LOAN RISK PREDICTION: AN APPLICATION FOR INVESTORS AT LENDING CLUB

Boris Sertic

July 24, 2017

ABSTRACT

This paper shows the application of Logistic Regression for predictions if the loan will be fully repaid or not, and how investors can use prediction models when deciding about their investment portfolio. The prediction model is built using historical data from Lending Club for period from 2007 until 2017. The original dataset has 111 variables and 1,321,880 observations.

The logistic regression model is built on 14 independent variables and 552,625 observations after data cleaning process. The model is tested on test data that has 165,788 observations, and results are analyzed using metrics from confusion matrices, ROC-curves, and AUC values. Furthermore, model is improved using different cut-off values and using synthetic data generation method to overcome the problem of imbalanced classification. Then, strategy table is built that investors can use when deciding about their investment portfolio. Finally, the results are compared with models of other authors. Results show that loan risk models have fairly poor performance mainly because of the imbalanced classification problem. The AUC value of loan risk models for Lending Club are in the range [0.54 - 0.713]. The best performance shows extreme gradient boosting, followed by logistic regression model.

Keywords: logistic regression, supervised machine learning, imbalanced classification, loan risk models

THEORETICAL FRAMEWORK

Data Mining

Data mining is a process that uses statistical, mathematical, artificial intelligence and machine-learning techniques to extract and identify useful information and subsequent knowledge from large databases. It is a process of discovering patterns and correlations in data. These patterns must be meaningful, i.e. the business must have some kind of advantage such as to increase revenue, reduce operational costs, take advantage over competitors in the market, and understand customer behavior.

Machine learning is algorithm or model that learns patterns and correlations in data, and then predicts similar patterns in new data. There are three different methods of machine learning: supervised machine learning (making predictions on labeled data), unsupervised machine learning (finding structures in unlabeled data) and reinforcement learning.

Data mining is all about predictions (classification and regression), association (link analysis and sequence analysis) and clustering (outlier analysis). Predictions are supervised machine learning methods and popular algorithms include decision trees, random forest, genetic algorithms, logistic regression, linear and nonlinear

regressions, and support vector machine. On the other hand, association and clustering are unsupervised machine learning methods. The most popular algorithms include Apriori and ZeroR for association, while K-means and SOM are popular algorithms for clustering.

There are several important steps of data mining that I will follow in this project. These steps include: business understanding, data understanding, data preparation, model building and testing and evaluation.

Classifications

Classification is one of the most popular and commonly used data mining tool in real life situations. Classification includes two important steps: 1) training the classifier, and 2) testing the classifier. The first step is learning phase where a classifier learns patterns and correlations from past data (training data). In the second step, the classifier is tested with new data where the class label is unknown. Therefore, classification is data mining tool where our model predicts to which class new observation belongs based on previous discovered patterns in training data. Three most used classifiers are logistic regression, linear discriminant analysis, and K-nearest neighbors. In this project I will use logistic regression as a classifier for loan status.

Logistic Regression

Logistic regression has the same idea as linear regression. The difference is that with logistic regression we try to predict qualitative response (output is categorical variable).

Linear regression is defined as:

$$y = p(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n (1)$$

To model logistic regression that will give output between 0 and 1 (probabilities) we must use sigmoid function which is defined as:

$$p = \frac{1}{1 + e^- y} (2)$$

If we change y in (2) with y in (1) we get:

$$p = \frac{1}{1 + e^-(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)} (3)$$

If our dependent variable have two classes, then (3) is conditional probability

$$P(Y = 1|x_1, x_2, \ldots, x_n) = \frac{1}{1 + e^-(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)} (4)$$

If we multiply the numerator and the denominator of (4) by $e^(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)$, we get

$$P(Y = 1|x_1, x_2, \ldots, x_n) = \frac{e^(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}{1 + e^(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)} (5)$$

On the other hand, we have that

$$P(Y = 0|x_1, x_2, \ldots, x_n) = 1 - P(Y = 1|x_1, x_2, \ldots, x_n) = 1 - \frac{e^(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}{1 + e^(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}$$

After basic algebra operations we get

$$P(Y = 0|x_1, x_2, \ldots, x_n) = \frac{1}{1 + e^(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)} (6)$$

If we divide (5) with (6) we will get odds in favor that $Y = 1|x_1, x_2, \ldots, x_n$

$$\frac{P(Y = 1|x_1, x_2, \ldots, x_n)}{P(Y = 0|x_1, x_2, \ldots, x_n)} = \frac{\frac{e^(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}{1 + e^(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}}{\frac{1}{1 + e^(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}} = e^(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n) (7)$$

Finally, if we take the logarithm of both sides in (7) we will get

$$ln(\frac{p(x)}{(1 - p(x)}) = (\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n) (8)$$

As I mentioned the quantity $\frac{p(x)}{1-p(x)}$ is called the odds in favor of $p(x)$. Since this quantity is inside of ln function, the domain of ln function is interval $(0, \infty)$. Therefore, the odd quantity can take any value between 0 and ∞. If the value of this quantity is very low (close to 0) means that probability of $p(x)$ is very low, while if the value of this quantity is very high (close to ∞) means that probabilty of $p(x)$ is very high. The whole expression on left side in (8) is called logit or log-odds, while the right side is called linear predictor which is combination of parameters $\beta_0 + \beta_1 + \ldots + \beta_n$ to be estimated and variables (features) $x_1, \ldots, x_n$.

These unknown parameters we obtain by general method called maximum likelihood. Maximum likelihood is a very general approach that is used to fit many of the non-linear models.If we want to obtain parameters $\beta_0 + \beta_1 + \ldots + \beta_n$ we can fit the logistic model in R and get these coefficients. β_0 parameter is intercept in the logistic model, while other parameters are coefficients for features in our model $x_1, . . , x_n$. The interpretation of these parameters is very simple. If we assume that one of variables in our model x_k goes up by 1, while everything else *ceteris paribus*, from (7) we can see that the odds are multiplied by $e^(\beta_k x_k)$. If β_k is negative, then $e^(\beta_k)$ is less than 1 which means that the odds will decrease if x_k increases. On the other hand, if β_k is positive, then $e^(\beta_k)$ is greater than 1 which means that the odds will increase as x_k increases.

Once we have computed unknown parameters $\beta_0 + \beta_1 + \ldots + \beta_n$ it is very simple to make predictions using logistic regression model. If we want to predict the probability of default we need only to plug in (5) all estimated parameters $\beta_0 + \beta_1 + \ldots + \beta_n$ together with values of our features obtained from data set. This will give us the probability of default. Finally, we can determine the threshold (cut-off value) based on which our model will classify this observation to class 1 or class 0.

Resampling Methods

Resampling methods involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model. Two most commonly used resampling methods are cross-validation and the bootstrap. Cross validation is a model validation technique for assessing how the results of an analysis will generalize to an independent data set. The main goal of cross validation is to split data into training set and test set, so we are able to test the model during the training phase. The classifiers learns from training set about pattern, and then we test the model on test set during the training phase. There are several types of cross validation: 1) exhaustive cross-validation (leave-p-out cross-validation and leave-one-out cross-validation), and 2) non-exhaustive cross-validation (k-fold cross-validation). Resampling methods can be computationally expensive, because they involve fitting the same statistical method multiple times using different subsets of the training data.

Classification Accuracy

There is no best learning method in the real life situations. Some methods perform better on some data sets, while others perform better on different but similar data sets. Deciding which method to use in a particular situation can be very challenging task. For this reason we need some measures that will allow us to evaluate different methods.

For classification models there are several measures of accuracy that we can compute from confusion matrix. These measures include: accuracy, precision, recall (true P rate), true N rate, FP rate and error. The confusion matrix is a contingency table of correct and incorrect classifications. The elements on the diagonal in the confusion matrix are called TP (true positives) and TN (true negatives), while elements on off-diagonal are called FN (false negative) and FP (false positive).

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

The accuracy is defined as percentage of correctly classified instances.

$$ACC = \frac{(TP + TN)}{(P + N)} (9)$$

The error is defined as percentage of incorrectly classified instances.

$$ERR = \frac{(FP + FN)}{(P + N)} = 1 - ACC$$

Precision or Positive Predictive Value (PPV) is defined as fraction of real positive instances among all predicted positive instances.

$$PPV = \frac{TP}{(TP + FP)}$$

Sensitivity (recall, True Positive Rate, hit rate) is defined as fraction of real positive instances (relevant instances) among all actual relevant instances.

$$TPR = \frac{TP}{P} = \frac{TP}{(TP + FN)}$$

Specificity or True Negative Rate is defined as fraction of real negative instances among all actual negative instances.

$$TNR = \frac{TN}{N} = \frac{TN}{(TN + FP)}$$

One of the most overall reliable measures for models is the ROC (Receiver Operating Characteristic) curve. The ROC-curve represents true positive rates against false positive rates. The closer the curve is to left up corner, the model performance is better. In the real life situations when we compare different ROC-curves from different

models, it is very common that is not clear which model performs better.

BUSINESS UNDERSTANDING

Business Description

Lending Club is an online market place where investors and borrowers meet. It is the biggest peer-to-peer lending service where the borrowers can apply for a loan within few minutes. The procedure starts by registering online for both, investors and borrowers. Lending Club employees check if investors and borrowers meet their criteria. The verification procedure takes about seven days. After successful verification process, the borrower's loan will appear on the platform and will be valid for 15 days. On the other hand, investors can browse available loans and decide where to invest and how to build their investment portfolio. The minimum investment is 25 USD per loan. They can view many different information before deciding where to invest. These information include loan information (amount, purpose, grade, interest rate, length, monthly payment), borrower information (job title, income, state), and credit information. Borrower's personal details such as first and second name, address, personal number are not available to investors . Lending Club has been in the business since 2007, and has issued over 26 billion dollars in loans to more than 1.5 million borrowers.

Business Problem

The main concerns of investors is how to minimize risks, and maximize their investments. The risk at Lending Club is higher for investors than in the banks since deposits are not protected. This can result in the loss for the investors. This case study has two main objectives:

1. To build a loan risk model for investors using supervised machine learning method (classification) that will help investors to predict if a borrower will repay the loan based on historical data provided by Lending Club.
2. To help investors when deciding which investment strategy to choose.

DATA PREPARATION

Data Collection

Data collection process is conducted by Lending Club employees who are in charge of approving loans to be online for investors. All data are available on their website. Therefore, I did not collect data, but downloaded the csv files.

Data Description

The original raw data is obtained from the Lending Club web site, and contains 1,321,847 observations and 111 variables. Each observation (record, row) represents one borrower and his/her information. The analyzed period is for 10 years from 2007 to 2017. The data are available in eight different csv files. The original dataset contains many different variables which Lending Club collects during different stages of loan process. Many of these variables are not interesting for building the loan risk model, neither for data analysis which would be interesting for investors. Our analysis and model will be based on 19 variables which are shown in Table 2. These variables are grouped into different categories: application category (variables that contain data which each borrower

provides during application process such as income, home ownership, etc) and behavioral category (variables that describe borrower's behavior such as grade, open credit lines, number of derogatory public records, account balance, etc).

Loan_status The dependent variable in our model will be loan_status. In the raw dataset it has 9 different values. However, in our model we will use two values Fully Paid which means that the loan was paid, and Charged Off which means that there is no longer a reasonable expectation of further payments. We will encode this variable and 1 will represent Fully Paid, while 0 will represent Charged Off.

Grade is variable which belongs to behavioral category. Lending Club categorized borrowers into seven different loan grades from A (best) to G (poor). This grade is assigned by Lending Club employees after the borrowerâ€ s credit history is analyzed.

Sub_grade is variable similar to grade. Lending Club assigns loan subgrade within each loan grade. There are five sub-grades for each grade. For example, for grade A subgrades are A1, A2, A3, A4, and A5.

Int_rate is variable which represents the annual rate on the loan.

Open_acc is the number of open credit lines in the borrowerâ€ s credit file.

Pub_rec represents the number of derogatory public records.

Emp_length represents borrower's employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.

Annual_inc represents self-reported annual income provided by the borrower during registration.

Home_ownership status is provided by the borrower during registration or obtained from the credit report.

Dti is a debt-to-income ratio calculated using the borrowerâ€ s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrowerâ€ s self-reported monthly income. Purpose represents self-reported purpose of loan provided by the borrower during application process.

Addr_state is the state where the borrower live and is provided by the borrower in the loan application.

Loan_amnt is the listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.

Installment is the monthly payment owed by the borrower if the loan originates.

Revol_bal is total credit revolving balance.

Revol_util is revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.

Delinq_2yrs is the number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years.

Inq_last_6mths is the number of inquiries in past 6 months.

Data Cleaning

Data cleaning is a crucial and most time consuming step in the process of building a model. Data scientists, according to interviews and expert estimates, spend from 50% to 80% of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.

First we will install all packages that we are going to use in this project.

```
#PART 1-Installing and loading R packages
install.packages("gmodels")
install.packages("lubridate")
install.packages("plyr")
install.packages("ggplot2")
install.packages("caTools")
install.packages("e1071")
install.packages("ROCR")
install.packages("caret")
install.packages("ROSE")
library(gmodels)
library(lubridate)
library(plyr)
library(ggplot2)
library(caTools)
library(e1071)
library(ROCR)
library(caret)
library(ROSE)
#END OF PART 1
```

```
#PART 2- Loading files into R data frame
#Loading data into R data frame
#It takes around 6 minutes to load
raw.data <- NULL
files<-list.files()
system.time(for(f in files) {
  temp<-read.csv(f, sep=",", dec=",", header = FALSE)
  raw.data<-rbind(raw.data,temp)
})

#END OF PART 2
```

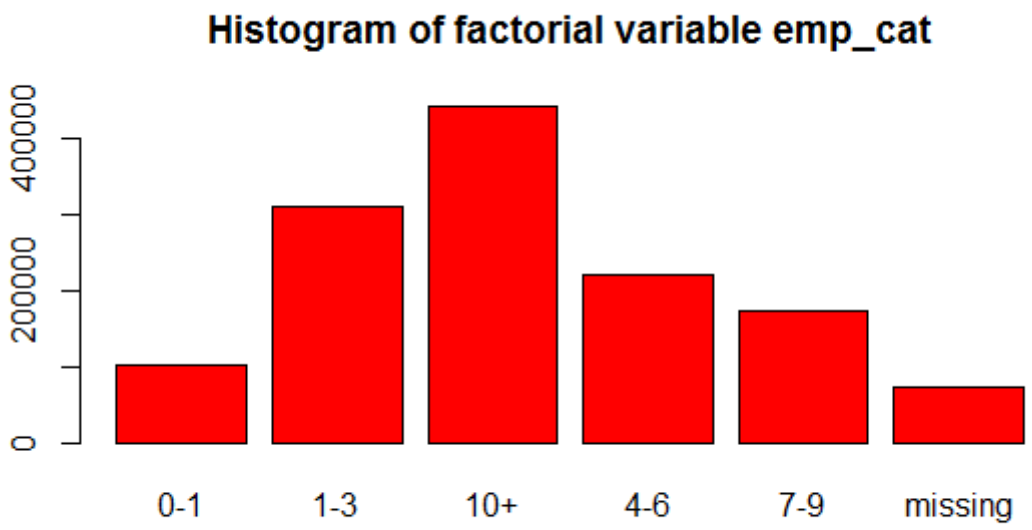
```
#PART 3 DATA CLEANING
#Exploring raw data
dim(raw.data) #111 variables 1.321.880 observations
class(raw.data) #Data frame
names(raw.data) #Column names are not copied.
head(raw.data, n=20)
tail(raw.data, n=20)
str(raw.data) #All features are factors
#Assign headers based on existing row in data frame
colnames(raw.data) <- as.character(unlist(raw.data[2,]))

#Deleting trash rows
arg <- raw.data$id=="id"
raw.data <- subset(raw.data, arg==FALSE)

#Selecting relevant features for model
features <- c("loan_status", "grade", "sub_grade", "open_acc","pub_rec", "dti", "delinq_
2yrs",
              "inq_last_6mths", "emp_length", "annual_inc", "home_ownership",  "purpose"
, "addr_state",
              "loan_amnt","int_rate", "installment", "issue_d", "revol_bal", "revol_uti
l")

raw.data <- subset(raw.data, select = features)

#Deleting empty rows
raw.data <- raw.data[!apply(raw.data == "", 1, all),]
```

Missing Values

Raw dataset has many missing values in many variables. The important part of data preprocessing phase is to properly deal with the missing values. There are three general methods how to manage missing values: delete rows / columns containing missing values, replace missing values, or keep missing values. When deleting missing values we can delete whole record containing missing value, or delete entire variable. The observations should be deleted only when there are few observations containing missing values, while variables should be deleted only if that variable contains a lot of missing values. In the case of replacement, the best practice is to use median imputation which means to replace missing values with the measure of the center of variable. The last method, keeping missing values, should be used when they can be the source of the valuable information

In my dataset the most problematic variable is emp_length which has 73.049 missing values. In this case, the best practice is to use coarse classification (putting variable in bins) since some models will delete observations with missing values as they cannot handle missing values. Coarse classification allows to simplify data and improve the interpretability of the model. Coarse classification requires to bin responses into groups that contain ranges of values . For this reason, I will introduce a new variable emp_cat that will replace emp_length variable. The new variable will be factor and will have 6 bins (<1, 1-3, 3-6, 6-9, 10+, missing). Picture shows histogram of factorial variable emp_cat including missing values as a level.

```
#Missing values for emp_length
summary(raw.data$emp_length) #There are 73.049 missing values

#Will keep missing values. Categorical variable. Will make six bins <1, 1-3, 3-6, 6-9, 1
0+, missing
options(scipen = 50)
plot(raw.data$emp_length, col="red")
raw.data$emp_cat <- rep(NA, length(raw.data$emp_length))
raw.data$emp_cat[which(raw.data$emp_length == "< 1 year")] <- "0-1"
raw.data$emp_cat[which(raw.data$emp_length == "1 year" | raw.data$emp_length=="2 years"
| raw.data$emp_length=="3 years")] <- "1-3"
raw.data$emp_cat[which(raw.data$emp_length == "4 years" | raw.data$emp_length=="5 years"
| raw.data$emp_length=="6 years")] <- "4-6"
raw.data$emp_cat[which(raw.data$emp_length == "7 years" | raw.data$emp_length=="8 years"
| raw.data$emp_length=="9 years")] <- "7-9"
raw.data$emp_cat[which(raw.data$emp_length == "10+ years")] <- "10+"
raw.data$emp_cat[which(raw.data$emp_length == "n/a")] <- "missing"
raw.data$emp_cat <- as.factor(raw.data$emp_cat)
plot(raw.data$emp_cat, col="red", main="Histogram of factorial variable emp_cat")
summary(raw.data$emp_cat)
raw.data$emp_length <- NULL
```



Picture 2. Histogram of factorial variable emp_cat after using coarse classification to handle missing values

Revol_util variable has 766 missing values. In this case, I replaced missing values with median. Other variables did not have missing values. Therefore, the process of data cleaning included converting type of variable into correct type and removing characters from values. For example, the variable int_rate was factor with values in format 12.3%. After data cleaning, the same variable is numeric in format 0.123.

Loan_status is the response variable in our model and deserves special attention. There are 9 different values for this variable. However, since we are going to build a model which will predict if the borrower is going to repay the loan or not, we are going to keep for our model only observations where loan_status contains values “Fully Paid“ or “Charged Off“. Moreover, we are going to encode this variable so that 1 represents Fully Paid, and 0 represents Charged Off. As a result, our dataset is reduced to 553,403 observations.

R code for preparing data for analysis

```
#Preparing data for analysis
#int_rate variable
class(raw.data$int_rate) #It is factor, should be numeric
raw.data$int_rate <- as.numeric(sub("%","",raw.data$int_rate)) #Taking out % sign and co
nverting into numeric
raw.data$int_rate <- raw.data$int_rate / 100
is.numeric(raw.data$int_rate) # TRUE
anyNA(raw.data$int_rate) #No missing values

#revol_util variable
class(raw.data$revol_util) #It is factor, should be numeric
raw.data$revol_util <- as.numeric(sub("%","",raw.data$revol_util)) #Taking out % sign an
d converting into numeric
raw.data$revol_util <- raw.data$revol_util / 100
is.numeric(raw.data$revol_util) # TRUE
anyNA(raw.data$revol_util) #There are missing values

index.NA <- which(is.na(raw.data$revol_util)) #766 missing values
raw.data$revol_util[index.NA] <- median(raw.data$revol_util, na.rm = TRUE) #All missing
 values replaced by median 0.542
anyNA(raw.data$revol_util) #No missing values

#revol_bal variable
class(raw.data$revol_bal) #It is factor, should be numeric
raw.data$revol_bal <- as.character(raw.data$revol_bal) #Converting into character
raw.data$revol_bal <- as.numeric(raw.data$revol_bal) # Converting into numeric
anyNA(raw.data$revol_bal) #No missing values

#installment variable
class(raw.data$installment) #It is factor, should be numeric
raw.data$installment <- as.character(raw.data$installment) #Converting into character
raw.data$installment <- as.numeric(raw.data$installment) #Converting into numeric
is.numeric(raw.data$installment) # TRUE
anyNA(raw.data$installment) #No missing values

#loan_amnt
class(raw.data$loan_amnt) #It is factor, should be numeric
raw.data$loan_amnt <- as.character(raw.data$loan_amnt) #Converting into character
raw.data$loan_amnt <- as.numeric(raw.data$loan_amnt) #Converting into numeric
is.numeric(raw.data$loan_amnt) # TRUE
anyNA(raw.data$loan_amnt) #No missing values

#annual_inc
class(raw.data$annual_inc) #It is factor, should be numeric
raw.data$annual_inc <- as.character(raw.data$annual_inc) #Converting into character
raw.data$annual_inc <- as.numeric(raw.data$annual_inc) #Converting into numeric
is.numeric(raw.data$annual_inc) # TRUE
anyNA(raw.data$annual_inc) #4 missing values
index.NA <- which(is.na(raw.data$annual_inc))
raw.data$annual_inc[index.NA] <- median(raw.data$annual_inc, na.rm = TRUE)
anyNA(raw.data$annual_inc) #No missing values

#laon_status
```

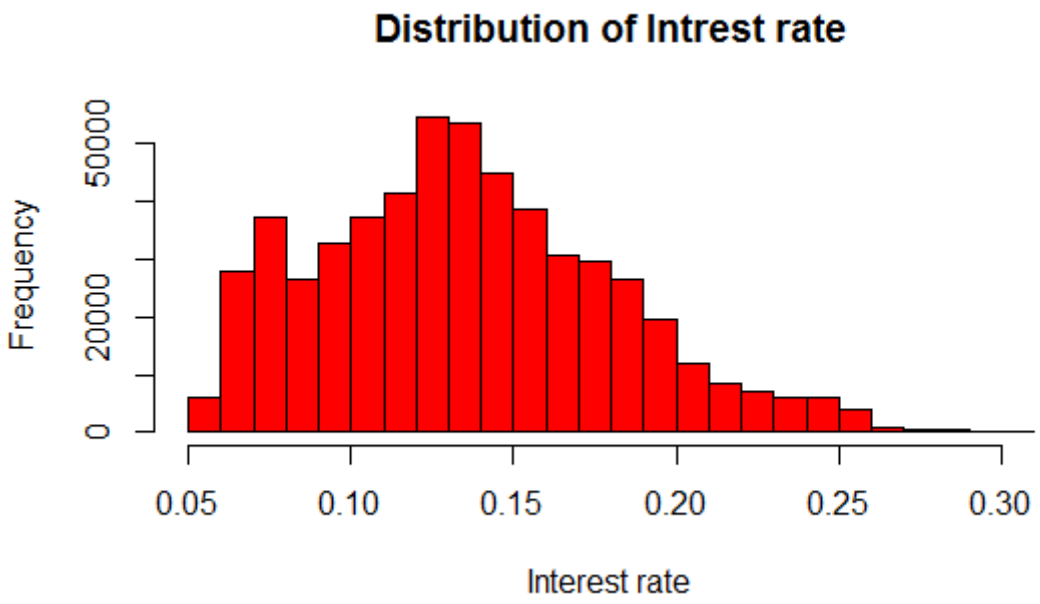
```
class(raw.data$loan_status) #It is factor
raw.data$loan_status <- as.character(raw.data$loan_status)
is.character(raw.data$loan_status)
#Taking only rows where laon_status is fully paid or charged off
arg <- raw.data$loan_status=="Fully Paid" | raw.data$loan_status=="Charged Off"
raw.data <- subset(raw.data, arg==TRUE) #Number of observations reduced to 553403

#Encoding loan_status 0 - Charged Off, 1 - Fully paid
raw.data$loan_status <- ifelse(raw.data$loan_status=="Fully Paid",1,0)
raw.data$loan_status <- as.integer(raw.data$loan_status) #Converting to integer
is.integer(raw.data$loan_status)
anyNA(raw.data$loan_status)

#dti
class(raw.data$dti) #It is factor, should be numeric
raw.data$dti <- as.character(raw.data$dti) #Converting into character
raw.data$dti <- as.numeric(raw.data$dti) #Converting into numeric
is.numeric(raw.data$dti) # TRUE
anyNA(raw.data$dti) #No missing values

#open_acc
class(raw.data$open_acc) #It is factor, should be numeric
raw.data$open_acc <- as.character(raw.data$open_acc) #Converting into character
raw.data$open_acc <- as.numeric(raw.data$open_acc) #Converting into numeric
is.numeric(raw.data$open_acc) # TRUE
anyNA(raw.data$open_acc) #No missing values

#pub_rec
class(raw.data$pub_rec) #It is factor, should be numeric
raw.data$pub_rec <- as.character(raw.data$pub_rec) #Converting into character
raw.data$pub_rec <- as.numeric(raw.data$pub_rec) #Converting into numeric
is.numeric(raw.data$pub_rec) # TRUE
anyNA(raw.data$pub_rec) #No missing values

#delinq_2yrs
class(raw.data$delinq_2yrs) #It is factor, should be numeric
raw.data$delinq_2yrs <- as.character(raw.data$delinq_2yrs) #Converting into character
raw.data$delinq_2yrs <- as.numeric(raw.data$delinq_2yrs) #Converting into numeric
is.numeric(raw.data$delinq_2yrs) # TRUE
anyNA(raw.data$delinq_2yrs) #No missing values

#inq_last_6mths
class(raw.data$inq_last_6mths) #It is factor, should be numeric
raw.data$inq_last_6mths <- as.character(raw.data$inq_last_6mths) #Converting into charac
ter
raw.data$inq_last_6mths <- as.numeric(raw.data$inq_last_6mths) #Converting into numeric
is.numeric(raw.data$inq_last_6mths) # TRUE
anyNA(raw.data$inq_last_6mths) #No missing values

str(raw.data)
```

EXPLORATORY DATA ANALYSIS

After data cleaning, we can start with exploratory data analysis (EDA) which is one of the most important step before building a model in order to detect potential outliers and mistakes. According to Seltman, we use EDA for detection of mistakes, checking of assumptions, preliminary selection of appropriate models, determining relationships among the explanatory variables, and assessing the direction and rough size of relationships between explanatory and outcome variables (Seltman, page 61, 2008).

Distribution of Interest Rate

The first thing what would be of interest for investors is distribution of interest rate. The best way to inspect distribution of interest rates is by plotting histogram and by computing 5-number summary statistics. Picture 3 shows distribution of interest rates, while in Picture 4 we can see 5-number summary statistics. From histogram, we can see that only few observations are with interest rate higher than 25%, while most of interest rates are between 12% and 15%. From 5-number summary statistics we can see that 75% of observations have interest rate lower than 16.49%, while 50% of all loans have interest rates between 10.16% and 16.49%. The minimum interest rate is 5.32%, while the maximum interest rate is 30.99%. Therefore, we can conclude that figures for interest rates make sense. They are a bit higher compared with banks' interest rates, but this is because these loans are not protected. On average, investor can expect interest rate of 13.61%.



Picture 3. Histogram of Interest Rate

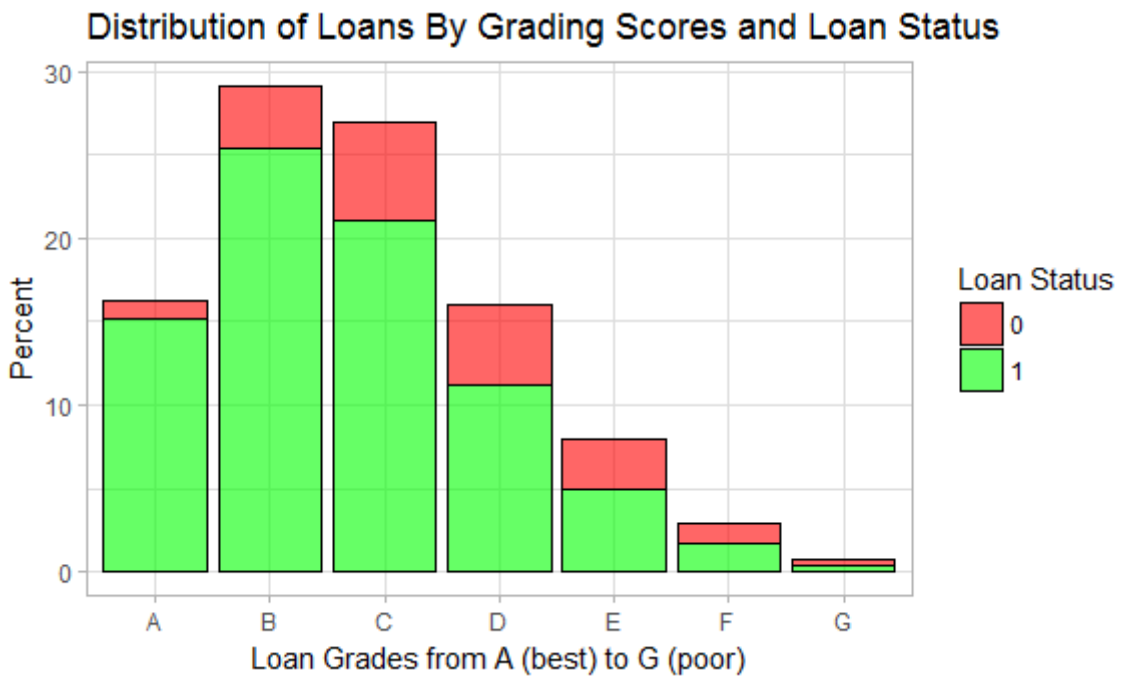
```
> summary(raw.data$int_rate)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0532  0.1016  0.1333  0.1361  0.1649  0.3099
```

Picture 4. 5-number summary statistics for interest rates

Distribution of Grading Scores and Loan Status

Distribution of grading score is another diagram that can be interesting for investors. First of all, investor can see the percentage of loans in terms of grading. Secondly, they can check if Lending Club Credit Approval Department controls seriously credit history of borrowers before making them available for investments. While making this diagram I noticed that it would be better if the response variable is factor instead of integer. Therefore, I made correction and converted loan_status variable to factor. Then, I made a histogram of grading scores colored by loan status and presented with percentages. Picture 5 shows distribution of loans by grading score

and loan status. Generally speaking, from the histogram we can see that proportion of loans which are not fully paid increases for poorer grades. For example, it is obvious from the Picture 6 that just few percentage of unpaid loans belongs to grade A, while for grades F and G this proportion increases significantly. Moreover, we can see that more than 50% of all loans are graded into scores B and C. For more details we can use CrossTable function which will produce contingency table. The results of this function are presented in Picture 6.



Picture 5. Distribution of Loans by Grading Scores and Loan Status

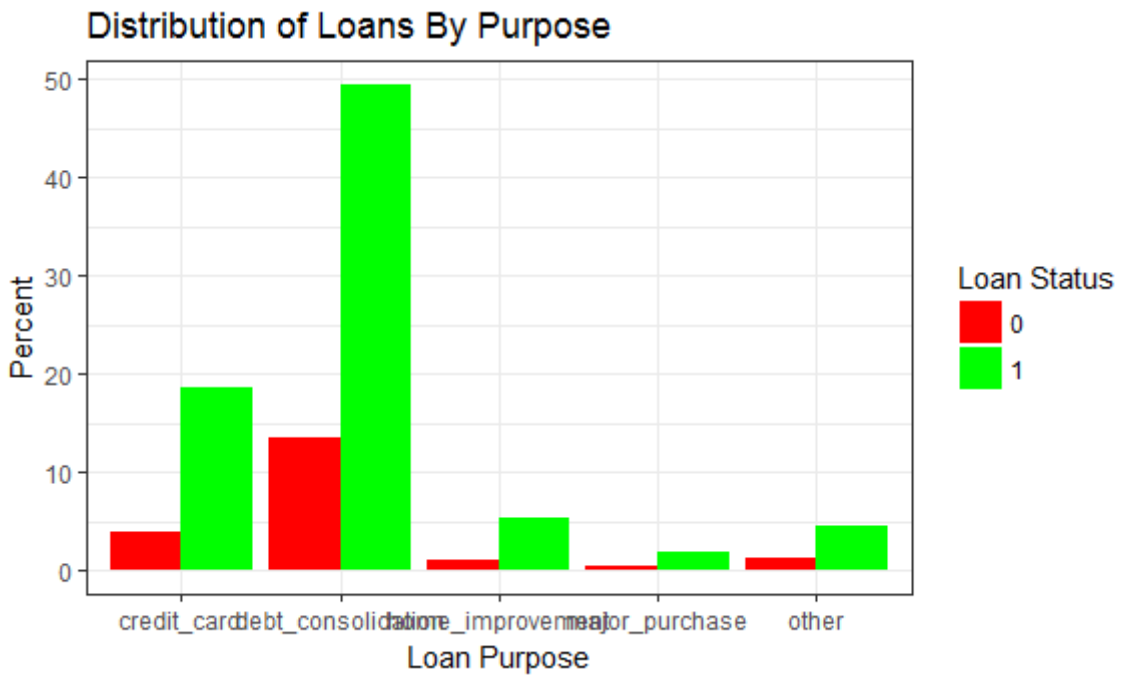
From Picture 5 we can see that 16.2% of all loans belongs to best grading score A, and that only 6.5% of A loans are not paid. Moreover, we can see that for last two grades F and G percentage of unpaid loans are 44.3% and 48.9%. However, F loans make only 3% of all loans, while G loans make 0.8% of all loans. Therefore, we can conclude that loans grading is correct as the percentage of unpaid loans increase as grade score quality decreases. Also, investors can be ensured that Lending Club takes seriously credit approval since only 3.8% of all loans belongs two lowest scoring grades. In addition, we used only two outcomes of response variable (Fully Paid and Charged Off) so we are not aware if only few installments are not paid for these poor loans. We can include other outcomes of response variable in our analysis, but that is not necessary as only 3.8% of all loans are classified in F and G.

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|-------------------------|


Total Observations in Table:  553403


               | raw.data$loan_status
raw.data$grade |         0 |         1 | Row Total |
---------------|-----------|-----------|-----------|
             A |      5786 |     83815 |     89601 |
               |     0.065 |     0.935 |     0.162 |
---------------|-----------|-----------|-----------|
             B |     20804 |    140474 |    161278 |
               |     0.129 |     0.871 |     0.291 |
---------------|-----------|-----------|-----------|
             C |     32629 |    116626 |    149255 |
               |     0.219 |     0.781 |     0.270 |
---------------|-----------|-----------|-----------|
             D |     26209 |     62088 |     88297 |
               |     0.297 |     0.703 |     0.160 |
---------------|-----------|-----------|-----------|
             E |     16985 |     27356 |     44341 |
               |     0.383 |     0.617 |     0.080 |
---------------|-----------|-----------|-----------|
             F |      7269 |      9150 |     16419 |
               |     0.443 |     0.557 |     0.030 |
---------------|-----------|-----------|-----------|
             G |      2058 |      2154 |      4212 |
               |     0.489 |     0.511 |     0.008 |
---------------|-----------|-----------|-----------|
  Column Total |    111740 |    441663 |    553403 |
---------------|-----------|-----------|-----------|
```

Picture 6. Contingency Table Using Cross Table Function

Distribution of Loans by Purpose

The next interesting analysis is to check distribution of loans by purpose. There are 14 different factors for this variable. Picture 7 shows 5 most common purposes of taking loans by borrowers at Lending Club. We can see that more than half of loans are taken due to debt consolidation, while around 20% for repaying credit card debts. On the other hand, from Table 1 we can see that the most risky loans are for small businesses as 30% of these loans are not fully paid. However, these loans make only 1.4% of all loans. We can conclude that data make sense since this kind of loans are mostly created for refinancing previous debts and debts on credit cards since credit cards have higher interest rates. In addition, we see that house credits make only 0.5% of all credits which makes sense since banks offer much better loans for houses. Therefore, we can ensure investors that data is valid.



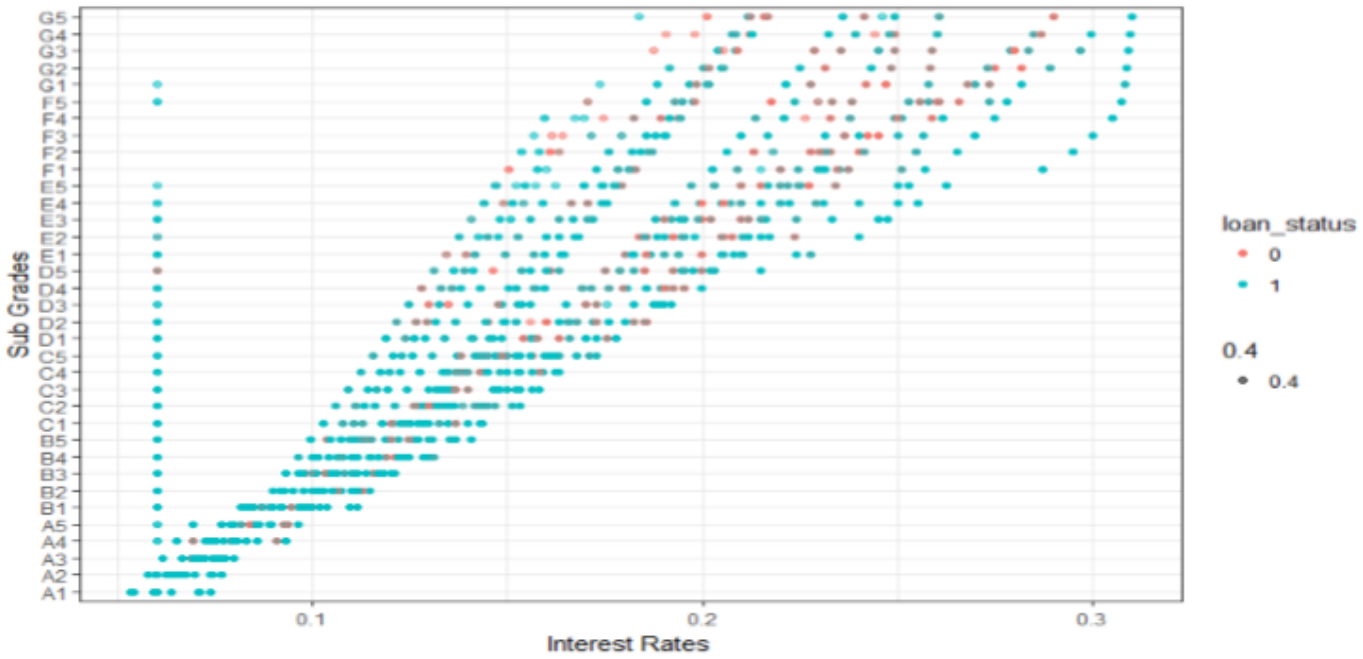
Picture 7. Histogram for Distribution of Loans by Most Common Purposes

car	896	5551	6447
	0.139	0.861	0.012
credit_card	20200	97074	117274
	0.172	0.828	0.212
debt_consolidation	69913	257604	327517
	0.213	0.787	0.592
educational	56	270	326
	0.172	0.828	0.001
home_improvement	5927	27734	33661
	0.176	0.824	0.061
house	619	2352	2971
	0.208	0.792	0.005
major_purchase	2109	10018	12127
	0.174	0.826	0.022
medical	1297	4511	5808
	0.223	0.777	0.010
moving	950	2996	3946
	0.241	0.759	0.007
other	6393	23143	29536
	0.216	0.784	0.053
renewable_energy	107	344	451
	0.237	0.763	0.001
small_business	2347	5384	7731
	0.304	0.696	0.014

Table 1. Contingency Table Using Cross Table Function

Relation between Grades and Interest Rate

Another important observation to check validity of dataset is to examine relation between interest rates and grade scores. Better grades should have lower interest rates, while poorer grades should have higher interest rates. The best way to control this relation is by making a scatter diagram.



Picture 8. Scatter Diagram Showing Relation between Interest Rates and Sub Grades

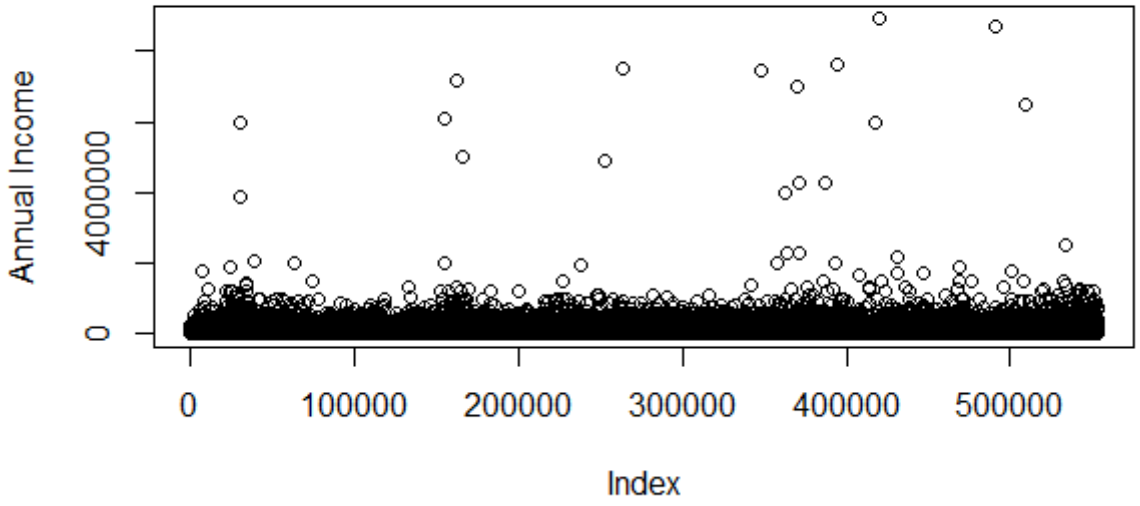
Picture 8 shows clearly that interest rates get higher as grade loan gets poorer. We can see that all loans graded with A (A1 – A5) have interest rates less than 10%. Also we can see that there are no many unpaid loans in this range. Also we can see that unpaid loans increase significantly for interest rates higher than 20%. However, two interesting facts draw attention. First, it seems that all loans with the highest interest rates (>30%) are fully paid. Detail inspection shows that there are in total 71 loans with these interest rates, and 91.5% of these loans are fully paid. Second, we can see some potential mistakes in our data that should be investigated. Almost every sub-grade has one observation with the same interest rate. By detail inspection of these records we can conclude that this is caused due to human mistake error when handling application or data entry process since all these loans share similar characteristics in term of other features such as dti, and pub_rec which shows credit history and behavior of borrower. Therefore, we can delete these observations. In total 103 observations were deleted from dataset.

Outlier Detection

If we observe 5-number summary statistic or if we make a boxplot or scatter plot of annual income, we can see that there are many outliers. As it is shown in Picture 11 the maximum value of annual income is $ 8,900,000 USD, while median is $ 64,000 USD, and 75% of all data is below $ 45,000 USD. Detail inspection shows that a borrower with annual income of $ 8,900,000 USD requested a loan of only $ 18,000 USD for debt consolidation. Therefore, it is clear that in this variable we have outliers due to mistakes. The best way to get clear picture is to plot a scatter diagram that is shown in Picture 12. As we can see from Picture 12, there are around 25 outliers.

```
> summary(raw.data$annual_inc) #There are potential outliers
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0   45000   64000   74470   90000 8900000
```

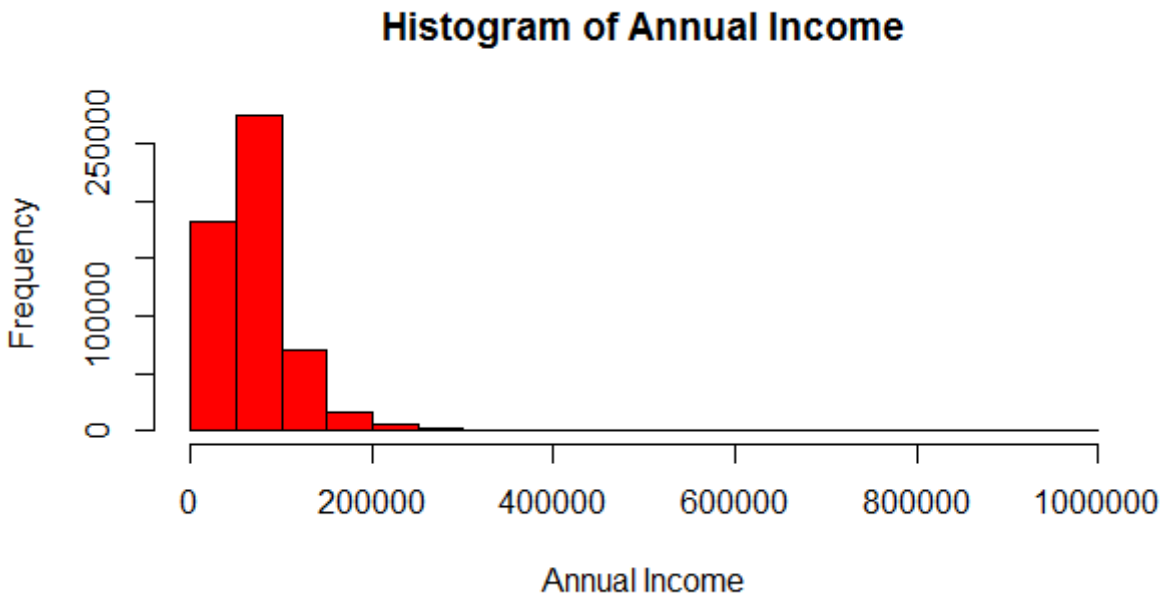
Picture 11. 5-number Summary Statistics for Annual Income



Picture 12. Scatter plot of Annual Income to Detect Outliers

There are two main methods how we can handle these outlier. The first method is using the rule of thumb where upper limit is computed as 1.5 * IRQ, where IRQ = 3rd Quartile – 1st Quartile. Therefore, upper limit for outliers would be $ 157,500 USD.

The second method is using expert judgment. According to the official statistics, the median income of US household is $ 55,775 USD. Therefore, in this case all annual incomes which are larger than $ 1,000,000 USD I will consider as outliers and will remove from dataset. There are 91 observations that meet this criteria and which were deleted.



Picture 13. Distribution of Annual Income

Picture 13 depicts distribution of annual income after upper outliers were removed. We can see that distribution is not normal and it is skewed to the right with most observations that fall within range between 50,000 and 100,000. This make sense since annual income usually follows this shape (only small percentage of population

have extremely high income), while majority of population falls within range which is in line with official statistics. Therefore, we can conclude that there is no unusual behavior in this feature.

The same producers was conducted for all other 17 variables. The outliers and missing values were treated in accordance to methods described before.

Multicollinearity

The final step of exploratory data analysis is to check correlations among all variables in order to decide which features we will include in the model. The main reason for doing this step is to identify features which are highly correlated. This can make problems to model due to multicollinearity. The test for multicollinearity is shown in Picture 14. As we can see there is a strong positive relations between loan amount and installment (0.953837272), and loan amount and annual income (0.42590538). This makes sense since installment will be determine on loan amount since the number of installments are fixed. Therefore, we can exclude installment from our model. Moreover, we can exclude sub_grade from our model as it is correlated with grade feature.

```
> cor(raw.data[, sapply(raw.data, class) != "factor"]) #Checking multicollinearity
                open_acc     pub_rec         dti  delinq_2yrs inq_last_6mths  annual_inc   loan_amnt    int_rate installment   revol_bal  revol_util
open_acc      1.00000000 -0.02017412  0.30238154  0.054870133    0.119054701  0.17716495  0.19703451  0.00716295  0.18728414  0.23553517 -0.13445308
pub_rec      -0.02017412  1.00000000 -0.04001767 -0.010656080    0.057046856 -0.01801314 -0.08111109  0.06305221 -0.06980527 -0.11540750 -0.08130215
dti           0.30238154 -0.04001767  1.00000000 -0.005466742   -0.003451065 -0.21575272  0.03961463  0.17625294  0.03769738  0.15071305  0.19108049
delinq_2yrs   0.05487013 -0.01065608 -0.00546674  1.000000000    0.025865776  0.06394978  0.00422376  0.06544480  0.01416584 -0.03174310 -0.01722904
inq_last_6mths 0.11905470  0.05704686 -0.00345107  0.025865776   1.000000000  0.05368842 -0.00943231  0.22370100  0.01242174 -0.01624004 -0.08288378
annual_inc    0.17716495 -0.01801314 -0.21575272  0.063949777    0.053688422  1.00000000  0.43517973 -0.07521848  0.42590538  0.38212965  0.04049021
loan_amnt     0.19703451 -0.08111109  0.03961463  0.004223763   -0.009432309  0.43517973  1.00000000  0.16570909  0.95383727  0.34842395  0.10339128
int_rate      0.00716295  0.06305221  0.17625294  0.065444798    0.223700998 -0.07521848  0.16570909  1.00000000  0.16086285 -0.01523279  0.28628434
installment   0.18728414 -0.06980527  0.03769738  0.014165842    0.012421737  0.42590538  0.95383727  0.16086285  1.00000000  0.33497494  0.12569414
revol_bal     0.23553517 -0.11540750  0.15071305 -0.031743099   -0.016240037  0.38212965  0.34842395 -0.01523279  0.33497494  1.00000000  0.24535328
revol_util   -0.13445308 -0.08130215  0.19108049 -0.017229040   -0.082883782  0.04049021  0.10339128  0.28628434  0.12569414  0.24535328  1.00000000
```

Picture 14. Multicollinearity

R Code for Exploratory Data Analysis

```
#PART 4 EXPLORATORY DATA ANALYSIS
# Distribution of Interest rate
hist(raw.data$int_rate, col = "red", main = "Distribution of Intrest rate", xlab = "Inte
rest rate")
summary(raw.data$int_rate)

#Turning loan_status to factor
raw.data$loan_status <- factor(raw.data$loan_status)

#Distribution of grade scores
#Histogram of grade score colored by loan_status in percentage
plot1 <- ggplot(raw.data,aes(x=grade, y=((..count..)/sum(..count..))*100))
plot1 <- plot1 + geom_histogram(aes(fill=loan_status), color="black", stat = "count", al
pha=0.6)
plot1 <- plot1 + theme_light()
plot1 <- plot1 + scale_fill_manual("Loan Status",values = c("red", "green")) +
  labs(y="Percent", x="Loan Grades from A (best) to G (poor)")
plot1 <- plot1 + ggtitle("Distribution of Loans By Grading Scores and Loan Status")
plot1

#Making Contingency Table to check percentage of grading score in relation with unpaid l
oans
CrossTable(raw.data$grade, raw.data$loan_status,prop.r = TRUE, prop.c = FALSE, prop.t =
FALSE,
           prop.chisq = FALSE )

#Taking the highest loan purposes
arg <- raw.data$purpose == "credit_card" | raw.data$purpose == "debt_consolidation" |
        raw.data$purpose == "home_improvement" | raw.data$purpose == "major_purchase" |
raw.data$purpose == "other"
j <- subset(raw.data, arg==TRUE)

#Making distribution of loans by purpose
plot2 <- ggplot(j,aes(x=purpose, y=((..count..)/sum(..count..))*100))
plot2 <- plot2 + geom_bar(aes(fill=loan_status), position = "dodge", stat = "count")
plot2 <- plot2 + theme_bw()
plot2 <- plot2 + scale_fill_manual("Loan Status",values = c("red", "green")) +
  labs(y="Percent", x="Loan Purpose")
plot2 <- plot2 + ggtitle("Distribution of Loans By Purpose")
plot2

#Making Contingency Table to check percentage of grading score in relation with unpaid l
oans
CrossTable(raw.data$purpose, raw.data$loan_status,prop.r = TRUE, prop.c = FALSE, prop.t
= FALSE,
           prop.chisq = FALSE )

#Making scatter diagram to control relation between interest rates and loans grades
plot3 <- ggplot(raw.data, aes(x=int_rate, y=sub_grade)) + geom_point(aes(color=loan_stat
us, alpha=0.4))
plot3 <- plot3 + theme_bw() + scale_fill_manual("Loan Status", values = c("red", "green"
)) +
  labs(y="Sub Grades", x="Interest Rates")
```

```
plot3

#Deleting detected outliers
arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="G1"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="F5"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="E5"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="E4"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="E3"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="E2"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="E1"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="D5"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="D4"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="D3"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="D2"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="D1"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="C5"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="C4"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="C3"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="C2"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="C1"
raw.data <- subset(raw.data, arg==FALSE)
```

```
arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="B5"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="B4"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="B3"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="B2"
raw.data <- subset(raw.data, arg==FALSE)

arg <- raw.data$int_rate==0.06 & raw.data$sub_grade=="B1"
raw.data <- subset(raw.data, arg==FALSE)

#5-number summary statistics for annual income
summary(raw.data$annual_inc) #There are potential outliers

#Plotting scatter diagram to detect outliers
plot(raw.data$annual_inc, ylab = "Annual Income")

#Removing outliers
index.outliers <- which(raw.data$annual_inc > 1000000) #91 outliers detected
raw.data <- raw.data[-index.outliers,] #Outliers deleted

#Histogram for Annual Income
hist(raw.data$annual_inc, col="red", xlab = "Annual Income", main = "Histogram of Annual
Income")

#Removing outliers for dti
summary(raw.data$dti)
plot(raw.data$dti)

outliers_upperlimit <- quantile(raw.data$dti, 0.75) + 1.5 * IQR(raw.data$dti) # upper_li
mit = 40.8
index.outliers.dti <- which(raw.data$dti > outliers_upperlimit | raw.data$dti < 0 ) #470
outliers
raw.data <- raw.data[-index.outliers.dti,] #Removing observations

#Removing outliers for open_acc
summary(raw.data$open_acc)
plot(raw.data$open_acc)
index.outliers2 <- which(raw.data$open_acc > 50 | raw.data$open_acc <0 ) #41 outliers
raw.data <- raw.data[-index.outliers2,] #Removing observations

#Removing outliers for pub_rec
summary(raw.data$pub_rec)
plot(raw.data$pub_rec)
index.outliers3 <- which(raw.data$pub_rec > 20 | raw.data$pub_rec <0 ) #8 outliers
raw.data <- raw.data[-index.outliers3,] #Removing observations

#Removing outliers for delinq_2yrs
summary(raw.data$delinq_2yrs)
plot(raw.data$delinq_2yrs)
```

```
index.outliers4 <- which(raw.data$delinq_2yrs > 20 | raw.data$delinq_2yrs <0 ) #7 outlie
rs
raw.data <- raw.data[-index.outliers4,] #Removing observations

#No detecetd outliers for inq_last_6mths
summary(raw.data$inq_last_6mths)
plot(raw.data$inq_last_6mths)

#No detecetd outliers for installment
summary(raw.data$installment)
plot(raw.data$installment)

#Removing outliers for revol_bal
summary(raw.data$revol_bal)
plot(raw.data$revol_bal)
index.outliers5 <- which(raw.data$revol_bal > 500000 | raw.data$revol_bal <0 ) #56 outli
ers
raw.data <- raw.data[-index.outliers5,] #Removing observations

#Removing outliers for revol_util
summary(raw.data$revol_util)
plot(raw.data$revol_util)
index.outliers6 <- which(raw.data$revol_util > 2 | raw.data$revol_util <0 ) #2 outliers
raw.data <- raw.data[-index.outliers6,] #Removing outliers

#No detecetd outliers for loan_amnt
summary(raw.data$loan_amnt)
plot(raw.data$loan_amnt)

#Multicollinearity
cor(raw.data[, sapply(raw.data, class) != "factor"]) #Checking multicollinearity
```

MODEL BUILDING

After we have analyzed data, we can conclude which features we will include in the model. Therefore, our model will have the following 14 independent variables: grade, int_rate, open_acc, pub_rec, dti, revol_bal, revol_util, delinq_2yrs, inq_last_6mths, emp_cat, annual_inc, home_ownership, purpose and loan_amnt, while our dependent variable is loan_status.

Logistic regression model is part of supervised machine learning method. Therefore, before building the model we need to split our data into training and test set. First, we are going to train the classifier on the training set. Secondly, we are going to test how good our classifier is on the test set where label class is unknown. Finally, we are going to compute several important performance metrics (predictive accuracy, error, sensitivity, specificity, recall). In addition, we will plot the ROC curve to determine how really our model is good.

The best way to split our data into training and test set is by cross-validation method (exhaustive cross-validation or non-exhaustive cross-validation) that uses multiple test sets. However, cross-validation method is very time consuming process. For this reason, in this project, I have decided to use single test set method where I split dataset into training set (70%) and test set (30%) using sample.split function.

Building the logistic regression model is straight forward in R. I have used glm function to make a classifier and provided the training set to learn the classifier. Then, I used predict function and test set to get prediction which I stored in the vector. The results are shown and discussed in the next part.

R code for Model Building

```
#PART 5 MODEL BUILDING AND MODEL EVALUATION

loan.model <- subset(raw.data, select = c(1,2,4:11,13,14,17:19))
anyNA(loan.model) # No missing values
dim(loan.model) #14 features + 1 response, 552,625 observations

#Splitting data set into training and test set
set.seed(123) #making results reproduciable


sample <- sample.split(loan.model$loan_status, 0.7)
train.data <- subset(loan.model, sample==TRUE)
test.data <- subset(loan.model, sample==FALSE)

#LOGISTIC REGRESSION

logistic.regressor <- glm(loan_status ~ ., family = "binomial", data = train.data)
summary(logistic.regressor)

#Predicting outcomes on test data
prob_pred <- predict(logistic.regressor, newdata = test.data, type = "response")
summary(prob_pred)

#Cut-off value = 0.5
pred_cut_off <- ifelse(prob_pred > 0.5, 1,0) #Setting cut-off to be at 0.5
table(test.data$loan_status,pred_cut_off )
pred <- prediction(pred_cut_off,test.data$loan_status)
perf <- performance(pred, "tpr", "fpr")
#Printing AUC Value
perf1 <- performance(pred, "auc")
print(perf1@y.values[[1]])
#Plotting the ROC-curve
roc.curve(test.data$loan_status, pred_cut_off,col="red", main="The ROC-curve for Model w
ith cut-off=0.5")
text(0.6,0.2,paste("AUC=0.52"))
confusionMatrix(test.data$loan_status,pred_cut_off )

#Cut-off value = 0.8
pred_cut_off <- ifelse(prob_pred > 0.8, 1,0) #Setting cut-off to be at 0.8
table(test.data$loan_status,pred_cut_off )
pred <- prediction(pred_cut_off,test.data$loan_status)
perf <- performance(pred, "tpr", "fpr")

#Printing AUC Value
perf1 <- performance(pred, "auc")
print(perf1@y.values[[1]])
#Plotting the ROC-curve
roc.curve(test.data$loan_status, pred_cut_off,col="red", main="The ROC-curve for Model w
ith cut-off=0.8")
text(0.6,0.2,paste("AUC=0.65"))
confusionMatrix(test.data$loan_status,pred_cut_off )

#Plotting proportion of fully paid vs charged off loans
options(scipen=20)
```

```
barchart(train.data$loan_status, main="Proportion of Fully Paid and Charged Off Loans (T
raining Set)", xlab="Number of Loans")

#Assuming investor wants to finance top 20% of new loans in his portfolio
cutoff <- quantile(prob_pred, 0.8)
pred_cut_20 <- ifelse(prob_pred > cutoff, 1,0)
true.value <- as.character(test.data$loan_status)
true.value <- as.integer(true.value)
true_and_pred <- cbind(true.value, pred_cut_20)

accepted_loans <- true_and_pred[pred_cut_20==1,1]
bad_rate <- (sum(accepted_loans==0) / length(accepted_loans))*100 #6.69% of bad loans in
his portfolio

#Building Strategy Table
accept_rate <- sort(seq(0,0.99,by=0.05), decreasing = TRUE)
cutoff <- c()
bad_rate <- c()
for(i in 1:length(accept_rate)) {
  cutoff[i] <- quantile(prob_pred, accept_rate[i])
  pred_cut <- ifelse(prob_pred > cutoff[i], 1,0)
  true.value <- as.character(test.data$loan_status)
  true.value <- as.integer(true.value)
  true_and_pred <- cbind(true.value, pred_cut)
  accepted_loans <- true_and_pred[pred_cut==1,1]
  bad_rate[i] <- (sum(accepted_loans==0) / length(accepted_loans))
}

#Making Strategy Table
strategy <- cbind(1 - accept_rate, cutoff, bad_rate)
colnames(strategy) <- c("Accept Rate","Cut-off Value", "Bad Rate")
strategy <- as.data.frame(strategy)


#Plotting Strategy Curve
curve <- as.matrix(strategy[-2])
curve[,2] <- curve[,2]
plot(curve, type="l",col="dark red", lwd=3, main="Strategy Curve")

#IMPROVING MODEL BY BALANCED DATA
#Making balanced data using SDG method
balanced.data <- ROSE(loan_status ~ ., data = train.data, seed = 1)$data
table(balanced.data$loan_status) #Now we have almost 50% 50%

#Building new logistic regression model
rose.regressor <- glm(loan_status ~ ., family = "binomial", data = balanced.data)
summary(rose.regressor)

#Making predictions on test set
prob_pred_rose <- predict(rose.regressor, newdata = test.data, type="response")
hist(prob_pred_rose)

#Evaluating new model
roc.curve(test.data$loan_status, prob_pred_rose, col="dark red", main="The ROC-curve for
Improved Model")
```

```
text(0.6,0.2,paste("AUC=0.704"))
#END OF PART 5
```

RESULTS

Confusion matrix for logistic model with cut-off value 0.5

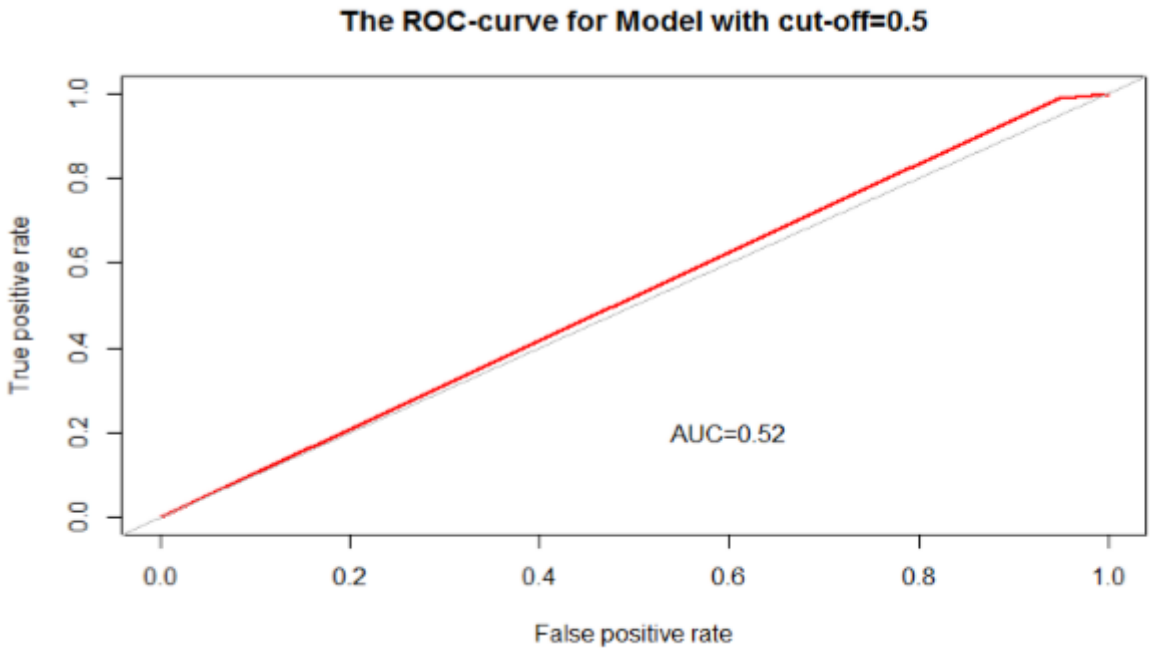
CUT-OFF = 0.5	AUC = 0.52	PREDICTED CLASS		
		0	1	
ACTUAL CLASS	0	TP = 1721	FN = 31732	33453
	1	FP = 1365	TN = 130970	132335
		3086	162702	165788
MEASURES DERIVED FROM CONFUSION MATRIX				
ACCURACY	(TP+TN) / (TP+FN+FP+TN)	80%		
SENSITIVITY (PRECISION)	TP / (TP + FP)	55.7%		
TRUE P RATE (RECALL)	TP / (TP + FN)	5.14%		
TRUE N RATE	TN / (FP + TN)	98.96%		
FP RATE	FP / (FP + TN)	1.03%		
ERROR	1 - ACCURACY	20%		

Table 6. Confusion Matrix for Logistic Model with Cut-off = 0.5

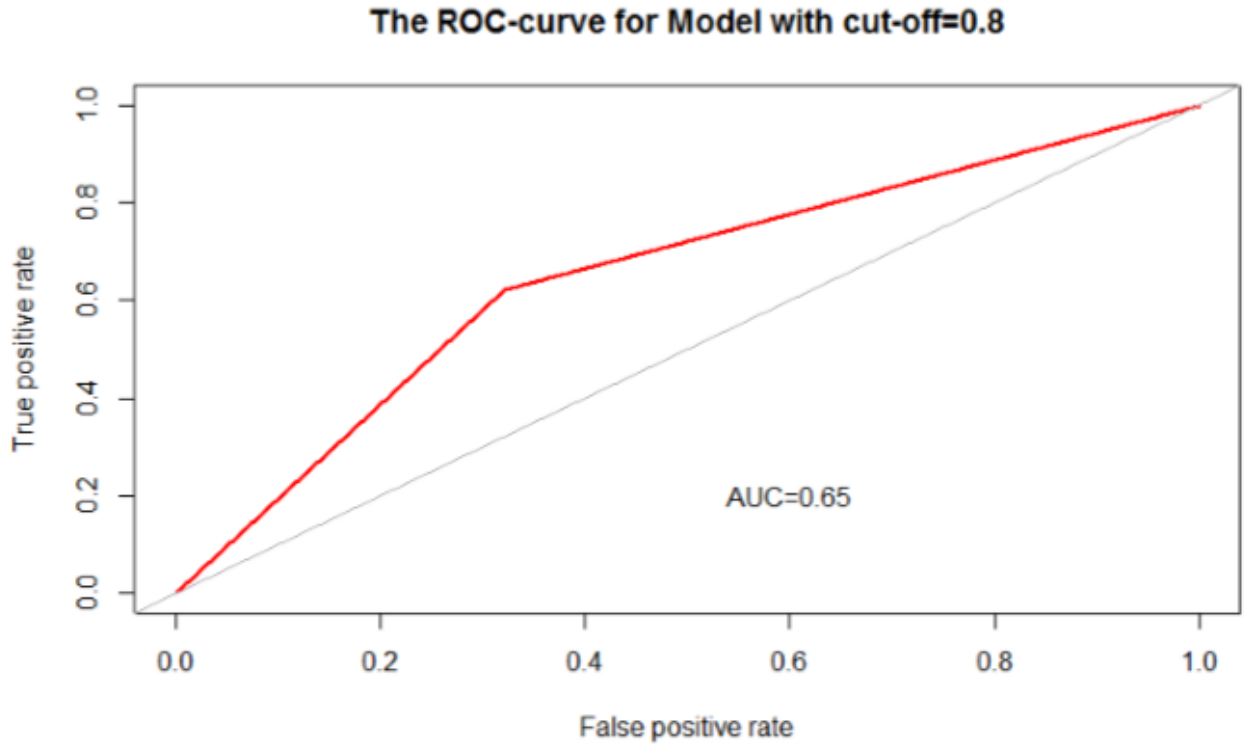
Confusion matrix for logistic model with cut-off value 0.8

CUT-OFF = 0.8	AUC = 0.65	PREDICTED CLASS		
		0	1	
ACTUAL CLASS	0	TP = 22695	FN = 10758	33453
	1	FP = 50233	TN = 82102	132335
		72928	92860	165788
MEASURES DERIVED FROM CONFUSION MATRIX				
ACCURACY	(TP+TN) / (TP+FN+FP+TN)	63.2%		
SENSITIVITY (PRECISION)	TP / (TP + FP)	31.1%		
TRUE P RATE (RECALL)	TP / (TP + FN)	67.84%		
TRUE N RATE	TN / (FP + TN)	62.04%		
FP RATE (DETECTION RATE)	FP / (FP + TN)	37.95%		
ERROR	1 - ACCURACY	36.8%		

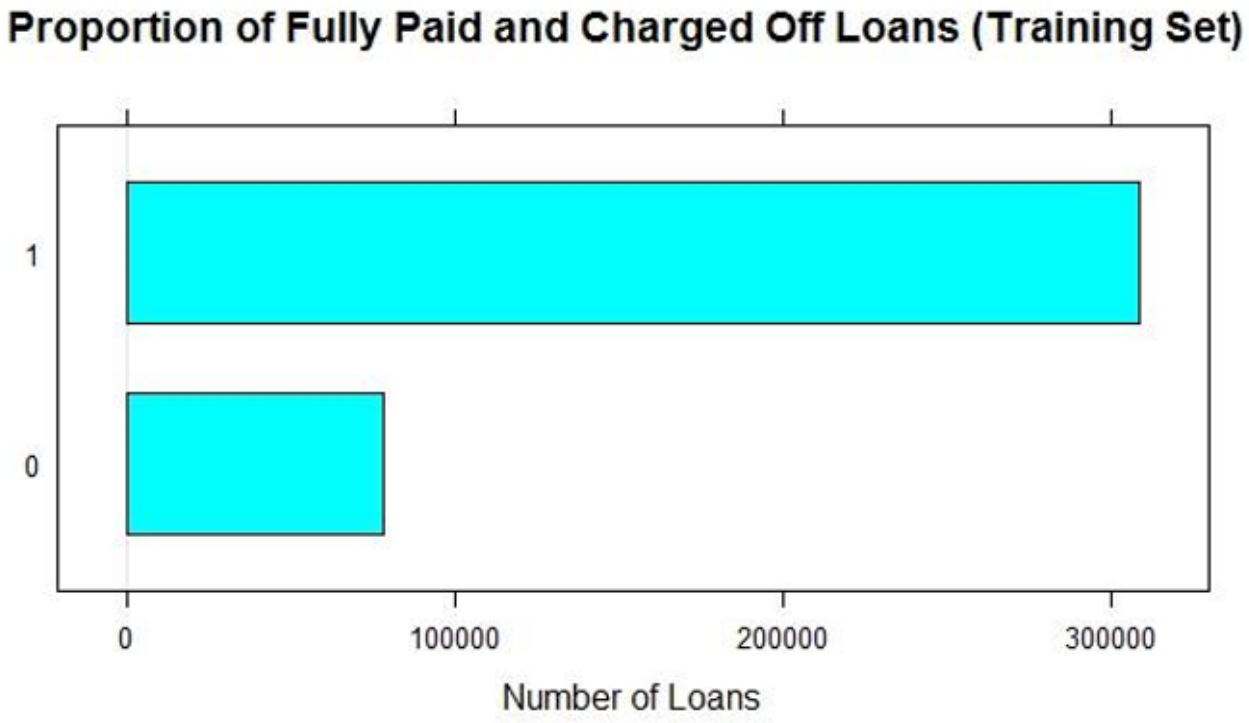
Table 7. Confusion Matrix for Logistic Model with Cut-off = 0.8



Picture 15. The ROC-curve for Logistic Model with Cut-off value 0.5



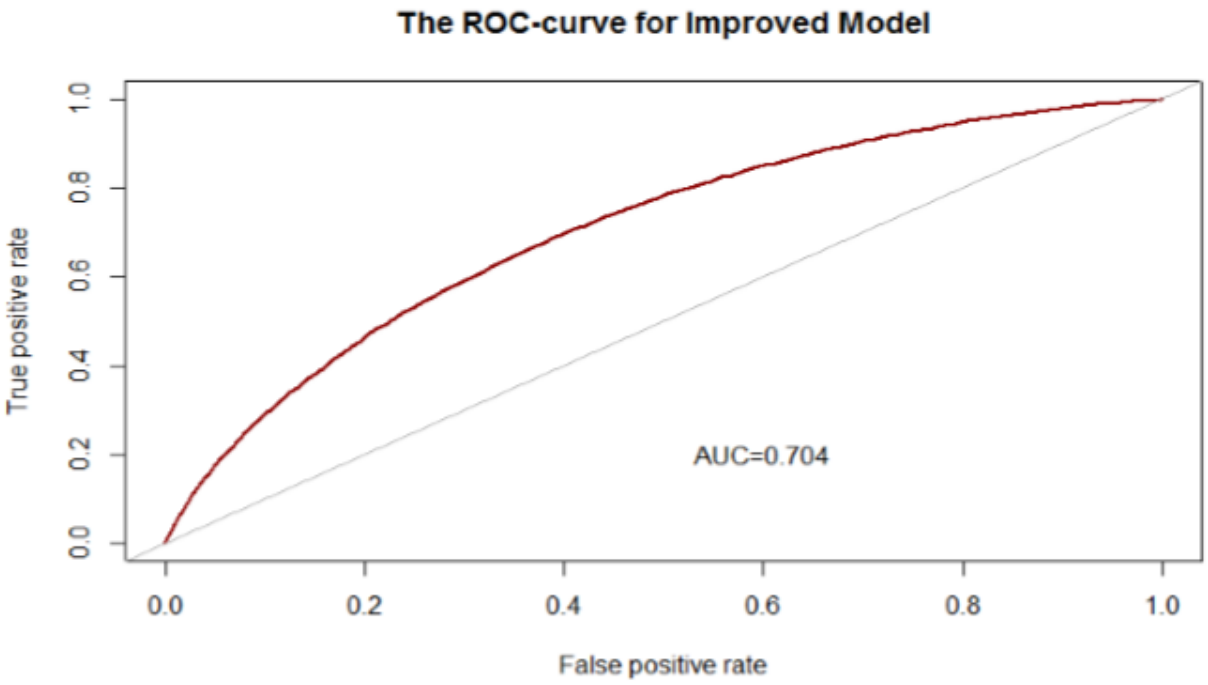
Picture 16. The ROC-curve for Logistic Model with Cut-off Value 0.8



Picture 17. Imbalanced Dataset

```
> summary(prob_pred)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2214  0.7243  0.8202  0.7978  0.8926  0.9999
```

Figure 1. 5-number Summary Statistics of Predicted Values with cut-off Value 0.5

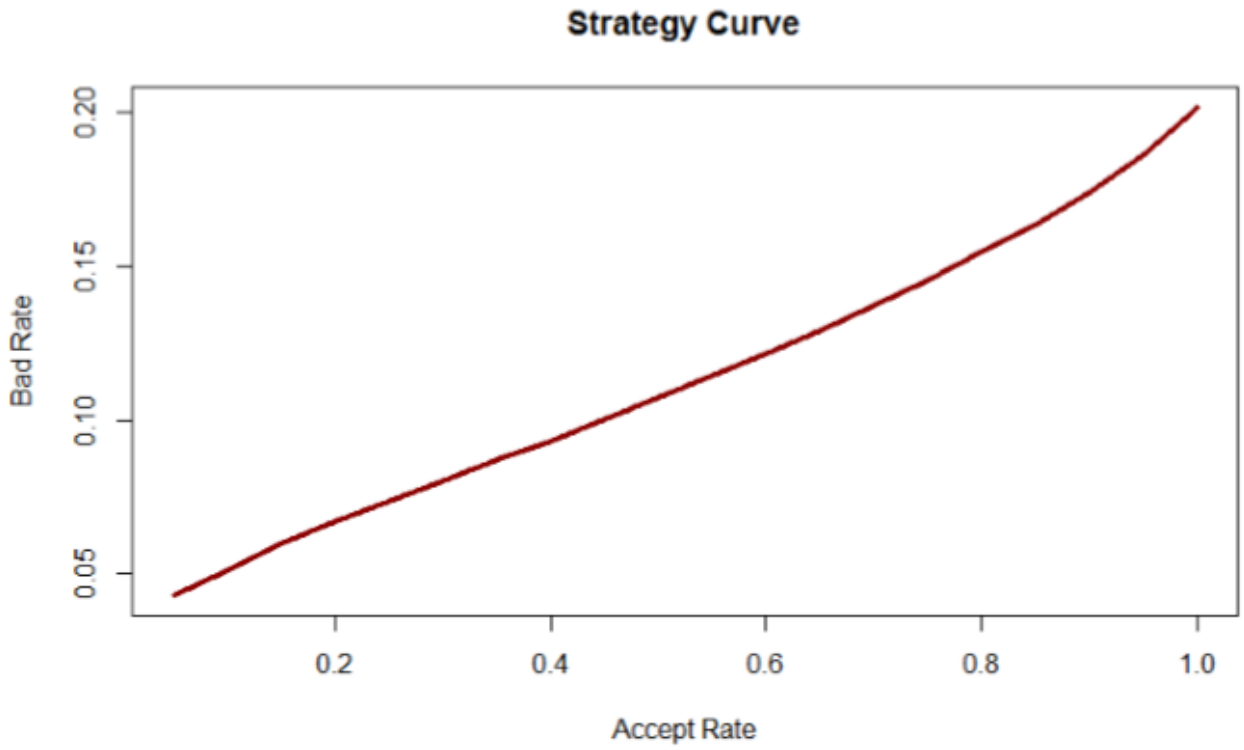


Picture 18. The ROC-curve for Improved Model with Balanced Data

Investment Strategy Table

	Accept Rate	Cut-off Value	Bad Rate
1	0.05	0.9493619	0.04306393
2	0.10	0.9357153	0.05126968
3	0.15	0.9213234	0.05991395
4	0.20	0.9064575	0.06692201
5	0.25	0.8925556	0.07373272
6	0.30	0.8792734	0.08032250
7	0.35	0.8659465	0.08721952
8	0.40	0.8516701	0.09341778
9	0.45	0.8366196	0.10039542
10	0.50	0.8202318	0.10743842
11	0.55	0.8034194	0.11437439
12	0.60	0.7856397	0.12140983
13	0.65	0.7672702	0.12930347
14	0.70	0.7467973	0.13718107
15	0.75	0.7242773	0.14527791
16	0.80	0.6979522	0.15468597
17	0.85	0.6666796	0.16347689
18	0.90	0.6255659	0.17394393
19	0.95	0.5661179	0.18633887
20	1.00	0.2213981	0.20177698

Picture 19. Investment Strategy Table for Investors



Picture 21. Strategy Curve

DISCUSSION

The two most important measures to evaluate models are the accuracy and error. The accuracy for logistic model with cut-off 0.5 is 80% which would lead us to misleading conclusion that this model is very good. However, this is not true. If we take a look the true P rate it is only 5.14%. This means that our model, for loans which are not fully paid, predicted correctly only in 5.14% cases. From the confusion matrix we can see that the model classified 1,721 loans as Charged Off, while in the test set there were in total 33,453 charged off loans. Therefore, although the accuracy is very good (80%) with perfect prediction of loans that will be fully paid (98.96%), this model would be very dangerous to use by investors. The main reason why investors would like to use this model is to avoid investments in loans that would not be fully repaid. For this reason, the ROC curve will give the real picture about the quality of our model. From Picture 15 we can see that the area under the curve is equal to 0.52. Therefore, it is clear from this picture that this model is very poor as it is not able to classified correctly loans that are charged off. The main reason for this poor model performance is because the dataset is imbalanced. Our model learned from the training set. If we take a look Picture 17, the proportion of fully paid and charged off loans, we can see that only 20% of all loans are labeled as charged off in the training set. Therefore, charged off loan is very rare event in our training data set. Moreover, from summary statistics of predicted vector in Figure 1, we can see that the minimum predicted value is 0.2214, while 1st quartile is 0.7243 (only 25% of data is less than 0.7243). Therefore, we can see that setting cut-off value to 0.5 would be very poor decision.

When we change the cut-off value to 0.8, we get different results. From Table 7 we can see that the overall model accuracy decreased by 16.8% compared to previous one. However, the true P rate improved significantly since it increased by 62.7% (67.84% compared to 5.14%). The true N rate dropped to 62.04% (compared to previous 98.96%). However, if we look at the ROC-curve in Picture 16, we can see that the area under the curve increased to 0.65. Although it is still far from very good performance, this model is better for use by investors since it is able to classify much better potential bad loans which is main concern for investors.

The best solution for investors would be to choose the cut-off value depending on which investment strategy they want to play. Deciding which cut-off to choose is very important because based on that value it will be determined the ratio between true P rate (bad loans classified as good loans) and true N rate (good loans classified as bad loans). If we look at the ROC-curve in the picture 16, investors who like risky strategies would be on the ROCcurve close to lower left corner because true P rate is very low, while conservative investors would be on the curve close to upper right corner where true N rate is very low. Therefore, this model will allow them to specify how many loans they should have in their investment portfolio if they do not want to exceed a certain percentage of bad loans in their portfolio. For example, if investor wants to finance the best 20% of new loans (test set), then he would need to take cut-off value which is in this case equal to 80% quintile in a probability distribution, i.e. 0.9064575. Based on this cut-off value we can compare the real loan status from the test set with the predicted values. Where predicted value is equal to 1 (investor accepted to invest in loan), we will store in the new vector accepted_loans the true class from test set (can be 0 or 1). Now, we can compute the bad rate which is ratio of accepted loans which are not fully paid with total number of accepted loans. In this scenario, if the test set was new loans where investor can invest, and if he decided on cut-off value to be 0.9064575, then out of 165,788 new loans, he would accept to finance 33,158 loans (20% of all loans), and he would have 2,219 bad loans in his portfolio. In other words, his portfolio would be composed of 6.69% bad loans, and 93.31% good loans. This we can compute for different accept rate. The results are presented in Picture 19. From this strategy table investors now can easily determine the percentage of loans they want to finance and cut-off value based on their investment strategy. For example, the risky investor may decide to have maximum 16% of bad loans in his portfolio. Based on strategy table in Picture 19, he would accept to finance 80% of all new loans. Therefore, the correct cut-off value would be 0.6979522. On the other hand, a conservative investor may decide to have maximum 5% of bad loans in his portfolio. From strategy table we can see that he would accept only 10% of new loans to finance. The correct cut-off value would be 0.9357153.

Picture 21 represents data visualization of strategy table which plots bad rate against accept rate.

So far in our discussion we were focused on how to improve our model only by changing cut-off value which would lead to different investment strategies. The next question is if we are possible to improve our model once we know the cause of poor performance?

Imbalanced data are very common in the real life situations. It refers to problem where one class outnumber other class. In this case, we have 79.8% observations labeled as Fully Paid, while 20.2% labeled as Charged Off. There are several methods to deal with imbalanced data. In this project, I have used Synthetic Data Generation algorithm using ROSE package in R. As we can see from Picture 18, the overall model performance improved since the AUC value increased to 0.704.

Several different authors have tried to build this kind of model for Lending Club using different machine learning algorithms. Wu in his paper has used dataset from 2007 – 2011, and build the model with 20 features using random forest and logistic regression. According to his results, after model tuning, logistic regression performed better with AUC value 0.7321, while random forest had AUC value of 0.7156 (Wu, 2011). The reason why he got slightly better results might be in the fact that he used data only for 5 years, while I used data for 10 years. On the other hand, O'Rourke has built a model based on 18 features using decision tree algorithm. Before model improvement, the AUC value was 0.622, while after balancing data the AUC value increased to 0.681 (O'Rourke, 2016). Several others authors have also compared different models using SVM (AUC=0.698), random forest (AUC = 0.705), logistic regression (AUC = 0.704), extreme gradient boosting (AUC = 0.713). The best results were achieved using extreme gradient boosting. However, compared complexity of this algorithm with simplicity of logistic regression, and comparing model performance of these two algorithms, it is clear that best model would be logistic regression.

Finally, from Picture 20 we can see that the most important feature is grade. This is not a surprise because this feature reflects behavior of borrower and it is classified by their credit history. Moreover, we can see that all behavioral features such as debt-to-income ratio, the number of inquiries by creditors during the past 6 months are important features in our model. As expected, the interest rate is also important feature in the model as higher rates are related to more risky loans. The purpose of loan is also very important. We saw that loans for small business purpose had the highest rate among all loans that were not fully paid.

CONCLUSIONS

In this project, I have developed a model using logistic regression to predict if a borrower will repay the loan based on historical data provided by Lending Club and to help investors when deciding which investment strategy to choose. After model building, testing, and data analysis, we can make the following conclusions:

1. It is clear that these models are not perfect, and that they showed to have fairly poor performance. The selection of cut-off value (discrimination threshold) is very important because if investors are going to use this model to decide on which loan to invest or not, the choice of cut-off value will determine which applicants will get a loan and which not. However, it is good that investors can decide the percentage of bad rate that they are willing to accept in their portfolio, and based on that they can decide very easy the percentage of the new loans that they want to finance.
2. The classification accuracy is very problematic measure for imbalanced classification which is very common scenario in investment and banking sector, and can lead us to make wrong decisions. For these kind of models, the better measure is the ROC-curve which shows true positive rates against false positive rates.
3. Loan risk models show excellent performance in predicting true negative instances, but very poor performance in predicting true positive instances due to imbalanced classification. We can improve model performance by applying several different methods such as under sampling, over sampling, synthetic data generation, and cost sensitive learning.
4. There are other methods that can be used for classification which include discriminant analysis, support vector machines, random forest and neural networks. In addition, new very popular method is survival analysis where probabilities of default change over time, and features that change over time can be included. This new method would be interesting to conduct in future projects.

REFERENCES

[1] Howard J. Seltman, Experimental Design and Analysis (2015).

[2] Tibshirani Robert. An Introduction to Statistical Learning with Applications in R (2015). Springer.

[3] Lending Club. https://www.lendingclub.com/info/download-data.action (https://www.lendingclub.com/info/download-data.action) LoanData 2007-2017. May 2017.

[4] Jiayu Wu. Loan Default Prediction Using Lending Club Data (2014).

[5] Hasan Felyeh. Data Mining Lecture Slides (2017). Dalarna University.

[6] Practical Guide to deal with Imbalanced Classification Problems in R. https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalancedclassification-problems/ (https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalancedclassification-problems/) . May 2017.

[7] Predict LendingClub's Loan Data. https://rstudio-pubsstatic.s3.amazonaws.com/203258_d20c1a34bc094151a0a1e4f4180c5f6f.html (https://rstudio-pubsstatic.s3.amazonaws.com/203258_d20c1a34bc094151a0a1e4f4180c5f6f.html) May 2017.

[8] Lending Club - Predicting Loan Outcomes. https://rpubs.com/torourke97/190551 (https://rpubs.com/torourke97/190551) May 2017.

[9] Lending Club Review for New Investors. http://www.lendacademy.com/lending-clubreview/ (http://www.lendacademy.com/lending-clubreview/) May 2017.

[10] Cleaning Data: An Example Using Data from Lending Club. http://financerecipes.com/cleaning_data.html (http://financerecipes.com/cleaning_data.html) May 2017.

[11] Introduction to Machine Learning. https://www.datacamp.com/courses/introductionto-machine-learning-with-r (https://www.datacamp.com/courses/introductionto-machine-learning-with-r) . Data Camp. May 2017.

[12] Cleaning Data in R. https://www.datacamp.com/courses/cleaning-data-in-r (https://www.datacamp.com/courses/cleaning-data-in-r) . Data Camp. May 2017

[13] Importing Data in R. https://www.datacamp.com/courses/importing-data-in-r-part-1 (https://www.datacamp.com/courses/importing-data-in-r-part-1) . Data Camp. May 2017.

[14] Machine Learning Toolbox. https://www.datacamp.com/courses/machine-learningtoolbox (https://www.datacamp.com/courses/machine-learningtoolbox) . Data Camp. May 2017.

[15] Exploratory Data Analysis. https://www.datacamp.com/courses/exploratory-dataanalysis (https://www.datacamp.com/courses/exploratory-dataanalysis) . May 2017.

[16] Credit Risk Modelling in R. https://www.datacamp.com/courses/introduction-tocredit-risk-modeling-in-r (https://www.datacamp.com/courses/introduction-tocredit-risk-modeling-in-r) . May 2017.