LendingClub

Machine Learning Algorithms of Lending Risk

Lin Jia
Shiqi Li
Spencer You
Mo Zou

Executive summary

Lending Club is the largest online credit marketplace that facilitates personal and business loans. However, Lending Club is having difficulties attracting new investors due to a number of reasons: increasing loan delinquency rate, lack of interest alignment and choice overloaded interface.

Our Project is aiming at predicting the risk of different loans, maximizing profits for both investors and lending club, and developing interfaces to filter loans with prediction. In order to achieve these goals, three steps of analysis are executed and consist of the whole project: risk measurement, profit calculation and application demonstration.

In the risk measurement part, our group explored the data set and found different situations of messy data. Data preprocessing was completed by data cleaning and feature transformation. SMOTE method was used to solve the imbalance problem in the dataset. Later, two metrics (accuracy rate and AUC method) were chosen to evaluate models' performance. Several machine learning algorithms - Logistic Regression, Support Vector Machine, Random Forest and Gradient Boosting were adopted to predict the loan status. Eventually, Gradient Boosting was chosen as the model based on its highest AUC and Accuracy Rate. Based on the threshold value of 0.5, the model predicted the top 10 significant features with the top 3 being number of mortgage accounts, interest rate and inquiry of credit cards in the last six months.

In the profit calculation part, the objective of financial analysis was to calculate the best cut-off value of the model and compare the total profits before and after the prediction model. The optimal cut-off value was 0.7 and model prediction can reach an average profit of $327.

In the application demo section, Shiny interface and Tableau dashboard were able to provide selected loan information in both micro and macro perspective.

To sum up, we managed to achieve the original purpose of our project, which is to use some machine learning methods to predict and reduce lending risk, via this project. We also improved adjusted our model based on accuracy metrics as well as business criteria, so that the interests of different parties could be aligned in the business model. More customers could be attracted to Lending Club based on our reasonable recommendation system and interactive visualization system.

1. Background

1.1 Introduction of Lending Club

Lending Club[1] is an online platform that facilitates personal and business loans. The underlying business concept is peer-to-peer lending, which is the practice of lending money to individuals through online platforms that match lenders with borrowers. Due to the lower overhead cost compared to traditional financial institutions, Lending Club is able to pass along the savings to borrowers with a lower interest rate and to investors with a favorable return.

1.2 Problems of Lending Club

However, Lending Club is faced with increasing difficulties due to following reasons:

- There is a **trend of increasing delinquency rate on its loans**. The charge-off rate increased from 4.58% to 6.31% , an increase of 38% for its lower-graded loans from 2013 to the first quarter of 2015[2]. This is mainly because Lending Club overemphasizes the significance of FICO score in its proprietary loan-selection algorithm, while overlooks other features, and thus inaccurately predicts some borrower's ability to repay the loan.
- Since Lending Club is not responsible for the bad debt incurred on the platform, investors would bear the cost if a borrower stopped his or her payment for the loan. This results in **a lack of alignment of interest** due to the unilateral contracts between lenders (investors) and loan originators (Lending Club).
- **The loan selection experience on Lending Club is not user-friendly** due to the complicated filters for loan selection. There are over 50 filters for investors to choose from and they are not intuitive. Investors often experience choice-overload when they try to determine which filters to use to pick their preferred loans.

1.3 Project Objective

Like any other financial investments, investors may lose their money due to the risk of default since the loans are unsecured. The objectives of our project are threefold that correspond to the problems Lending Club is facing:

- Incorporate more variables and use machine learning algorithms to predict the risk of loan status more accurately.
- Based on the more accurate loan-selection algorithms, conduct financial analysis to fine-tune the model to maximize an investor's return.
- Build a more user-friendly interface for Lending Club to facilitate the loan selection process for investors.

2. Data Exploration and Preprocessing

2.1 Data Exploration

The database chosen for this project was downloaded from the Lending Club website[3]. Only loans that were issued between 2012 to 2013 are included in our analysis. This is due to the fact that the term of a loan on Lending Club is either 36 months or 60 months. By constraining to this particular time frame, we ensured that the outcome of each loan is available at the time of the analysis and this facilitates us to determine the accuracy of our algorithm.

The database contains over 100,000 records(rows) and 98 features(columns). The 98 independent variables could be divided into three buckets: borrowers' information (such as location, home type, employment), financial metrics (such as FICO score, Debt-to-Equity ratio, bankruptcy history), as well as loan characteristics (such as loan amount, term, interest rate, grade).

Through data exploration, we have found that most of the loan amount on Lending Club is from $5000 to $25000 (Figure 1). Lending Club determines the loan interest rate based on its own grading system. Interest rate increases as loan grade decreases due to the increased risk associated with the loan (Table 1). It is worth noting that the majority of the borrowers on Lending Club have a grading of C or above (Figure 2), this suggests that the majority of the loans on Lending Club are from prime consumer borrowers. This gives investor confidence to invest.

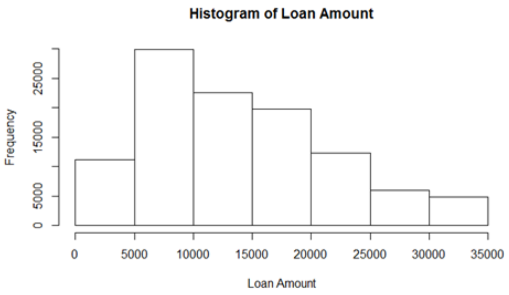


Figure 1 Histogram of Loan Amount

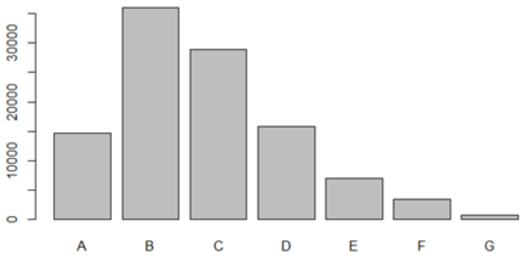


Figure 2 Lending Club Grading System

Table 1 Correlation of Grade and Interest Rate of Lending Club

Grade	A	B	C	D	E	F	G
Interest Rate	5.32%~ 7.97%	9.44%~ 11.99%	12.62%~ 16.02%	17.09%~ 21.45%	22.91%~ 26.30%	28.72%~ 30.75%	30.79%~ 30.99%

The loans on Lending Club are not evenly distributed geographically, concentration can be observed in few states such as California, Florida and New York.

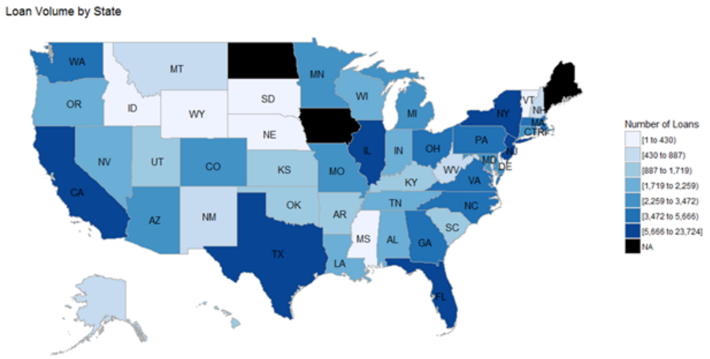


Figure 3 Distribution of Number of Loans geographically

2.2 Data Preprocessing

2.2.1 Feature Transformation

The basic idea for the project is to use multiple independent variables (x's) to predict the loan status (y). Our original dataset is messy containing many features that need to clean and transform.

To process the missing values, we first viewed their distribution and frequency. For those features with few missing records, we delete these rows without hurting the majority of information. For other features, we either conducted mean imputation or deleted the columns.

Secondly, we managed to deal with the problem of multiple classes. For example, there are 7 different situations within the loan status values, including charge off, default, late, in process of paying, etc. It would be hard to build a clear and accurate model if we trained a classification model with 7 outcomes. To simplify the model, we transferred it into a binary value, with 0 representing all bad scenarios and 1 representing good ones.

Thirdly we also had features with repetitive meanings, which we transformed into one single feature with business insights. For example, there were 2 different features regarding with FICO score, FICO range high and FICO range low, and we averaged them to get one FICO score number. This would remove the redundant and repetitive columns.

2.2.2 Solution to imbalance problem

Data imbalance was a huge problem for our project. In this dataset, it contains 5 times number of y=1 compared with y=0, which will generate bias analysis of models' performance. Typically, there are two ways to deal with the imbalance problem--undersampling and oversampling, and both of them are to generate a new dataset with equivalent number of y. After some trials, we adopted SMOTE method (Synthetic Minority

Over-sampling Technique), which is basically a combination of these two methods. SMOTE will generate new records based on existing data points in the minority class, and randomly selects points from the majority. After SMOTE method, data preprocessing was completed with a well clean dataset for further analysis.

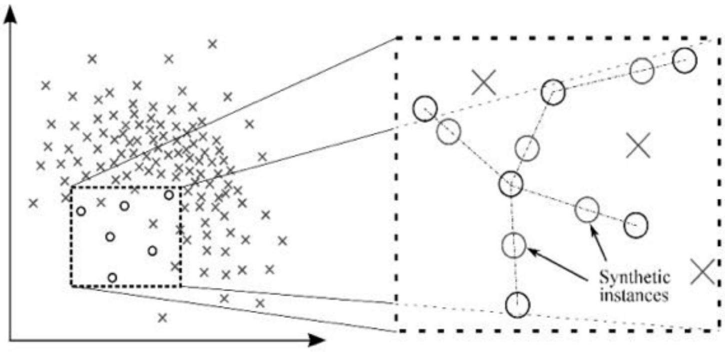


Figure 3 Introduction of SMOTE method to solve imbalance problem

2.2.3 Metrics

There are two criteria to evaluate our model's performance, which makes our model selection process logically rigorous and thorough.

The first metric is accuracy rate which means the correct predictions over total predictions. This method is the most straightforward one, and provides reasonable idea of business insights. We chose AUC method as another index. AUC stands for Area Under ROC Curve, and ROC shows the relationship between true and false positive rate, with different cut-off values. The reason we also used AUC is that it contains more information, and importantly, insensitive to data imbalance. With the dataset and metrics in place, we furtherly trained various machine learning algorithms for our project.

3. Model Building

In order to select the most appropriate model to predict the loan status, we have tried several machine learning algorithms - Logistic Regression, Support Vector Machine, Random Forest and Gradient Boosting. Briefly speaking, Logistic Regression is a classification algorithm used to estimate the probability of a binary response based on the predictors. Support Vector Machine attempts to find a hyperplane that divides two classes with the largest margin. Random Forest takes the average of different trees trained by taking bootstrap sample, while Gradient Boosting try to correct its errors iteratively by learning to predict the residual.

Table 2 AUC and Accuracy for Different Algorithms

	LR	SVM	RF	GB
AUC	0.68	0.62	0.61	0.68
Accuracy	0.62	0.63	0.7	0.75

Gradient Boosting was chosen as our model based on its highest AUC and Accuracy Rate.

Based on the threshold value of 0.5, the algorithm calculated the top 10 significant features with the top 3 features of number of mortgage accounts, interest rate and inquiry of credit cards in the last six months. However, this threshold value is not optimal considering the profit for total investor groups. Financial Analysis needs to be conducted.

Figure 4 Feature Contributions

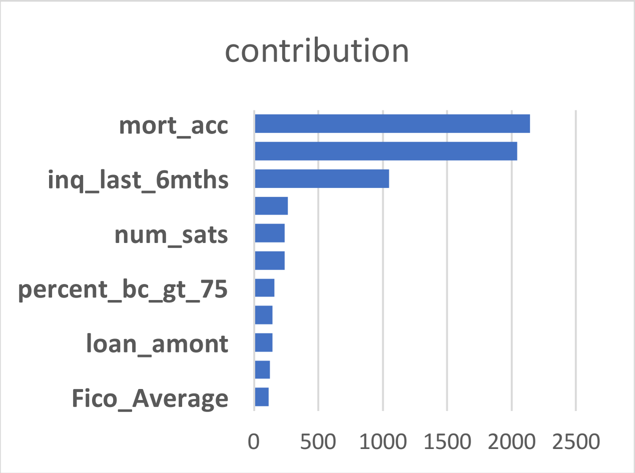


Table 3 Confusion Matrix(cut-off=0.5)

	Actual 0	Actual 1
Predict 0	3070	8297
Predict 1	4637	35280

4. Financial Analysis

4.1 Cost-Benefit Analysis

The objective of the financial analysis is to calculate the best cut-off value of the model to maximize investors' profits. The profits before and after the adjustment of the cut-off value are also compared. Three assumptions are made before our analysis. **A.** We assume the worst case scenario --- all loans that are late are unrecoverable and thus our model is highly conservative. **B.** Before our project, investors will invest all the loans posted on Lending Club while after our project, investor groups will invest the loans predicted to be a good loan. **C.** Therefore, before our project, the total profits for investors are (FN+TP)* Interest Rate - (TN+FP). In other words, investors would lose money due to the delinquent loans (TN+FP). While after our project, investors would only invest in the loans predicted to be paid in the future. Therefore, the total profit is TP*Interest Rate - FP.

4.2 Cut-off Selection

With the objective to maximize investors' total profit, the optimal cut-off value was selected for the model. After running a loop function, we obtained the optimal cut-off value of 0.7. Intuitively, our model tried to reduce the FP rate since it is very costly if the borrowers end up not paying the loan later.

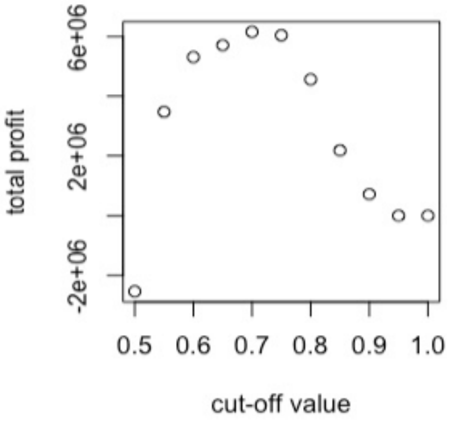


Figure 5 Cost-Benefit Analysis to Select Optimal Cut-off

The original profit per investors was $ -593. The reason this number being negative is that our model is highly conservative and we categorized 20% of loans to be default whereas in reality, the actual default rate is around 6% for lower-grade loans and 1.5% for high-grade loans on Lending Club[4]. After our prediction, the average profit reached $ 327.

Our prediction model not only maximize the profits for investors, but also help Lending Club to acquire more investors and increase its commission fees.

5. Data Visualization

5.1 Scenario Introduction

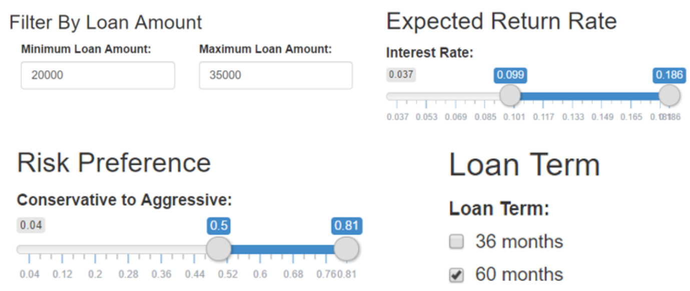
The build up model has a high accuracy in selecting the investment choice and predicting the expected return. Moving forward, our project tried to use the model to select the investment choice that meet investors expectation. The team chose two different types of investors to test our model, conservative Shiqi and aggressive Professor Parker. Shiqi was interested in short term investment (36 months), and she only had $5,000 dollars to invest as a new graduate. She has a low risk tolerance and only expect to receive a positive return higher that what banks can offer. As for Professor Parker, he is more interested in long term investment (60 months). He had a decent amount of savings and decided to invest $30,000 dollars. Additionally, Professor Parker expected a high return on the investment and can take on high riskiness.

Table 4 Scenario Introduction

Shiqi	Professor Parker
Short term investment (36 months)	Long term investment (60 months)
$5,000	$30,000
Low Risk Tolerance	High Risk Tolerance
Positive Return	High Return

5.2 Shiny Interface

The team designed a Shiny interface that would be able to select different requirements for investors like Shiqi and Professor Parker. Shiny interface had four criteria for the investors to choose, including Filter by Loan Amount, Expected Return Rate, Risk Preference and Filter By Loan Term. Filter By Loan Amount is the selection that allows people to choose the expected loan amount they want to invest in. In Professor Parker's situation, Loan Amount was chosen to be $20,000 to $35,000. Expected Return Rate is the predicted return on invest. As Professor Parker wanted high return, the return rate is set as 0.099 and above. Risk Preference means the riskiness in investment, and it varies from conservative to aggressive. As Professor Parker has high risk tolerance, the riskiness is set as 0.5 and above. Filter by Loan Term helped investor to distinguish the two different Loan Term, 36 months and 60 months. In Professor Parker's situation, the Loan Term is 60 months.



Filter By Loan Amount

Minimum Loan Amount: 20000

Maximum Loan Amount: 35000

Expected Return Rate

Interest Rate: 0.037 0.099 0.186

Risk Preference

Conservative to Aggressive: 0.04 0.5 0.81

Loan Term

Loan Term:

□ 36 months

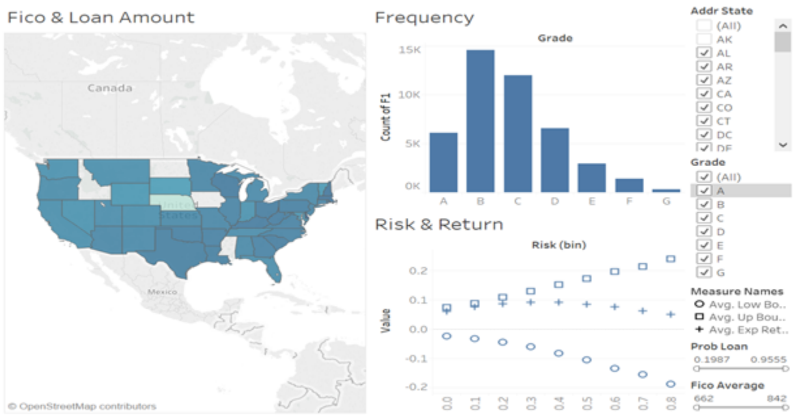
☑ 60 months

Figure 6 Filters on Our Shiny APP

Once all the criteria were entered into the Shiny Interface, the algorithm run through the whole data set, and showed the loans that met the criterions. As shown in the graph below, 672 loans were recommended for Professor Parker. Most of the grade are in the range from E to G, and interest rate are more than 0.20. Also, there were other information about the loans shown on the interface, like up_boundary, low_boundary, emp_length, annual_inc and etc which provided more information for making investment decisions.

5.3 Tableau Dashboard

Shiny interface can show loan information on a micro scale; however, in term of looking at the data in a macro perspective, like FICO score & Average Loan Amount, Number of borrowers in every grade and relationship between risk and return, Tableau can do exactly all of these. On the right side of the screenshot shown below, there were filters for choosing Address State, Grade, Probability of Loan Payment and FICO Average Score. Once all the filters were completed, the map on the left side can show average FICO Score and Average Loan Amount for the selected states. On the top right of the screenshot, it showed the distribution of borrower's amount in different grades. On the bottom right of the screenshot, the graph demonstrated the relationship between risk and return. The upper ones were the average up boundary for expected return under different risk, the lower ones were the average lower boundary for expected return under different risk, and the ones in the middle were the average expected return for investment under different risk.



Fico & Loan Amount

Frequency

Risk & Return

Addr State

□ (All)
□ AK
☑ AL
☑ AR
☑ AZ
☑ CA
☑ CO
☑ CT
☑ DC
☑ DE

Grade

☑ (All)
☑ A
☑ B
☑ C
☑ D
☑ E
☑ F
☑ G

Measure Names

○ Avg. Low Bo..
□ Avg. Up Bou..
+ Avg. Exp Ret..

Prob Loan 0.1987 0.9555

Fico Average 662 842

Figure 7 Data Analysis on Tableau

6. Conclusion & Next Steps

At the beginning of the report, the team sums up the problems that Lending is facing: **increasing delinquency rate** on the loans it has issued, **lack of interest alignment** between the platform and investors and **complicated user-interface**. Our team has tackled and come up with a solution to each of the problems: the **gradient boosting algorithm** achieves very high accuracy and AUC and was selected as the best algorithm to predict the risk of a loan. Through a financial analysis, the optimal cut-off value of the algorithm was set to be 0.7 because it maximizes investor's total return. This algorithm is **beneficial to both investors and the platform** because it allows Lending Club to predict the delinquency rate more accurately and boost investor's' return at the same time. Finally, our team has developed a Shiny app and Tableau interface to facilitate the visualization and the loan selection process for investors.

In terms of future steps, our team is considering to further build upon the solutions that we already have. For instance, adding more advanced machine learning techniques such as regulation to further decrease the overfitting that is present in our model is a good idea. A new business model that involves a contract between Lending Club and its investors is also a possibility: Lending Club guarantees an increase in return for investors and in turn charges a higher service fee. In this way, both parties are better off. Finally, an interface that combines the Shiny app and Tableau will enable investors to visualize and filter loans with minimal effort.

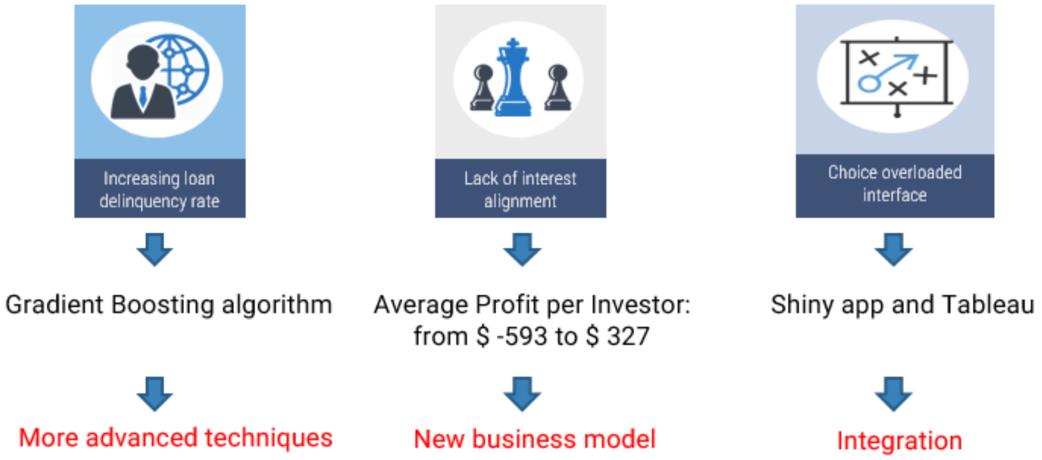

Increasing loan delinquency rate
Lack of interest alignment
Choice overloaded interface
Gradient Boosting algorithm
Average Profit per Investor: from $ -593 to $ 327
Shiny app and Tableau
More advanced techniques
New business model
Integration

Figure 8 Conclusion and Next Step

7. Code

# (1). Data Preprocessing in Python

```
# import different library
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")

# read data
data = pd.read_excel("/Users/shiqi/Desktop/LCDataDictionary.xlsx", sheetname='LoanStats')
data1 = pd.read_csv("/Users/shiqi/Desktop/LoanStats3b_securev1.csv",encoding = "ISO-8859-1")
data1.head()

# match the meanings of features to different features in order
df = pd.DataFrame(columns=('feature', 'description')) #create a dateframe
for i in range(data1.shape[1]):
    for j in range(data.shape[0]):
        if data1.columns[i] == data.loc[j][0]:
            df.loc[i]={'feature':data1.columns[i],'description':data.loc[j][1]}
df.to_csv("/Users/shiqi/Desktop/name.csv")
#match the sequence of column names in data1 to the description in data

# Text Analysis for feature “employment title”
!pip install wordcloud
from os import path
from wordcloud import WordCloud
%matplotlib inline
import matplotlib.pyplot as plt
cloud = WordCloud(width=1440, height=1080).generate(" ".join(data1["title"].astype(str)))
plt.figure(figsize=(20, 15))
plt.imshow(cloud)
plt.axis('off')

# Get the data1 with useful features we selected manually with business analysis
feature = pd.read_csv("/Users/shiqi/Desktop/features.csv")
ls_feature = feature["Column"].tolist()
data1 = data1[ls_feature]
data1.to_csv("/Users/shiqi/Desktop/clean_data.csv")
#get the missing value summary data
df_isnull = pd.DataFrame(data1.isnull().sum())
df_isnull.to_csv("/Users/shiqi/Desktop/isnull.csv")

# Missing values
# delete rows if the number of missing values is low
delete_row=['loan_amnt',"bc_util","mort_acc","num_sats","percent_bc_gt_75"]
data1.dropna(subset=delete_row,axis=0,inplace=True)
data1["mths_since_last_delinq"].replace(np.nan,84,inplace=True)
data1.ix[data1["mths_since_last_delinq"]>84,"mths_since_last_delinq"] = 84
data1["mths_since_last_delinq"].value_counts()
data1.isnull().sum()
```

```
# delete columns if the number of missing values is high
delete_column = ["mths_since_last_record",
"mths_since_last_major_derog","num_tl_op_past_12m","pct_tl_nvr_dlq"]
data1.drop(delete_column, axis=1, inplace=True)
data1.isnull().sum()

# Data transformation
# “term”: extract first two digit from str
data1["term"] = data1["term"].str[:3]
data1["term"] = data1["term"].astype(int)

# “int_rate”: extract percentage from str
data1['int_rate']=data1['int_rate'].str.extract('(\d+\.\d+)').astype(float) #extract float
data1['int_rate']=data1['int_rate']/100

# “emp_length”: extract float from str and get rid of ‘n/a’
data1["emp_length"].replace('n/a', np.nan, inplace=True)
data1.dropna(subset=["emp_length"],axis=0,inplace=True)
data1['emp_length']=data1['emp_length'].str.extract('(\d+)')
data1['emp_length'].astype(float)

# “loan_status”: change into binary variable
data1["loan_status"].replace(["Fully Paid","Current","In Grace Period"],1,inplace= True)
data1["loan_status"].replace(["Charged Off","Late (31-120 days)","Late (16-30
days)","Default"],0,inplace= True)
data1["loan_status"].value_counts()

# “fico_range_low”, “fico_range_high” : take average
data1["Fico_Average"]= (data1["fico_range_low"]+data1["fico_range_high"])/2
data1.drop("fico_range_low",axis=1,inplace=True)
data1.drop("fico_range_high",axis=1,inplace=True)
data1["Fico_Average"].dropna(axis=0, inplace= True)

# “revol_util”: extract float from str
data1["revol_util"]=data1["revol_util"].str.extract('(\d+\.\d+)').astype(float) #extract float
data1["revol_util"]=data1["revol_util"]/100

# “pub_rec_bankruptcies”, change into binary variable
data1["pub_rec_bankruptcies"].replace([1,2,3,4,5,6,7,8],1,inplace=True)

# Get rid of unexpected missing values and write data
data1.dropna(subset=["revol_util"],axis=0,inplace=True)
data1.to_csv("/Users/shiqi/Desktop/newdata.csv")
```

# (2). Preparation of Model Building in R

# 2.1 remove the unexpected columns--------

```
data <- read.csv("/Users/shiqi/Desktop/Project/dataset/original_data.csv")
drop <- c("X","Unnamed..0","addr_state")
data[drop] <- NULL
```

# 2.2 Split dataset to training and testing (1:1)----

```
library(caret)
set.seed(1234)
splitIndex <- createDataPartition(data$loan_status, p = .50, list = FALSE, times = 1)
traindata <- data[ splitIndex,]
testdata <- data[-splitIndex,]
prop.table(table(traindata$loan_status))
dim(traindata) # training and testing data contain 51284 rows respectively
```

# 2.3 Solution to imbalance problem (SMOTE)-----

```
# oversampling and undersampling for the training dataset
library(DMwR)
traindata$loan_status <- as.factor(traindata$loan_status)
traindata <- SMOTE(loan_status ~ ., traindata, perc.over = 100, perc.under=200)
write.csv(traindata,"/Users/shiqi/Desktop/Project/dataset/aftersmote.csv")
table(traindata$loan_status)
# numbers of "0" and "1" are equal after SMOTE with 15546 records each
dim(traindata)
# so the traindata currently contains 31092 rows
```

# (3). Model Training

# 3.1 Logistic Regression------------

```
lg.1 <- glm(loan_status~.,data=traindata, family="binomial")
traindata$pred1 <- predict(lg.1, newdata=traindata, type='response')
summary(lg.1)
library(pROC)
roc1 <- roc(traindata$loan_status, traindata$pred1, plot=TRUE, print.thres=TRUE)
print(roc1)
traindata$pred1 <- NULL
traindata$X <- NULL
# training AUC= 0.698
testdata$pred1 <- predict(lg.1, newdata=testdata, type='response')
roc2 <- roc(testdata$loan_status, testdata$pred1, plot=TRUE, print.thres=TRUE)
print(roc2)
# testing AUC= 0.6754
testdata$pred1.2[testdata$pred1<0.5] <- 0
testdata$pred1.2[testdata$pred1>=0.5] <- 1
mean(testdata$loan_status == testdata$pred1.2)
# testing Accuracy = 0.6228648
varImp(lg.1, scale=FALSE)
lg.1$coefficients
```

# 3.2 Support Vector Machine---------

```
library(e1071)
fit_svm <- svm(loan_status~., data=traindata)
testdata$pred3 <- predict(fit_svm, newdata = testdata, type="prob")
testdata$pred3 <- as.numeric(as.character(testdata$pred3))
roc(testdata$loan_status, testdata$pred3, plot=TRUE, print.thres=TRUE)
# testing AUC = 0.6243
mean(testdata$loan_status == testdata$pred3)
# testing Accuracy = 0.6325
```

# 3.3 Random Forest----------------

```
library(randomForest)
fit_rf <- randomForest(loan_status~., data=traindata, ntree=200)
testdata$pred5 <- predict(fit_rf, newdata = testdata)
testdata$pred5 <- as.numeric(as.character(testdata$pred5))
roc(testdata$loan_status, testdata$pred5, plot=TRUE, print.thres=TRUE)
# testing Auc = 0.6144
mean(testdata$loan_status == testdata$pred5)
# testing Accuracy = 0.7045472
```

# 3.4 Gradient Boosting -------------

```
fitControl <- trainControl(method="repeatedcv", number = 4, repeats = 4)
fit <- train(loan_status~., data=traindata, method="gbm", trControl = fitControl, verbose = FALSE)
testdata$pred2 <- predict(fit, newdata = testdata, type="prob")
roc(testdata$loan_status, testdata$pred2[, 2], plot=TRUE, print.thres=TRUE)
# testing AUC = 0.6793
testdata$pred2.3[testdata$pred2[, 2]<0.5] <- 0
testdata$pred2.3[testdata$pred2[, 2]>=0.5] <- 1
mean(testdata$loan_status == testdata$pred2.3)
# testing Accuracy = 0.7477966

# Conclusion: Gradient Boosting is selected as our model
# based on its highest AUC and Accuracy
```

# (4). Financial Analysis-----------------

# 4.1 Cut-off Selection------------

```
revenue_ls <- c()
ilist <- seq(0.5, 1, 0.05) #generate list from 0.5 to 1 with incremental of 0.05
revenue_ls <- c()

# run a loop tp calculate the profits for different cut-offs
for(i in ilist) {
  testdata$pred2.3[testdata$pred2[, 2]< i] <- 0
  testdata$pred2.3[testdata$pred2[, 2]>=i] <- 1
  revenue <- sum(testdata$loan_amnt[testdata$pred2.3==1 &
testdata$loan_status==1]*testdata$int_rate[testdata$pred2.3==1 & testdata$loan_status==1])
  cost <- sum(testdata$loan_amnt[testdata$pred2.3==1 & testdata$loan_status==0])
  total_after <- revenue-cost
  revenue_ls <- c(revenue_ls,total_after)
}
ilist
# Plot the relationship of cut-off and profit
plot(ilist,revenue_ls, xlab = "cut-off value", ylab = "total profit")

# Plot the relationship of cut-off and profit
max(revenue_ls)

testdata$pred2.3[testdata$pred2[, 2]<0.7] <- 0
testdata$pred2.3[testdata$pred2[, 2]>=0.7] <- 1

# testing Accuracy = 0.4669098 after changing cut-off to 0.7
mean(testdata$loan_status == testdata$pred2.3)
confusionMatrix(data=testdata$pred2.3, testdata$loan_status)

# Column contribution
varImp(fit, scale=FALSE)
plot(testdata$mort_acc,testdata$pred2[, 2])

# 4.2 Cost-Benefit Analysis-------------
#Profit Before Our Project
total_before <-
sum(testdata$loan_amnt[testdata$loan_status==1]*testdata$int_rate[testdata$loan_status==1]) -
sum(testdata$loan_amnt[testdata$loan_status==0])
# total_before/ #investor = $ -593

# Profit After Our Project
revenue_after <- sum(testdata$loan_amnt[testdata$pred2.3==1 &
testdata$loan_status==1]*testdata$int_rate[testdata$pred2.3==1 & testdata$loan_status==1])
cost_after <- sum(testdata$loan_amnt[testdata$pred2.3==1 & testdata$loan_status==0])
total_after <- revenue_after - cost_after
# total_before/ #investor = $ 327
```

# (5). New Data Prediction-----------

```
# new data needed to predict
predictdata <- read.csv("/Users/shiqi/Desktop/Project/dataset/newdata.csv")
predictdata$X <- NULL
predictdata$Unnamed..0 <- NULL
# predict the data based on gradient boosting
predictdata$prob_loan_status <- predict(fit, newdata = predictdata, type="prob")
predictdata$loan_status[predictdata$prob_loan_status[, 2]<0.7] <- 0
predictdata$loan_status[predictdata$prob_loan_status[, 2]>=0.7] <- 1
predictdata$prob_loan <- predictdata$prob_loan_status[, 2]
predictdata$prob_loan_status <- NULL
# write the csv
write.csv(predictdata,"/Users/shiqi/Desktop/Project/dataset/afterpredict.csv")
sum(is.na(predictdata))
```

# (6). Data Preparation for Shiny--------

```
after <- read.csv("/Users/shiqi/Desktop/Project/dataset/afterpredict.csv")
hist(after$prob_loan)
# generate the column of "expected return rate"
after$exp_return <- after$int_rate * after$prob_loan
# generate the column of "up_boundary"
after$up_boundary <- 0.2403
after$up_boundary[after$grade == 'E'] <- 0.2007
after$up_boundary[after$grade == 'D'] <- 0.1727
after$up_boundary[after$grade == 'C'] <- 0.1416
after$up_boundary[after$grade == 'B'] <- 0.1095
after$up_boundary[after$grade == 'A'] <- 0.0743
# generate the column of "low_boundary"
after$low_boundary <- 0.0523
after$low_boundary[after$grade == 'E'] <- 0.0634
after$low_boundary[after$grade == 'D'] <- 0.0682
after$low_boundary[after$grade == 'C'] <- 0.0710
after$low_boundary[after$grade == 'B'] <- 0.0656
after$low_boundary[after$grade == 'A'] <- 0.0495
# write the csv for shiny APP
write.csv(after, "/Users/shiqi/Desktop/Project/dataset/shinydata.csv")
```

# (7). Shiny App Development--------

```
# Prep -----------------
library(tidyverse)
# library(readr)
# library(dplyr)
library(leaflet)
library(shiny)

listings <- read_csv("afterpredict.csv")


# UI -------------------
# Define UI for application that draws a histogram
ui <- fluidPage(
  title = "Lending Club",
  titlePanel("Lending Club"),
  tabsetPanel(
    tabPanel("Investor", dataTableOutput("table"))
  ),
  hr(),
  fluidRow(
    column(4,
           h3("Filter By Loan Amount"),
           column(6,
                  numericInput("minLoan", label = "Minimum Loan Amount:", value = 1000)
           ),
           column(6,
         numericInput("maxLoan", label = "Maximum Loan Amount:", value = 35000)
      )
  ),
  column(4,
      h3("Filter By Interest Rate"),
      checkboxGroupInput("gradefilter", label="Return Grade:",
               choices = c("A", "B", "C", "D", "E", "F"),
               selected = c("A", "B", "C", "D", "E", "F"))
  ),
  column(4,
      h3("Filter By Loan Term"),
      checkboxGroupInput("termfilter", label="Loan Term:",
               choices = c("36 months"=36, "60 months"=60),
               selected = c("36 months"=36, "60 months"=60))
  ),
  column(4,
      h3("Filter By Probability of Fully Paid"),
      sliderInput("paidfilter", label = "Pr (Fully Paid):",
            min = 0, max = 1, value = c(0,1))
  )
 ),
 hr(), #horizontal row
 p("Data from", a("Lending Club", href= "https://www.lendingclub.com/site/home", target = "_blank"))
)

# Server ------------------------
# Define server logic required to draw a histogram
server <- function(input, output) {

 df <- reactive({
  df <- listings %>%
   filter((loan_amnt >= input$minLoan & loan_amnt <= input$maxLoan) &
       (prob_loan >= input$paidfilter[1] & loan_status <= input$paidfilter[2]) &
       (grade %in% input$gradefilter)&
       (term %in% input$termfilter)) #within the checkbox numeric values
  return(df)
 })


 # Data Table --------------------
 output$table <- renderDataTable({df()})

}

# Run the app -------------------------
# Run the application
shinyApp(ui = ui, server = server)
rsconnect::setAccountInfo(name='shiqili',
             token='96B020622C170A87ED72494BFAA69C74',
             secret='WWy6cxpmY1tFXRkyWE//TUNtOkf9ZQ9YToCV4zM7')
```