CREDIT MODELLING RISK IN R

First Step-

Data Preprocessing

Data we have

Training set:



Structure of training set

str(training\_set)

Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 614 obs. of 13 variables:

$ Loan\_ID : chr "LP001002" "LP001003" "LP001005" "LP001006" ...

$ Gender : chr "Male" "Male" "Male" "Male" ...

$ Married : chr "No" "Yes" "Yes" "Yes" ...

$ Dependents : chr "0" "1" "0" "0" ...

$ Education : chr "Graduate" "Graduate" "Graduate" "Not Graduate" ...

$ Self\_Employed : chr "No" "No" "Yes" "No" ...

$ ApplicantIncome : int 5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...

$ CoapplicantIncome: num 0 1508 0 2358 0 ...

$ LoanAmount : int NA 128 66 120 141 267 95 158 168 349 ...

$ Loan\_Amount\_Term : int 360 360 360 360 360 360 360 360 360 360 ...

$ Credit\_History : int 1 1 1 1 1 1 1 0 1 1 ...

$ Property\_Area : chr "Urban" "Rural" "Urban" "Urban" ...

$ Loan\_Status : chr "Y" "N" "Y" "Y" ...

Explore training set

Load gmodels package to use **Crosstable function which will tell the ratio of default loans**

CrossTable(training\_set$Loan\_Status)

Cell Contents

|-------------------------|

| N |

| N / Table Total |

|-------------------------|

Total Observations in Table: 614

| N | Y |

|-----------|-----------|

| 192 | 422 |

| 0.313 | 0.687 |

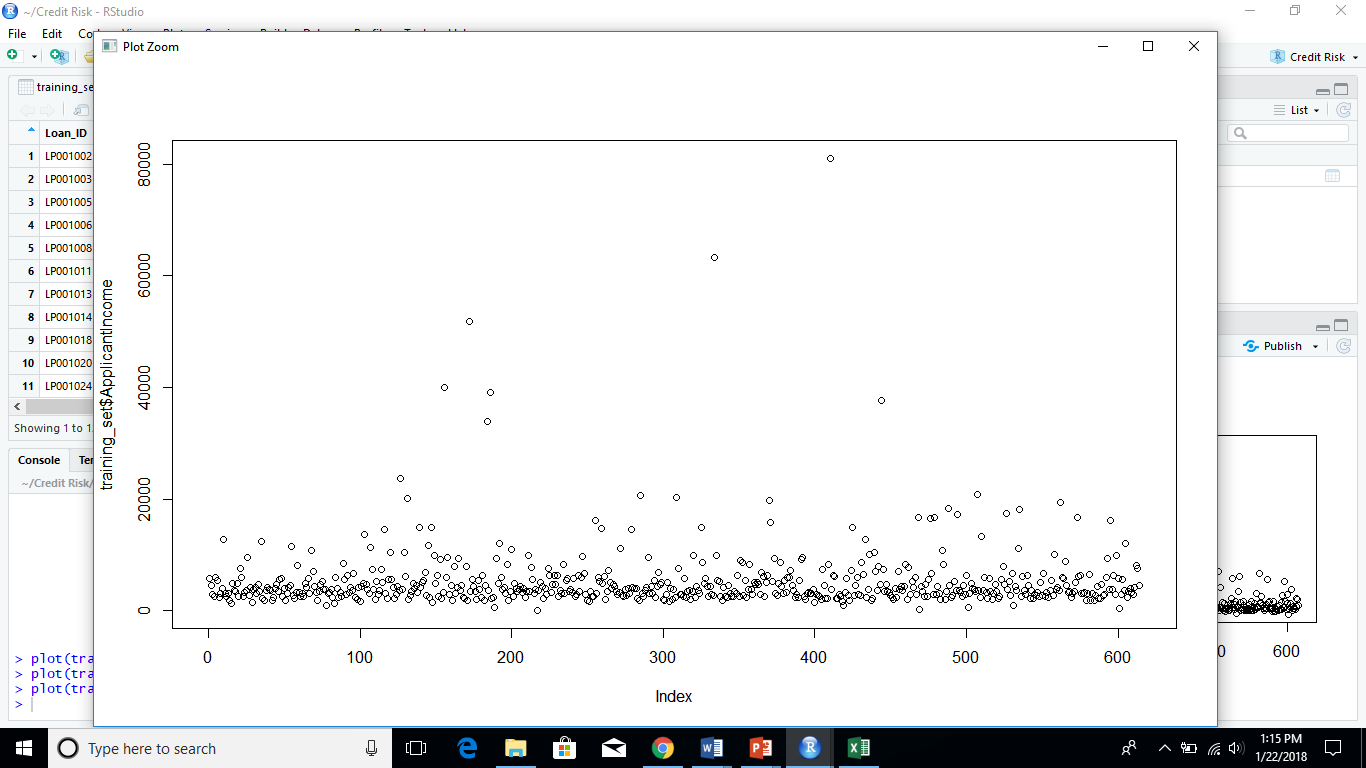
|-----------|-----------|

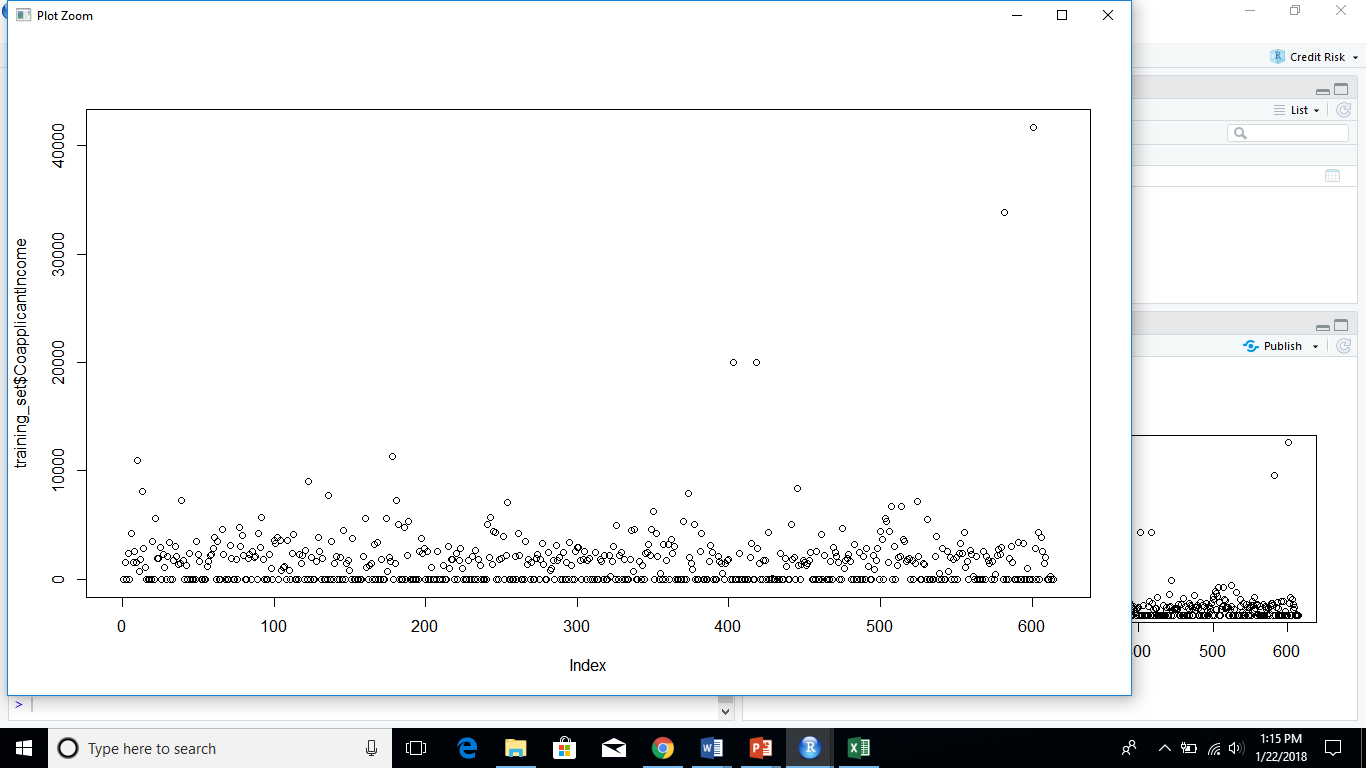
422 loans were paid back whereas 192 were not paid back

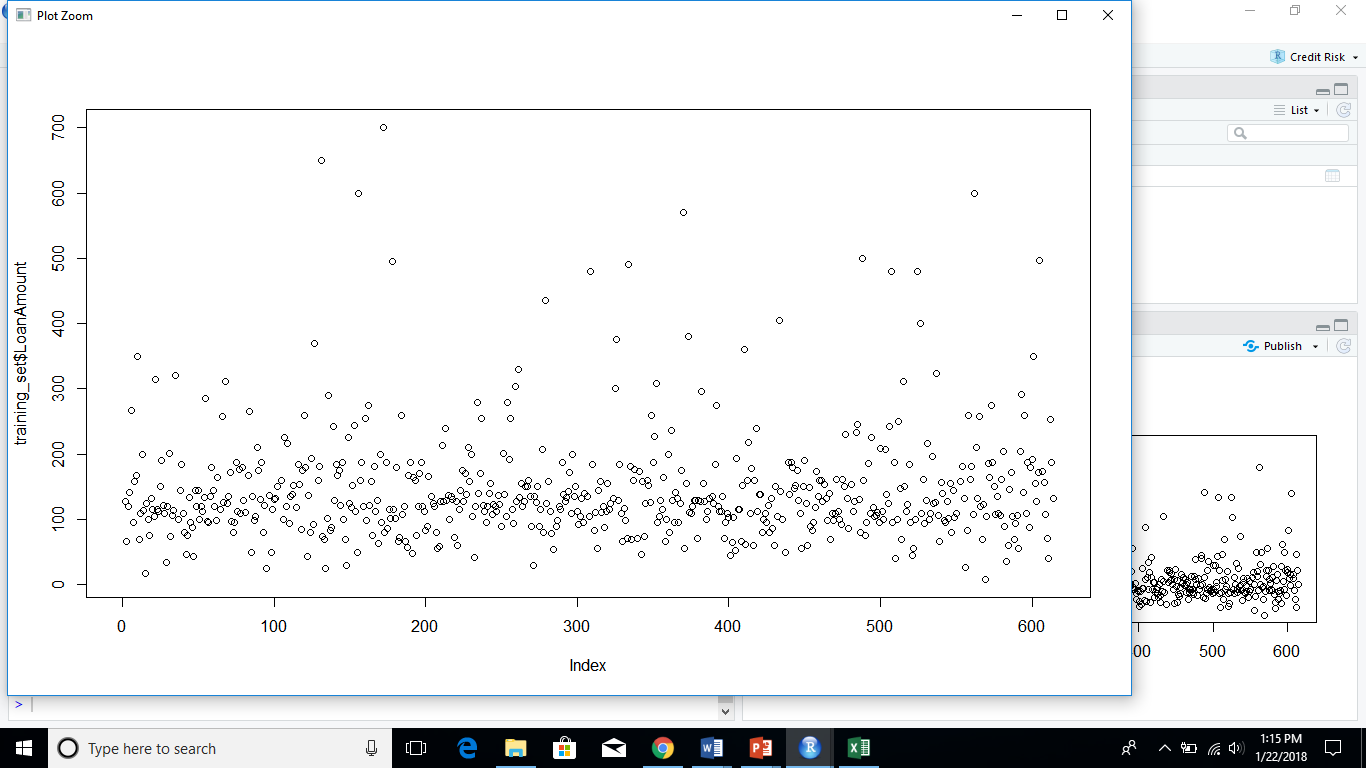
We need to analyze the data and remove the outliers (the values which show vast difference from other values.

In our dataset we can have outliers in Applicant Income, CoApplicant Income and loan amount so we will check for outliers and remove if any outliers present.

We plotted the three variables and did not find any outliers.







Now we will find the missing values and try to remove or replace it

Removing NA values from Gender

removegenderna<-which(is.na(training\_set$Gender))

training\_set<- training\_set[-removegenderna,]

Removing NA values from Married

removemarriedna<-which(is.na(training\_set$Married))

> training\_set<- training\_set[-removemarriedna,]

Replacing the NA values of dependents to 0

removedependentsna<-which(is.na(training\_set$Dependents))

> training\_set$Dependents[removedependentsna]<-0

Replacing NA values of SelfEmployed to No

Replace the NA values in Loan\_Amount\_term to median of column

|  |
| --- |
| removetermna<-which(is.na(training\_set$Loan\_Amount\_Term))  > medianterm<-median(training\_set$Loan\_Amount\_Term)  > medianterm<-median(training\_set$Loan\_Amount\_Term,na.rm = TRUE)  > training\_set$Loan\_Amount\_Term[removetermna]<-medianterm |
|  |
| |  | | --- | |  | |

The data is preprocessed now

str(training\_set)

Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 598 obs. of 13 variables:

$ Loan\_ID : chr "LP001002" "LP001003" "LP001005" "LP001006" ...

$ Gender : chr "Male" "Male" "Male" "Male" ...

$ Married : chr "No" "Yes" "Yes" "Yes" ...

$ Dependents : chr "0" "1" "0" "0" ...

$ Education : chr "Graduate" "Graduate" "Graduate" "Not Graduate" ...

$ Self\_Employed : chr "No" "No" "Yes" "No" ...

$ ApplicantIncome : int 5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...

$ CoapplicantIncome: num 0 1508 0 2358 0 ...

$ LoanAmount : int NA 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 : chr "Urban" "Rural" "Urban" "Urban" ...

$ Loan\_Status : chr "Y" "N" "Y" "Y" ...

Divide the training set in train and test set

|  |
| --- |
| train\_set<-sample(1:nrow(training\_set),2/3 \*nrow(training\_set))  > traininga\_set<-training\_set[train\_set,]  > test\_set<- training\_set[-train\_set,] |
|  |
| |  | | --- | | > | |

Convert loan\_status in numeric put 1 where there is Y and o where we have N

convto0<-which(traininga\_set$Loan\_Status=="N")

> traininga\_set$Loan\_Status[convto0]<-0

> View(traininga\_set)

> convto1<-which(traininga\_set$Loan\_Status=="Y")

> traininga\_set$Loan\_Status[convto1]<-1

> convto0t<-which(test\_set$Loan\_Status=="N")

> test\_set$Loan\_Status[convto0t]<-0

> convto1t<-which(test\_set$Loan\_Status=="Y")

> test\_set$Loan\_Status[convto1t]<-1

Convert to numeric

traininga\_set$Loan\_Status<-as.numeric(as.character(traininga\_set$Loan\_Status))

> test\_set$Loan\_Status<-as.numeric(as.character(test\_set$Loan\_Status))

LOGISTIC REGRESSION

Logistic Regression is a classification algorithm. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. To represent binary / categorical outcome, we use dummy variables. You can also think of logistic regression as a special case of linear regression when the outcome variable is categorical, where we are using log of odds as dependent variable. In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function.

Apply Logistic Regression on training dataset

Take all factors

log\_predict<-glm(formula= Loan\_Status ~ ApplicantIncome + CoapplicantIncome + Gender+ Dependents +Education + Self\_Employed + LoanAmount + Loan\_Amount\_Term +Credit\_History+ Property\_Area,family = "binomial",data=traininga\_set)

> log\_predict

Call: glm(formula = Loan\_Status ~ ApplicantIncome + CoapplicantIncome +

Gender + Dependents + Education + Self\_Employed + LoanAmount +

Loan\_Amount\_Term + Credit\_History + Property\_Area, family = "binomial",

data = traininga\_set)

Coefficients:

(Intercept) ApplicantIncome CoapplicantIncome

-6.896e-01 6.356e-06 -3.168e-05

GenderMale Dependents1 Dependents2

2.519e-02 -2.626e-01 8.157e-01

Dependents3+ EducationNot Graduate Self\_EmployedYes

2.945e-01 -5.155e-01 2.888e-02

LoanAmount Loan\_Amount\_Term Credit\_History

-6.762e-05 -1.142e-03 2.163e+00

Property\_AreaSemiurban Property\_AreaUrban

8.129e-01 6.187e-01

Degrees of Freedom: 384 Total (i.e. Null); 371 Residual

Null Deviance: 462.5

Residual Deviance: 377.6 AIC: 405.6

Summary

summary(log\_predict)

Call:

glm(formula = Loan\_Status ~ ApplicantIncome + CoapplicantIncome +

Gender + Dependents + Education + Self\_Employed + LoanAmount +

Loan\_Amount\_Term + Credit\_History + Property\_Area, family = "binomial",

data = traininga\_set)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.2882 -0.7530 0.5423 0.6766 1.8883

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -6.896e-01 8.678e-01 -0.795 0.4268

ApplicantIncome 6.356e-06 2.326e-05 0.273 0.7847

CoapplicantIncome -3.167e-05 4.018e-05 -0.788 0.4305

GenderMale 2.519e-02 3.460e-01 0.073 0.9420

Dependents1 -2.626e-01 3.636e-01 -0.722 0.4701

Dependents2 8.157e-01 4.410e-01 1.850 0.0644 .

Dependents3+ 2.945e-01 4.986e-01 0.591 0.5547

EducationNot Graduate -5.155e-01 3.018e-01 -1.708 0.0876 .

Self\_EmployedYes 2.888e-02 3.982e-01 0.073 0.9422

LoanAmount -6.762e-05 1.960e-03 -0.035 0.9725

Loan\_Amount\_Term -1.142e-03 2.048e-03 -0.558 0.5770

Credit\_History 2.163e+00 2.839e-01 7.619 2.56e-14 \*\*\*

Property\_AreaSemiurban 8.129e-01 3.156e-01 2.576 0.0100 \*\*

Property\_AreaUrban 6.187e-01 3.249e-01 1.904 0.0569 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 462.49 on 384 degrees of freedom

Residual deviance: 377.57 on 371 degrees of freedom

AIC: 405.57

Number of Fisher Scoring iterations: 4

Now prediction with test data

prediction<- predict(log\_predict,newdata = test\_set,type="response")

> range(prediction)

[1] 0.1186485 0.9481618

Verify your model we will take cutoff ad 20% Confusion matrix

table(test\_set$Loan\_Status,pred\_cutoff\_20)

pred\_cutoff\_20

0 1

0 3 62

1 1 126

#calcullating accuracy of model

> Accuracy<-(3+126)/(62+1+126+3)

> Accuracy\*100

[1] 67.1875

67 percent

Finding the right cutoff value the strategy curve

Obtain the cutoff rate of acceptance rate =80%

|  |
| --- |
| prediction<- predict(log\_predict,newdata = test\_set,type="response")  > cutoff<-quantile(prediction,0.8)  > cutofffinal<-ifelse(prediction>cutoff,1,0)  > accepted\_status<-test\_set$Loan\_Status[cutofffinal==0]  > sum(accepted\_status/length(accepted\_status))  [1] 0.6405229 |
|  |
| |  | | --- | |  | |

Bad rate=0.6405

ROC curve

Sensitivity vs 1-specificity

Load pROC library

library(pROC)

ROC\_predict<- roc(test\_set$Loan\_Status,prediction)

> plot(ROC\_predict)

> auc(ROC\_predict)

Area under the curve: 0.6514