A PROJECT REPORT

On:

Predicting The Eligibility For Home-Loan Sanction.

Course Project of Statistical & AI Techniques in Data Mining : (MTH552A)

Submitted By;

Krishnendu Paul (211322)

Under Supervision of Dr. Amit Mitra

DEPARTMENT OF MATHEMATICS AND STATISTICS

Acknowledgement

Real learning comes from a practical work.We would like to thank our instructor of the course Dr. Amit Mitra (Department of Mathematics and Statistics, IIT KANPUR), for providing us constant guidance and motivation for this project, without which it would have been an impossible task to accomplish.We would like to thank our department professors for teaching all the necessary topics with immense care which was needed to make the project fruitful.

We also take this opportunity to thank the authors and publishers of the various books and journals we have consulted .Without those this work would not have been completed.

Indeed it has been a great learning experience and has also provided us with a practical insight of the theoretical knowledge gathered during the course lecture.

Contents

1 Introduction:

Buying a house is one of the biggest dreams come true for many people and an extravagant affair altogether. Giving life to such a dream requires a lot of effort from the client' end and the best one can do to accommodate the home in their budget is through a home loan. A real estate loan can be chosen to buy a new home/apartment or a land in which we build the house, and even for renovation, extension, and repairs to an existing house.

The most common kind of real estate loan available for buying a home. There are many housing finance companies like public banks, and private banks that provide residential loans where we borrow money to buy the home of your choice and pay back the loan by monthly instalments.

Most of us prefer taking a loan from a bank or a trusted non-bank finance company (NBFN) that is linked to government policies and is trustworthy.

A typical loan is made up of three components, principal or the borrowed amount, rate of interest and tenure or duration for which the loan is availed.

2 Problem Description:

On the basis of the guarantee provided, loans are of two kinds, such as secured loans and unsecured loans.

Lenders often issue loans guaranteed by a particular personal asset. It could be a house, a car, a boat, or stocks and bonds. When property is used to secure a loan then the lender maintains ownership rights in the asset until the loan gets repaid. This means if you do not repay the loan or otherwise meet the terms of your loan agreement, the lender has the legal right to seize and sell the property in order to repay the loan. This asset is sometimes called loan collateral.

Unsecured loan does not refer to a particular property as collateral on the loan. Instead, the loan is issued on the basis of our ability to repay the loan. we might have to provide information about your income, savings, employment, or credit history. Some common types of unsecured loans include credit cards, student loans, and personal loans etc.

Customers are required to submit a loan application once the bank has validated their eligibility.

We need to automate the loan qualification process based on the customer information provided when completing the online application form and to automate this process, one needs to identify the customers segments, those are eligible for loan amount so that banks can specifically target these customers.

3 Objective:

The purpose of the work is to automate the home-loan eligibility process based on customer information.

Before proceeding into theoretical analysis, we use a method to analyze the characteristics of the variables visually, named as Exploratory Data Analysis.

It's a classification issue in terms of whether a home-loan is going to be approved or not. In this project, we compare between FIVE Supervised Classification Algorithm namely-Logistic Regression , Decision Tree , Random Forest , Bagging(Bootstrap aggregating) and Support Vector Machine (SVM) , then with help of the best model we will predict if the home-loan should be granted or not on to the customers listed on test set.

After that we checking the top most significant regressors in that model.

4 Description of Data:

The Data set is collected from Kaggle.There are two files within the dataset train.csv and test.csv . train.csv to fit the model and the test.csv to predict the outcome as required.

In this data set Loan_Status is our response variable and others are regressor variable.

```
   Loan_ID Gender Married Dependents    Education Self_Employed ApplicantIncome
1 LP001002   Male      No          0     Graduate            No             5849
2 LP001003   Male     Yes          1     Graduate            No             4583
3 LP001005   Male     Yes          0     Graduate           Yes             3000
4 LP001006   Male     Yes          0 Not Graduate            No             2583
5 LP001008   Male      No          0     Graduate            No             6000
6 LP001011   Male     Yes          2     Graduate           Yes             5417
  CoapplicantIncome LoanAmount Loan_Amount_Term Credit_History Property_Area
1                 0         NA              360              1         Urban
2              1508        128              360              1         Rural
3                 0         66              360              1         Urban
4              2358        120              360              1         Urban
5                 0        141              360              1         Urban
6              4196        267              360              1         Urban
  Loan_Status
1           Y
2           N
3           Y
4           Y
5           Y
6           Y
```

Data Table:

Variable Name	Description
Loan ID	Loan reference number (unique ID)
Gender	Male/Female
Married	Applicant Married (Y/N)
Dependents	Number of family members
Education	Applicant Education (Graduate/Under Graduate)
Self Employed	Applicant employment status(Y/N)
Applicant Income	Applicant's income
Coapplicant Income	Additional applicant's income
LoanAmount	Loan amount in thousands
Loan Amount Term	Term of loan (in days)
Credit History	Record of previous credit history
Property Area	The location of property(Urban/Semi Urban/Rural)
Loan Status	Loan approved (Y/N)

So let us check on both train and test data set wheather the response variable (Loan Status) is present or not in our given data set.

Train Data:

```
 [1] "Loan_ID"           "Gender"            "Married"
 [4] "Dependents"        "Education"         "Self_Employed"
 [7] "ApplicantIncome"   "CoapplicantIncome" "LoanAmount"
[10] "Loan_Amount_Term"  "Credit_History"    "Property_Area"
[13] "Loan_Status"
```

Test Data:

```
 [1] "Loan_ID"           "Gender"            "Married"
 [4] "Dependents"        "Education"         "Self_Employed"
 [7] "ApplicantIncome"   "CoapplicantIncome" "LoanAmount"
[10] "Loan_Amount_Term"  "Credit_History"    "Property_Area"
```

From the output we can see the response variable "Loan Status" is not present in test data set, so we can't check the accuracy of different models based on this test data set.

5 Data Pre-processing & Missing Values Imputation:

5.1 Data Pre-processing:

We use a data mining technique to turn the raw data gathered from diverse sources into cleaner information that's suitable for work. We know that Raw data can have missing or inconsistent values as well as present a lot of redundant information. These problemes should be taken care of, otherwise the final output would be plagued with faulty insights. This is true for more sensitive analysis that can be more affected by very small mistakes. Here we will check if there are missing values and incorporate them with suitable alternatives.

```
   Loan_ID             Gender            Married           Dependents
 Length:614         Length:614         Length:614         Length:614
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character




  Education         Self_Employed      ApplicantIncome CoapplicantIncome
```

```
Length:614         Length:614         Min.   :  150   Min.   :    0
Class :character   Class :character   1st Qu.: 2878   1st Qu.:    0
Mode  :character   Mode  :character   Median : 3812   Median : 1188
                                      Mean   : 5403   Mean   : 1621
                                      3rd Qu.: 5795   3rd Qu.: 2297
                                      Max.   :81000   Max.   :41667


  LoanAmount    Loan_Amount_Term Credit_History   Property_Area
Min.   :  9.0   Min.   : 12      Min.   :0.0000   Length:614
1st Qu.:100.0   1st Qu.:360      1st Qu.:1.0000   Class :character
Median :128.0   Median :360      Median :1.0000   Mode  :character
Mean   :146.4   Mean   :342      Mean   :0.8422
3rd Qu.:168.0   3rd Qu.:360      3rd Qu.:1.0000
Max.   :700.0   Max.   :480      Max.   :1.0000
NA's   :22      NA's   :14       NA's   :50
Loan_Status
Length:614
Class :character
Mode  :character
```

- Here we observe that some of the variables are categorical,while others are numerical.
- In the given data frame contains missing values in different columns.Most of the regressor variables like Gender, Married, Dependents, Self employed ,Loan amount, Loan Amount term and Credit History have missing values.Credit History contains maximum number of

missing values.

5.2 Missing Values Imputation:

- Our first work will be replace the strings with numbers for categorical inputs and only then impute the missing values.
- For the categorical variables,here we will replace the missing entries with mode. On the other hand due to the presence of outliers, we will replace missing entries with median for quantitative variables.

```
#4.Changing from character to numeric (train_data)
train_data = subset(train_dt, select = -c(Loan_ID))
train_data$Gender <- ifelse(train_data$Gender == "Male", 1, 0)
train_data$Married <- ifelse(train_data$Married == "Yes", 1, 0)
dependents <- c("0" = 0, "1" = 1, "2" = 2, "3+" = 3)
train_data$Dependents <- dependents[train_data$Dependents]
train_data$Education <- ifelse(train_data$Education == "Graduate", 1, 0)
train_data$Self_Employed <- ifelse(train_data$Self_Employed == "Yes", 1, 0)
area <- c("Urban" = 2, "Semiurban" = 1, "Rural" = 0)
train_data$Property_Area <- area[train_data$Property_Area]
train_data$Loan_Status <- ifelse(train_data$Loan_Status == "Y", 1, 0)


#5.Changing from character to numeric (test_data)
test_data = subset(test_dt, select = -c(Loan_ID))
test_data$Gender <- ifelse(test_data$Gender == "Male", 1, 0)
test_data$Married <- ifelse(test_data$Married == "Yes", 1, 0)
dependents <- c("0" = 0, "1" = 1, "2" = 2, "3+" = 3)
test_data$Dependents <- dependents[test_data$Dependents]
test_data$Education <- ifelse(test_data$Education == "Graduate", 1, 0)
```

```r
test_data$Self_Employed <- ifelse(test_data$Self_Employed == "Yes", 1, 0)
area <- c("Urban" = 2, "Semiurban" = 1, "Rural" = 0)
test_data$Property_Area <- area[test_data$Property_Area]

attach(train_data)

#6. Calculate mode
Mode <- function(x, na.rm)
{
xtab <- table(x)
xmode <- names(which(xtab == max(xtab)))
if (length(xmode) > 1) xmode <- ">1 mode"
return(xmode)
}

#7. Imputing with mode
m1 <- Mode(Gender, na.rm = T)
train_data[is.na(train_data$Gender), "Gender"] <- m1
m11 <- Mode(test_data$Gender, na.rm = T)
test_data[is.na(test_data$Gender), "Gender"] = m11
m2 <- Mode(Married, na.rm = T)
train_data[is.na(train_data$Married), "Married"] <- m2
m21 <- Mode(test_data$Married, na.rm = T)
test_data[is.na(test_data$Married), "Married"] = m21
m3 <- Mode(Dependents, na.rm = T)
train_data[is.na(train_data$Dependents), "Dependents"] <- m3
m31 <- Mode(test_data$Dependents, na.rm = T)
test_data[is.na(test_data$Dependents), "Dependents"] = m31
```

```r
m4 <- Mode(Self_Employed, na.rm = T)
train_data[is.na(train_data$Self_Employed), "Self_Employed"] <- m4
m41 <- Mode(test_data$Self_Employed, na.rm = T)
test_data[is.na(test_data$Self_Employed), "Self_Employed"] = m41
m5 <- Mode(Credit_History, na.rm = T)
train_data[is.na(train_data$Credit_History), "Credit_History"] <- m5
m51 <- Mode(test_data$Credit_History, na.rm = T)
test_data[is.na(test_data$Credit_History), "Credit_History"] = m51

#8. Imputing with median
m6 <- median(LoanAmount, na.rm = T)
train_data[is.na(train_data$LoanAmount), "LoanAmount"] <- 6
m61 <- median(test_data$LoanAmount, na.rm = T)
test_data[is.na(test_data$LoanAmount), "LoanAmount"] <- m61
m7 <- median(Loan_Amount_Term, na.rm = T)
train_data[is.na(train_data$Loan_Amount_Term), "Loan_Amount_Term"] <- m7
m71 <- median(test_data$Loan_Amount_Term, na.rm = T)
test_data[is.na(test_data$Loan_Amount_Term), "Loan_Amount_Term"] = m71
train_data$Gender <- ifelse(train_data$Gender == "1", 1, 0)
train_data$Married <- ifelse(train_data$Married == "1", 1, 0)
dependents <- c("0" = 0, "1" = 1, "2" = 2, "3" = 3)
train_data$Dependents <- dependents[train_data$Dependents]
train_data$Self_Employed <- ifelse(train_data$Self_Employed == "1", 1, 0)
train_data$Credit_History <- ifelse(train_data$Credit_History == "1", 1, 0)

test_data$Gender <- ifelse(test_data$Gender == "1", 1, 0)
dependents <- c("0" = 0, "1" = 1, "2" = 2, "3" = 3)
test_data$Dependents <- dependents[test_data$Dependents]
```

```r
test_data$Self_Employed <- ifelse(test_data$Self_Employed == "1", 1, 0)

test_data$Credit_History <- ifelse(test_data$Credit_History == "1", 1, 0)
```

5.3 Classify the Variables:

```
'data.frame': 614 obs. of  12 variables:
 $ Gender           : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Married          : num  0 1 1 1 0 1 1 1 1 1 ...
 $ Dependents       : num  0 1 0 0 0 2 0 3 2 1 ...
 $ Education        : num  1 1 1 0 1 1 0 1 1 1 ...
 $ Self_Employed    : num  0 0 1 0 0 1 0 0 0 0 ...
 $ ApplicantIncome  : int  5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...
 $ CoapplicantIncome: num  0 1508 0 2358 0 ...
 $ LoanAmount       : num  6 128 66 120 141 267 95 158 168 349 ...
 $ Loan_Amount_Term : num  360 360 360 360 360 360 360 360 360 360 ...
 $ Credit_History   : num  1 1 1 1 1 1 1 0 1 1 ...
 $ Property_Area    : num  2 0 2 2 2 2 2 1 2 1 ...
 $ Loan_Status      : num  1 0 1 1 1 1 1 0 1 0 ...
```

- **Categorical Variables:**

The categotical variables have data fields that can be divided into definite groups. In this case, Gender (Make or Female), Married (Yes or No), Self_Employed (Yes or No), Loan_Status (Y or N) are the categorical variables.

- **Ordinal Variables:**

For ordinal variables we can divided into groups, but these groups have some kind of order. In this study, Dependents (0 or 1 or 2 or 3+), Education (Graduate or Not Graduate), Credit_History (0 or 1), Property_Area (Urban or Semi Urban or Rural) are the ordinal variables.

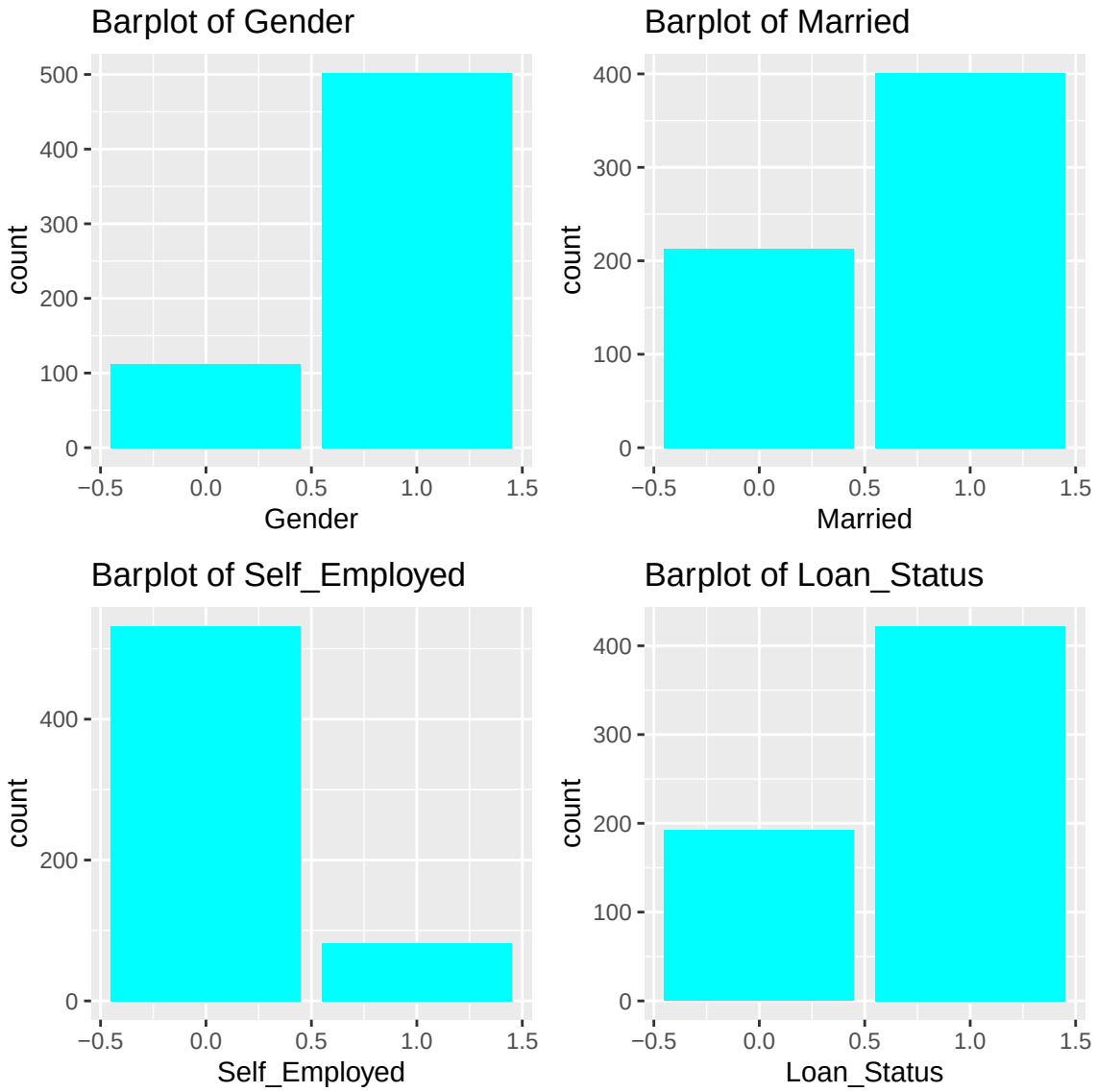
- **Numerical Variables:**

Here these variables can take up any value within a given range. In this case, ApplicantIncome, LoanAmount, CoapplicantIncome, Loan Amount Term are the numerical variable.

6 Exploratory Data Analysis (EDA):

Exploratory data analysis is an approach of analysing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modelling or hypothesis testing task.

6.1 Univariate Analysis:



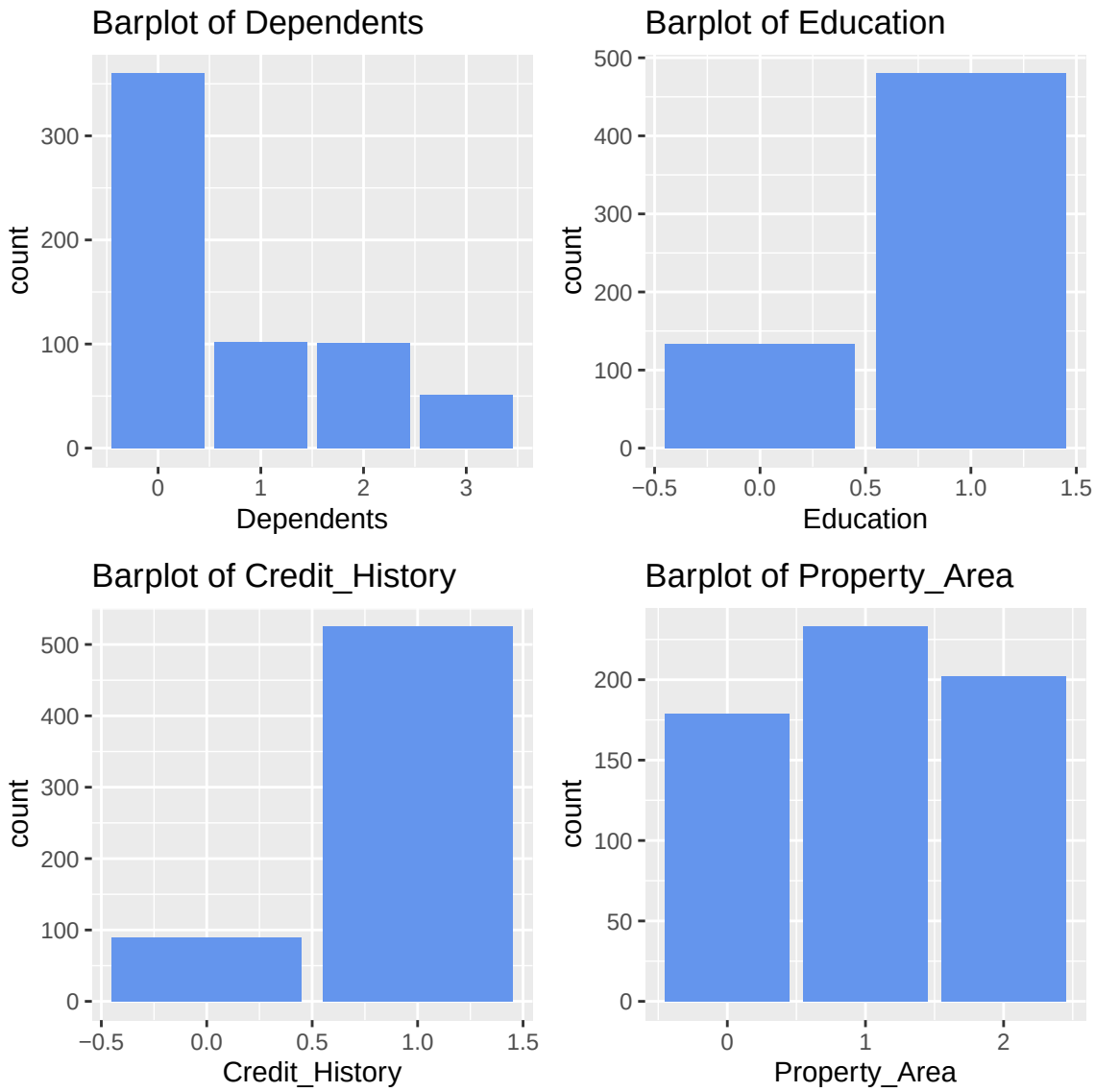
Observations:

• Here 79.64% of them are male.

•Here 64.82% of them are married.

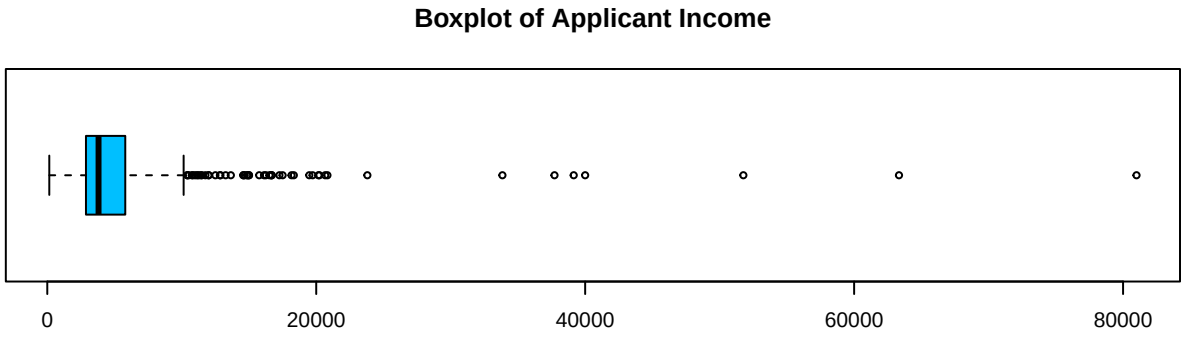
•Here 81.43% of them are self-employed.

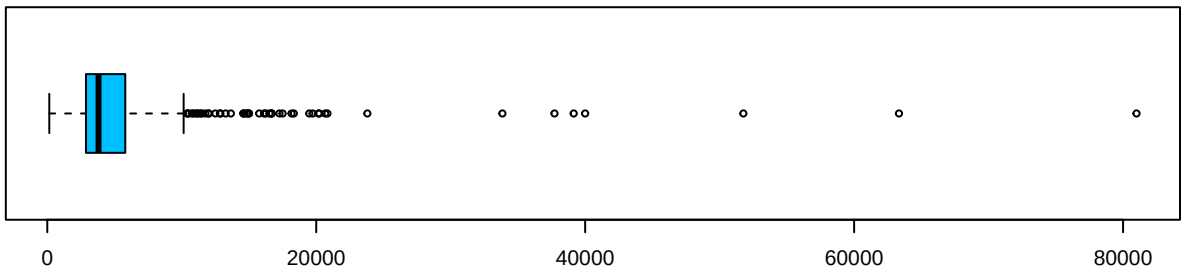
•Here 68.72% of their loans have been approved.

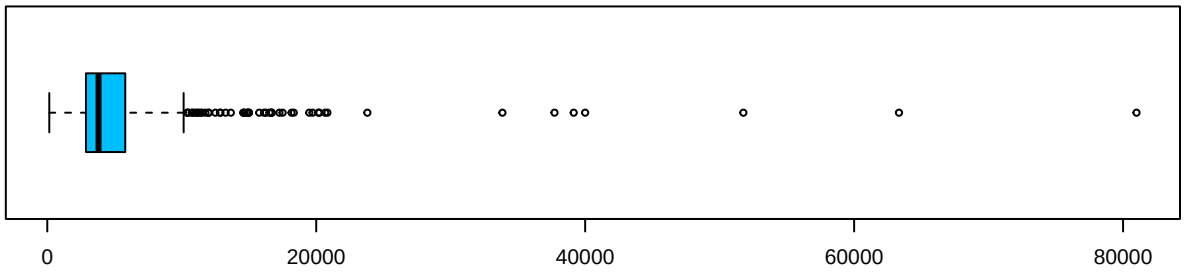


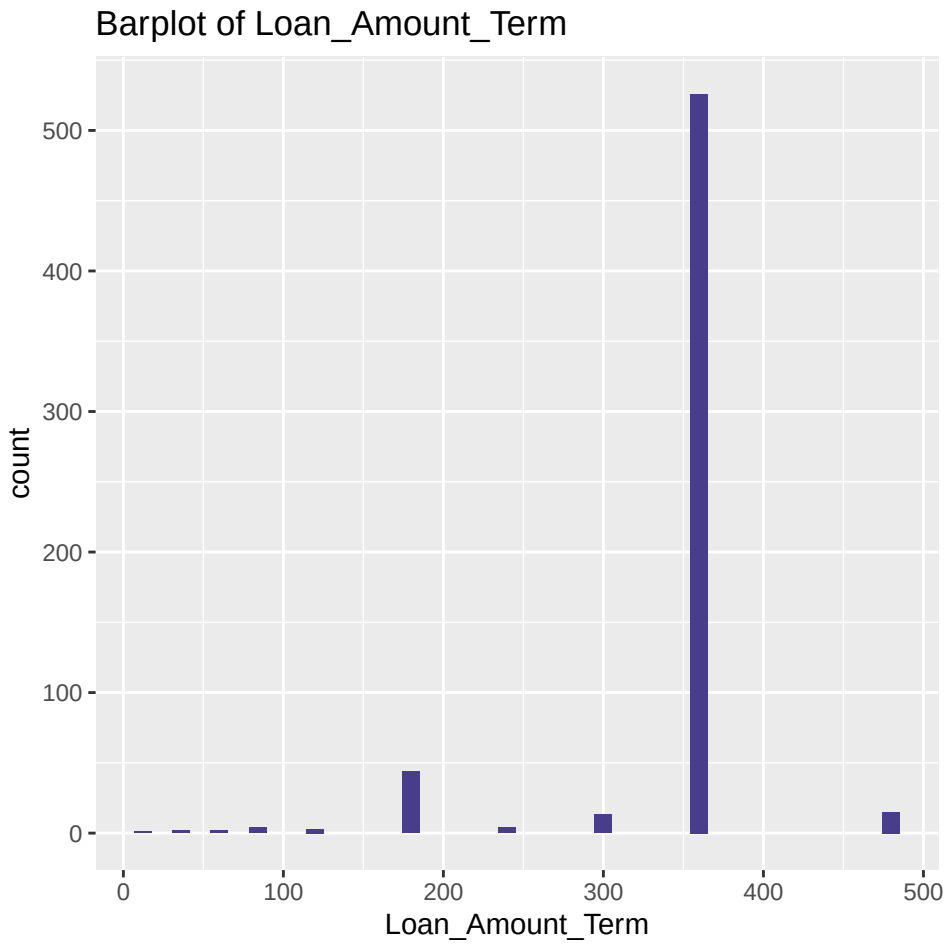
Observations:

- maximum (~ 56%) has no dependent in his/her family.
- maximum (~ 78%) of them are graduated.
- maximum (~ 77%) of their credit history meets guidelines.
- maximum (~ 69%) of them belong to semi-urban or urban area.


Boxplot of Applicant Income
0
20000
40000
60000
80000


Boxplot of Coapplicant Income
0
20000
40000
60000
80000


Boxplot of LoanAmount
0
20000
40000
60000
80000



Observations:

•For these data income of applicant mainly lies in the range of 10000-40000 with some outliers.

• For these data Income of co-applicant is lesser than income of applicant and is within the range of 5000-15000, again with some outliers.

• For these data amount of loan is mostly concentrated between 250-500.

• For these data aore than 80% of the applicants took loan for 360 months.

6.2 Bivariate Analysis :

Observations:

•We can see that there is no substantial difference between male and female approval rates.

• We can see that married applicants have a slightly higher chance of loan approval.

• We can see that there is no substantial difference in the loan approval rates for self-employed and not self-employed.

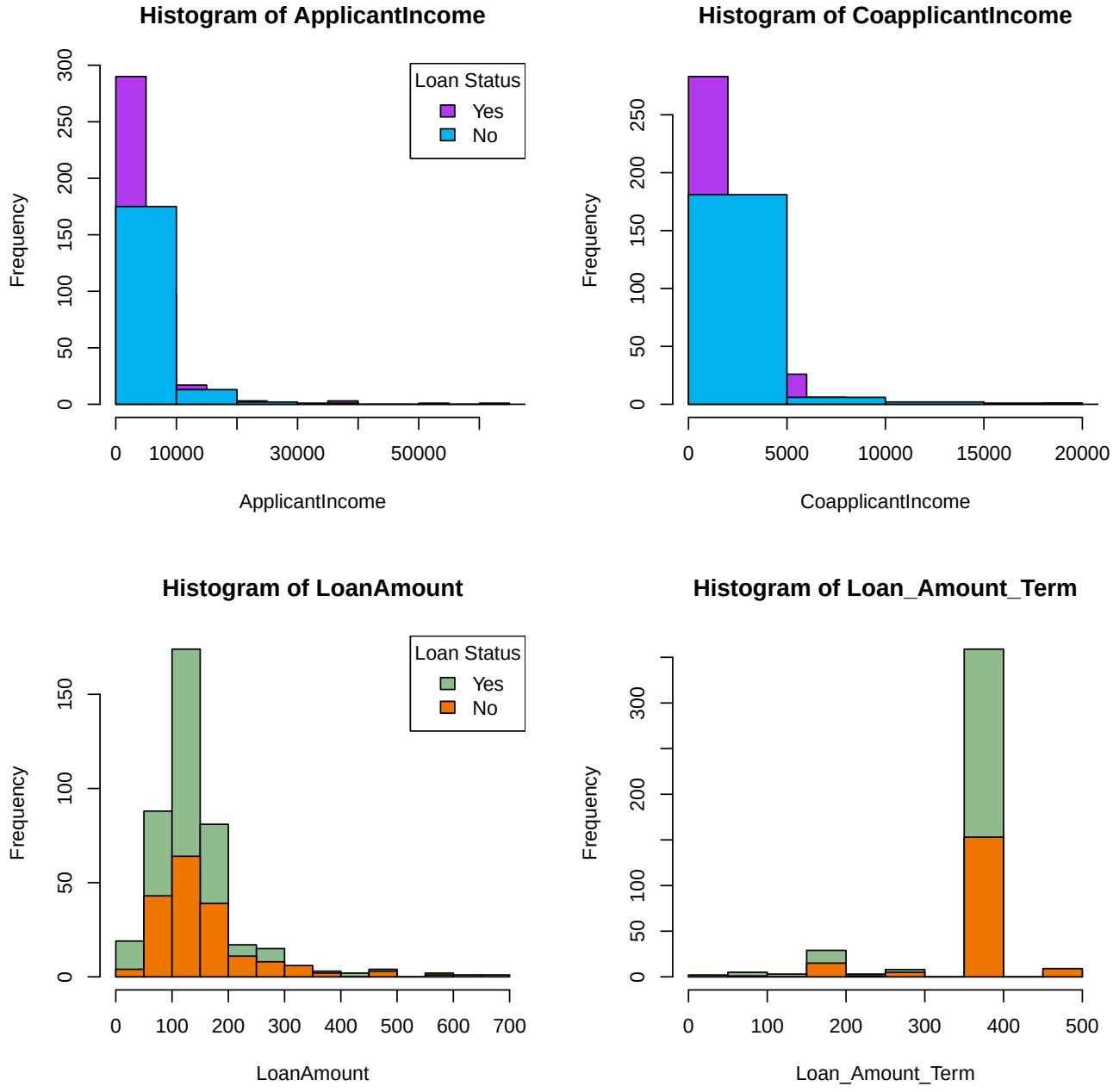
•Here Applicants with no dependence have slightly higher chances of approval.

•Here Graduates have higher chance of loan approval compared to non-graduates.

•From the plot we see applicants with properties in semi-urban areas have higher loan approval rates. Here we expect the applicants with credit history 1 have higher rates of approval. Let us check the claim by ploting grouped bar diagram.

- It is very clear applicants with credit history 1 have higher rates of approval; whereas with credit history 0 have negligible chance of acceptance.

Now we will plot histograms of quantitative variables by grouping the loan status.



Observations:

•For lower applicant income group there is a higher chances of rejection and same statement is applicable for lower co-applicant income group.

•We cannot draw any conclusion by observing the histograms of loan amount and number of months for which loan is given.

7 Applying Models:

At first we need to scale the continuous variables of both train and test data set.

Since our test set does not contain response variable, we will split the train set in 8:2and name them train_set and test_set. After checking accuracy on both these data sets, we will apply the best model on test set to perform our prediction.

7.1 Logistic Regression :

7.1.1 Theory:

Logistic Regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems

7.1.2 Performance:

- Misclassification rate on train_set:

```
[1] 0.2016293
```

- Misclassification rate on test_set:

```
[1] 0.1382114
```

7.2 Classification tree :

7.2.1 Theory:

Classification tree or decision tree is a multistage decision process. Rather than using the complete set of features jointly make an output decision, different subset of features is used at different levels of tree.

For a classification tree we predict that each observation belongs to the most commonly occurring class of training observations in the region which it belongs. A tree has its starting point called the root node. There are internal nodes connected by branches, from each node there emerges two branches for two types of decisions (Yes/No) And finally there are the terminal nodes which classify a feature vector into a class label.

7.2.2 Performance:

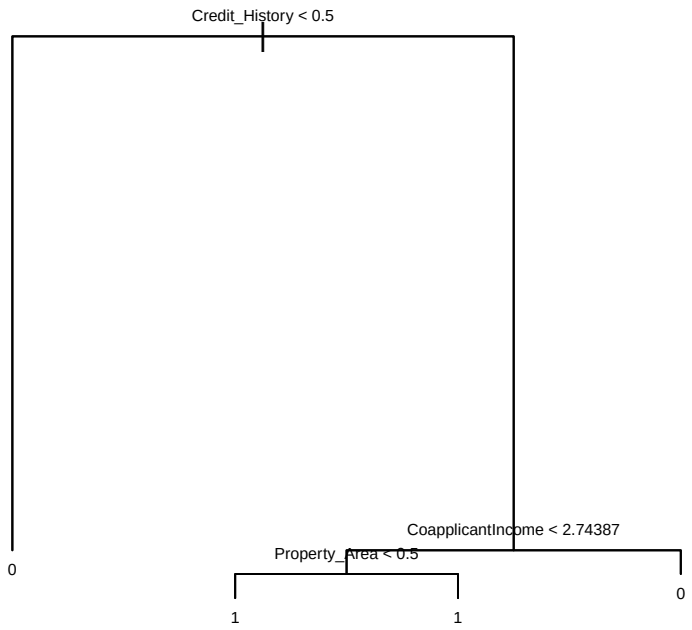
- Misclassification rate on train_set:

```
[1] 0.1856678
```

- Misclassification rate on test_set:

```
[1] 0.1856678
```

7.2.3 Diagram of Decision Tree:



7.3 Random Forest:

7.3.1 Theory:

Random Forest is a supervised learning algorithm that is used to classify as well as regression. However, it is mostly used for classification purposes. we know that a forest is composed of trees and that more trees means a more robust forest.

This classifier also use "MAJORITY VOTING RULE" to assign a feature vector to a particular class.

7.3.2 Performance:

- Misclassification rate on train_set:

```
randomForest 4.7-1
Type rfNews() to see new features/changes/bug fixes.


Attaching package:  'randomForest'
The following object is masked from 'package:ggplot2':
      margin
The following object is masked from 'package:gridExtra':
      combine


Call:
 randomForest(formula = as.factor(Loan_Status) ~ ., data = train_set,      ntree = 500, mtr
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3


        OOB estimate of  error rate: 22.61%
Confusion matrix:
   0   1 class.error
0 67  91  0.57594937
1 20 313  0.06006006
```

- Misclassification rate on test_set:

```
Call:
 randomForest(formula = as.factor(Loan_Status) ~ ., data = test_set,      mtry = sqrt(11),
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 13.01%
Confusion matrix:
   0  1 class.error
0 21 13  0.38235294
1  3 86  0.03370787
```

7.4 Bagging(Bootstrap aggregating):

7.4.1 Theory:

Bagging, also known as Bootstrap aggregating, is an ensemble learning technique that helps to improve the performance and accuracy of machine learning algorithms. It is used to deal with bias-variance trade-offs and reduces the variance of a prediction model. Bagging avoids overfitting of data and is used for both regression and classification models, specifically for decision tree algorithms.

7.4.2 Performance:

- Misclassification rate on train_set:

```
Call:
 randomForest(formula = as.factor(Loan_Status) ~ ., data = train_set,      mtry = 11, impor
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 11

        OOB estimate of  error rate: 24.24%
Confusion matrix:
   0   1 class.error
0 71  87   0.5506329
1 32 301   0.0960961
```

- Misclassification rate on test_set:

```
Call:
 randomForest(formula = as.factor(Loan_Status) ~ ., data = test_set,      mtry = 11, import
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 11

        OOB estimate of  error rate: 14.63%
Confusion matrix:
   0  1 class.error
0 22 12  0.35294118
1  6 83  0.06741573
```

7.5 Support Vector Machine (SVM):

7.5.1 Theory:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

7.5.2 Performance:

- Misclassification rate on train_set:

```
[1] 0.1905537
```

- Misclassification rate on test_set:

```
[1] 0.1300813
```

8 Prediction :

```
  [1] 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1
 [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 1 1 0 0 1 0 1 1 1 1
 [75] 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1
[112] 1 1 1 1 1 0 0 0 0 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0
[149] 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1
[186] 1 1 1 1 1 1 1 0 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
[223] 1 1 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 0 1 0 1 0 1 0 1 1 1 1 0 1 1 1 1 0 0 1 1
[260] 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 0 1 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1
[297] 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
[334] 1 1 0 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

predicted_loan_status
  0   1
 72 295
```

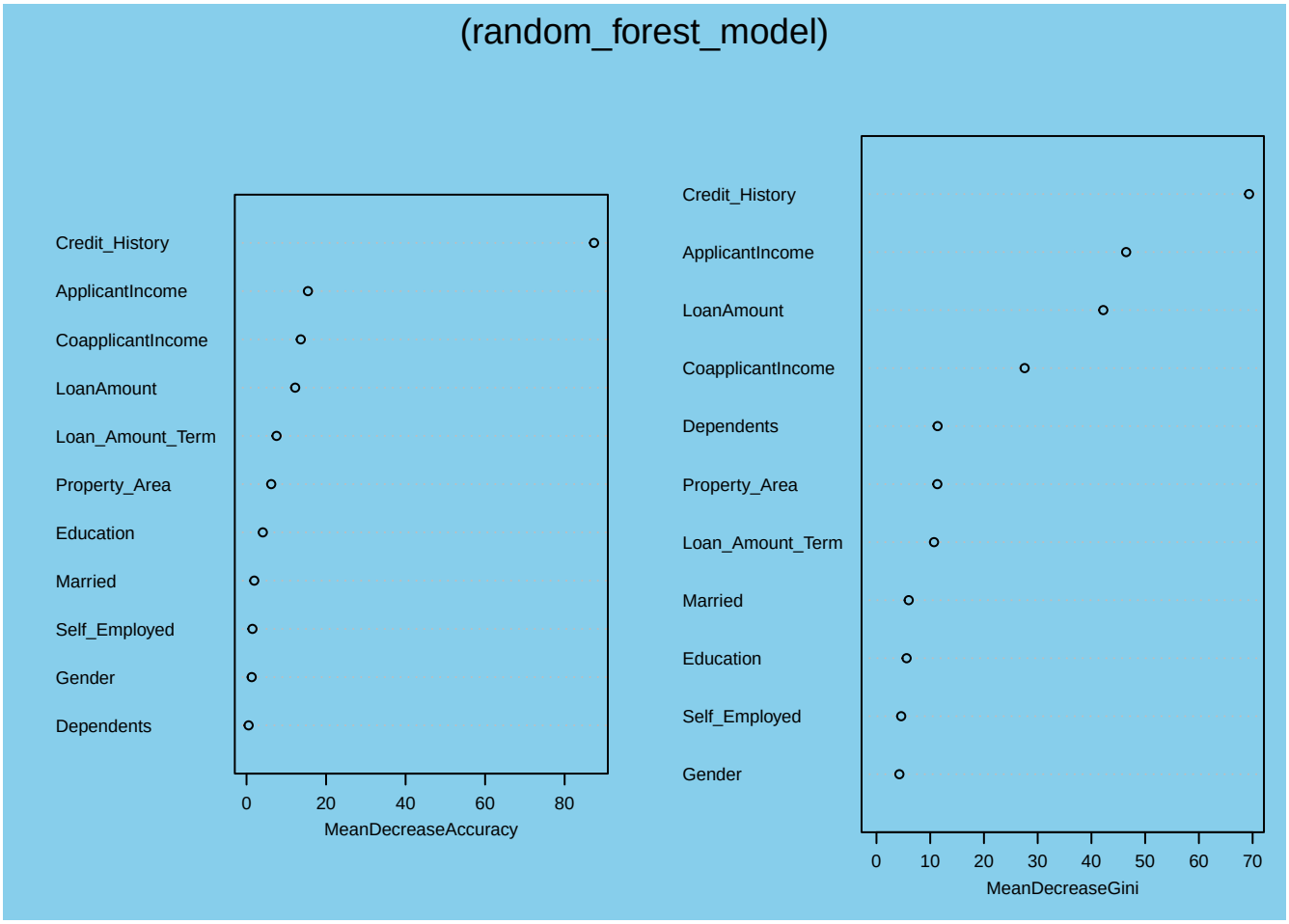
From the above output we get the frequency of loan_status for test data.

So now we checking for the 100th applicant in test data should be given a loan he/she applied for.

```
[1] 1
```

Therefore the output is 1, that means "Yes" according to our notation. So 100th applicant in test data set should be given the loan he applied for based on the information provided.

9 Conclusion:



From the graph we can see Credit History plays the most significant role in explaining variations among responses for both of the models and Applicant Income and Coapplicant Income and Loan Amount secured in the top most important features in both these models. Hence we conclude credit history of the individual, income of the applicant and income of the coapplicant and amount of loan play vital roles for predicting one's loan status that anyone applied for.

- Next, we have to compared the models discussed in this paper by their misclassification rates shown in the following table.

Model Name	Train_Set	Test_Set
Logistic Regression	0.2016	0.1382
Decision Tree	0.1857	0.1857
Random Forest	0.2261	0.1301
Bagging(Bootstrap aggregating)	0.2424	0.1463
SVM	0.1905	0.1308

Among all the models discussed in this report Random Forest shows the lowest misclassification rate in case of test set.

Hence we conclude that Random Forest is most preferable classification model for predicting the loan_status of given individuals on the basis of their provided information.

10 BIBLIOGRAPHY:

- MTH 552A Statistical And AI Techniques in DATA MINING - Dr. Amit Mitra's class Lectures and Lecture Notes
- An Introduction to Statistical Learning with Application in R, Second Edition, Hastie & Tibshirani.
- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013.
- https://www.kaggle.com/datasets/shaijudatascience/loan-prediction-practice-av-competition