated with each other.



From the above heatmap graph we can see that there are no features in our dataset that cross the correlation threshold of 0.9 and thus we can conclude that there is not much correlation between the independent features.

- Scaling the dataset

When we train a model using a Machine Learning algorithm the dataset needs to be scaled so that each feature contributes proportionately to the final distance between the data.

For example, if the range of one feature is much higher than the range of the other features, that particular feature will dominate the other features.

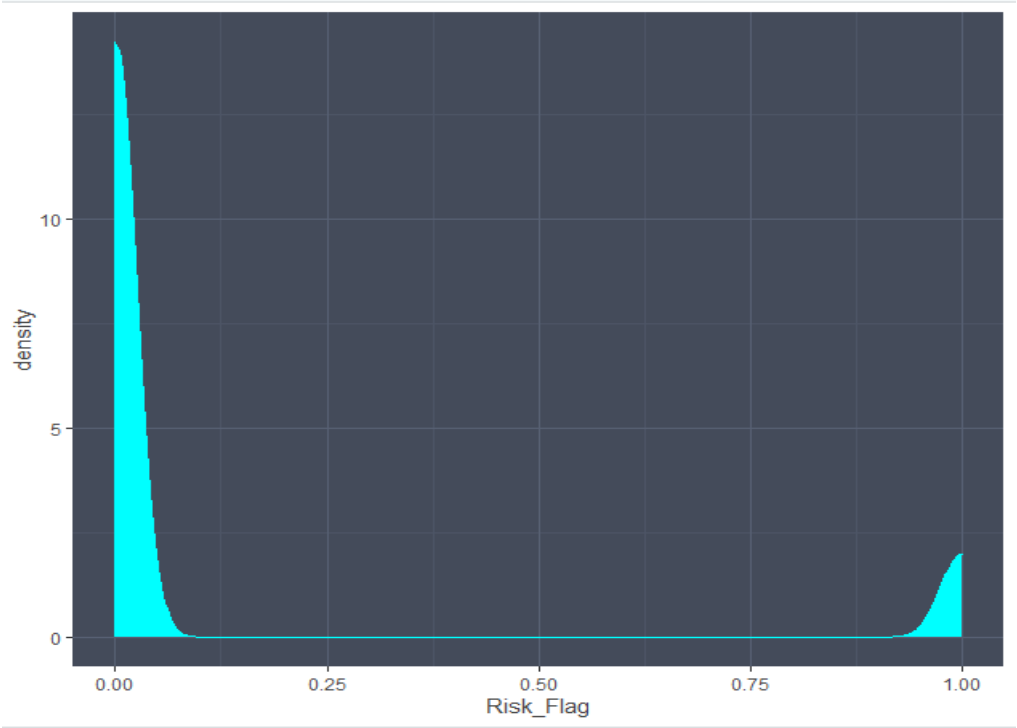
Therefore, when we scale the dataset, it ensures that the values of all the features lie within the same range/scale.

Also, for algorithms that use gradient descent or make use of Euclidean distances scaling the dataset helps in faster convergence and faster calculation of the distance.

We have used the normalization technique to scale the dataset where each feature is scaled to a range of [0,1].

- Balancing the data

When we are solving a classification problem it is important the number of classes in the training dataset are balanced otherwise the model will be biased.



Our dataset is highly imbalanced with 87.7% of the data points belonging to class 0 and 12.3% of the data points belonging to class 1.

We trained the model without balancing the data initially, we got a good accuracy score but the precision and recall was very less. To overcome this, we made use of undersampling and oversampling techniques using ROSE library to balance the dataset and our precision and recall score improved.

3. Supervised Learning Algorithms

Supervised Learning Algorithms try to learn function (the training vector) that maps an input to an output based on example input-output pairs.

Since we are trying to solve a classification problem where we are trying to classify a customer as a defaulter or non-defaulter, the supervised learning algorithms that we used are Decision Trees and XGBoost.

In decision trees the data is continuously split according to a certain parameter until the final decision (the final classification outcome) is reached. A decision tree is defined by two entities, namely nodes and leaves. The leaves are the decisions or the final outcomes. And the nodes are where the data is split. When the tree is built, it is constructed by recursively evaluating different features and using the feature that best splits the data at each node using criteria such as gini impurity and information gain.

Decision Trees tend to be low bias and high variance models, they are prone to overfitting. Therefore, we have made use of XGBoost algorithm.

XGBoost is a decision tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. Gradient Boosting models are models that are built sequentially by minimizing the errors from the previous models while increasing the influence of high performing models. We use XGBoost algorithms as they are known to increase speed and performance of the model through parallel processing, tree pruning and minimizing overfitting.

Neural Networks

We have also trained the model using Neural Networks, which is a subset of machine learning where the algorithm itself learns the underlying relationships in the data with less manual human intervention and can very easily adapt to changes in the input data.

We made use of Keras library to build and train the neural network for 100 epochs.

4. Deviations from Project Plan

We have not deviated from the plan. As planned, we trained two supervised learning algorithms (Decision Trees and XGBoost) and Neural Networks.

5. Model Evaluation Methods

We have randomized and split our data into training (70%) and testing (30% data) sets. We train the models using the training set and evaluate model performance using the test set using the test data and the trained model, we predict the likely outcomes and then derive Accuracy, Recall and Precision from the Confusion matrix.

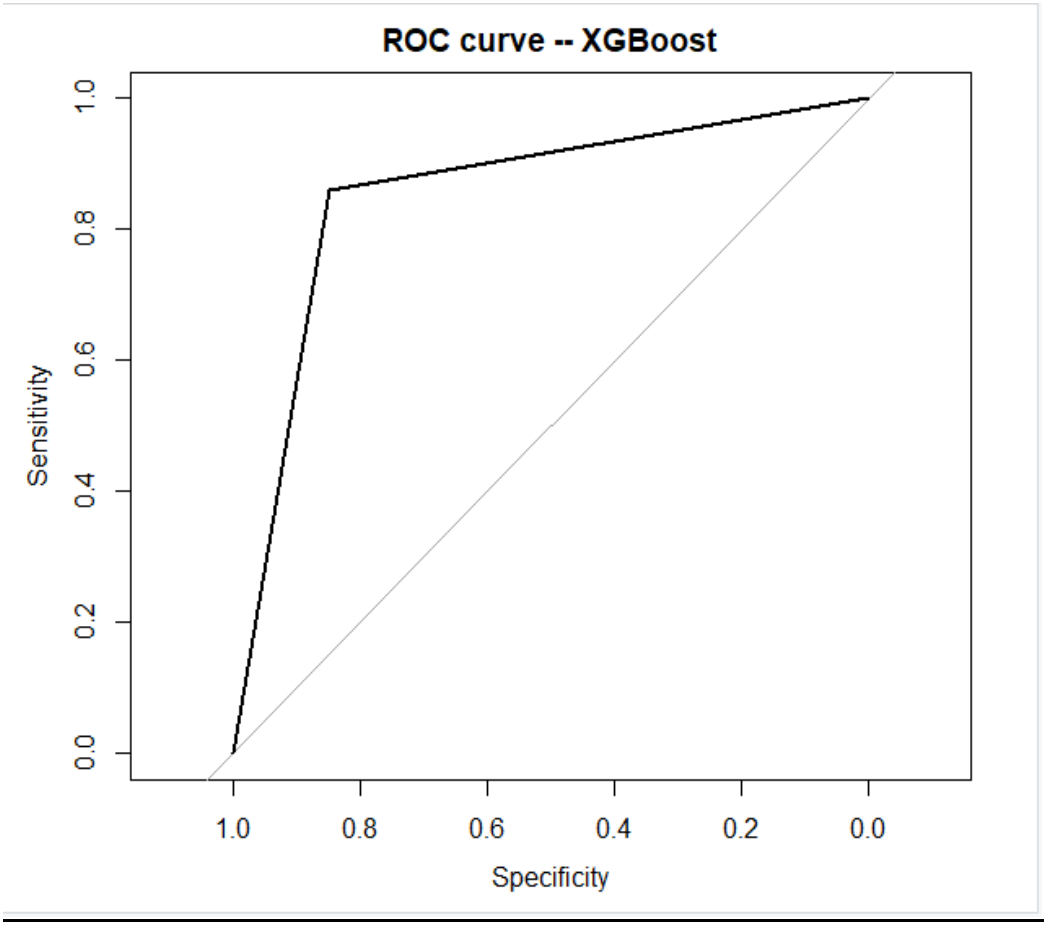
Accuracy is the number of correct predictions from all the predictions made. Recall is the number of positive class predictions made out of all the actual positive examples in the dataset. Precision is the number of positive class predictions made out of all the predicted positive class.

We have also plotted the ROC (Receiver Operating Characteristic) curve to summarise the TPR (True Positive Rate) and FPR (False Positive Rate) for the predictive model using different thresholds. The best threshold has the highest true positive rate together with the lowest false positive rate. In addition to that we have calculated and plotted the training and validation accuracy and loss for the neural network model

RESULTS

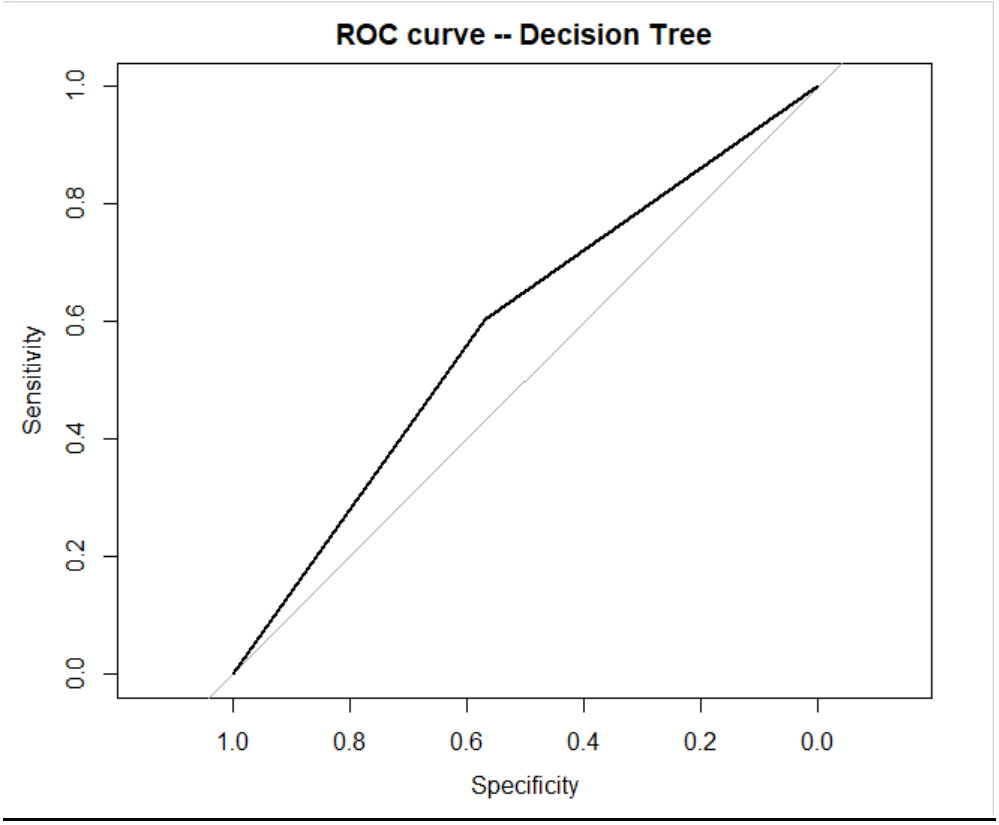
Model: XGBOOST

- Accuracy = 85.5%
- Precision = 85.86%
- Recall = 85%
- AUC score = 85.5%



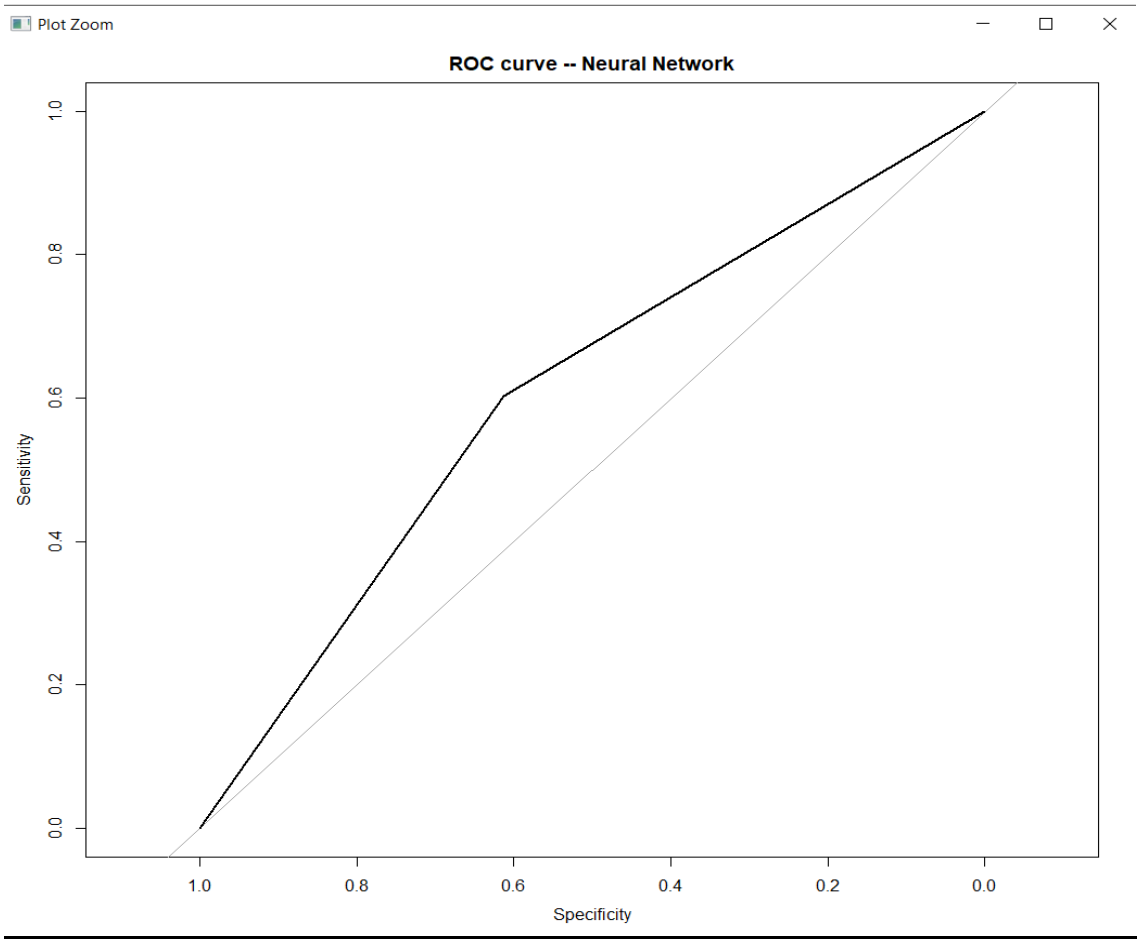
Model: DECISION TREE

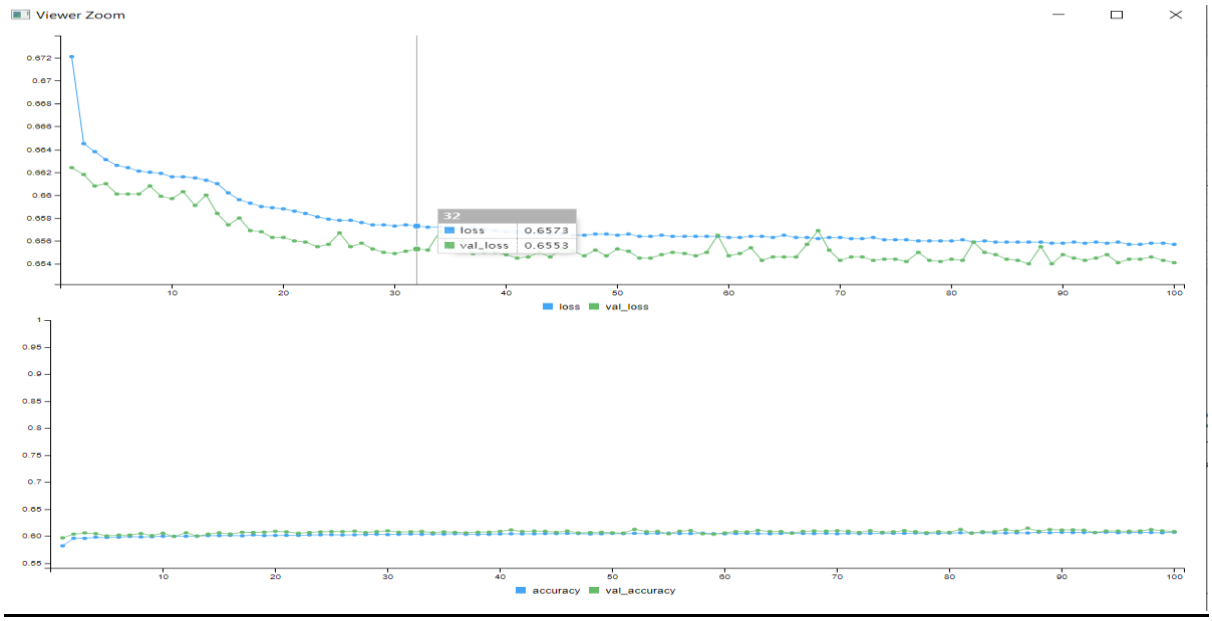
- Accuracy = 58.65
- Precision = 58.98
- Recall = 56.90
- AUC score = 58.65



Model: NEURAL NETWORKS

- Accuracy = 60.71%
- Precision = 60.66%
- Recall = 61.27%
- AUC score = 60.75%





DISCUSSION

From the above results we can see that given the parameters we are able to predict if the person would be a defaulter with almost 85% of accuracy using XGBoost model, so that the bank can make additional checks before sanctioning the loan to the person or reject the loan itself.

False Positive is a case where our model predicts that a person is a defaulter but in reality he is not a defaulter. It is important to reduce the False Positive because an honest customer could be rejected the loan. Therefore, when False Positive is important the model should have good precision. Our XGBoost Model has a Precision of 85.86%

False Negative is a case when the customer is actually a defaulter but the model predicts otherwise. It is also important to reduce False Negatives because if the bank grants the loan and the customer fails to fulfill it, the bank will suffer financial losses. Therefore, when False Negative is important the model should have good recall. Our XGBoost Model has a recall of 85%

CONCLUSION

From the above results we can conclude that XGBoost performs the best with respect to all the evaluation metrics such as accuracy, precision, recall and aucscore compared to other models. Decision Tree model performed poorly and even neural network could not better the results of XGBoost after training for 100 epochs. Since all the metrics are important in solving this classification problem XGBoost model is the best model to be used when solving such problem as it is more generalized and robust.

In Future, we would like to implement a retraining approach. There maybe changes in the defaulter's behaviour and this will result in data drift and the accuracy of the model to predict the defaulters might start to decrease. Therefore, it is important to keep training the model with new data.

REFERENCES

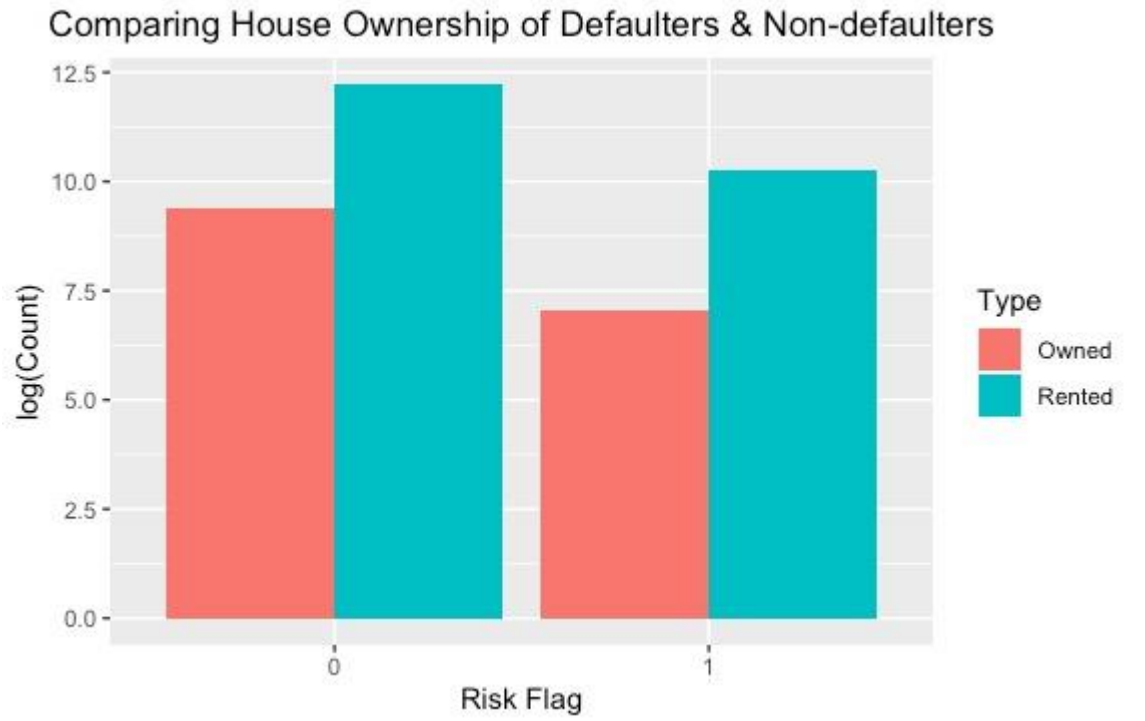
http://cs229.stanford.edu/proj2017/final-reports/5195587.pdf

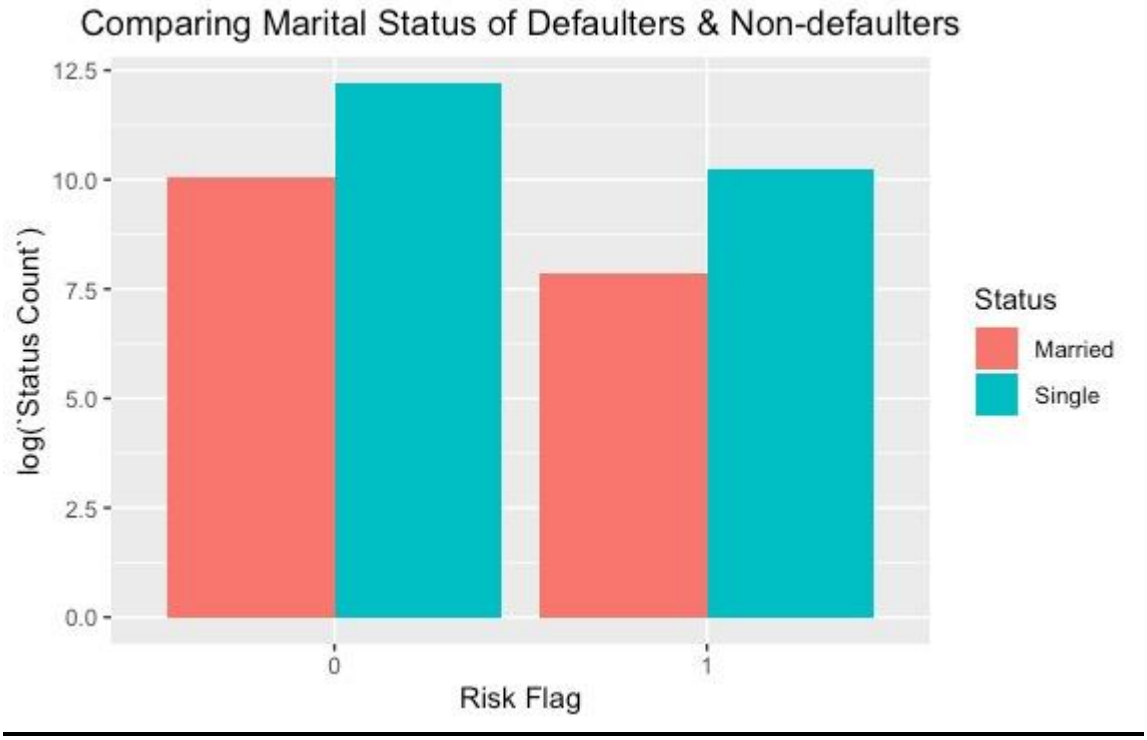
https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012042

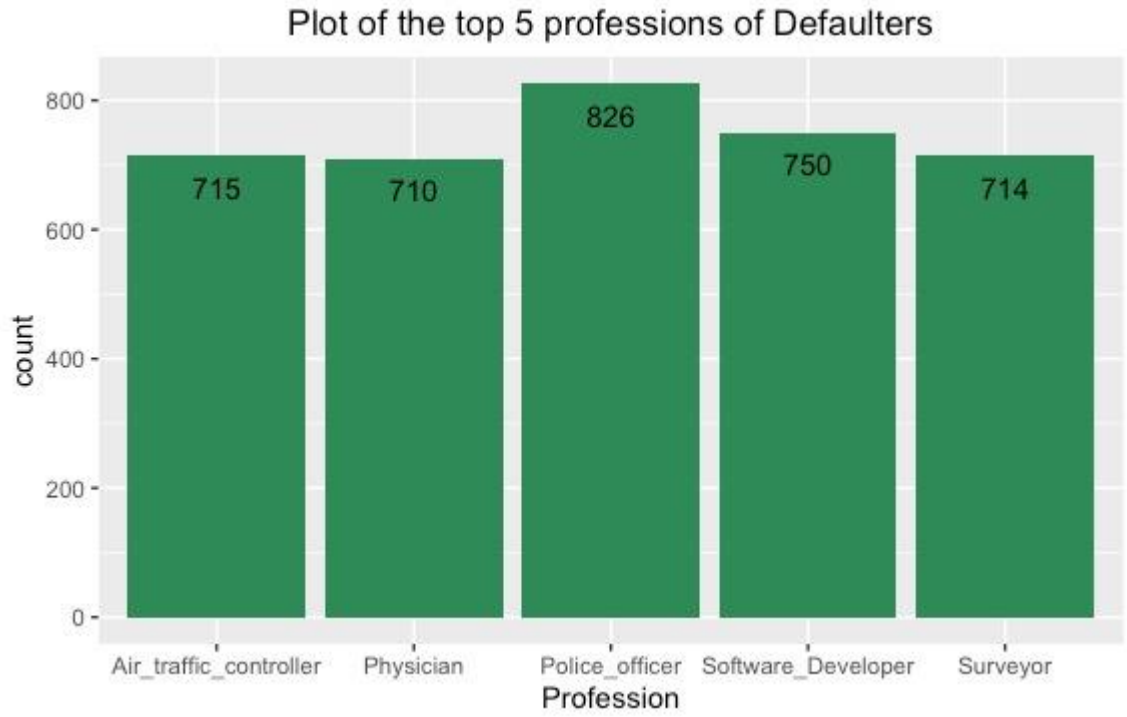
https://r-charts.com/part-whole/pie-chart-percentages-ggplot2/

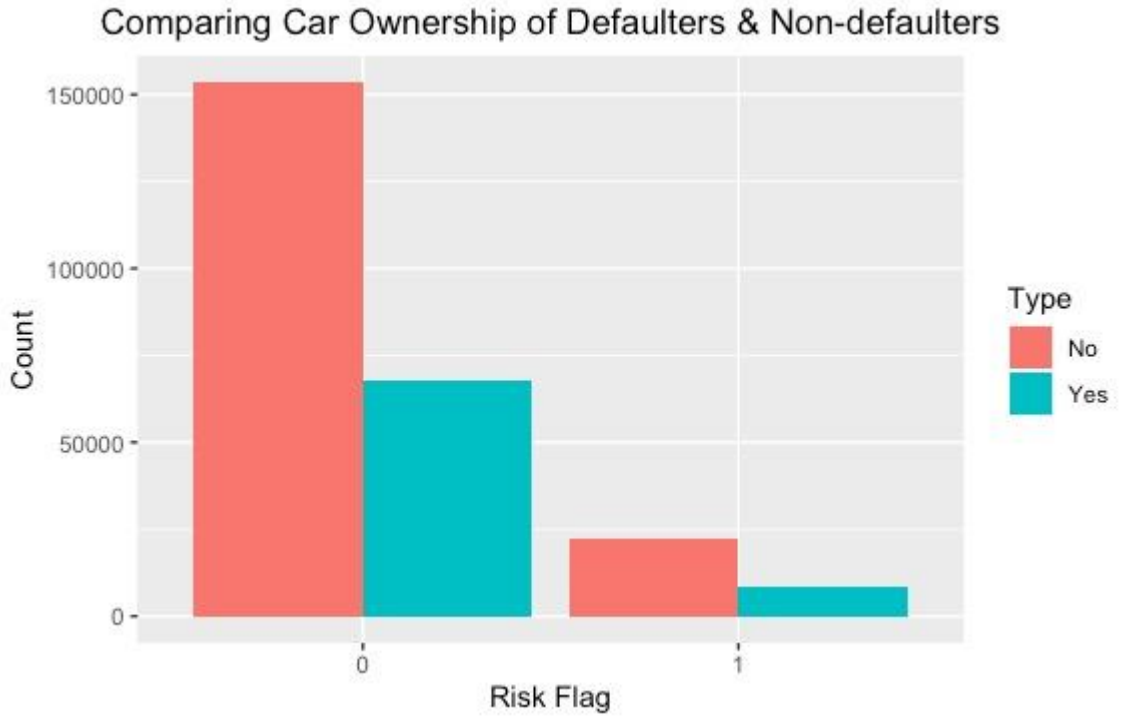
APPENDIX

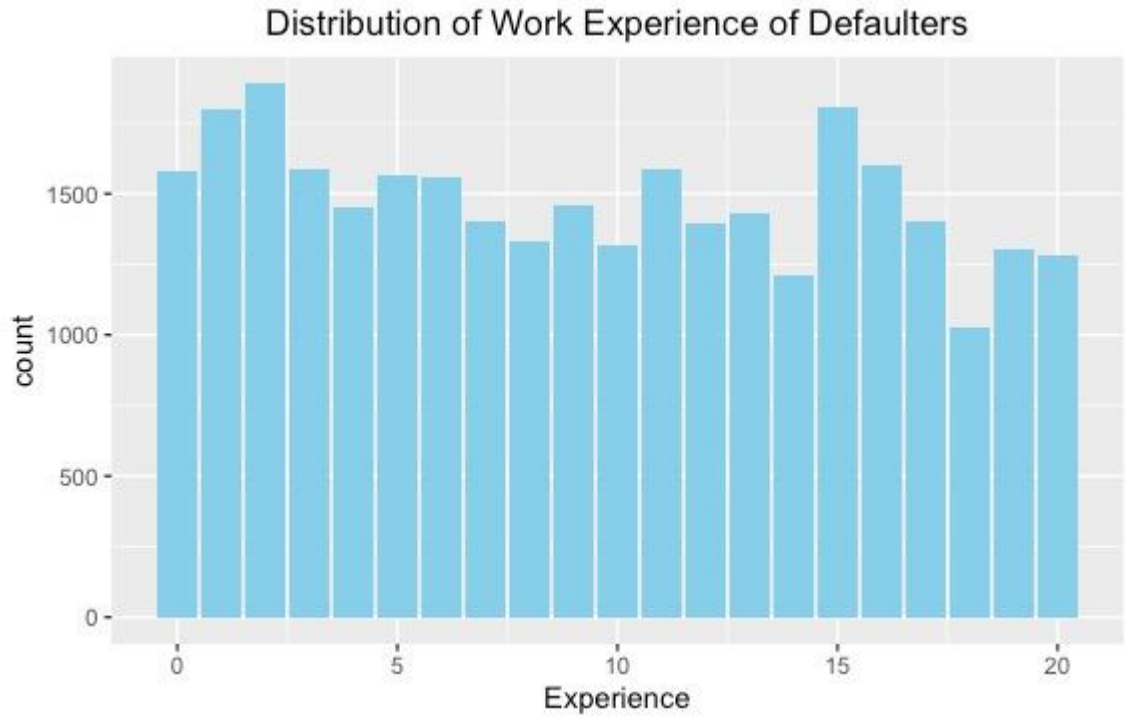
Insight visualizations:

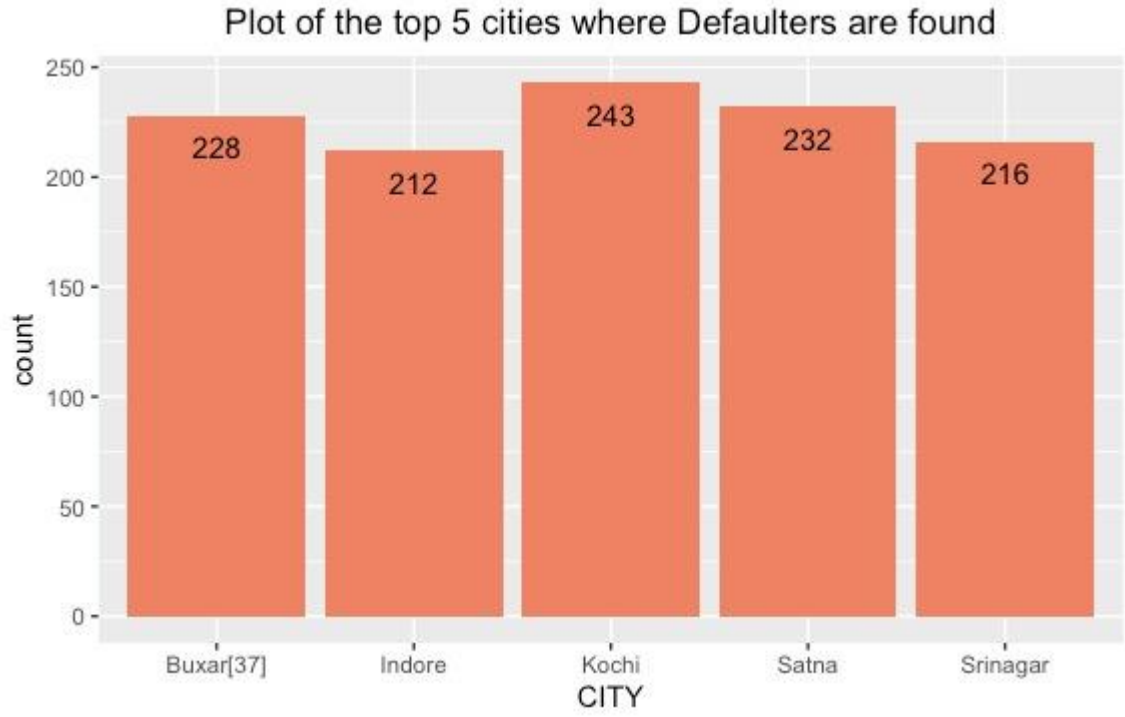


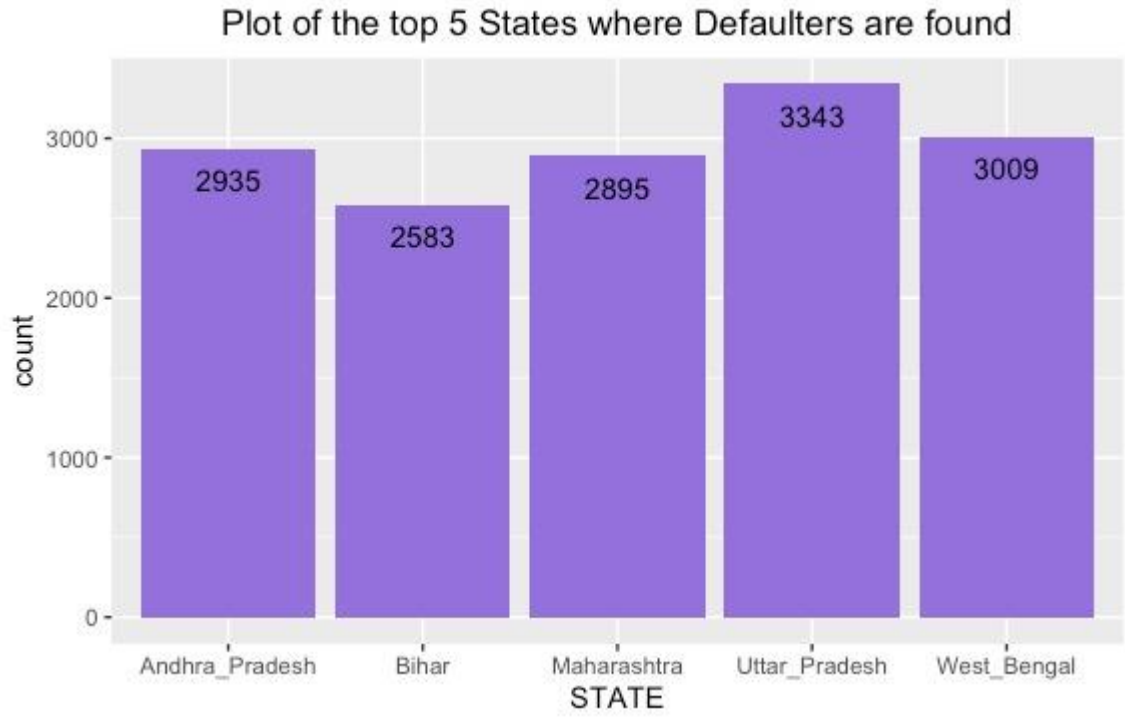


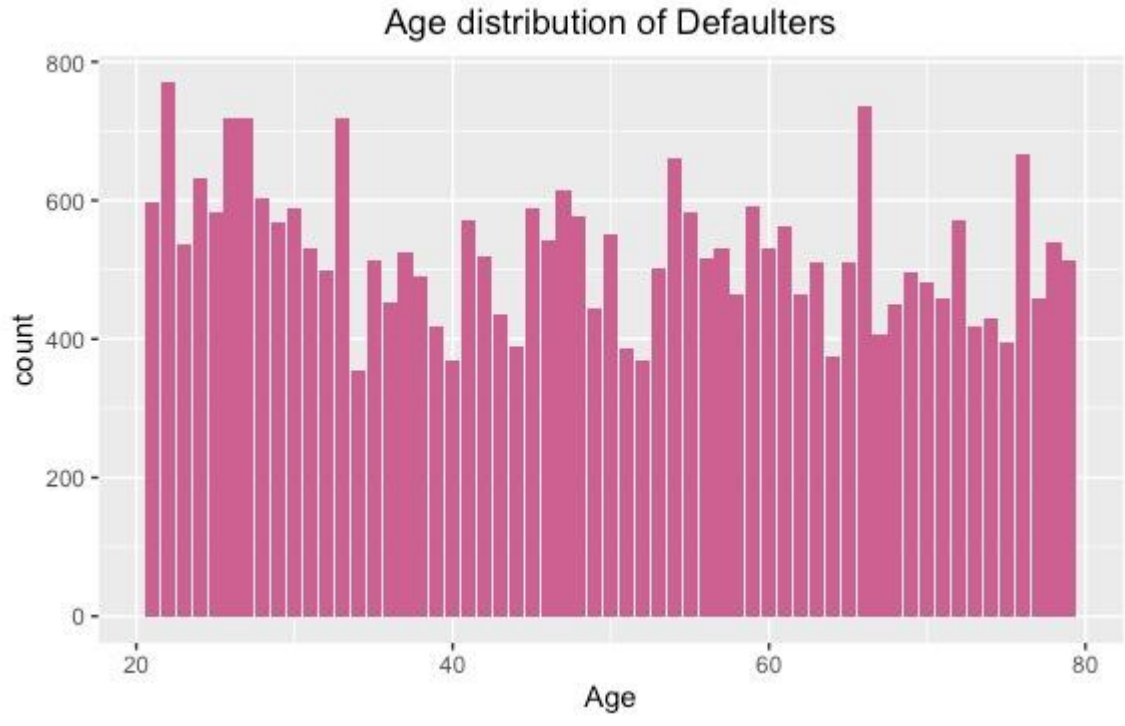

Plot of the top 5 professions of Defaulters
800
600
400
200
0
count
715
710
826
750
714
Air_traffic_controller
Physician
Police_officer
Software_Developer
Surveyor
Profession


Comparing Car Ownership of Defaulters & Non-defaulters
150000
100000
50000
0
Count
0
1
Risk Flag
Type
No
Yes

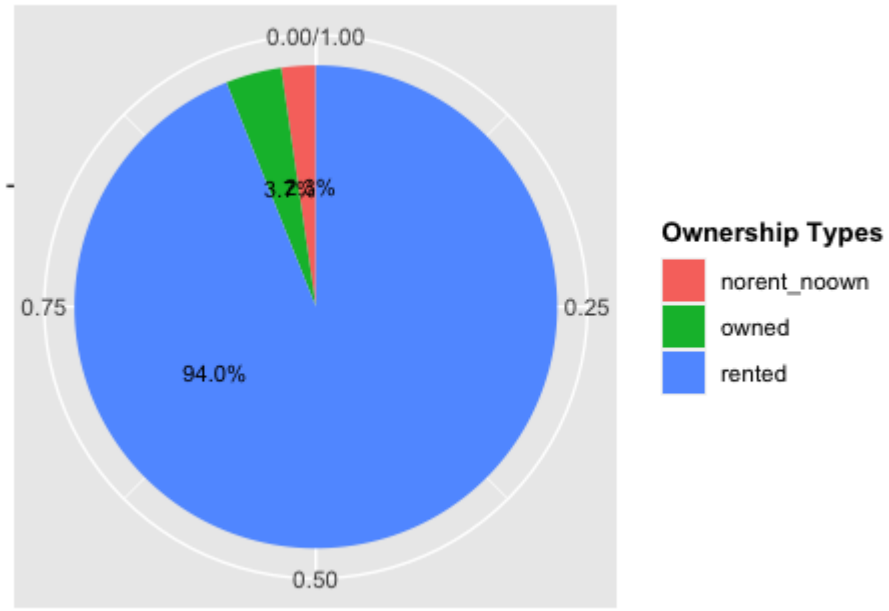

Distribution of Work Experience of Defaulters
count
1500
1000
500
0
0
5
10
15
20
Experience


Plot of the top 5 cities where Defaulters are found
count
250
200
150
100
50
0
228
212
243
232
216
Buxar[37]
Indore
Kochi
Satna
Srinagar
CITY

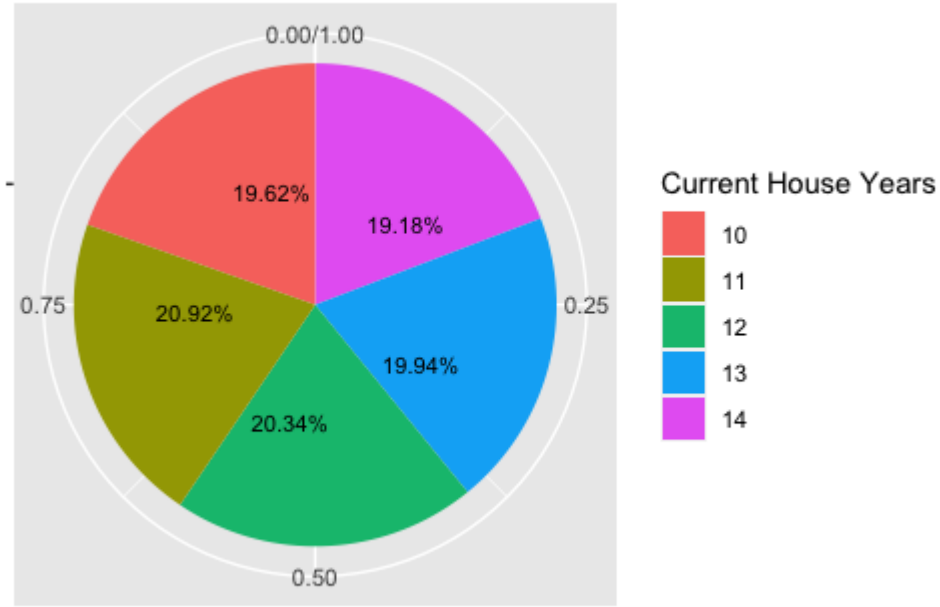

Plot of the top 5 States where Defaulters are found
count
3000
2000
1000
0
2935
2583
2895
3343
3009
Andhra_Pradesh
Bihar
Maharashtra
Uttar_Pradesh
West_Bengal
STATE

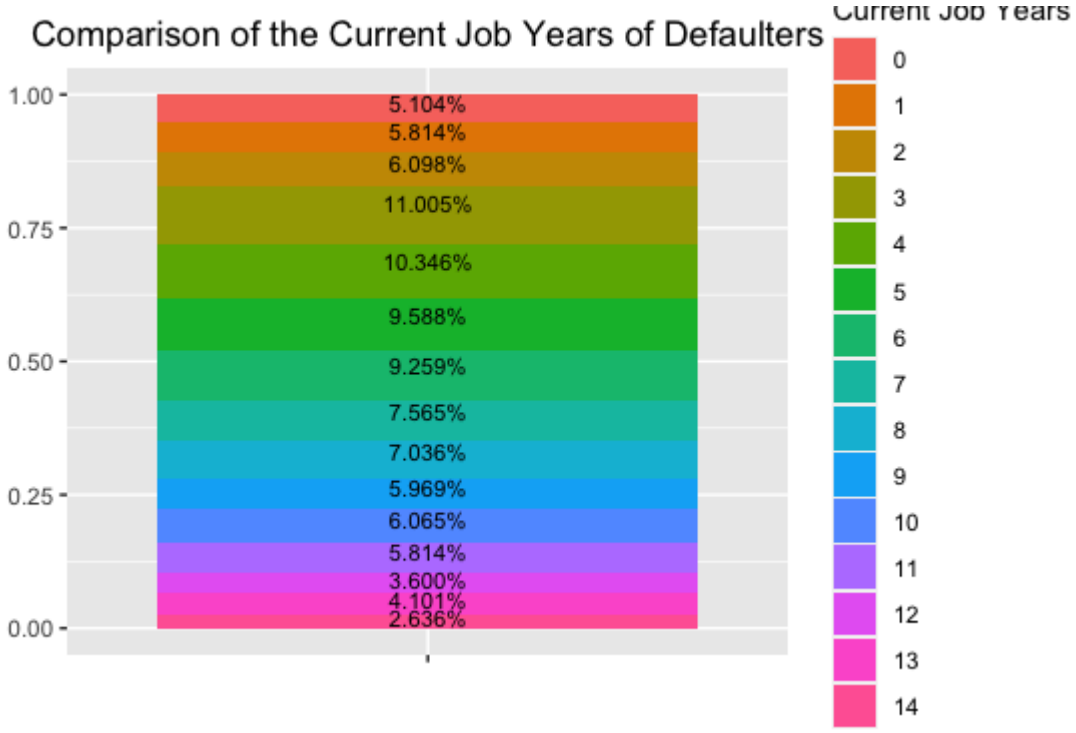

Age distribution of Defaulters
count
800
600
400
200
0
20
40
60
80
Age

Comparison of the House Ownership of Defaulters



Comparison of the Current House Years of Defaulters




Comparison of the Current Job Years of Defaulters
1.00
0.75
0.50
0.25
0.00
5.104%
5.814%
6.098%
11.005%
10.346%
9.588%
9.259%
7.565%
7.036%
5.969%
6.065%
5.814%
3.600%
4.101%
2.636%
Current Job Years
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14