Complete Study of Loan Prediction

R Studio

2021-13-07

Introduction

Estimating the probability that an individual would default on their loan, is useful for banks to decide whether to sanction a loan to the individual or not. We introduce an effective prediction technique that helps the banker to predict the credit risk for customers who have applied for loan.

Investors (lenders) provide loans to borrowers in exchange for the promise of repayment with interest. ... The interest rate is provided to us for each borrower. Therefore, so we'll address the second question indirectly by trying to predict if the borrower will repay the loan by its mature date or not.

Table of Content

➢ Problem Statement

Predict Loan Eligibility for Dream Housing Finance Company

Dream Housing Finance company deals in all kinds of home loans. They have presence across all urban, semi urban and rural areas. Customer first applies for home loan and after that company validates the customer eligibility for loan.

Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have provided a dataset to identify the customers segments that are eligible for loan amount so that they can specifically target these customers.

➢ Data Dictionary

Train file: CSV containing the customers for whom loan eligibility is known as 'Loan_Status'

Variable	**Description**
Loan_ID	Unique Loan ID
Gender	Male/ Female
Married	Applicant married (Y/N)
Dependents	Number of dependents
Education	Applicant Education (Graduate/ Under Graduate)
Self_Employed	Self employed (Y/N)
ApplicantIncome	Applicant income
CoapplicantIncome	Coapplicant income
LoanAmount	Loan amount in thousands
Loan_Amount_Term	Term of loan in months
Credit_History	credit history meets guidelines
Property_Area	Urban/ Semi Urban/ Rural
Loan_Status	(Target) Loan approved (Y/N)

Test file: CSV containing the customer information for whom loan eligibility is to be predicted

Variable	**Description**
Loan_ID	Unique Loan ID
Gender	Male/ Female
Married	Applicant married (Y/N)
Dependents	Number of dependents

Education	Applicant Education (Graduate/ Under Graduate)
Self_Employed	Self employed (Y/N)
ApplicantIncome	Applicant income
CoapplicantIncome	Coapplicant income
LoanAmount	Loan amount in thousands
Loan_Amount_Term	Term of loan in months
Credit_History	credit history meets guidelines
Property_Area	Urban/ Semi Urban/ Rural

➢ Evaluation Metric

Your model performance will be evaluated on the basis of your prediction of loan status for the test data (test.csv), which contains similar data-points as train except for the loan status to be predicted. Your submission needs to be in the format as shown in sample submission.

We at our end, have the actual loan status for the test dataset, against which your predictions will be evaluated. We will use the **Accuracy** value to judge your response.

➢ Tools and Techniques

We have used the following Analytical techniques / methodology for analysing the Data

1. Summary of Statistics for each variable
2. Using Graphs and Box Plots to visually represent them
3. Identification of significant Metrological factors through correlation and regression methodology
4. Using Multiple Linear Regression & Neural Network for Model Development
5. Tools used: R & Excel
6. Techniques: Box Plot, Histogram, Bar Chart, Line Chart, Visual Clues, Correlation Matrix,
7. Multiple Linear Regression, Artificial Neural Network
8. We have used R Programming environment and Microsoft Excel for our analysis

➢ Analytics approach

The Analytical Approach will involve the following (not necessarily in the order) activities:

1. Data extraction from Primary Data source
2. Data cleaning and data preparation
3. Study each of the variables by exploring the data
4. Study the variables for its relevance for the study
5. Identifying Y variable(s).

6. Division of data into train and test
7. Model Development
8. Final Model
9. Model Validation & Model Validation on Test
10. Intervention Strategies and recommendations

```
library(readxl)
Train_Loan_Prediction <- read_excel("Train_Loan_Prediction.xlsx")
View(Train_Loan_Prediction)

attach(Train_Loan_Prediction)
summary(Train_Loan_Prediction)
```

```
##    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.:100.2   1st Qu.:360      1st Qu.:1.0000   Class :character
##  Median :129.0   Median :360      Median :1.0000   Mode  :character
##  Mean   :146.4   Mean   :342      Mean   :0.8422
##  3rd Qu.:164.8   3rd Qu.:360      3rd Qu.:1.0000
##  Max.   :700.0   Max.   :480      Max.   :1.0000
##                  NA's   :14       NA's   :50
##  Loan_Status
##  Length:614
##  Class :character
##  Mode  :character
##
##
```

```
##
##

str(Train_Loan_Prediction)

## tibble [614 x 13] (S3: tbl_df/tbl/data.frame)
##  $ Loan_ID          : chr [1:614] "LP001002" "LP001003" "LP001005" "LP001006" ...
##  $ Gender           : chr [1:614] "Male" "Male" "Male" "Male" ...
##  $ Married          : chr [1:614] "No" "Yes" "Yes" "Yes" ...
##  $ Dependents       : chr [1:614] "0" "1" "0" "0" ...
##  $ Education        : chr [1:614] "Graduate" "Graduate" "Graduate" "Not Graduate" ...
##  $ Self_Employed    : chr [1:614] "No" "No" "Yes" "No" ...
##  $ ApplicantIncome  : num [1:614] 5849 4583 3000 2583 6000 ...
##  $ CoapplicantIncome: num [1:614] 0 1508 0 2358 0 ...
##  $ LoanAmount       : num [1:614] 146 128 66 120 141 ...
##  $ Loan_Amount_Term : num [1:614] 360 360 360 360 360 360 360 360 360 360 ...
##  $ Credit_History   : num [1:614] 1 1 1 1 1 1 1 0 1 1 ...
##  $ Property_Area    : chr [1:614] "Urban" "Rural" "Urban" "Urban" ...
##  $ Loan_Status      : chr [1:614] "Y" "N" "Y" "Y" ...

dim(Train_Loan_Prediction)

## [1] 614  13
```

Let’s try with Smart EDA

```
library(SmartEDA)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

library(ISLR)


ExpData(Train_Loan_Prediction,type = 1)

##                                           Descriptions     Value
## 1                                   Sample size (nrow)       614
## 2                              No. of variables (ncol)        13
## 3                    No. of numeric/interger variables         5
## 4                              No. of factor variables         0
## 5                                No. of text variables         8
## 6                             No. of logical variables         0
## 7                          No. of identifier variables         1
## 8                                No. of date variables         0
## 9             No. of zero variance variables (uniform)         0
## 10               %. of variables having complete cases 53.85% (7)
```

```
## 11   %. of variables having >0% and <50% missing cases 46.15% (6)
## 12 %. of variables having >=50% and <90% missing cases     0% (0)
## 13          %. of variables having >=90% missing cases     0% (0)
```

```
ExpNumStat(Train_Loan_Prediction,by ="A",Outlier=TRUE,round= 2)
```

```
##              Vname Group  TN nNeg nZero nPos NegInf PosInf NA_Value
## 1   ApplicantIncome   All 614    0     0  614      0      0        0
## 2 CoapplicantIncome   All 614    0   273  341      0      0        0
## 4  Loan_Amount_Term   All 614    0     0  600      0      0       14
## 3        LoanAmount   All 614    0     0  614      0      0        0
##   Per_of_Missing       sum min   max    mean median      SD   CV     IQR
## 1           0.00 3317724.0 150 81000 5403.46 3812.5 6109.04 1.13 2917.50
## 2           0.00  995444.9   0 41667 1621.25 1188.5 2926.25 1.80 2297.25
## 4           2.28  205200.0  12   480  342.00  360.0   65.12 0.19    0.00
## 3           0.00   89892.4   9   700  146.40  129.0   84.04 0.57   64.50
##   Skewness Kurtosis   LB.25%   UB.75% nOutliers
## 1     6.52    60.04 -1498.75 10171.25        50
## 2     7.47    84.26 -3445.88  5743.12        18
## 4    -2.36     6.61   360.00   360.00        88
## 3     2.72    10.80     3.50   261.50        41
```

#OBSERVATIONS:
1. Dependent Variable: Loan status
2. All independent variables are numeric or integer except ID Gender, Martial status, Education, Employment & loan status is categorical
4. Max value for Loan amount is very high compared to 3rd Qu - Possibility of outliers?
5. Similar outlier possibility found in ApplicantIncome & CoapplicantIncome
6. Missing Values present in Loan_Amount_Term & LoanAmount

7.As there is missing Values in Dependent variable (Loan Amount),   have to treat it

```
library(Hmisc)

library(tidyverse)

library(dplyr)
```

##Lets have close Data Introduction

##Lets Change Male=1 & Female =2

```
Train_Loan_Prediction$Gender[Train_Loan_Prediction$Gender=="Male"] <- "1"
Train_Loan_Prediction$Gender[Train_Loan_Prediction$Gender=="Female"] <- "2"

Train_Loan_Prediction$Gender=as.factor(Train_Loan_Prediction$Gender)
```

##Lets Change Yes=1 & NO =0

```
Train_Loan_Prediction$Married[Train_Loan_Prediction$Married=="Yes"] <- "1"
Train_Loan_Prediction$Married[Train_Loan_Prediction$Married=="No"] <- "0"

Train_Loan_Prediction$Married=as.factor(Train_Loan_Prediction$Married)

Train_Loan_Prediction$Dependents=as.factor(Train_Loan_Prediction$Dependents)
Train_Loan_Prediction$Credit_History=as.factor(Train_Loan_Prediction$Credit_History)

str(Train_Loan_Prediction)
```

```
## tibble [614 x 13] (S3: tbl_df/tbl/data.frame)
##  $ Loan_ID          : chr [1:614] "LP001002" "LP001003" "LP001005" "LP001006" ...
##  $ Gender           : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Married          : Factor w/ 2 levels "0","1": 1 2 2 2 1 2 2 2 2 2 ...
##  $ Dependents       : Factor w/ 4 levels "0","1","2","3+": 1 2 1 1 1 3 1 4 3 2 ...
##  $ Education        : chr [1:614] "Graduate" "Graduate" "Graduate" "Not Graduate" ...
##  $ Self_Employed    : chr [1:614] "No" "No" "Yes" "No" ...
##  $ ApplicantIncome  : num [1:614] 5849 4583 3000 2583 6000 ...
##  $ CoapplicantIncome: num [1:614] 0 1508 0 2358 0 ...
##  $ LoanAmount       : num [1:614] 146 128 66 120 141 ...
##  $ Loan_Amount_Term : num [1:614] 360 360 360 360 360 360 360 360 360 360 ...
##  $ Credit_History   : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 2 2 ...
##  $ Property_Area    : chr [1:614] "Urban" "Rural" "Urban" "Urban" ...
##  $ Loan_Status      : chr [1:614] "Y" "N" "Y" "Y" ...
```

```
summary(Train_Loan_Prediction)
```

```
##    Loan_ID           Gender    Married    Dependents  Education
##  Length:614         1   :489   0   :213   0   :345   Length:614
##  Class :character   2   :112   1   :398   1   :102   Class :character
##  Mode  :character   NA's: 13   NA's:  3   2   :101   Mode  :character
##                                           3+  : 51
##                                           NA's: 15
##
##
##  Self_Employed      ApplicantIncome CoapplicantIncome   LoanAmount
##  Length:614         Min.   :  150   Min.   :    0     Min.   :  9.0
##  Class :character   1st Qu.: 2878   1st Qu.:    0     1st Qu.:100.2
##  Mode  :character   Median : 3812   Median : 1188     Median :129.0
##                     Mean   : 5403   Mean   : 1621     Mean   :146.4
##                     3rd Qu.: 5795   3rd Qu.: 2297     3rd Qu.:164.8
##                     Max.   :81000   Max.   :41667     Max.   :700.0
##
##  Loan_Amount_Term Credit_History Property_Area      Loan_Status
##  Min.   : 12      0   : 89       Length:614         Length:614
```

```
##  1st Qu.:360      1   :475       Class :character   Class :character
##  Median :360      NA's: 50       Mode  :character   Mode  :character
##  Mean   :342
##  3rd Qu.:360
##  Max.   :480
##  NA's   :14
```

```
summary(Credit_History)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  1.0000  1.0000  0.8422  1.0000  1.0000      50
```

```
#Treatment of Missing Data
#Replace LoanAmount missing values from mean


Train_Loan_Prediction$LoanAmount[which(is.na(Train_Loan_Prediction$LoanAmount))
]=mean(Train_Loan_Prediction$LoanAmount, na.rm = TRUE)
summary(LoanAmount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     9.0   100.2   129.0   146.4   164.8   700.0
```

```
str(Loan_Status)
```

```
##  chr [1:614] "Y" "N" "Y" "Y" "Y" "Y" "Y" "N" "Y" "N" "Y" "Y" "Y" "N" "Y" .

..
```

```
#converting in Loan Status, Factors from character

Train_Loan_Prediction$Loan_Status=as.factor(Train_Loan_Prediction$Loan_Status)

Loan_Status

summary(Loan_Status)
```

```
##    Length     Class      Mode
##       614 character character
```

```
dim(Train_Loan_Prediction)
```

```
## [1] 614  13
```

```
summary(Train_Loan_Prediction)
```

```
##    Loan_ID           Gender    Married    Dependents  Education
##  Length:614         1   :489   0   :213   0   :345   Length:614
##  Class :character   2   :112   1   :398   1   :102   Class :character
##  Mode  :character   NA's: 13   NA's:  3   2   :101   Mode  :character
```

```
##                                   3+  : 51
##                                   NA's: 15
##
##
##  Self_Employed      ApplicantIncome CoapplicantIncome   LoanAmount
##  Length:614         Min.   :  150   Min.   :    0     Min.   :  9.0
##  Class :character   1st Qu.: 2878   1st Qu.:    0     1st Qu.:100.2
##  Mode  :character   Median : 3812   Median : 1188     Median :129.0
##                     Mean   : 5403   Mean   : 1621     Mean   :146.4
##                     3rd Qu.: 5795   3rd Qu.: 2297     3rd Qu.:164.8
##                     Max.   :81000   Max.   :41667     Max.   :700.0
##
##  Loan_Amount_Term Credit_History Property_Area      Loan_Status
##  Min.   : 12      0   : 89       Length:614         N:192
##  1st Qu.:360      1   :475       Class :character   Y:422
##  Median :360      NA's: 50       Mode  :character
##  Mean   :342
##  3rd Qu.:360
##  Max.   :480
##  NA's   :14
```

```
##@@@@@@@@@@@@@@@@@
library(SmartEDA)
library(ISLR)
ExpCatStat(Train_Loan_Prediction, Target = "Loan_Status")
```

```
## Warning in chisq.test(tb): Chi-squared approximation may be incorrect
```

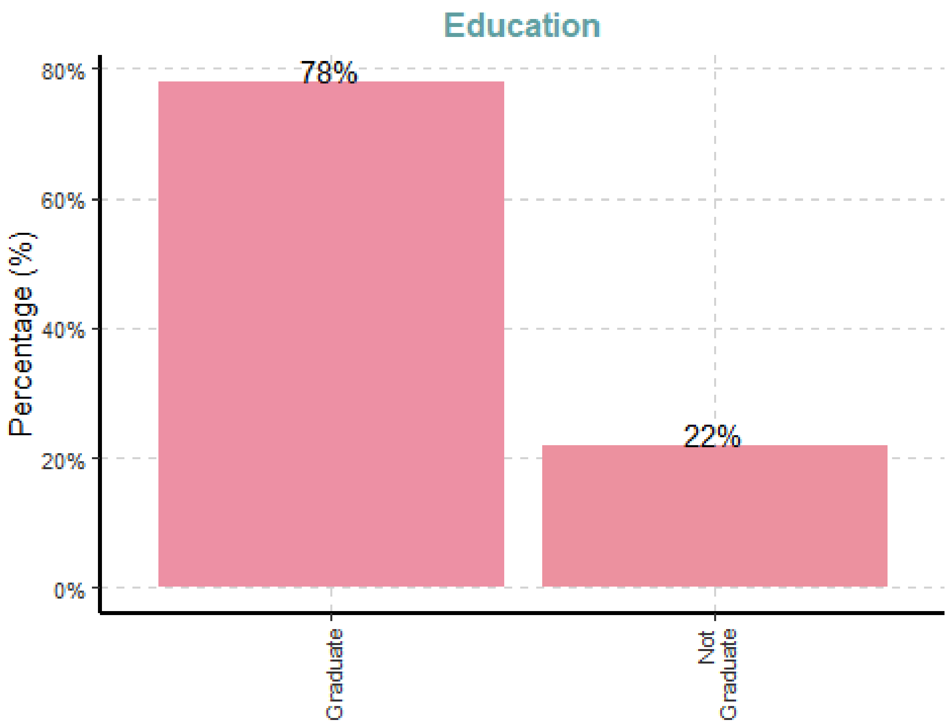
```
##            Variable      Target Unique Chi-squared p-value df IV Value
## 1         Education Loan_Status      2       4.091   0.043  1        0
## 2     Self_Employed Loan_Status      3       0.000   1.000  1        0
## 3     Property_Area Loan_Status      3      12.298   0.002  2        0
## 4            Gender Loan_Status      3       0.140   0.709  1        0
## 5           Married Loan_Status      3       4.475   0.034  1        0
## 6        Dependents Loan_Status      5       3.158   0.368  3        0
## 7    Credit_History Loan_Status      3     174.637   0.000  1        0
## 8  Loan_Amount_Term Loan_Status     11      14.013   0.122  9        0
## 9   ApplicantIncome Loan_Status     10       4.318   0.889  9        0
## 10 CoapplicantIncome Loan_Status      6       3.008   0.699  5        0
## 11       LoanAmount Loan_Status     10       5.117   0.824  9        0
##    Cramers V Degree of Association Predictive Power
## 1       0.08             Very Weak   Not Predictive
## 2       0.00             Very Weak   Not Predictive
## 3       0.14                  Weak   Not Predictive
## 4       0.02             Very Weak   Not Predictive
## 5       0.09                  Weak   Not Predictive
## 6       0.07             Very Weak   Not Predictive
```

```
## 7       0.56              Strong   Not Predictive
## 8       0.15                Weak   Not Predictive
## 9       0.08           Very Weak   Not Predictive
## 10       0.07           Very Weak   Not Predictive
## 11       0.09                Weak   Not Predictive
```

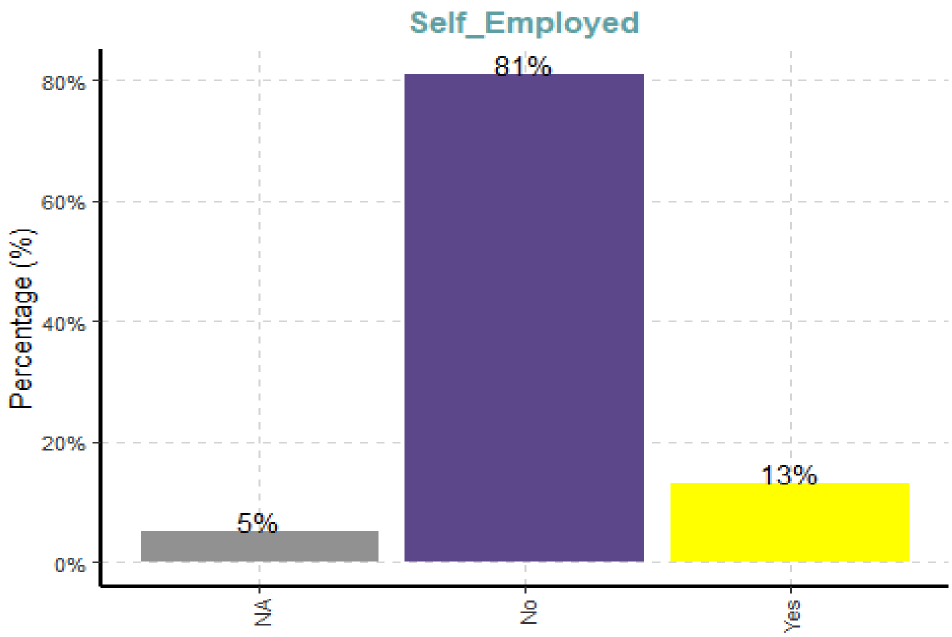
Its show that credit_History variables has strong Degree of Association

ExpCatViz(Train_Loan_Prediction)

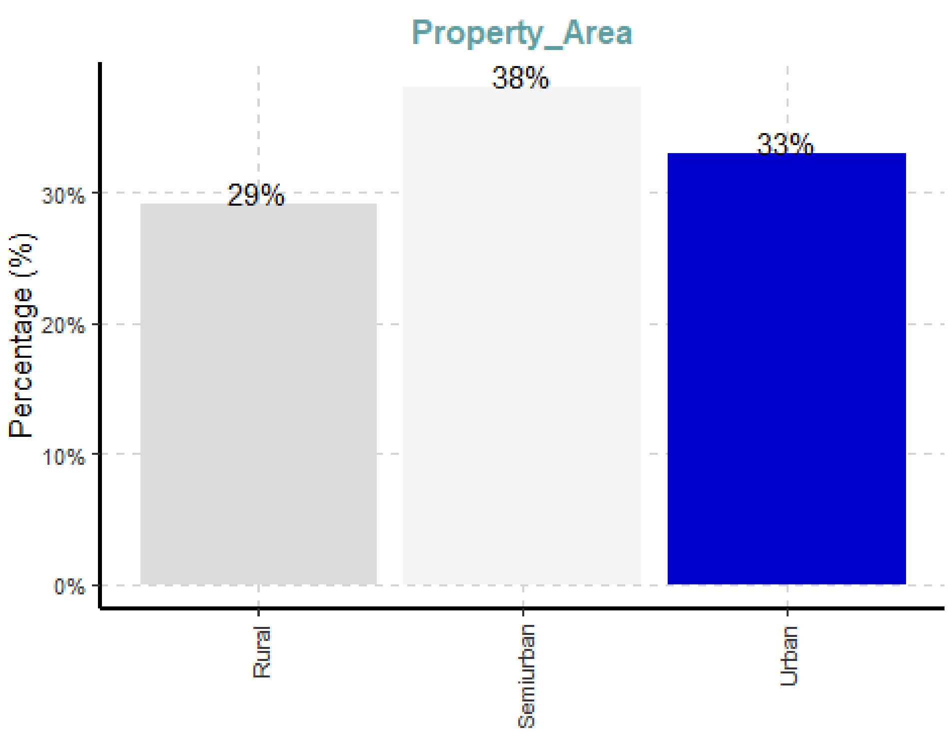
[[1]] 78% if total data is educated



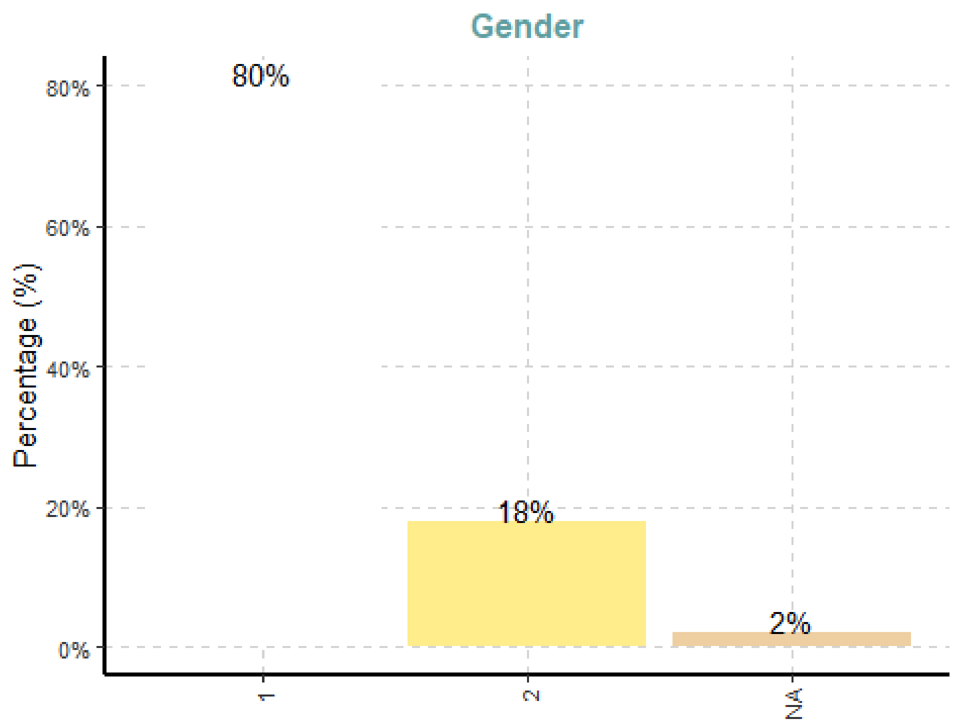
[[1]] Major Part (81%) is Job class and 5% data is Not Available



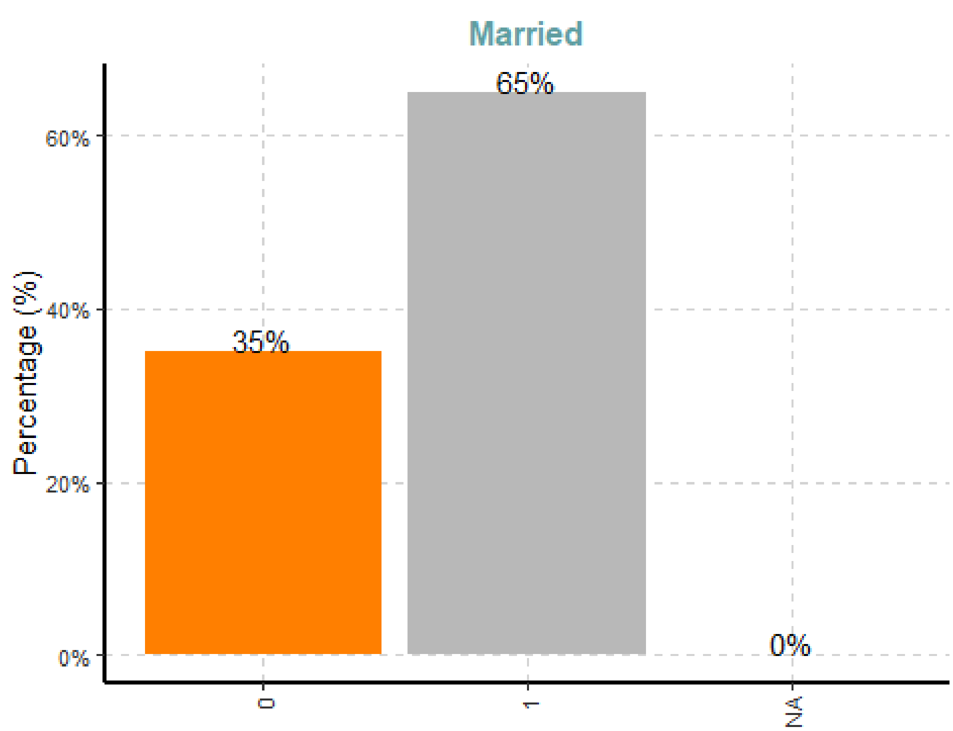
[[1]] Property Area of data is almost equally, hence we can say that bank customers are in all over area, and are positive part of bank.



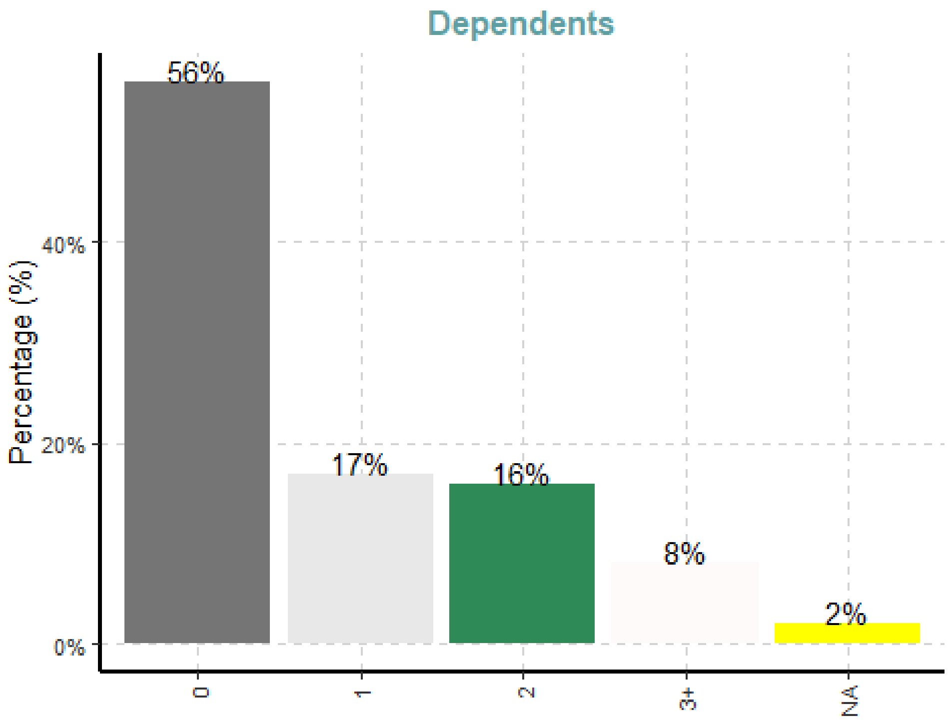
[[1]] Males are 80% of total data, whereas we do not have 2% data.



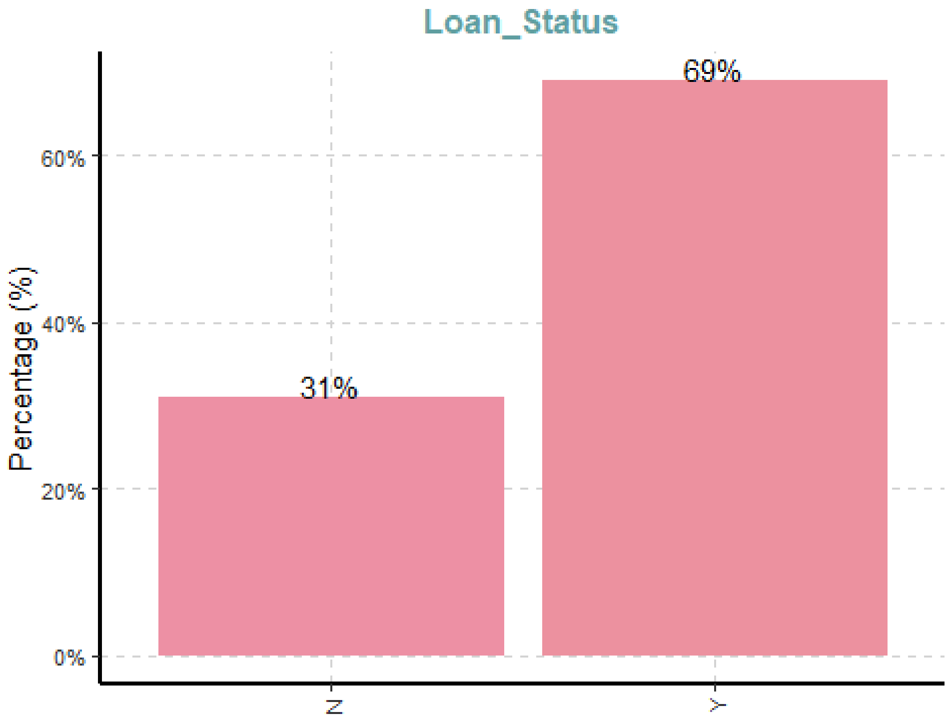
[[1]] 65% of the customers are married.



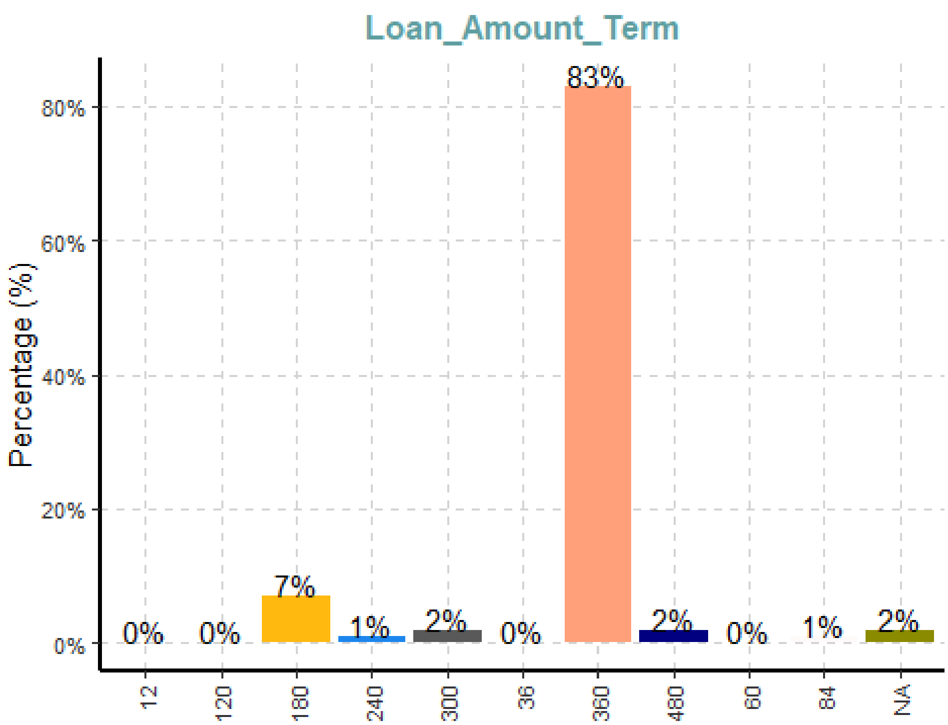
[[1]] 56% have no Dependents, which is good as no dependents will be more flexible is paying back the loan EMI.



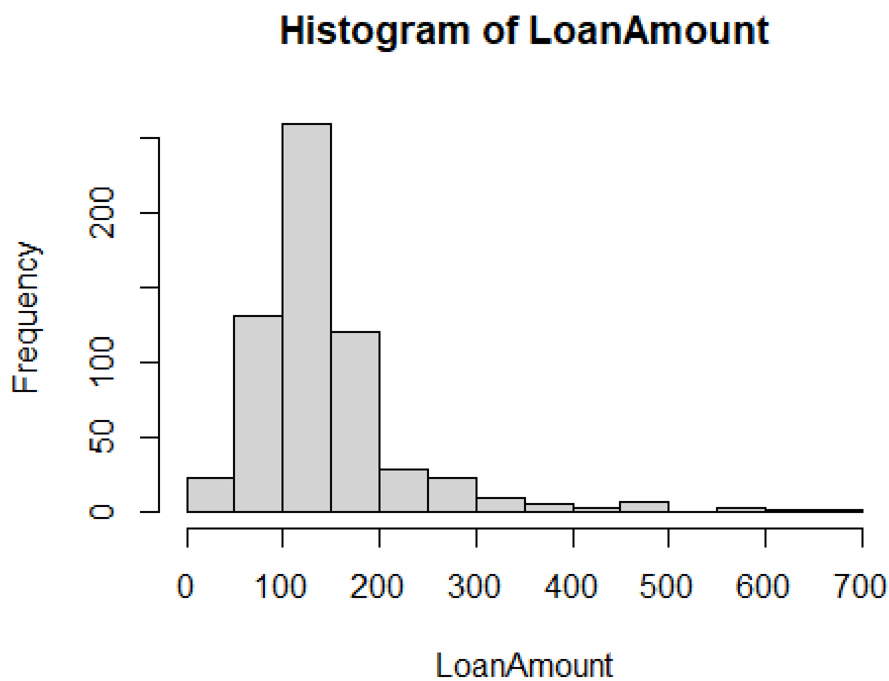
[[1]] 69% Customer Loan are approved



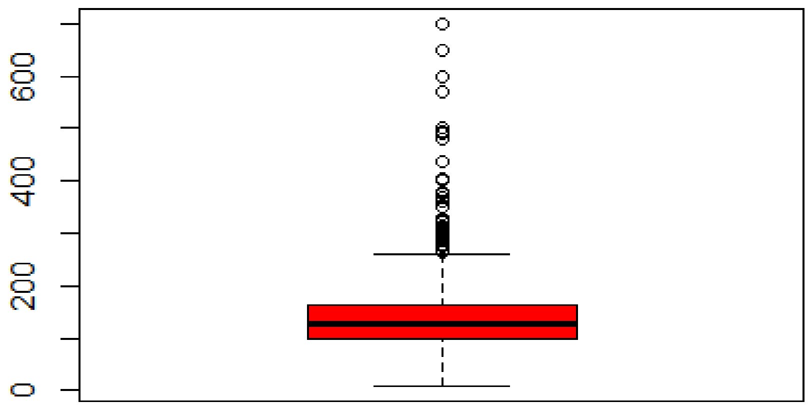
[[1]] 83% Customer Loan are for 30 years only 7% for 15 years,2% for 25 & 40 years, and 1% for 20 & 7 years



```
##OBSERVATION
#In total observation we have 79% graduate
#86% Job class people rest Self-employed
#we have 29% rural area Property, 39% semi urban, 32% Urban

# Building Histogram to understand
hist(LoanAmount)
```

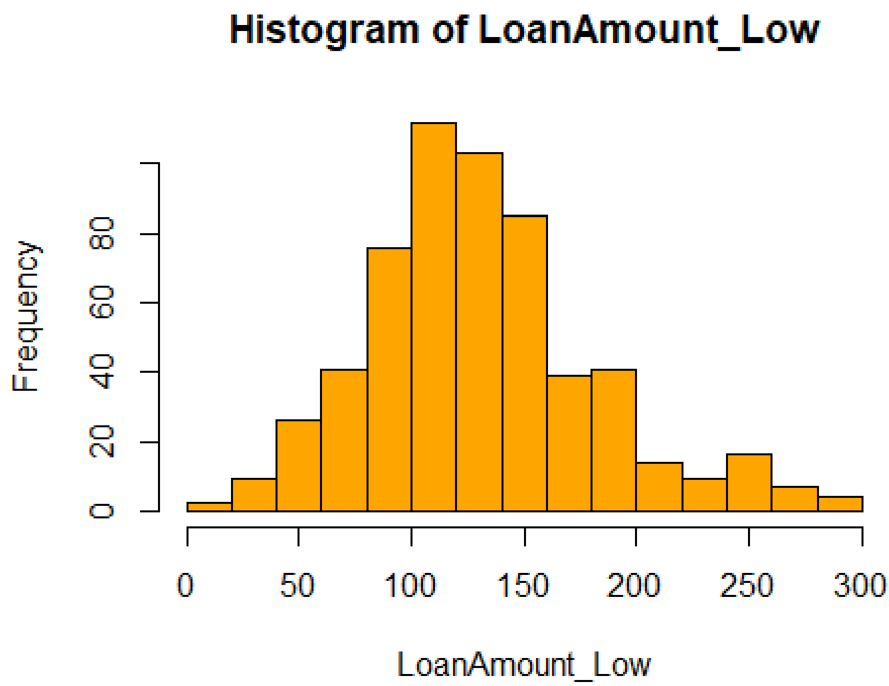



```
#Observation: Outlier(s) affecting histogram

# Building Boxplot to understand was is affecting


boxplot(LoanAmount,col = "red")
```

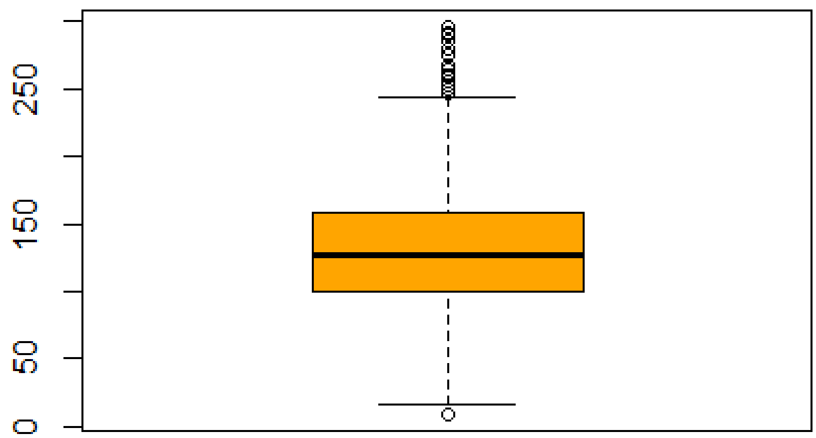


#OBSERVATIONS:
Most of the Loan amount are at the low end - some outlier very far out
For now, let us examine only low loan Amount (< 300)

```
LoanAmount_Low = LoanAmount[LoanAmount < 300]
hist(LoanAmount_Low, col = "orange")
```




```
boxplot(LoanAmount_Low, col = "orange")
```

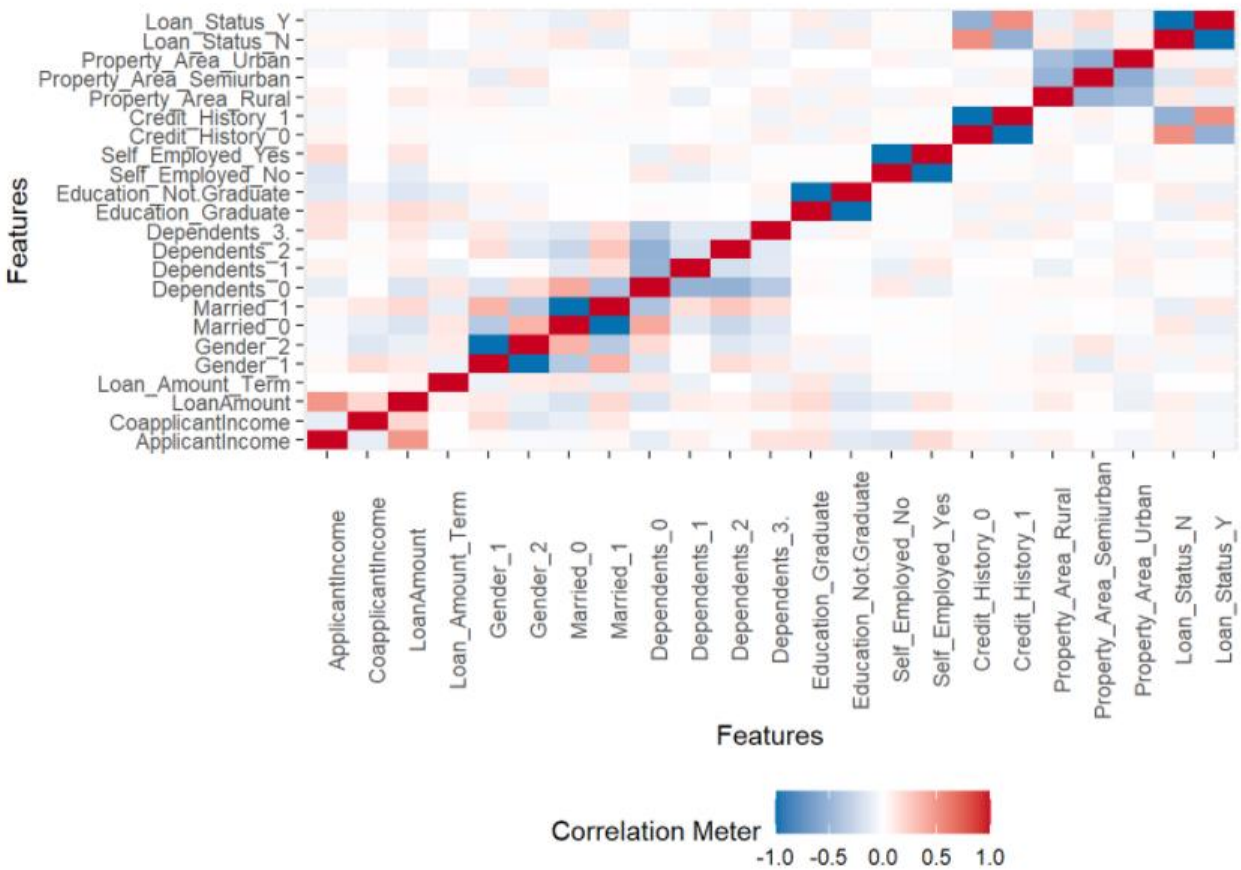



```
str(LoanAmount_Low)
```

```
##  num [1:584] 146 128 66 120 141 ...

str(LoanAmount)

##  num [1:614] 146 128 66 120 141 ...
```

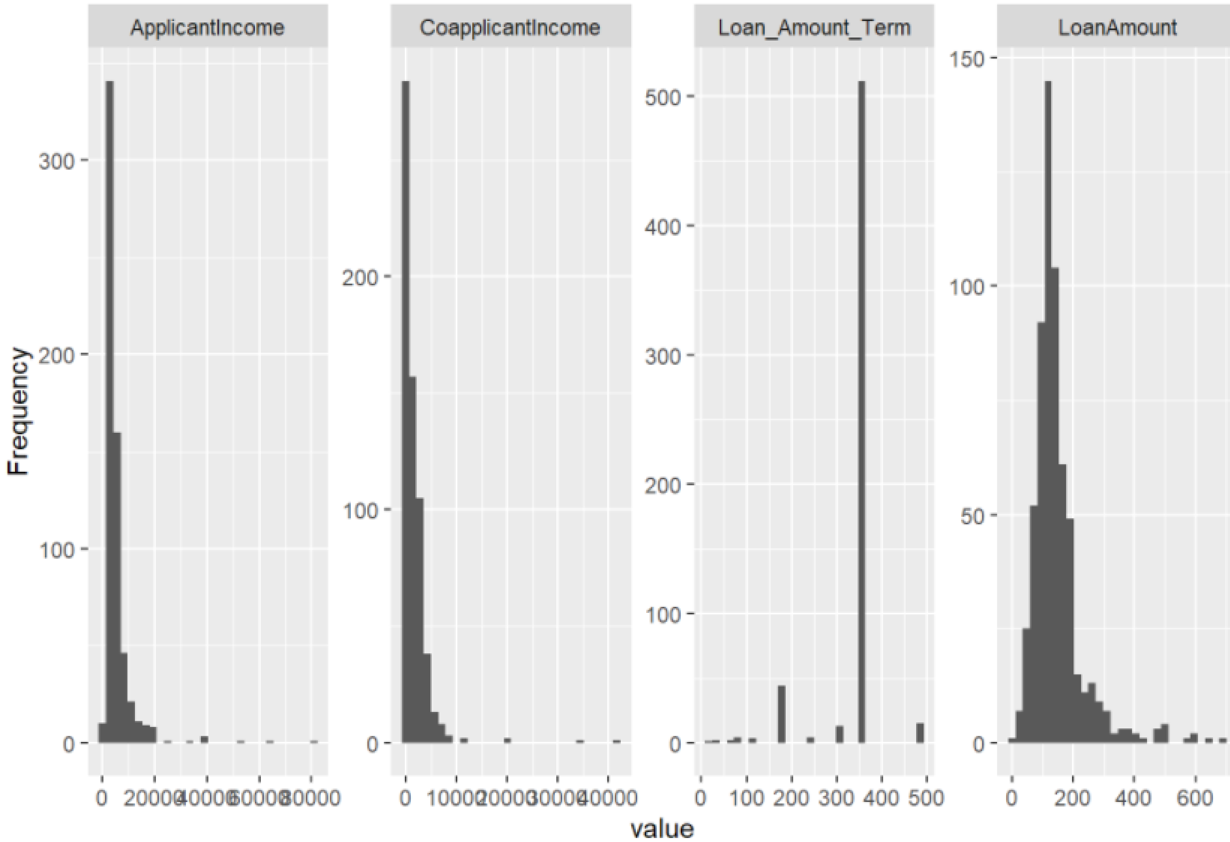
##30 observations are outliers hence, we cannot remove it

```
# Correlation check
library(DataExplorer)
plot_correlation(na.omit(Train_Loan_Prediction), maxcat = 5L)

## Loan_ID: 499 categories
```



#Visible that Loanstatus is correlated with credit history of customer

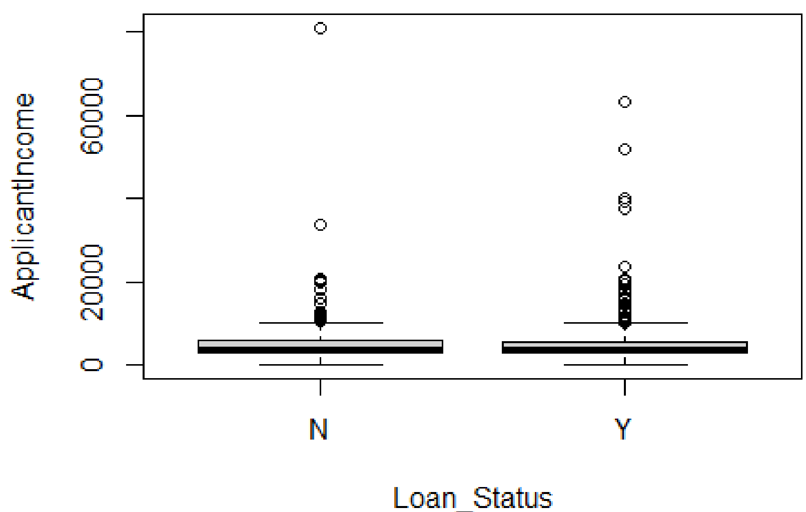
```
plot_histogram(Train_Loan_Prediction)
```



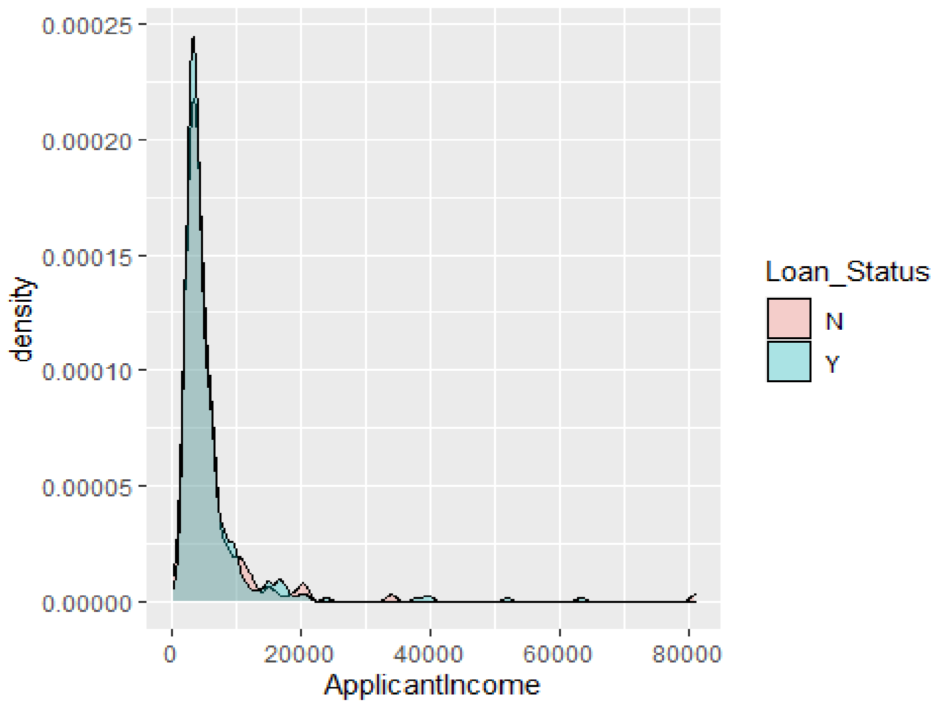

```
library(ggplot2)
attach(Train_Loan_Prediction)

## The following objects are masked from Train_Loan_Prediction (pos = 19):
##
##     ApplicantIncome, CoapplicantIncome, Credit_History, Dependents,
##     Education, Gender, Loan_Amount_Term, Loan_ID, Loan_Status,
##     LoanAmount, Married, Property_Area, Self_Employed

ggplot(Train_Loan_Prediction, aes(x = Credit_History)) +geom_density(aes(fill = Loan_
Status), alpha = 0.3)
```

Seems that Credit_History has major impact on the status of loan Status

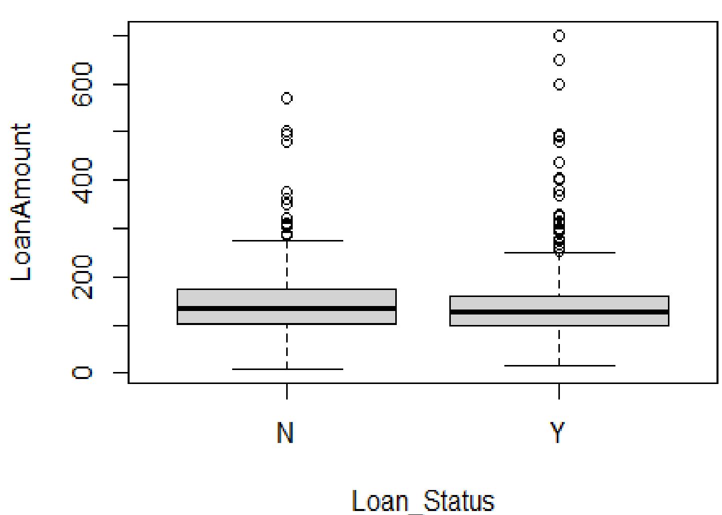
```
boxplot(ApplicantIncome~Loan_Status)
```



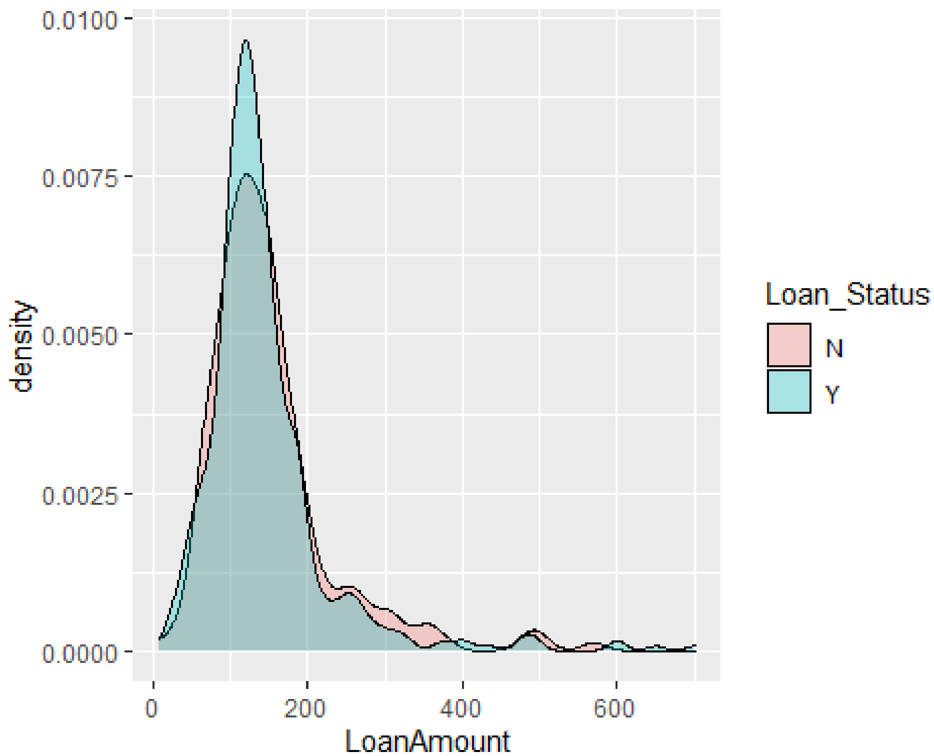
```
ggplot(Train_Loan_Prediction, aes(x = ApplicantIncome)) +geom_density(aes(fill = Loan_Status), alpha = 0.3)
```



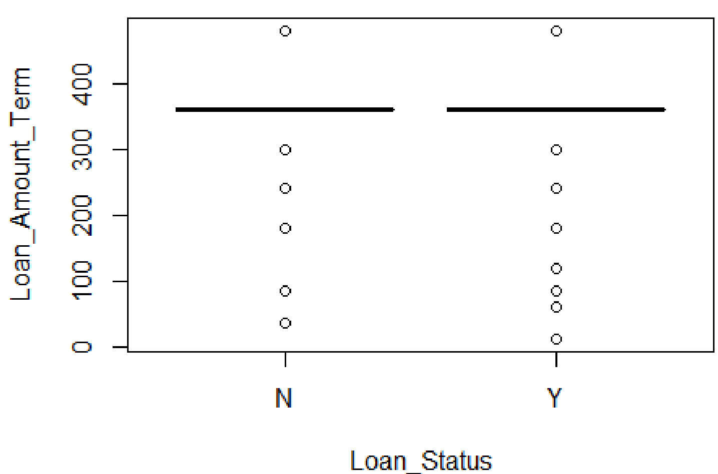

```
## Seems that applicant Income has no impact on the status of loan

boxplot(LoanAmount~Loan_Status)
```



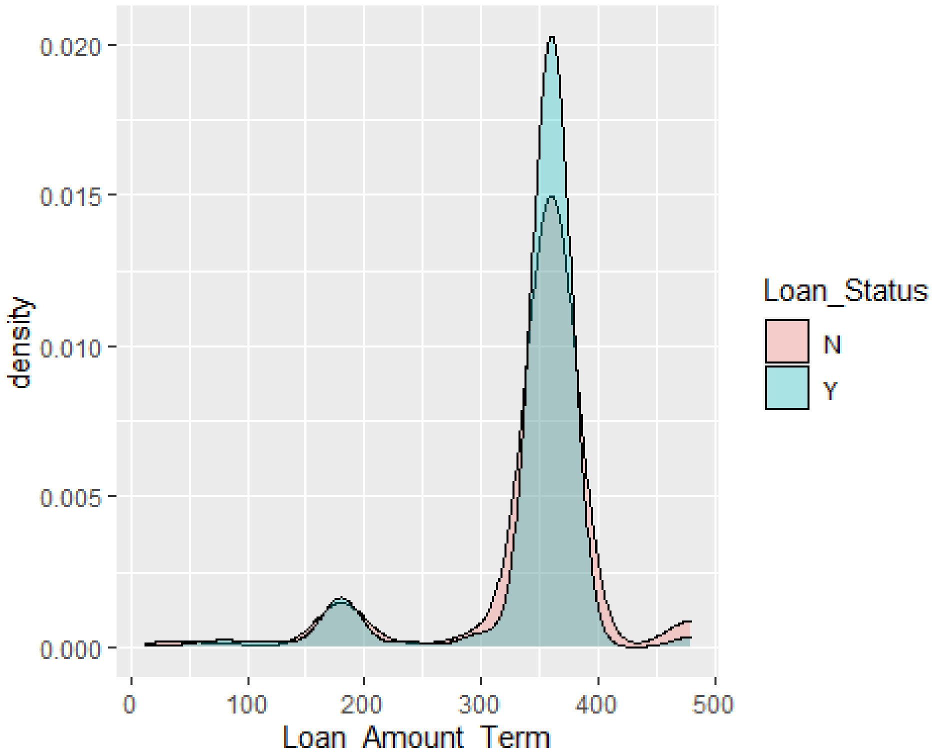
```
ggplot(Train_Loan_Prediction, aes(x = LoanAmount)) +geom_density(aes(fill = Loan_Status), alpha = 0.3)
```




```
## Seems that applicant Income has no impact on the status of loan

boxplot(Loan_Amount_Term~Loan_Status)
```



```
ggplot(Train_Loan_Prediction, aes(x = Loan_Amount_Term)) +geom_density(aes(fill =
Loan_Status), alpha = 0.3)

## Warning: Removed 14 rows containing non-finite values (stat_density).
```




```
## Seems that Loan_Amount_Term has no impact on the status of loan



library(readxl)
Test_Loan_Prediction <- read_excel("Test_Loan_Prediction.xlsx")
View(Test_Loan_Prediction)

##Lets Change Male=1 & Female =2
Test_Loan_Prediction$Gender[Test_Loan_Prediction$Gender=="Male"] <- "1"
Test_Loan_Prediction$Gender[Test_Loan_Prediction$Gender=="Female"] <- "2"

Test_Loan_Prediction$Gender=as.factor(Test_Loan_Prediction$Gender)

##Lets Change Yes=1 & NO =0

Test_Loan_Prediction$Married[Test_Loan_Prediction$Married=="Yes"] <- "1"
Test_Loan_Prediction$Married[Test_Loan_Prediction$Married=="No"] <- "0"

Test_Loan_Prediction$Married=as.factor(Test_Loan_Prediction$Married)
```

```
Train_Loan_Prediction$Dependents=as.factor(Train_Loan_Prediction$Dependents)
Train_Loan_Prediction$Credit_History=as.factor(Train_Loan_Prediction$Credit_History)

str(Test_Loan_Prediction)

## tibble [367 x 12] (S3: tbl_df/tbl/data.frame)
##  $ Loan_ID          : chr [1:367] "LP001015" "LP001022" "LP001031" "LP001035" ...
##  $ Gender           : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 2 1 1 1 ...
##  $ Married          : Factor w/ 2 levels "0","1": 2 2 2 2 1 2 1 2 2 1 ...
##  $ Dependents       : chr [1:367] "0" "1" "2" "2" ...
##  $ Education        : chr [1:367] "Graduate" "Graduate" "Graduate" "Graduate" ...
##  $ Self_Employed    : chr [1:367] "No" "No" "No" "No" ...
##  $ ApplicantIncome  : num [1:367] 5720 3076 5000 2340 3276 ...
##  $ CoapplicantIncome: num [1:367] 0 1500 1800 2546 0 ...
##  $ LoanAmount       : num [1:367] 110 126 208 100 78 152 59 147 280 123 ...
##  $ Loan_Amount_Term : num [1:367] 360 360 360 360 360 360 360 360 240 360 ...
##  $ Credit_History   : num [1:367] 1 1 1 NA 1 1 1 0 1 1 ...
##  $ Property_Area    : chr [1:367] "Urban" "Urban" "Urban" "Urban" ...

summary(Test_Loan_Prediction)

##    Loan_ID           Gender    Married  Dependents         Education        
##  Length:367         1   :286   0:134   Length:367         Length:367        
##  Class :character   2   : 70   1:233   Class :character   Class :character  
##  Mode  :character   NA's: 11           Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##  Self_Employed      ApplicantIncome CoapplicantIncome   LoanAmount   
##  Length:367         Min.   :    0   Min.   :    0     Min.   : 28.0  
##  Class :character   1st Qu.: 2864   1st Qu.:    0     1st Qu.:100.2  
##  Mode  :character   Median : 3786   Median : 1025     Median :125.0  
##                     Mean   : 4806   Mean   : 1570     Mean   :136.1  
##                     3rd Qu.: 5060   3rd Qu.: 2430     3rd Qu.:158.0  
##                     Max.   :72529   Max.   :24000     Max.   :550.0  
##                                                       NA's   :5      
##  Loan_Amount_Term Credit_History   Property_Area     
##  Min.   :  6.0    Min.   :0.0000   Length:367        
##  1st Qu.:360.0    1st Qu.:1.0000   Class :character  
##  Median :360.0    Median :1.0000   Mode  :character  
##  Mean   :342.5    Mean   :0.8254                     
##  3rd Qu.:360.0    3rd Qu.:1.0000                     
##  Max.   :480.0    Max.   :1.0000                     
##  NA's   :6        NA's   :29
```

```
#Treatment of Missing Data
#Replace LoanAmount missing values from mean
Test_Loan_Prediction$LoanAmount[which(is.na(Test_Loan_Prediction$LoanAmount))]
=mean(Test_Loan_Prediction$LoanAmount, na.rm = TRUE)
LoanAmount

##   [1] 146.2 128.0  66.0 120.0 141.0 267.0  95.0 158.0 168.0 349.0  70.0 95
………………………………………………………………………………………………………………
…………….## [613] 187.0 133.0

summary(LoanAmount)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     9.0   100.2   129.0   146.4   164.8   700.0

str(Loan_Status)

##  Factor w/ 2 levels "N","Y": 2 1 2 2 2 2 2 1 2 1 ...

Test_Loan_Prediction$Credit_History=as.factor(Test_Loan_Prediction$Credit_History)
# Deleting rest missing observations
Train_Loan_Prediction = Train_Loan_Prediction[complete.cases(Train_Loan_Prediction
), ]
Train_Loan_Prediction = na.omit(Train_Loan_Prediction)


dim(Test_Loan_Prediction)

## [1] 367  12

summary(Test_Loan_Prediction)

##    Loan_ID          Gender    Married  Dependents        Education
##  Length:367        1   :286   0:134   Length:367        Length:367
##  Class :character   2   : 70   1:233   Class :character   Class :character
##  Mode  :character   NA's: 11          Mode  :character   Mode  :character
##
##
##
##
##  Self_Employed      ApplicantIncome CoapplicantIncome   LoanAmount
##  Length:367         Min.   :    0   Min.   :    0     Min.   : 28.0
##  Class :character   1st Qu.: 2864   1st Qu.:    0     1st Qu.:101.0
##  Mode  :character   Median : 3786   Median : 1025     Median :126.0
##                     Mean   : 4806   Mean   : 1570     Mean   :136.1
##                     3rd Qu.: 5060   3rd Qu.: 2430     3rd Qu.:157.5
##                     Max.   :72529   Max.   :24000     Max.   :550.0
##
##  Loan_Amount_Term Credit_History Property_Area
```

```
##  Min.   :  6.0    0   : 59       Length:367
##  1st Qu.:360.0    1   :279       Class :character
##  Median :360.0    NA's: 29       Mode  :character
##  Mean   :342.5
##  3rd Qu.:360.0
##  Max.   :480.0
##  NA's   :6
```

```
library(glmnet)
```

Model Building - Logistic regression

```
library(car)

dim(Train_Loan_Prediction)

## [1] 499  13

dim(Test_Loan_Prediction)

## [1] 367  12
```

fit a logistic regression model with the training dataset

*#**Model1**#*

```
log.model=glm(Loan_Status ~Credit_History+LoanAmount+ApplicantIncome+Gender+
Married+Dependents+Loan_Amount_Term+CoapplicantIncome,
   data = Train_Loan_Prediction,family = binomial(link = "logit"))
summary(log.model)

##
## Call:
## glm(formula = Loan_Status ~ Credit_History + LoanAmount + ApplicantIncome +
##     Gender + Married + Dependents + Loan_Amount_Term + CoapplicantIncome,
##     family = binomial(link = "logit"), data = Train_Loan_Prediction)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0318  -0.4522   0.5965   0.7134   2.4910
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -2.317e+00  7.839e-01  -2.956  0.00311 **
## Credit_History1    3.600e+00  4.196e-01   8.580  < 2e-16 ***
## LoanAmount        -2.382e-03  1.678e-03  -1.420  0.15571
## ApplicantIncome    4.103e-06  2.617e-05   0.157  0.87542
## Gender2           -1.167e-01  3.157e-01  -0.370  0.71156
```

```
## Married1          6.074e-01  2.756e-01   2.204  0.02752 *
## Dependents1       -2.241e-01  3.272e-01  -0.685  0.49349
## Dependents2        1.923e-01  3.552e-01   0.541  0.58823
## Dependents3+       5.315e-02  4.645e-01   0.114  0.90890
## Loan_Amount_Term   1.806e-04  1.825e-03   0.099  0.92118
## CoapplicantIncome -4.121e-05  4.110e-05  -1.003  0.31598
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 623.06  on 498  degrees of freedom
## Residual deviance: 476.74  on 488  degrees of freedom
## AIC: 498.74
##
## Number of Fisher Scoring iterations: 4
```

```
# Check for multicollinearity
vif(log.model)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## Credit_History    1.013950  1        1.006951
## LoanAmount        1.493945  1        1.222271
## ApplicantIncome   1.423104  1        1.192939
## Gender            1.190926  1        1.091296
## Married           1.394458  1        1.180872
## Dependents        1.298897  3        1.044550
## Loan_Amount_Term  1.034747  1        1.017225
## CoapplicantIncome 1.115719  1        1.056276
```

```
#**Model2**#
log.mode2=glm(Loan_Status ~Credit_History+LoanAmount+ApplicantIncome+Gender+Married,
              data = Train_Loan_Prediction,family = binomial(link = "logit"))
summary(log.mode2)
```

```
##
## Call:
## glm(formula = Loan_Status ~ Credit_History + LoanAmount + ApplicantIncome +
##     Gender + Married, family = binomial(link = "logit"), data = Train_Loan_Prediction)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9597  -0.4609   0.6041   0.7080   2.4696
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -2.292e+00  4.806e-01  -4.770 1.84e-06 ***
```

```
## Credit_History1  3.594e+00  4.183e-01   8.591  < 2e-16 ***
## LoanAmount      -2.757e-03  1.628e-03  -1.694   0.0903 .
## ApplicantIncome  8.320e-06  2.568e-05   0.324   0.7459
## Gender2         -1.066e-01  3.084e-01  -0.346   0.7295
## Married1         6.129e-01  2.538e-01   2.415   0.0157 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 623.06  on 498  degrees of freedom
## Residual deviance: 478.80  on 493  degrees of freedom
## AIC: 490.8
##
## Number of Fisher Scoring iterations: 4
```

```
# Check for multicollinearity
vif(log.mode2)
```

```
##  Credit_History      LoanAmount ApplicantIncome          Gender         Married
##        1.009365        1.409551        1.357235        1.138841        1.186732
```

```
#**Model3**#
log.mode3=glm(Loan_Status ~Credit_History+LoanAmount+Married,
              data = Train_Loan_Prediction,family = binomial(link = "logit"))
summary(log.mode3)
```

```
##
## Call:
## glm(formula = Loan_Status ~ Credit_History + LoanAmount + Married,
##     family = binomial(link = "logit"), data = Train_Loan_Prediction)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9518  -0.4677   0.6052   0.7147   2.5357
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -2.320759   0.463925  -5.002 5.66e-07 ***
## Credit_History1  3.591247   0.417888   8.594  < 2e-16 ***
## LoanAmount      -0.002484   0.001404  -1.769   0.0768 .
## Married1         0.636977   0.238224   2.674   0.0075 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 623.06  on 498  degrees of freedom
```

```
## Residual deviance: 479.02  on 495  degrees of freedom
## AIC: 487.02
##
## Number of Fisher Scoring iterations: 4
```

 # Check for multicollinearity

```
vif(log.mode3)

## Credit_History     LoanAmount       Married
##       1.007460       1.040424       1.046323
```

#Model 1 AIC:498.74
#Model 2 AIC:490.8
#Model 3 AIC:487.02

#we always prefer model with minimum AIC value hence, preferred model is Model 3
 # log Likelihood ratio test- Measure the goodness of fit of three models - compare the models with intercept with predicted model

```
library(zoo)

library(lmtest)

attach(Test_Loan_Prediction)

## The following objects are masked from Train_Loan_Prediction (pos = 9):
##
##     ApplicantIncome, CoapplicantIncome, Credit_History, Dependents,
##     Education, Gender, Loan_Amount_Term, Loan_ID, LoanAmount, Married,
##     Property_Area, Self_Employed

## The following objects are masked from Train_Loan_Prediction (pos = 26):
##
##     ApplicantIncome, CoapplicantIncome, Credit_History, Dependents,
##     Education, Gender, Loan_Amount_Term, Loan_ID, LoanAmount, Married,
##     Property_Area, Self_Employed

Test_Loan_Prediction$Credit_History= as.factor(Test_Loan_Prediction$Credit_History)
```

to predict using logistic regression model, probabilities obtained

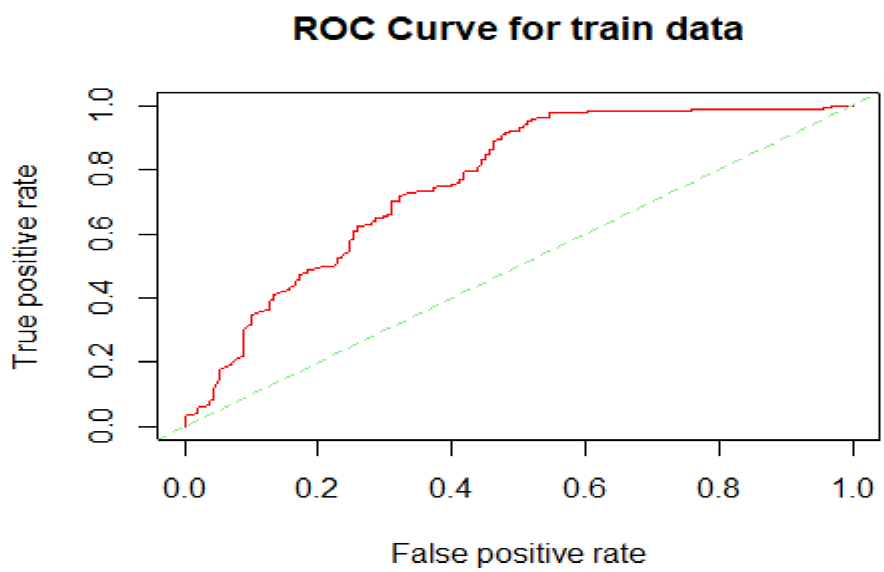
```
log.predictions <- predict(log.mode3, Test_Loan_Prediction, type="response")
```

Look at probability output

```
head(log.predictions, 10)

##         1         2         3         4         5         6         7         8
## 0.8367551 0.8312545 0.8007354        NA 0.7458832 0.8220024 0.7547228 0.11416
79
##         9        10
## 0.7706677 0.7241253
```

```
##Below we are going to assign our labels with decision rule that if the prediction is greater than 0.5, assign it 1 else 0.
log.prediction.rd <- ifelse(log.predictions > 0.5, 1, 0)
head(log.prediction.rd, 10)

##  1  2  3  4  5  6  7  8  9 10
##  1  1  1 NA  1  1  1  0  1  1

library(pscl)

## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis

pR2(log.model)

## fitting null model for pseudo-r2

##          llh      llhNull           G2     McFadden         r2ML         r2CU
## -238.3716368 -311.5285018  146.3137300    0.2348320    0.2541374    0.3563854

# Odds Ratio
exp(coef(log.model))

##       (Intercept)   Credit_History1        LoanAmount   ApplicantIncome
##        0.09853211       36.61219491        0.99762127        1.00000410
##           Gender2          Married1       Dependents1       Dependents2
##        0.88983126        1.83572647        0.79926616        1.21207118
##      Dependents3+  Loan_Amount_Term CoapplicantIncome
##        1.05458461        1.00018059        0.99995879

# Probability (credit History shows highest agomt all)
exp(coef(log.model))/(1+exp(coef(log.model)))

##       (Intercept)   Credit_History1        LoanAmount   ApplicantIncome
##        0.08969434        0.97341288        0.49940461        0.50000103
##           Gender2          Married1       Dependents1       Dependents2
##        0.47085223        0.64735668        0.44421786        0.54793498
##      Dependents3+  Loan_Amount_Term CoapplicantIncome
##        0.51328361        0.50004514        0.49998970

# Accuracy | Base Line Model
nrow(Train_Loan_Prediction[Train_Loan_Prediction$Loan_Status == 1,])/nrow(Train_Loan_Prediction)

## [1] 0
```

```
library(caret)

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

## The following object is masked from 'package:survival':
##
##     cluster

library(InformationValue)

## Warning: package 'InformationValue' was built under R version 4.0.5

##
## Attaching package: 'InformationValue'

## The following objects are masked from 'package:caret':
##
##     confusionMatrix, precision, sensitivity, specificity

pred = predict(log.mode3, Train_Loan_Prediction, type="response")
y_pred_num = ifelse(pred>0.5,1,0)
y_pred = factor(y_pred_num, levels=c(0,1))
y_actual = Train_Loan_Prediction$Loan_Status
pred <- predict(log.mode3, Train_Loan_Prediction, type = "response")
confusionMatrix(Train_Loan_Prediction$Loan_Status, pred)

##    N   Y
## 0 67   7
## 1 91 334

#lest plot ROC

library(ROCR)
train.roc <- prediction(pred, Train_Loan_Prediction$Loan_Status)
plot(performance(train.roc, "tpr", "fpr"),
     col = "red", main = "ROC Curve for train data")
abline(0, 1, lty = 8, col = "lightgreen")
```




```
# AUC
train.auc = performance(train.roc, "auc")
train.area = as.numeric(slot(train.auc, "y.values"))
train.area

## [1] 0.7620829

#76.20% is AUC

# Gini - Area covered by ROC and mean line (more area cover to 1 is better)
train.gini = (2 * train.area) - 1
train.gini

## [1] 0.5241657

# Calibrating thresold levels to increase sensitivity

### Model Building - KNN

library(trainR)

# Normalize variables
scale = preProcess(Train_Loan_Prediction, method = "range")

train.norm.data = predict(scale, Train_Loan_Prediction)
test.norm.data = predict(scale, Test_Loan_Prediction)

knn_fit = train(Loan_Status ~., data = train.norm.data, method = "knn",
            trControl = trainControl(method = "cv", number = 3),
            tuneLength = 10)
```

```
knn_fit = train
knn_fit$besttune$k
```

➢ Recommendations: End Note

1. We have 80% male customers, we should also target females at urban areas, so that we can have ration of 50:50 Male Female.
2. As we have seen that most of our customers are job bases, we should more focus on business class, as they can bear interest and take high number of loan amount.
3. 31% of our customers are not eligible/approved for loan, we should give small amount of loan or other securities loan offers to those customer’s, so that they won’t switch to other banks.
4. Our bank should focus on Long term loans.
5. Our loan Amount falls in between 0-300 range, we should keep some attractive offers for the range of 500-1000, For Ex: Zero Processing Fees, Zero documentation charges, early possession etc.