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.

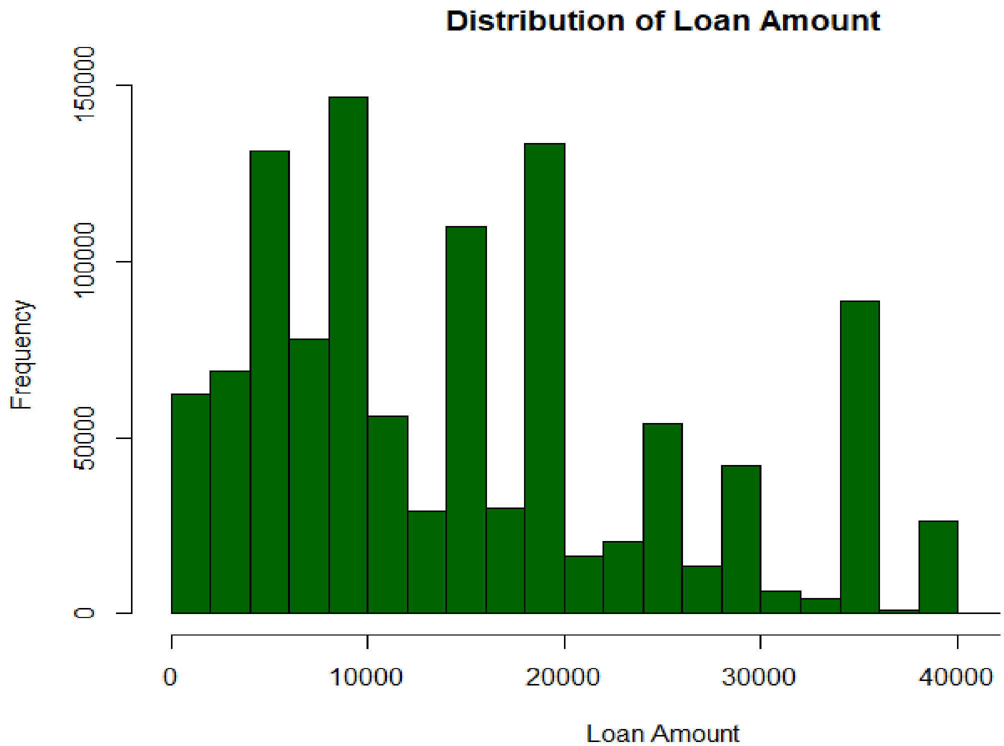
Due to lack of thorough background checks and standard models to analyze the potential borrowers, today there are a lot of people who are unable to pay off their debt and thus their debt keeps on increasing. This will make sure that the lending club won't be a success. If this problem is ignored, it 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.

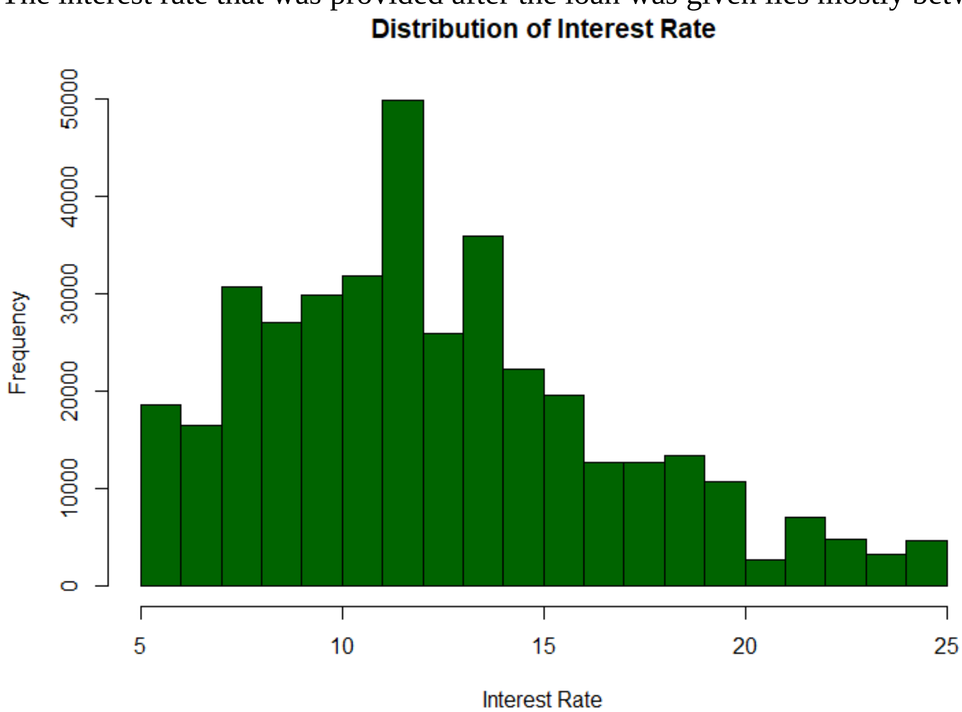
The second objective will be to build models to decide the interest allocated to a certain borrower. This can be done by one prediction model considering all borrowers or clustering the borrowers and then using the prediction algorithms based on the clusters. Then calculate the accuracy measures and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

II. Data Exploration and Visualization

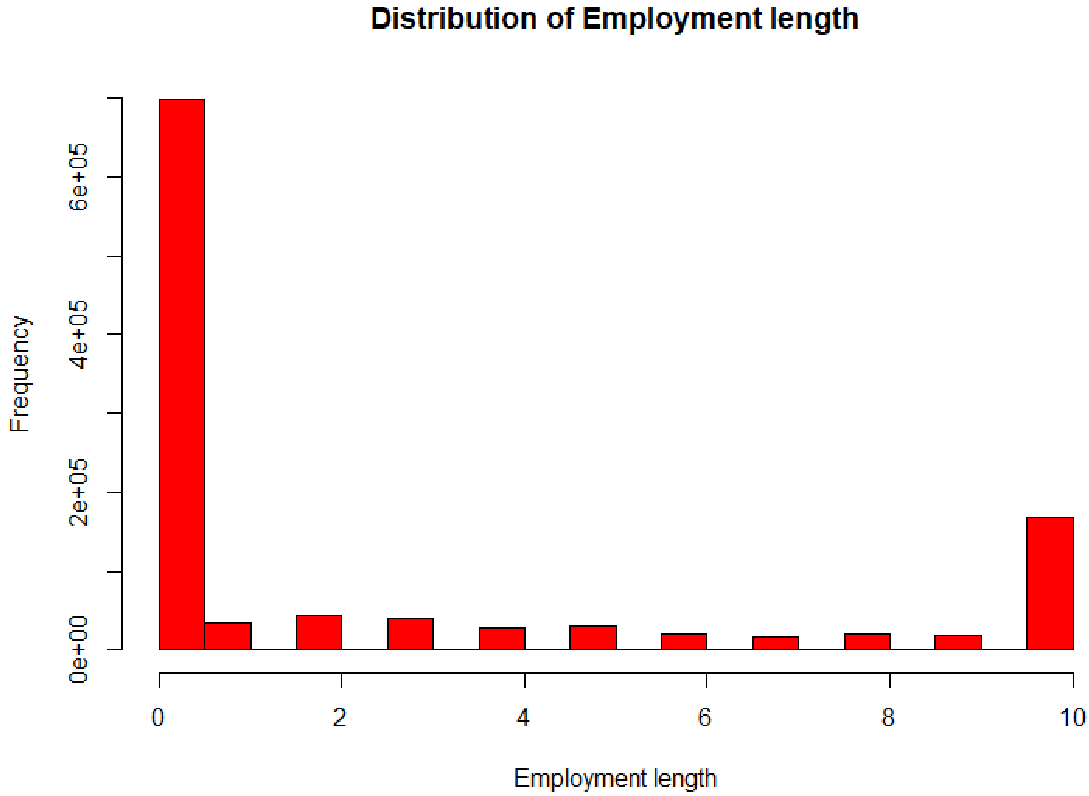
The data set includes two types of data, the loan details of the applications that were accepted for lending club and the details of the applications that were rejected. By performing Exploratory data analysis on these datasets, we have learned the following, the distribution of the loan amount is mostly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



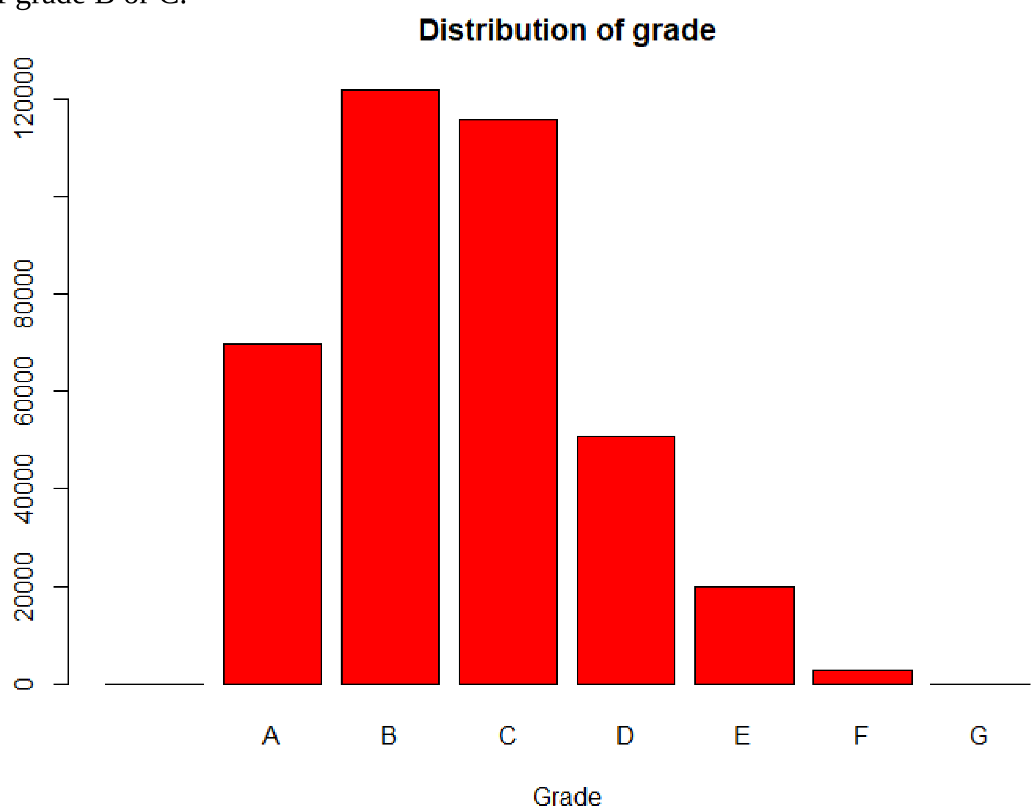
The interest rate that was provided after the loan was given lies mostly between 10%-15%



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,

```
> summary(LoanStats.step2)
   risk_score       loan_amnt          term                       purpose          addr_state
 Min.   :667.0   Min.   : 1000   Min.   :36.00   debt_consolidation:215402   CA     : 50626
 1st Qu.:677.0   1st Qu.: 8000   1st Qu.:36.00   credit_card       : 82197   TX     : 32553
 Median :692.0   Median :12300   Median :36.00   home_improvement  : 27686   NY     : 31257
 Mean   :701.2   Mean   :14596   Mean   :41.86   other             : 24783   FL     : 27448
 3rd Qu.:717.0   3rd Qu.:20000   3rd Qu.:36.00   major_purchase    :  9278   IL     : 15479
 Max.   :847.5   Max.   :38000   Max.   :60.00   medical           :  4755   NJ     : 13964
                                                 (Other)           : 15983   (Other):208757
 debt_settlement_flag      dti              annual_inc            grade          sub_grade          int_rate
  :     0             Min.   : -1.00   Min.   :      200   B      :121704   B5     : 27943   Min.   : 5.32
 N:378460             1st Qu.: 12.30   1st Qu.:   48000   C      :115534   C1     : 27779   1st Qu.: 9.16
 Y:  1624             Median : 18.18   Median :   67000   A      : 69573   B4     : 26210   Median :11.49
                      Mean   : 18.82   Mean   :   79702   D      : 50689   B3     : 23504   Mean   :12.49
                      3rd Qu.: 24.79   3rd Qu.:   95000   E      : 19798   C2     : 23224   3rd Qu.:14.99
                      Max.   :999.00   Max.   :9573072   F      :  2782   B1     : 23124   Max.   :24.99
                                                          (Other):     4   (Other):228300
```

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

```
> summary(RejectStats)
   loan_amnt                   purpose           risk_score          dti                 addr_state
 Min.   : 1000   debt_consolidation:382268   Min.   :300.0   Min.   :     -1.0   CA     : 91972
 1st Qu.: 6000   credit_card       :111440   1st Qu.:606.0   1st Qu.:     12.5   TX     : 69576
 Median :13550   other             : 56676   Median :640.0   Median :     24.4   NY     : 59436
 Mean   :15555   home_improvement  : 39668   Mean   :638.8   Mean   :    148.5   FL     : 58532
 3rd Qu.:20000   Debt consolidation: 35800   3rd Qu.:673.0   3rd Qu.:     42.6   PA     : 28284
 Max.   :50000   car               : 19196   Max.   :990.0   Max.   :317800.0   IL     : 27208
                 (Other)           :106168                                       (Other):416208
   emp_length          status
 Min.   : 0.0000   Min.   :0
 1st Qu.: 0.0000   1st Qu.:0
 Median : 0.0000   Median :0
 Mean   : 0.6938   Mean   :0
 3rd Qu.: 0.0000   3rd Qu.:0
 Max.   :10.0000   Max.   :0
```

The correlation between the terms used for prediction models ,

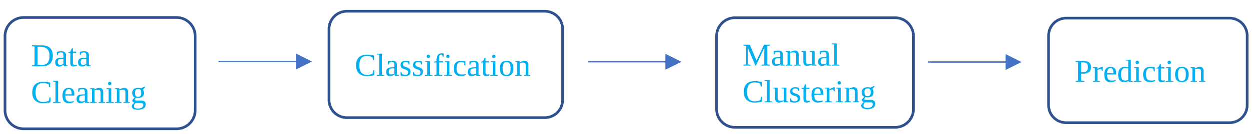
```
> cor(LoanStats.step2[,c(1,2,3,7,8,11)])
              risk_score  loan_amnt        term         dti  annual_inc    int_rate
risk_score   1.000000000 0.08679570 0.006305818 -0.06643431  0.06175167 -0.34564519
loan_amnt    0.086795702 1.00000000 0.369790538  0.02131044  0.31121694  0.14019340
term         0.006305818 0.36979054 1.000000000  0.05184095  0.05820556  0.38115471
dti         -0.066434312 0.02131044 0.051840948  1.00000000 -0.14575435  0.19350343
annual_inc   0.061751673 0.31121694 0.058205560 -0.14575435  1.00000000 -0.08819392
int_rate    -0.345645185 0.14019340 0.381154705  0.19350343 -0.08819392  1.00000000
```

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.

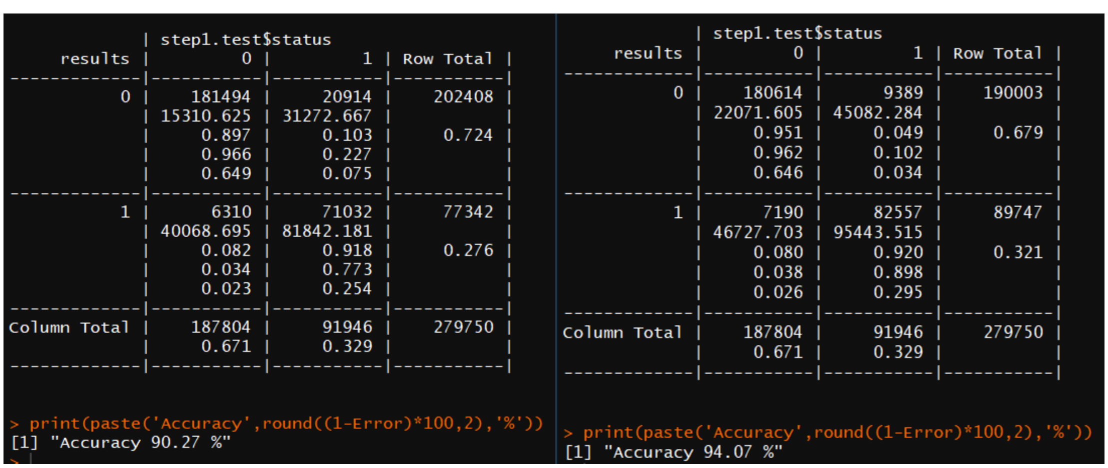
The first step is classification of a person to decide eligibility for loan, we build a logistic regression model and Random forest model for classification. We then test out these two models based on the training and test data set to check for accuracy and then select the most accurate one. Then we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



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 %



```
             | step1.test$status
     results |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |    181494 |     20914 |    202408 |
             | 15310.625 | 31272.667 |           |
             |     0.897 |     0.103 |     0.724 |
             |     0.966 |     0.227 |           |
             |     0.649 |     0.075 |           |
-------------|-----------|-----------|-----------|
           1 |      6310 |     71032 |     77342 |
             | 40068.695 | 81842.181 |           |
             |     0.082 |     0.918 |     0.276 |
             |     0.034 |     0.773 |           |
             |     0.023 |     0.254 |           |
-------------|-----------|-----------|-----------|
Column Total |    187804 |     91946 |    279750 |
             |     0.671 |     0.329 |           |
-------------|-----------|-----------|-----------|

> print(paste('Accuracy',round((1-Error)*100,2),'%'))
[1] "Accuracy 90.27 %"
```

```
             | step1.test$status
     results |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |    180614 |      9389 |    190003 |
             | 22071.605 | 45082.284 |           |
             |     0.951 |     0.049 |     0.679 |
             |     0.962 |     0.102 |           |
             |     0.646 |     0.034 |           |
-------------|-----------|-----------|-----------|
           1 |      7190 |     82557 |     89747 |
             | 46727.703 | 95443.515 |           |
             |     0.080 |     0.920 |     0.321 |
             |     0.038 |     0.898 |           |
             |     0.026 |     0.295 |           |
-------------|-----------|-----------|-----------|
Column Total |    187804 |     91946 |    279750 |
             |     0.671 |     0.329 |           |
-------------|-----------|-----------|-----------|

> print(paste('Accuracy',round((1-Error)*100,2),'%'))
[1] "Accuracy 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,

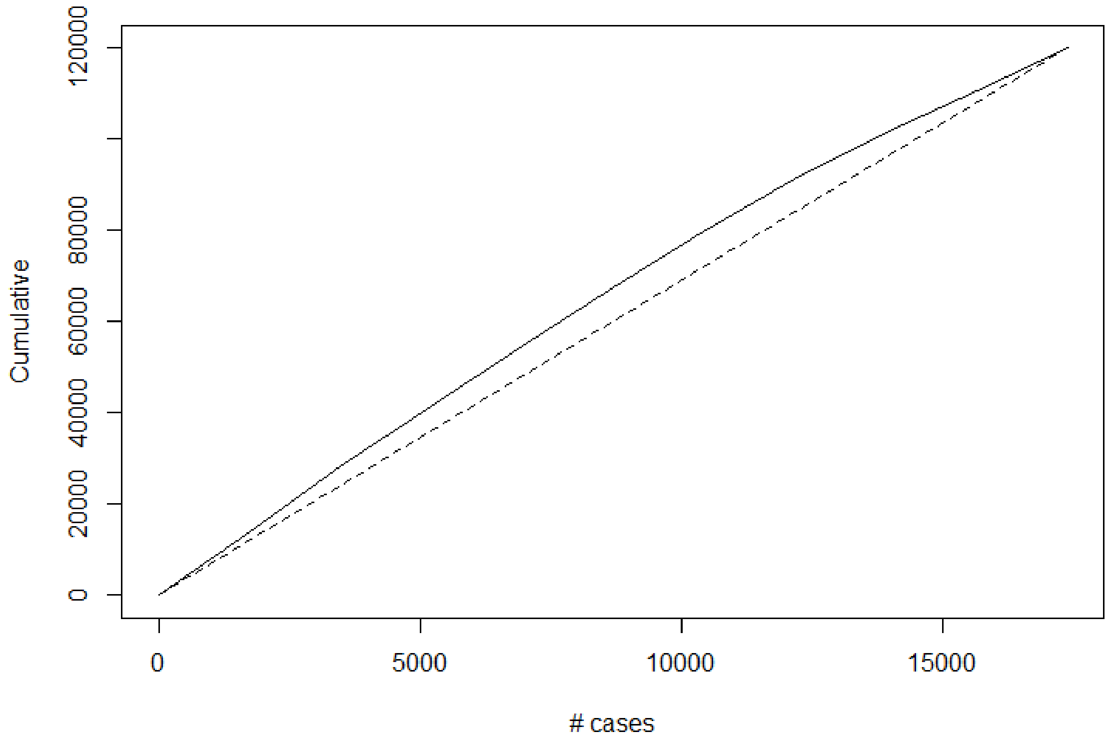
Cluster A	Cluster B	Cluster C	Cluster D	Cluster E	Cluster F	Cluster G
69574	121704	115534	50689	19798	2782	4

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,

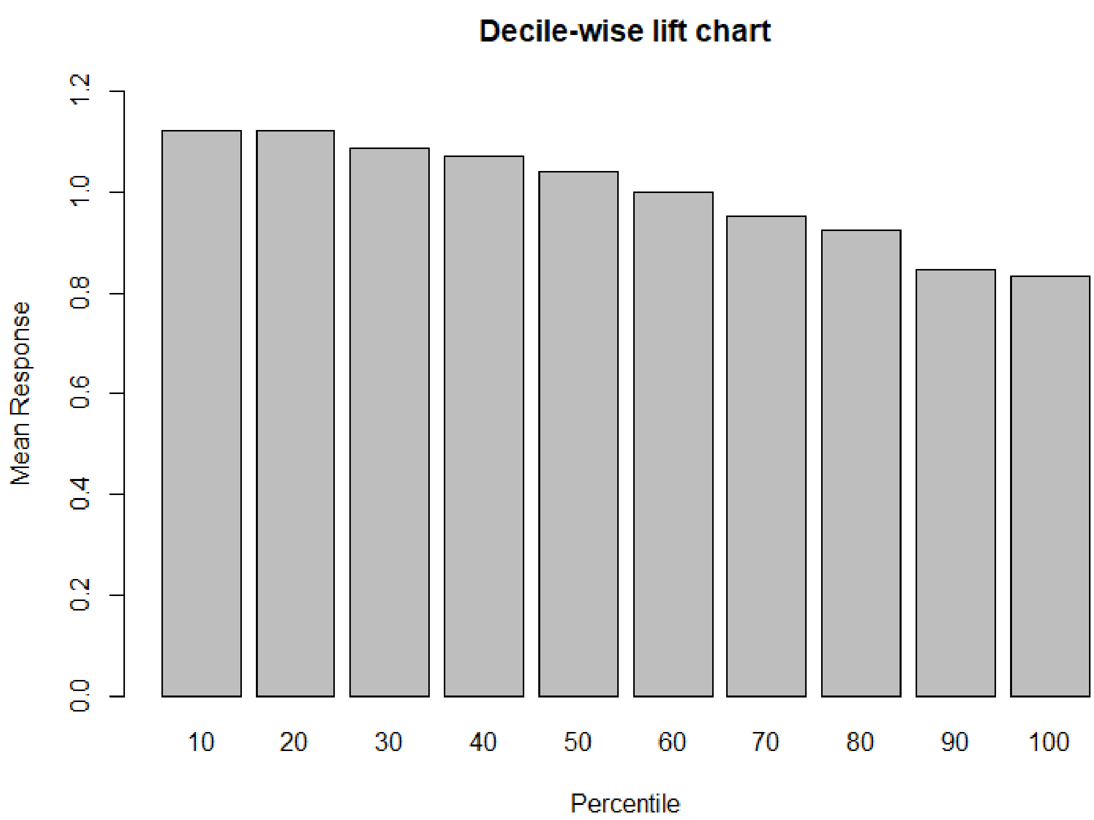
	Linear Regression				
	ME	RMSE	MAE	MPE	MAPE
Cluster A	0.000180503	0.2168954	0.1638487	-0.07850966	2.215986
Cluster B	1.10E-05	0.3802295	0.2603964	-0.1447836	2.612056
Cluster C	0.000128785	0.326936	0.2580412	-0.0737104	1.940845
Cluster D	0.00120558	0.6599789	0.521682	-0.1627899	2.890379
Cluster E	0.003628251	1.571917	1.417222	-0.5178821	6.475595
Cluster F	-0.0122289	0.7870952	0.2876092	-0.409065	1.560907
Cluster G	-18.83632	18.83632	18.83632	-313.9387	313.9387

	Random Forest				
	ME	RMSE	MAE	MPE	MAPE
Cluster A	0.007598558	0.3322382	0.2106659	-0.1084212	2.889621
Cluster B	-0.003312823	0.4170571	0.2733922	-0.1778681	2.749125
Cluster C	0.002795977	0.7175134	0.5095746	-0.1537561	2.824861
Cluster D	0.009818451	0.7197631	0.5102357	-0.112858	2.824586
Cluster E	-0.03072936	1.749533	1.441166	-0.6452882	6.597159
Cluster F	-0.007794894	0.8396221	0.2969098	-0.3867222	1.604093
Cluster G	0	0	0	0	0

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")
```