**USE CASE STUDY REPORT**

**Group No**.: Group 24

**Student Names**: Gaurav Korgaonkar, Yongshen Bai

Executive Summary

The main objective of this case study was to determine the eligibility of a borrower to be considered for a loan from the lending club and then based on his statistics predict the interest rate that should be allocated to the borrower. Machine learning models were built and trained using historical data that was obtained from the lending club website. These models were then tested using validation data sets to check the accuracy and the best models were selected.

# I. Background and Introduction

Money lending is the world’s oldest profession, and thus began the concept of a Lending club. A Lending club, as the name suggests, lends money to customers and expects a return with interest. While investing in a Lending club, these investors expect all the borrowers to pay back the amount with proper interest to make it profitable. But, this is not always the case. Currently millions of borrowers in the United States are defaulting on their loans which amount to billions of dollars. This is the problem we would like to tackle using this case study.

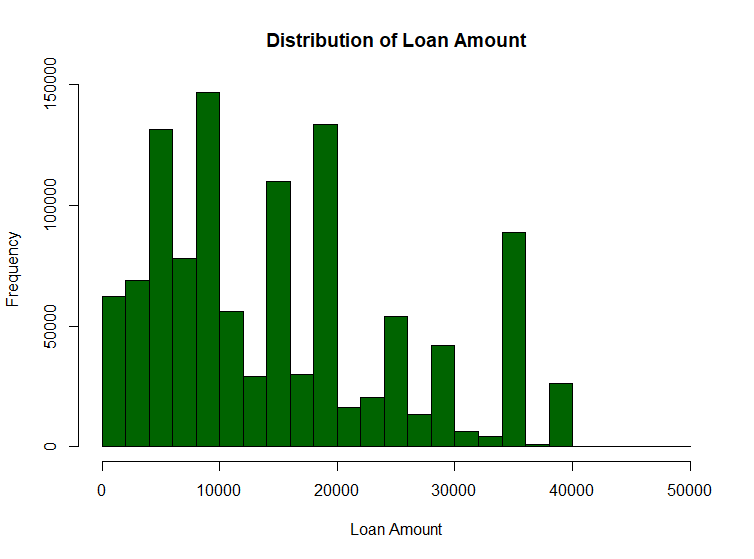
Due to lack of thorough background checks and standard models to analyze the potential borrowers, today there are a lot of people who are unable to pay off their debt and thus their debt keeps on increasing. This will make sure that the lending club won’t be a success. If this problem is ignored, it will lead to more of people with increasing debt and loss of investment. Our proposed solution to this problem is using the historical data already available to the lending club and build standard models based on this data.

The first objective is to analyze all the parameters in the dataset, deal with the missing variables and determine the important ones that determine the decision based on their contribution and correlation with others to get an appropriate number of parameters for the models and build classification models that will generate a flag whether to give a loan or not.

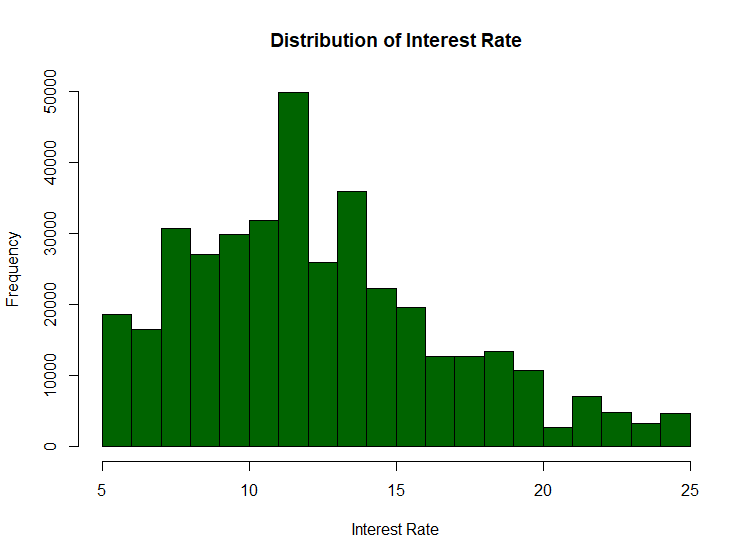
The second objective will be to build models to decide the interest allocated to a certain borrower. This can be done by one prediction model considering all borrowers or clustering the borrowers and then using the prediction algorithms based on the clusters. Then calculate the accuracy measures and try to improve the models.

# II. Data Exploration and Visualization

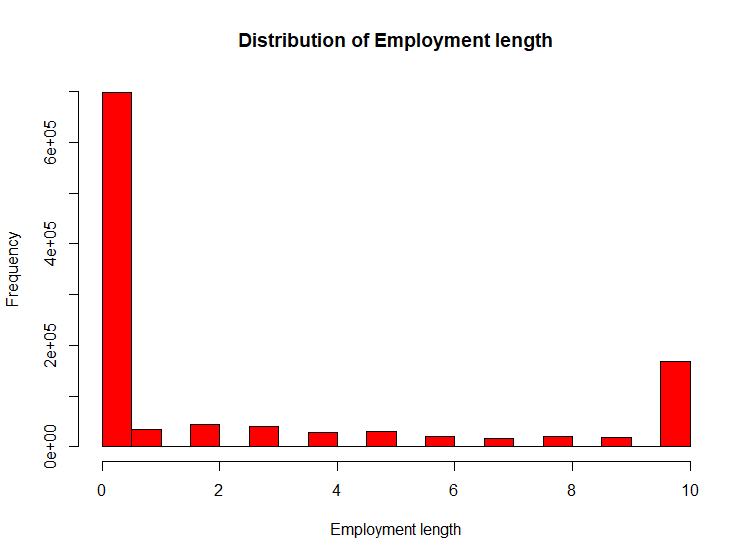
The data set includes two types of data, the loan details of the applications that were accepted for lending club and the details of the applications that were rejected. By performing Exploratory data analysis on these datasets, we have learned the following, the distribution of the loan amount is mostly based in the $5000-$20000 region.



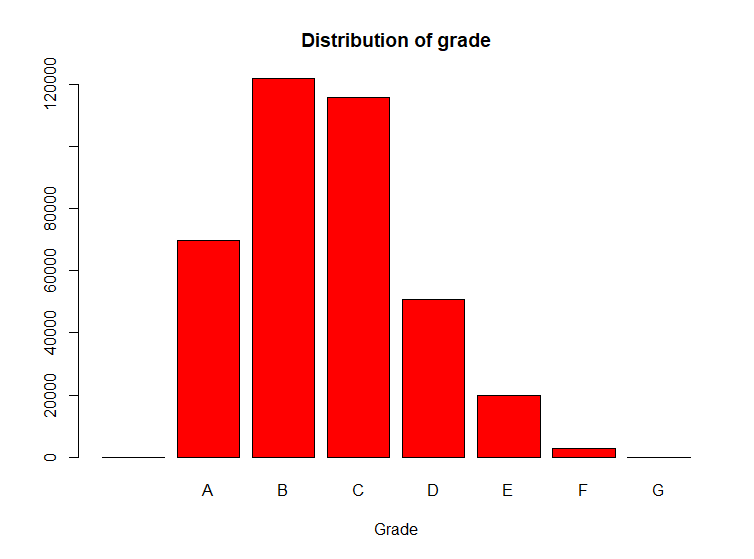
The interest rate that was provided after the loan was given lies mostly between 10%-15%



The employment length of the borrowers shows us that most them have worked 0 years.



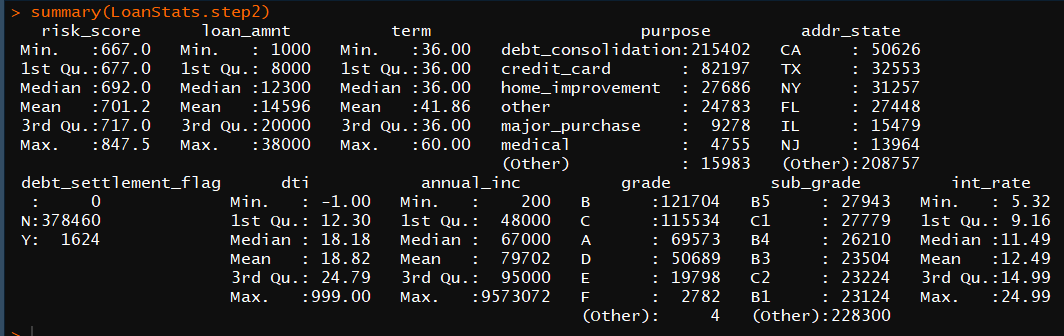
The distribution of the grade related to the loan application shows that most of the applications are of grade B or C.



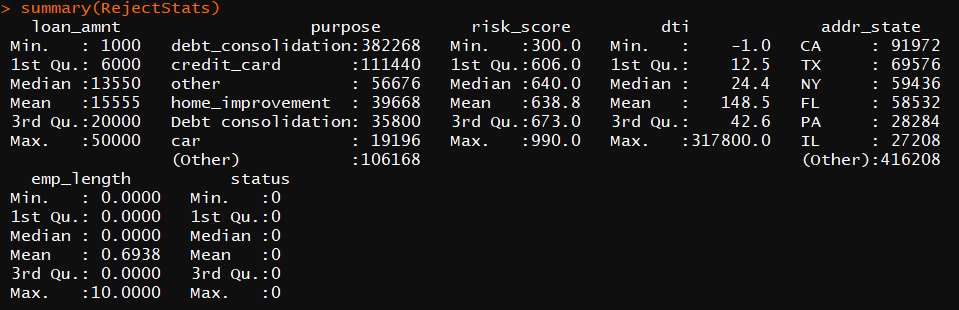
# III. Data Preparation and Preprocessing

As the project is based on the possibility of investing in a lending club, we need data related to past ventures of the lending club. This data is available on their website. A set of files in csv format that give data about declined loan applications and the data about the accepted loan applications with further details. We merged these files and then use that file for this case study.

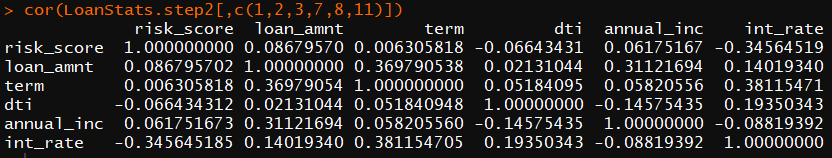
The summary of the loan applications that were accepted by the Lending Club,



The summary of the loan applications that were rejected by the Lending Club,



The correlation between the terms used for prediction models ,



# IV. Data Mining Techniques and Implementation

After analyzing the dataset, it was observed that it contains a lot of inconsistent and missing data. Hence, we had to use various data cleaning techniques to solve this issue. The inconsistent values were treated and the missing data was removed from the dataset.

The first step is classification of a person to decide eligibility for loan, we build a logistic regression model and Random forest model for classification. We then test out these two models based on the training and test data set to check for accuracy and then select the most accurate one. Then we segment data into clusters manually using categorical features.

Second step is predicting what interest rate to offer to that borrower, we need to build various prediction models for each cluster. Using Linear regression and random forest algorithms. Check for accuracy of the models based on MAE, RMS, MAPE for training and testing datasets. Then, we select the most accurate model for prediction.

Classification

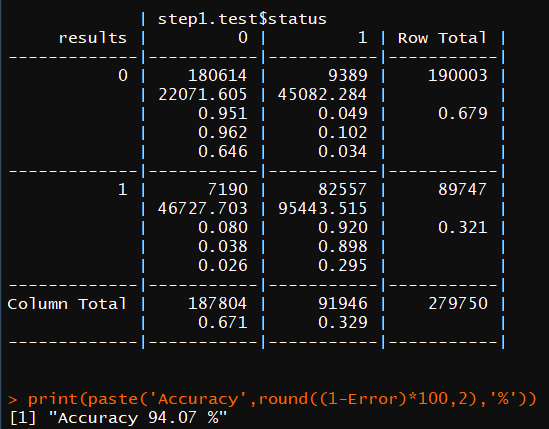
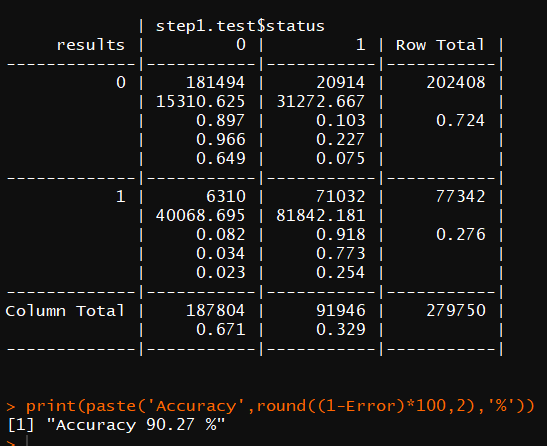
Prediction

Manual Clustering

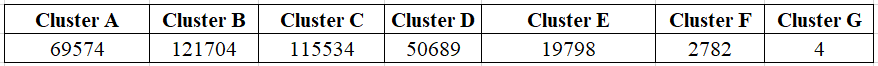
Data Cleaning

# V. Performance Evaluation

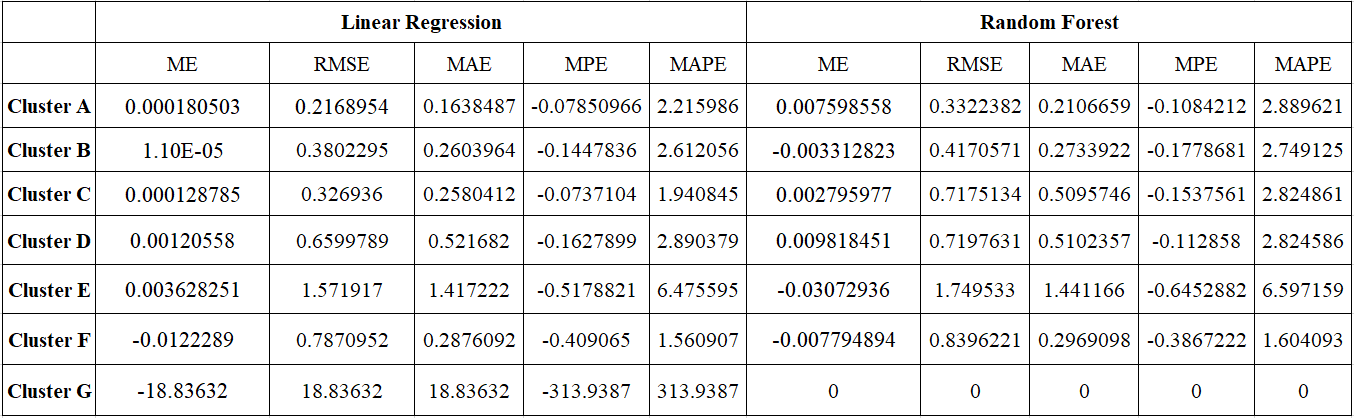
For classification problem, Logistic regression and Random Forest algorithms were used. Both models were evaluated for accuracy of results. The accuracy for logistic regression model is 90.27% and the accuracy for random forest model is 94.07 %

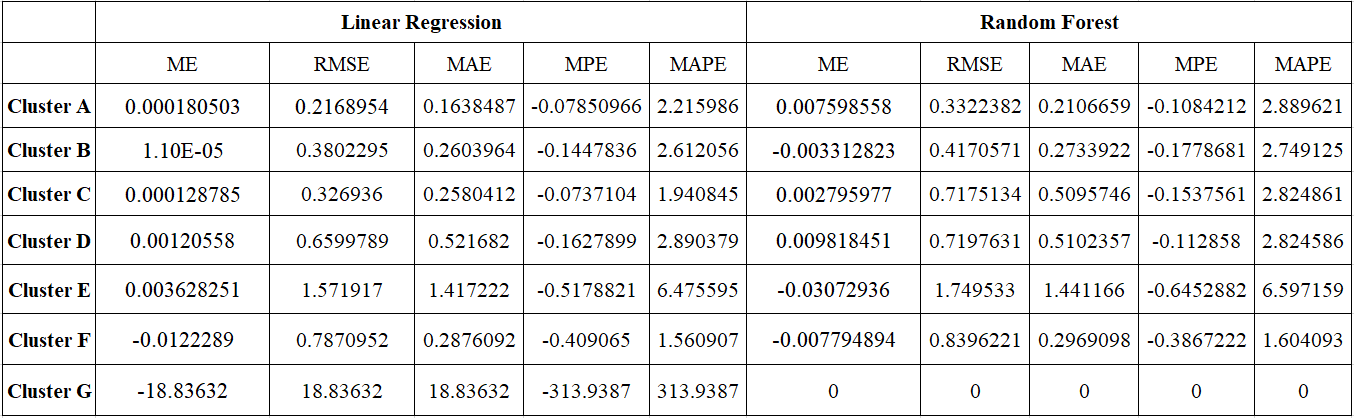
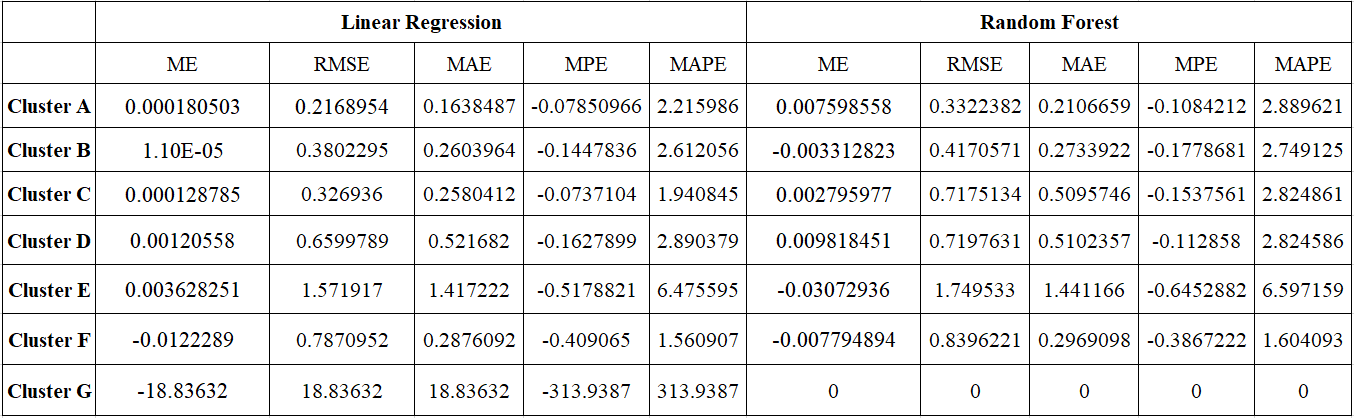


Clustering based on grade makes more sense as that variable derives the quality of the application. Thus, for this case study, we have clustered the dataset based on grades,

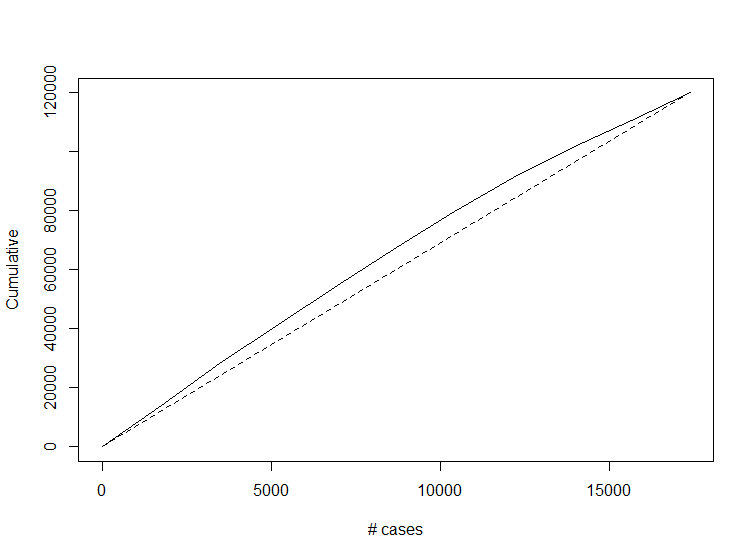


For Prediction problem, linear regression and random forest algorithms were used. Both models were evaluated for accuracy of results. The accuracy measures for these models are,

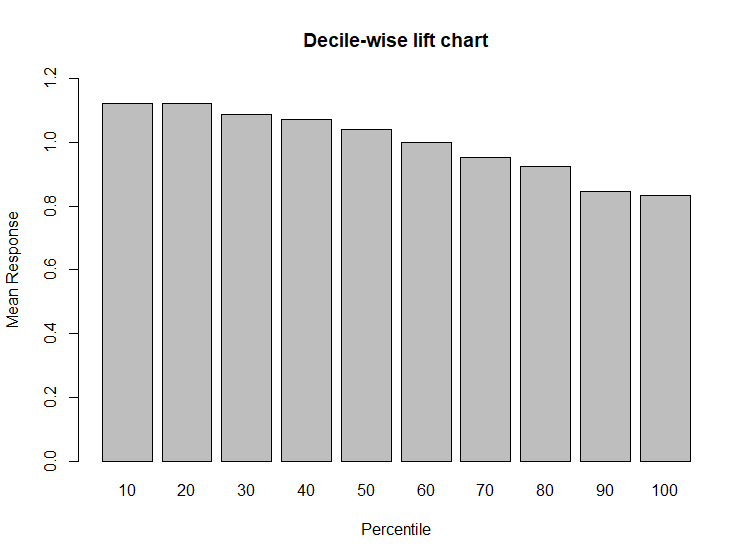




Then we plotted lift charts and decile-wise charts for models to analyze the accuracy. Here is the lift chart for cluster A,



Here is the decile-wise chart for cluster A,



# VI. Discussion and Recommendation

The records were manually clustered into grades because that made more sense than using a clustering algorithm as the interest were supposed to be calculated on how good the loan application is. As the results show, the random forest algorithm gives a better accuracy for classification problem than the logistic regression algorithm whereas the linear regression model gives better accuracy for the prediction algorithm than the random forest algorithm. Thus, the random forest algorithm must be used for classification problems and linear regression must be used to predict the interest rate of such records for better accuracy.

# VII. Summary

The objective is to determine whether a borrower should be allowed to loan from the lending club and then based on his statistics predict the interest rate that should be allocated to the borrower. Historical data was obtained from the lending club website and cleaned to be fed into machine learning models. Firstly, the data was tested in classification models for determining the status of their loan application and then the data was fed to prediction models for determining the interest allocated to that person. All the models were tested for accuracy measures and then the best models were decided.

# Appendix: R Code for use case study

### Data Cleaning

##LOAN STATS

L1<-read.csv("LoanStats\_securev1\_2016Q1.csv",skip=1)

L2<-read.csv("Q2.csv")

L3<-read.csv("LoanStats\_securev1\_2016Q3.csv",skip=1)

L4<-read.csv("LoanStats\_securev1\_2016Q4.csv",skip=1)

LoanStats<-rbind(L1,L2,L3,L4)

LoanStats<-LoanStats[LoanStats$fico\_range\_low > 660,]

LoanStats$risk\_score<- (LoanStats$fico\_range\_low + LoanStats$fico\_range\_high) / 2

LoanStats$term<-as.integer(LoanStats$term %>% str\_replace(' 36 months','36') %>% str\_replace(' 60 months','60'))

LoanStats$int\_rate<- as.double(str\_replace(LoanStats$int\_rate,'%',''))

LoanStats$status<-1

LoanStats$emp\_length<-gsub('\\+','',LoanStats$emp\_length)

LoanStats$emp\_length<-as.integer( LoanStats$emp\_length %>% str\_replace(' years','') %>% str\_replace('< 1 year','0') %>% str\_replace(' year','') )

LoanStats.step1<-LoanStats[,c('loan\_amnt', 'purpose', 'risk\_score', 'dti', 'addr\_state', 'emp\_length','status')]

LoanStats.step1<-LoanStats.step1[complete.cases(LoanStats.step1),]

head(LoanStats.step1)

##REJECTED

R1<-read.csv("RejectStats\_2016Q1.csv",skip = 1)

R2<-read.csv("RejectStats\_2016Q1.csv",skip = 1)

R3<-read.csv("RejectStats\_2016Q1.csv",skip = 1)

R4<-read.csv("RejectStats\_2016Q1.csv",skip = 1)

RejectStats<-rbind(R1,R2,R3,R4)

colnames(RejectStats)<-c('loan\_amnt','date' ,'purpose', 'risk\_score', 'dti', 'zip\_code', 'addr\_state', 'emp\_length', 'policy\_code')

RejectStats<- RejectStats[,c('loan\_amnt', 'purpose', 'risk\_score', 'dti', 'addr\_state', 'emp\_length')]

RejectStats$status<- 0

RejectStats$emp\_length<-gsub('\\+','',RejectStats$emp\_length)

RejectStats$emp\_length<-as.integer( RejectStats$emp\_length %>% str\_replace(' years','') %>% str\_replace('< 1 year','0') %>% str\_replace(' year','') )

RejectStats$dti<- as.double(str\_replace(RejectStats$dti,'%',''))

RejectStats<-RejectStats[complete.cases(RejectStats),]

head(RejectStats)

### EDA

eda<-rbind(LoanStats.step1,RejectStats)

hist(eda$loan\_amnt,xlab = 'Loan Amount',main = 'Distribution of Loan Amount',col='darkgreen')

hist(eda$emp\_length,col='red',xlab='Employment length',main = 'Distribution of Employment length')

hist(LoanStats$int\_rate,col='darkgreen',main = 'Distribution of Interest Rate',xlab = 'Interest Rate')

barplot(table(LoanStats$grade),col='red',main = 'Distribution of grade',xlab = 'Grade')

### Machine Learning models

#Step1

## Logistic glm

step1.set<-rbind(LoanStats.step1,RejectStats)

set.seed(101)

sample = sample.split(step1.set$status, SplitRatio = .75)

step1.train = subset(step1.set, sample == T)

step1.test = subset(step1.set, sample == F)

step1.model<-glm(status ~.,family=binomial(link='logit'),data=step1.train)

results<-predict(step1.model,newdata = step1.test)

results<-ifelse(results > 0.5,1,0)

gmodels::CrossTable(results,step1.test$status)

Error <- mean(results != step1.test$status)

print(paste('Accuracy',round((1-Error)\*100,2),'%'))

## RF

rf.model<-randomForest(status ~ loan\_amnt+risk\_score+dti+addr\_state+emp\_length,

data=step1.train,

importance=TRUE,

ntree=2)

results<-predict(rf.model,step1.test)

results<-ifelse(results > 0.5,1,0)

Error <- mean(results != step1.test$status)

print(paste('Accuracy',round((1-Error)\*100,2),'%'))

#Step 2

#Manual clustering

LoanStats.step2<-LoanStats[,c('risk\_score', 'loan\_amnt', 'term', 'purpose', 'addr\_state', 'debt\_settlement\_flag', 'dti', 'annual\_inc','grade' ,'sub\_grade', 'int\_rate')]

LoanStats.step2<-LoanStats.step2[complete.cases(LoanStats.step2),]

head(LoanStats.step2)

LoanStats.step2<-LoanStats.step2[LoanStats.step2$int\_rate<=25,]

LoanStats.step2<-LoanStats.step2[LoanStats.step2$loan\_amnt<=38000,]

boxplot(LoanStats.step2$int\_rate)

unique(LoanStats.step2$grade)

LoanStats.step2.A<-LoanStats.step2[LoanStats.step2$grade=='A',]

nrow(LoanStats.step2.A)

LoanStats.step2.B<-LoanStats.step2[LoanStats.step2$grade=='B',]

nrow(LoanStats.step2.B)

LoanStats.step2.C<-LoanStats.step2[LoanStats.step2$grade=='C',]

nrow(LoanStats.step2.C)

LoanStats.step2.D<-LoanStats.step2[LoanStats.step2$grade=='D',]

nrow(LoanStats.step2.D)

LoanStats.step2.E<-LoanStats.step2[LoanStats.step2$grade=='E',]

nrow(LoanStats.step2.E)

LoanStats.step2.F<-LoanStats.step2[LoanStats.step2$grade=='F',]

nrow(LoanStats.step2.F)

LoanStats.step2.G<-LoanStats.step2[LoanStats.step2$grade=='G',]

nrow(LoanStats.step2.G)

# Interest rate prediction

#Cluster A

#Linear

set.seed(101)

sample = sample.split(LoanStats.step2.A$int\_rate, SplitRatio = .75)

step2.train = subset(LoanStats.step2.A, sample == T)

step2.test = subset(LoanStats.step2.A, sample == F)

step2.model<-lm(int\_rate ~risk\_score+ loan\_amnt + term + purpose +addr\_state +dti +annual\_inc+sub\_grade,family = "binomial",data=step2.train)

results.lm<-predict(step2.model,newdata = step2.test)

accuracy(results.lm,step2.test$int\_rate)

## RF

rf2.model<-randomForest(int\_rate ~risk\_score+ loan\_amnt + term + purpose +addr\_state +dti +annual\_inc+sub\_grade,data=step2.train,

importance=TRUE,

ntree=2)

results.rf<-predict(rf2.model,step2.test)

accuracy(results.rf,step2.test$int\_rate)

#Cluster B

#Linear

set.seed(101)

sample = sample.split(LoanStats.step2.B$int\_rate, SplitRatio = .75)

step2.train = subset(LoanStats.step2.B, sample == T)

step2.test = subset(LoanStats.step2.B, sample == F)

step2.model<-lm(int\_rate ~risk\_score+ loan\_amnt + term+dti +annual\_inc+sub\_grade,data=step2.train)

summary(step2.model)

results.lm<-predict(step2.model,newdata = step2.test)

accuracy(results.lm,step2.test$int\_rate)

## RF

rf2.model<-randomForest(int\_rate ~risk\_score+ loan\_amnt + term+dti +annual\_inc+sub\_grade,data=step2.train,

importance=TRUE,

ntree=2)

results.rf<-predict(rf2.model,step2.test)

accuracy(results.rf,step2.test$int\_rate)

#Cluster C

#Linear

set.seed(101)

sample = sample.split(LoanStats.step2.C$int\_rate, SplitRatio = .75)

step2.train = subset(LoanStats.step2.C, sample == T)

step2.test = subset(LoanStats.step2.C, sample == F)

step2.model<-lm(int\_rate ~risk\_score+ loan\_amnt + term+dti +annual\_inc+sub\_grade,data=step2.train)

summary(step2.model)

results.lm<-predict(step2.model,newdata = step2.test)

accuracy(results.lm,step2.test$int\_rate)

## RF

rf2.model<-randomForest(int\_rate ~risk\_score+ loan\_amnt + term+dti +annual\_inc+sub\_grade,data=step2.train,

importance=TRUE,

ntree=2)

results.rf<-predict(rf2.model,step2.test)

accuracy(results.rf,step2.test$int\_rate)

#Cluster D

#Linear

set.seed(101)

sample = sample.split(LoanStats.step2.D$int\_rate, SplitRatio = .75)

step2.train = subset(LoanStats.step2.D, sample == T)

step2.test = subset(LoanStats.step2.D, sample == F)

step2.model<-lm(int\_rate ~risk\_score+ loan\_amnt + term+dti +annual\_inc+sub\_grade,data=step2.train)

summary(step2.model)

results.lm<-predict(step2.model,newdata = step2.test)

accuracy(results.lm,step2.test$int\_rate)

## RF

rf2.model<-randomForest(int\_rate ~risk\_score+ loan\_amnt + term+dti +annual\_inc+sub\_grade,data=step2.train,

importance=TRUE,

ntree=2)

results.rf<-predict(rf2.model,step2.test)

accuracy(results.rf,step2.test$int\_rate)

#Cluster E

#Linear

set.seed(101)

sample = sample.split(LoanStats.step2.E$int\_rate, SplitRatio = .75)

step2.train = subset(LoanStats.step2.E, sample == T)

step2.test = subset(LoanStats.step2.E, sample == F)

step2.model<-lm(int\_rate ~risk\_score+ loan\_amnt + term+dti +annual\_inc+sub\_grade,data=step2.train)

summary(step2.model)

results.lm<-predict(step2.model,newdata = step2.test)

accuracy(results.lm,step2.test$int\_rate)

## RF

rf2.model<-randomForest(int\_rate ~risk\_score+ loan\_amnt + term+dti +annual\_inc+sub\_grade,data=step2.train,

importance=TRUE,

ntree=2)

results.rf<-predict(rf2.model,step2.test)

accuracy(results.rf,step2.test$int\_rate)

#Cluster F

#Linear

set.seed(101)

sample = sample.split(LoanStats.step2.F$int\_rate, SplitRatio = .75)

step2.train = subset(LoanStats.step2.F, sample == T)

step2.test = subset(LoanStats.step2.F, sample == F)

step2.model<-lm(int\_rate ~risk\_score+ loan\_amnt + term+dti +annual\_inc+sub\_grade,data=step2.train)

summary(step2.model)

results.lm<-predict(step2.model,newdata = step2.test)

accuracy(results.lm,step2.test$int\_rate)

## RF

rf2.model<-randomForest(int\_rate ~risk\_score+ loan\_amnt + term+dti +annual\_inc+sub\_grade,data=step2.train,

importance=TRUE,

ntree=2)

results.rf<-predict(rf2.model,step2.test)

accuracy(results.rf,step2.test$int\_rate)

#Cluster G

#Linear

set.seed(101)

sample = sample.split(LoanStats.step2.G$int\_rate, SplitRatio = .75)

step2.train = subset(LoanStats.step2.G, sample == T)

step2.test = subset(LoanStats.step2.G, sample == F)

step2.model<-lm(int\_rate ~risk\_score+ loan\_amnt + term+dti +annual\_inc+sub\_grade,data=step2.train)

summary(step2.model)

results.lm<-predict(step2.model,newdata = step2.test)

accuracy(results.lm,step2.test$int\_rate)

library(gains)

gain <- gains(step2.test$int\_rate, results.lm, groups=10)

# plot lift chart

plot(c(0,gain$cume.pct.of.total\*sum(step2.test$int\_rate))~c(0,gain$cume.obs),

xlab="# cases", ylab="Cumulative", main="", type="l")

lines(c(0,sum(step2.test$int\_rate))~c(0, dim(step2.test)[1]), lty=2)

# compute deciles and plot decile-wise chart

heights <- gain$mean.resp/mean(step2.test$int\_rate)

midpoints <- barplot(heights, names.arg = gain$depth, ylim = c(0,9),

xlab = "Percentile", ylab = "Mean Response", main = "Decile-wise lift chart")