Columbia University

STAT W4201

Advanced Data Analysis

Will A Loan Default ?

A statistical study of credit analysis for Lending Club loans

---

Author:

Jiahong Hu (jh3561)

Supervisor:

Prof.Demissie Alemayehu

December 13, 2015

1 Background

Lending Club, located in San Francisco, is the worlds largest peer-to-peer lending platform. Online peer-to-peer lending is a relatively new practice of lending money without going through a traditional financial intermediary, such as a bank. However, compared to the traditional types of loans, peer-to-peer loans are usually unsecured personal loans and associated with high risk of default. Hence, our team is tempted to propose models to predict loan status (default/fully paid-off) based on the information before the loan initiation. Our approach could be served as a supplement tool to improve the companys overall risk control and to secure its long-term growth.

2 Objective

Our project is aimed to select relevant factors that have influential effects on a loaners ability to repay his/her loan based on historical loan data. Relied on those factors, we apply statistical classification models to predict whether a loan will default. We will evaluate the performance of the models based on two indicators: total classification error rate and false negative rate. We intend to reduce false negative rate because we want to avoid the situation where many default observations are classified as non-default, which is very risky to the business.

3 Data Source and Exploratory Data Analysis

3.1 Data Source

Our analysis is based on the public data from Lending Club. The dataset contains complete loan data for all loans issued through the time period between 2013 and 2014, with 53 variables and 260,000 observations in total. After performing preliminary data cleaning (missing values and outliers) and features selection based on the objectives of project, a new dataset with 14 variables and 50,000 observations is constructed for further analysis. Those 14 variables provide information about the loaners background and the characteristics of those loans. Table 6. in the Appendix provide the summary about 14 variables.

3.2 Exploratory Data Analysis (EDA)

The results of exploratory data analysis are presented here,

Step 1: Response Variable Loan Status (Y)

	Y=1 (Default and Charged off)	**Y=0** (Fully Paid)	Total
Number	10494	44629	55123
%	19%	81%	100%

Table 1: Summary of Response Variable

Table 1. shows that the sample default rate is approximately 19% and hence the imbalanced class classification problem should be paid attention to in the later stage.

Step 2: Response Variable Loan Status (Y) and Numeric Variables

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Annual income ($)	3,000	47,000	65,000	74,710	90,000	4,900,000
Debt to Income Ratio	0.00	11.40	16.86	17.26	22.86	39.99
Interest Rate (%)	0.06	0.11	0.14	0.14	0.17	0.26
Loan Amount ($)	1,000	8,000	12,000	14,090	19,200	35,000
Public Record	0.00	0.00	0.00	0.24	0.00	11.00

Table 2: Summary of Numeric Variables

We further look at the distribution of each numeric value and eliminated possible outliers.

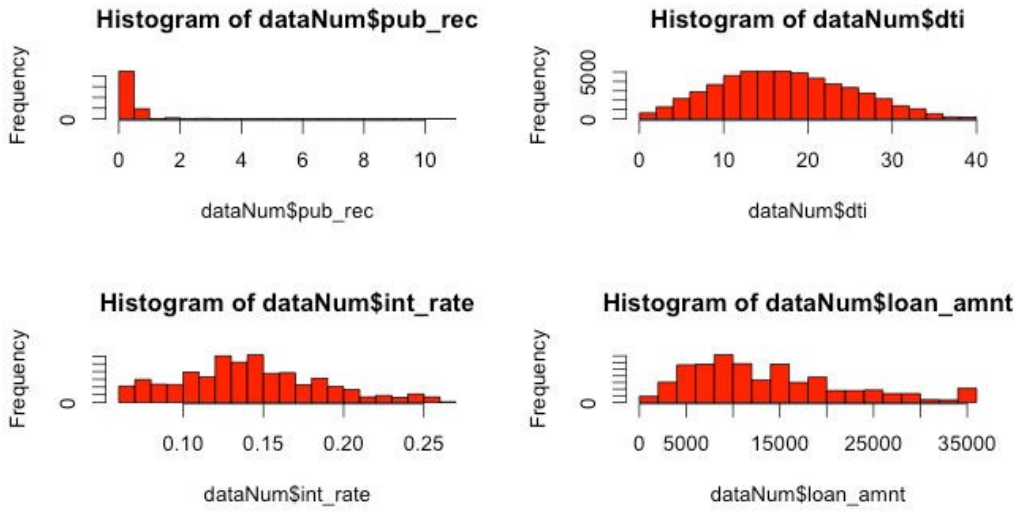
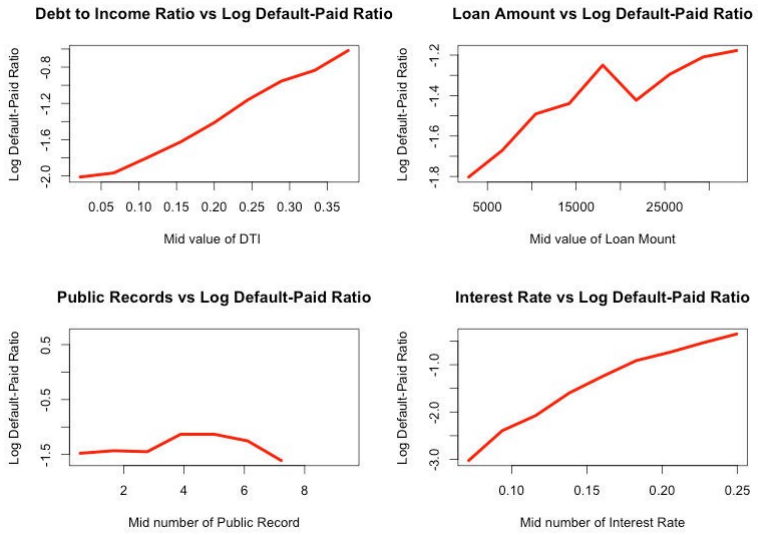


Figure 1: Histogram of Numeric Variables

Then, we examine the relationships between Y and each numeric variable separately.



The relationship between log default-to-paid ratio and explanatory variables such as: debt-to-income ratio, loan amount, public records, and interest rate are presented here [1],

• For the plot on upper left corner, there seems to be a strong positive linear relationship between debt-to-income ratio and log default-to-paid ratio.

• For the plot on the upper right corner, we can see the relationship between debt-to-income-ratio and loan amount generally follows a positive linearity with a peak near loan amount $1900.

• For the plot on the lower left corner, there is no general relationship between debt-to-income-ratio and public records. When the number of derogatory records is 0 or less than 2, the

corresponding default rate is relatively low. When the number of derogatory records is between 4 and 6, the default rate is the highest among all groups. When the number of derogatory records is more than 6, the chance of default is surprisingly low, which might be caused by the high probability of rejection of the people with more than 6 derogatory records.

• For the plot on the lower right corner, there seems to be a strong positive linear relationship between interest rate and debt-to-paid-ratio.

The findings above suggest that the debt-to-income ratio and interest rate may play an important role in our future predicting model.

Step 3 Response Variable Loan Status (Y) and Category Variables

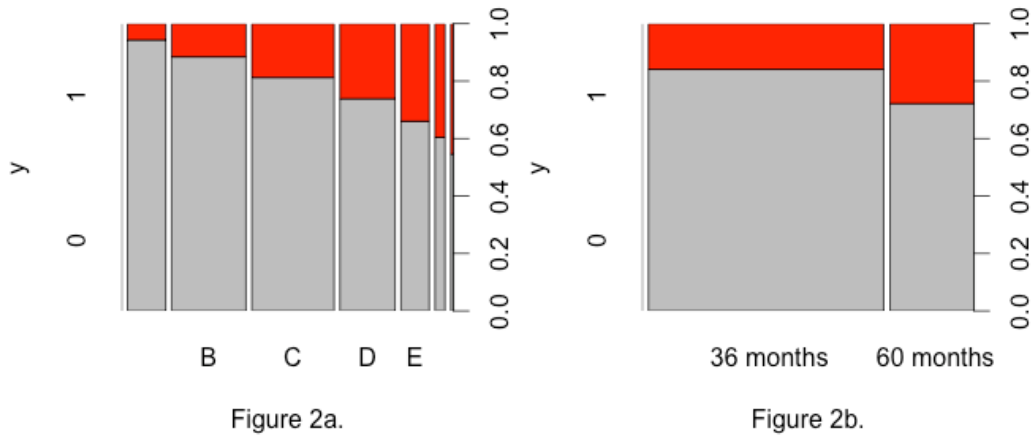


Figure 2a. Figure 2b.

Figure 2: Relationship between Loan Status with Loan Grade Category and Term

In Figure 2a, the area of each bar represents the relative sample size of observations in each category and the proportion of red area in each bar represents the rate of default in each Loan Grade category A, B, C, D, E, F. The majority of loans of Lending Clubs are rated as grade B and C. The higher the loan rating in alphabetical order, the lower the default rate.

Figure 2b shows that Landing Club granted significantly more loans with 36-month periods of payments than 60-month period of payments. Compared to loans with 36-month periods, loans with 60 months periods have higher default rate on average.

Moreover, the Kruskall-Wallis Rank Sum Test produced P-value less than 5% and thus we confirmed that differences of default rate among various Loan Grade Categories are significant. Wilcoxon Rank Sum Test also confirmed the difference of default rate between 36-month and 60-month term are statistically significant.

4 Modeling and Results

In this section, two classification methods, logistic regression and random forest, are used to identify high-risk individuals who will default.

4.1 Logistic Regression

Firstly, we used logistic regression model. The reason for choosing this model is,

- The response variable is binary. Hence, logistic regression is a neutral choice.
- Logistic regression could give us an explicit model, which is easy to interpret.
- Flexible thresholds could be used in logistic regression.

We randomly chose 20% of the data as the test data set and the remaining as the training data set to train the model. The result below shows that almost all variables are significant.

```
> summary(log_train)

Call:
glm(formula = y_train ~ ., family = binomial("logit"), data = x_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6019  -0.6657  -0.5080  -0.3418   2.8331

Coefficients: (1 not defined because of singularities)
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.065e+00  1.021e-01 -39.799  < 2e-16 ***
annual_inc  -4.443e-06  4.424e-07 -10.044  < 2e-16 ***
dti          3.335e-02  1.732e-03  19.250  < 2e-16 ***
emp_length  -2.030e-02  3.656e-03  -5.554 2.79e-08 ***
grade       -9.330e-02  3.891e-02  -2.398 0.016499 *
int_rate     1.541e+01  1.209e+00  12.739  < 2e-16 ***
loan_amnt    1.897e-05  2.022e-06   9.385  < 2e-16 ***
pub_rec     -6.760e-02  2.499e-02  -2.705 0.006835 **
purpose     -5.213e-02  2.753e-02  -1.893 0.058308 .
rent         1.946e-01  4.701e-02   4.141 3.46e-05 ***
mortgage    -1.715e-01  4.708e-02  -3.643 0.000269 ***
home                NA         NA      NA       NA
term         4.091e-03  1.420e-03   2.881 0.003968 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 3: Result of Logistic Regression

We then checked the goodness of fit by calculating the Pearson-residual, which is 41610, and the result of Pearson chi-square test shows the model is a good fit.

If we set the threshold at 50% for the probability of default as usual, to classify an observation, we would get 18.44% and 18.77% total error rate for training and test data, respectively. However, the false negative rate is 93.93% which is inconsistent with our primary objective of identifying high-risk individuals who will default. Therefore, we want to train our model with different thresholds from 0 to 1 to obtain an optimal point.

The output for training set as well as test set is as shown in Figure 4.

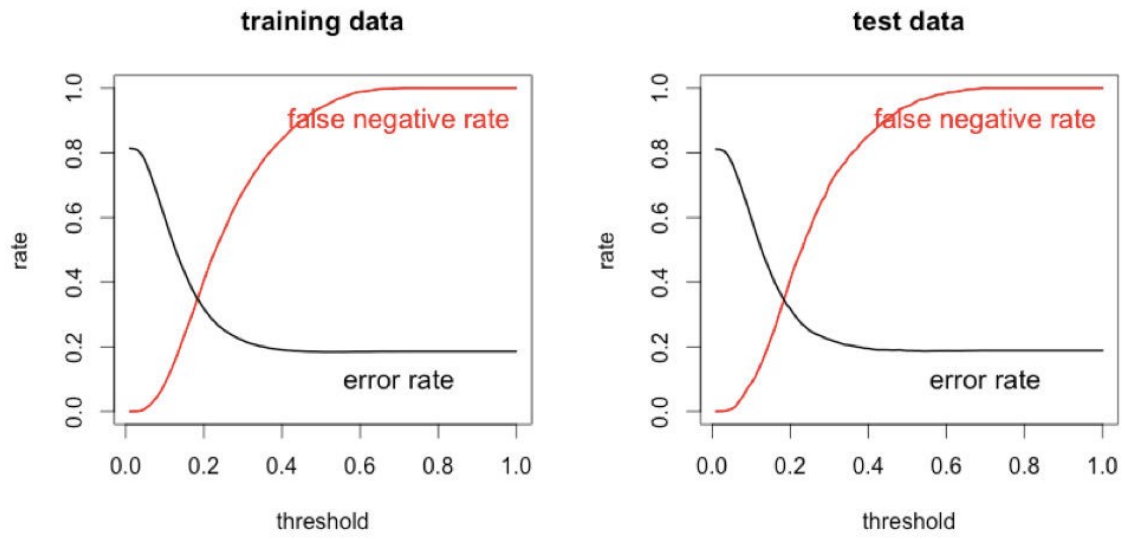


Figure 4: Total Error Rate and False Negative Rate

From Figure 4, we can tell that as the value of threshold point decreases, the false negative rate decrease, but at the large expense of increasing error rate.

Next, we wanted to check if any of the variables are useless and then checked the feature importance. We performed model selection using exhaustive method based on the value of BIC. The output is shown in Table 4.

row.names	annual_inc	dti	emp_length	grade	int_rate	loan_amnt	pub_rec	purpose	rent	mortgage	home	term
1 (1)					*							
2 (1)		*			*							
3 (1)		*			*				*			
4 (1)		*			*	*			*			
5 (1)	*	*			*	*			*			
6 (1)	*	*	*		*	*			*			
7 (1)	*	*	*		*	*			*			*
8 (1)	*	*	*		*	*	*		*			*
9 (1)	*	*	*		*	*	*		*		*	*

Table 3: Model selection and feature importance of Logistic Regression

The first column indicates how many variables we use in each step of the best model, and the * sign indicates if the variable is included in the model or not.

Here is another presentation of the result.

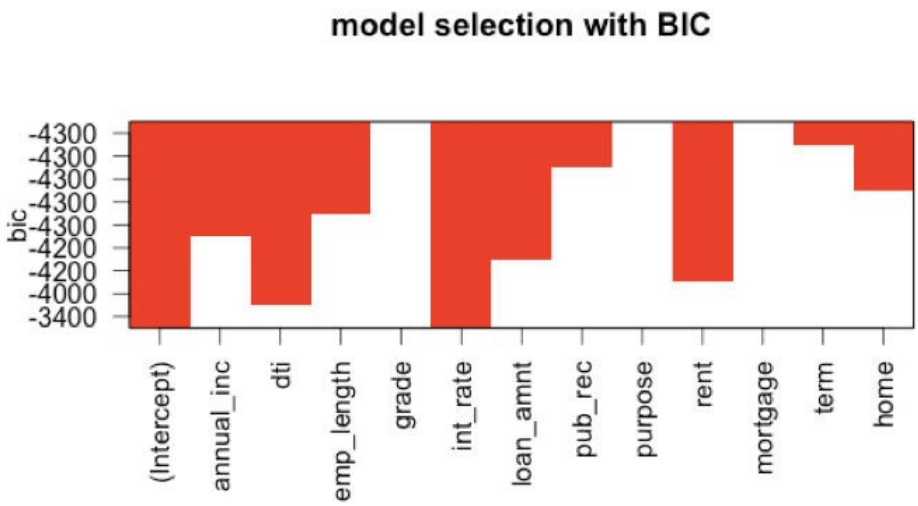


Figure 5: Feature Important Level of Logistic Regression

The number on the left shows the value deduction of BIC when the marked variables are included. Since the value of BIC always decreases when we add more variables, we do not need to delete any variables and we keep the previous model unchanged. Also, the most important variables are interest rate, debt-to-income ratio, home ownership, loan amount and annual income. The company could pay more attention to these features.

We then use the lasso regression to test the feature importance level. In our graph, it shows the most important variable is the interest rate. The secode is dti. The third is rent. Then other variables are almost similar important. The result is same as the result of the logistic regression.

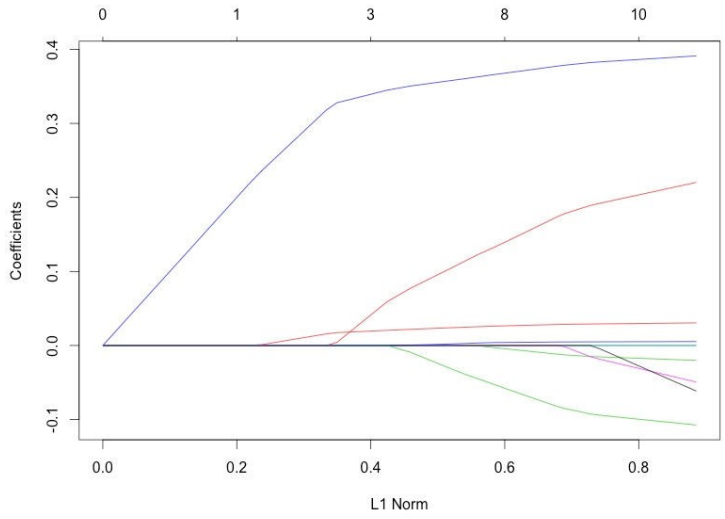


Figure 6: Result of Lasso Logistic Regression

4.2 Random Forest

The second classifier we use is Random Forest. The reason why we use Random Forest is,

- Unbalanced classes problem could be partially solved by up-sampling or down-sampling
- There is no underlying assumption about data in order to use random forest
- Random froest further reduces predictors correlation.

At first, we implemented Random Forest without balancing the data. The total error rate is 18.74% but the false negative rate is 93.78%. In order to decease a lower false negative rate, we manually selected equal number of default data and non-default data on each iteration to fit the model. Although the total error rate increases to 30.62%, the false negative rate decreases significantly to 42.39%.

row.names	error rate	false negative rate
training data without Down-sampling	0.1854515	0.9406574
test data without Down-sampling	0.1882655	0.9425113
training data with Down-sampling	0.2982319	0.4521038
test data with Down-sampling	0.2959271	0.4483106

Table 4: Summary of Random Forest

The feature importance in Random Forest model is shown in Figure 6. And the predictors are chose based on Gini Index.

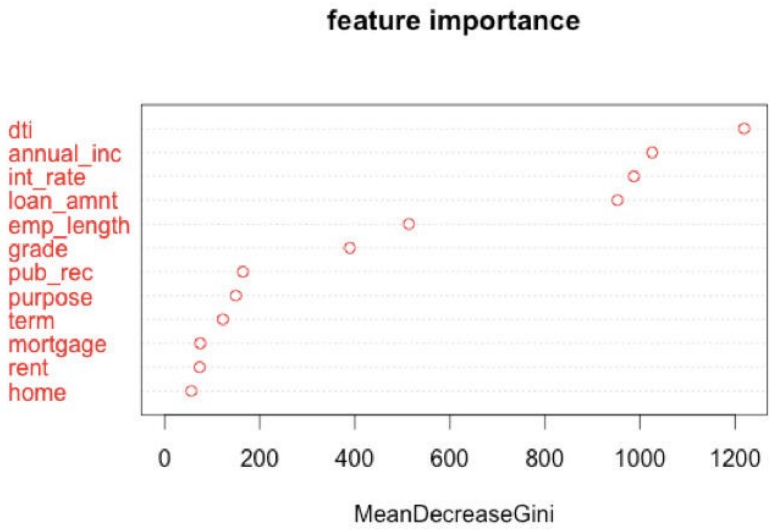


Table 5: Feature Importance Level of Random Forest

The most important features are debt-to-income ratio, annual income, interest rate, loan amount and employment length.

5 Discussion and Conclusion

Based on balanced random forest model, we achieve a relatively low error rate while keeping a reasonable false negative rate. As for the logistic regression model, to choose a optimal threshold point is rather a trade-off problem. We suggest the company to carefully select a threshold based on their objectives. If the company wants to expand business and attract more clients, the threshold value could be small; if the company cares more about the security of their money and intends to avoid high false negative rates, they should pick a high threshold value.

Combined these two models, though there is a minor difference, the most important features are debt-to-income ratio, annual income, interest rate. The company may apply our models not only to predict default loaner but pay more attention to above features when conducting pro-phase survey to a specifi client.

6 Appendix

No.	Variable Name	Type	Source	Note
Y	Loan Status	Binary		1: Default and Charged off 0: Fully Paid
X1	Annual Income	Numeric	Loaner	
X2	Debt To Income Ratio	Numeric	Both	[0,1]
X3	Interest Rate	Numeric	Loan	[0,1]; Interest Rate on the Loan
X4	Loan Amount	Numeric	Loan	$
X5	Public Record	Numeric	Loaner	Number of derogatory public record
X6	Employment Length	Ordinal	Loaner	[0,10]; In years; 0 = <1 and 10 = >10
X7	Loan Grade	Ordinal	Loan	A- G (A represents highest rating)
X8	Loan Sub Grade	Ordinal	Loan	A1-A5,…, G1-G5
X9	Purpose of the Loan	Category	Loan	1: Debt Consolidation 0: Others
X10	Term	Category	Loan	Number of Payments - Either 36 or 60 months
X11	Home Ownership Status	Category	Loan	Rent, Own, Mortgage

We plotted the response variable Loan Status (Y) against five category variables, including Loan Grade (X7), Sub Loan Grade (X8), Purpose of Loan (X9), Term (X10), and Home Ownership Status (X11). Figure 2. shows that there might exist certain relationships between response variable Loan Status (Y) and Loan Grade (X7) and between Loan status(Y) and Term (X10) but Figure 7. shows that there are no obvious relationships among Loan Status (Y) and Sub Loan Grade (X8), Purpose of Loan (X9), and Home Ownership Status (X11).

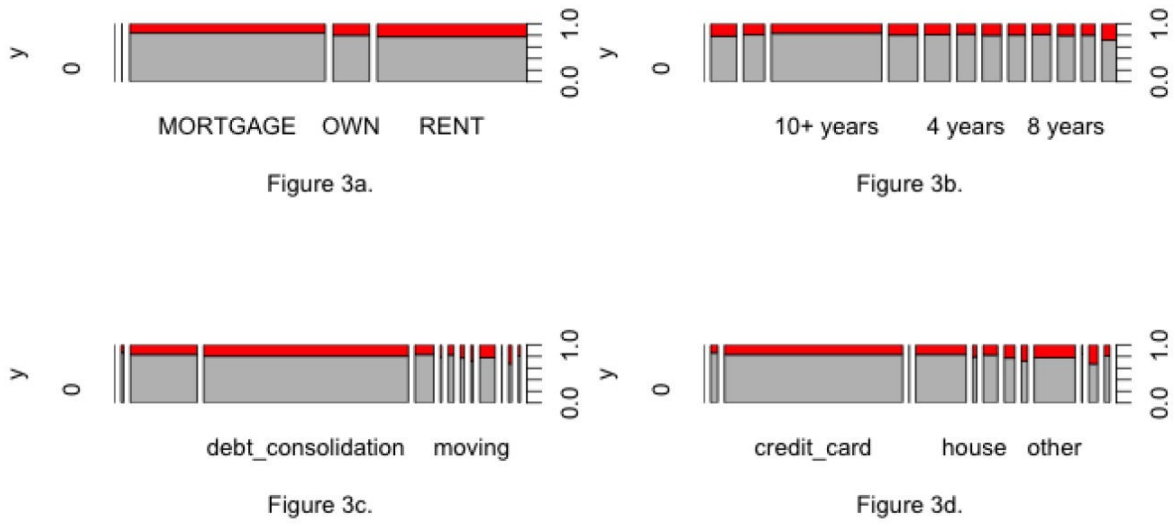


Figure 3a. Figure 3b.

Figure 3c. Figure 3d.

Figure 7: Summary of Our Dataset

```
########### EDA ###############
## numeric data
# histogram
# interest rate
temp = rep(0,length(data[,1]))
temp = as.numeric(substr(as.character(data[,7]),2,6))/100
data[,7] = temp

dataNum = data[,c(1,2,7,8,10)]
hist(dataNum$pub_rec, col="red")
hist(dataNum$dti, col="red")
hist(dataNum$int_rate, col="red")
hist(dataNum$loan_amnt, col="red")

# linear relationship
# dti
range(data$dti)
bin_dti = seq(min(data$dti),max(data$dti),length.out=10)

x0 = subset(data,data$dti>=bin_dti[1] & data$dti<bin_dti[2])
x0_prop = 1-length(which(x0$loan_status=='Fully Paid'))/nrow(x0)
x1 = subset(data,data$dti>=bin_dti[2] & data$dti<bin_dti[3])
x1_prop = 1-length(which(x1$loan_status=='Fully Paid'))/nrow(x1)
x2 = subset(data,data$dti>=bin_dti[3] & data$dti<bin_dti[4])
x2_prop = 1-length(which(x2$loan_status=='Fully Paid'))/nrow(x2)
x3 = subset(data,data$dti>=bin_dti[4] & data$dti<bin_dti[5])
x3_prop = 1-length(which(x3$loan_status=='Fully Paid'))/nrow(x3)
x4 = subset(data,data$dti>=bin_dti[5] & data$dti<bin_dti[6])
x4_prop = 1-length(which(x4$loan_status=='Fully Paid'))/nrow(x4)
x5 = subset(data,data$dti>=bin_dti[6] & data$dti<bin_dti[7])
x5_prop = 1-length(which(x5$loan_status=='Fully Paid'))/nrow(x5)
x6 = subset(data,data$dti>=bin_dti[7] & data$dti<bin_dti[8])
x6_prop = 1-length(which(x6$loan_status=='Fully Paid'))/nrow(x6)
x7 = subset(data,data$dti>=bin_dti[8] & data$dti<bin_dti[9])
x7_prop = 1-length(which(x7$loan_status=='Fully Paid'))/nrow(x7)
x8 = subset(data,data$dti>=bin_dti[9] & data$dti<bin_dti[10])
x8_prop = 1-length(which(x8$loan_status=='Fully Paid'))/nrow(x8)

z2 = cbind(x0_prop,x1_prop,x2_prop,x3_prop,x4_prop,x5_prop,x6_prop,x7_prop,x8_prop)

z2 = apply(z2, 2,function(x) log(x/(1-x)))

bin_dti_mid = (bin_dti[1:9] + bin_dti[2:10])/2
plot(bin_dti_mid,z2,main="Debt to Income Ratio vs Log Default-Paid Ratio",
     col="red",type="l",xlab="Mid value of DTI ",ylab="Log Default-Paid Ratio")

# Loan amount
range(data$loan_amnt)
bin_loan_amnt = seq(min(data$loan_amnt),max(data$loan_amnt),length.out=10)

x0 = subset(data,data$loan_amnt>=bin_loan_amnt[1] & data$loan_amnt<bin_loan_amnt[2])
x0_prop = 1-length(which(x0$loan_status=='Fully Paid'))/nrow(x0)

x1 = subset(data,data$loan_amnt>=bin_loan_amnt[2] & data$loan_amnt<bin_loan_amnt[3])
x1_prop = 1-length(which(x1$loan_status=='Fully Paid'))/nrow(x1)

x2 = subset(data,data$loan_amnt>=bin_loan_amnt[3] & data$loan_amnt<bin_loan_amnt[4])
x2_prop = 1-length(which(x2$loan_status=='Fully Paid'))/nrow(x2)

x3 = subset(data,data$loan_amnt>=bin_loan_amnt[4] & data$loan_amnt<bin_loan_amnt[5])
x3_prop = 1-length(which(x3$loan_status=='Fully Paid'))/nrow(x3)

x4 = subset(data,data$loan_amnt>=bin_loan_amnt[5] & data$loan_amnt<bin_loan_amnt[6])
x4_prop = 1-length(which(x4$loan_status=='Fully Paid'))/nrow(x4)

x5 = subset(data,data$loan_amnt>=bin_loan_amnt[6] & data$loan_amnt<bin_loan_amnt[7])
x5_prop = 1-length(which(x5$loan_status=='Fully Paid'))/nrow(x5)

x6 = subset(data,data$loan_amnt>=bin_loan_amnt[7] & data$loan_amnt<bin_loan_amnt[8])
x6_prop = 1-length(which(x6$loan_status=='Fully Paid'))/nrow(x6)

x7 = subset(data,data$loan_amnt>=bin_loan_amnt[8] & data$loan_amnt<bin_loan_amnt[9])
x7_prop = 1-length(which(x7$loan_status=='Fully Paid'))/nrow(x7)

x8 = subset(data,data$loan_amnt>=bin_loan_amnt[9] & data$loan_amnt<bin_loan_amnt[10])
x8_prop = 1-length(which(x8$loan_status=='Fully Paid'))/nrow(x8)

z3 = cbind(x0_prop,x1_prop,x2_prop,x3_prop,x4_prop,x5_prop,x6_prop,x7_prop,x8_prop)
z3 = apply(z3, 2,function(x) log(x/(1-x)))

bin_loan_amnt_mid = (bin_loan_amnt[1:9] + bin_loan_amnt[2:10])/2
plot(bin_loan_amnt_mid,z3,main="Loan Amount vs Log Default-Paid Ratio",
     col="red",type="l",xlab="Mid value of Loan Mount",ylab="Log Default-Paid Ratio")

# Public Record
range(data$pub_rec)
bin_pub_rec = seq(min(data$pub_rec),max(data$pub_rec),length.out=10)

x0 = subset(data,data$pub_rec>=bin_pub_rec[1] & data$pub_rec<bin_pub_rec[2])
x0_prop = 1-length(which(x0$loan_status=='Fully Paid'))/nrow(x0)

x1 = subset(data,data$pub_rec>=bin_pub_rec[2] & data$pub_rec<bin_pub_rec[3])
x1_prop = 1-length(which(x1$loan_status=='Fully Paid'))/nrow(x1)

x2 = subset(data,data$pub_rec>=bin_pub_rec[3] & data$pub_rec<bin_pub_rec[4])
x2_prop = 1-length(which(x2$loan_status=='Fully Paid'))/nrow(x2)

x3 = subset(data,data$pub_rec>=bin_pub_rec[4] & data$pub_rec<bin_pub_rec[5])
x3_prop = 1-length(which(x3$loan_status=='Fully Paid'))/nrow(x3)

x4 = subset(data,data$pub_rec>=bin_pub_rec[5] & data$pub_rec<bin_pub_rec[6])
x4_prop = 1-length(which(x4$loan_status=='Fully Paid'))/nrow(x4)

x5 = subset(data,data$pub_rec>=bin_pub_rec[6] & data$pub_rec<bin_pub_rec[7])
x5_prop = 1-length(which(x5$loan_status=='Fully Paid'))/nrow(x5)

x6 = subset(data,data$pub_rec>=bin_pub_rec[7] & data$pub_rec<bin_pub_rec[8])
x6_prop = 1-length(which(x6$loan_status=='Fully Paid'))/nrow(x6)

x7 = subset(data,data$pub_rec>=bin_pub_rec[8] & data$pub_rec<bin_pub_rec[9])
x7_prop = 1-length(which(x7$loan_status=='Fully Paid'))/nrow(x7)

x8 = subset(data,data$pub_rec>=bin_pub_rec[9] & data$pub_rec<bin_pub_rec[10])
x8_prop = 1-length(which(x8$loan_status=='Fully Paid'))/nrow(x8)

z6 = cbind(x0_prop,x1_prop,x2_prop,x3_prop,x4_prop,x5_prop,x6_prop,x7_prop,x8_prop)
z6 = apply(z6, 2,function(x) log(x/(1-x)))

bin_pub_rec_mid = (bin_pub_rec[1:9] + bin_pub_rec[2:10])/2
plot(bin_pub_rec_mid,z6,main="Public Records vs Log Default-Paid Ratio",
     col="red",type="l",xlab="Mid number of Public Record",ylab="Log Default-Paid Ratio")
```

```r
# interest rate
range(data$int_rate)
bin_int_rate= seq(min(data$int_rate),max(data$int_rate),length.out=10)

x0 = subset(data,data$int_rate>=bin_int_rate[1] & data$int_rate<bin_int_rate[2])
x0_prop = 1-length(which(x0$loan_status=='Fully Paid'))/nrow(x0)

x1 = subset(data,data$int_rate>=bin_int_rate[2] & data$int_rate<bin_int_rate[3])
x1_prop = 1-length(which(x1$loan_status=='Fully Paid'))/nrow(x1)

x2 = subset(data,data$int_rate>=bin_int_rate[3] & data$int_rate<bin_int_rate[4])
x2_prop = 1-length(which(x2$loan_status=='Fully Paid'))/nrow(x2)

x3 = subset(data,data$int_rate>=bin_int_rate[4] & data$int_rate<bin_int_rate[5])
x3_prop = 1-length(which(x3$loan_status=='Fully Paid'))/nrow(x3)

x4 = subset(data,data$int_rate>=bin_int_rate[5] & data$int_rate<bin_int_rate[6])
x4_prop = 1-length(which(x4$loan_status=='Fully Paid'))/nrow(x4)

x5 = subset(data,data$int_rate>=bin_int_rate[6] & data$int_rate<bin_int_rate[7])
x5_prop = 1-length(which(x5$loan_status=='Fully Paid'))/nrow(x5)

x6 = subset(data,data$int_rate>=bin_int_rate[7] & data$int_rate<bin_int_rate[8])
x6_prop = 1-length(which(x6$loan_status=='Fully Paid'))/nrow(x6)

x7 = subset(data,data$int_rate>=bin_int_rate[8] & data$int_rate<bin_int_rate[9])
x7_prop = 1-length(which(x7$loan_status=='Fully Paid'))/nrow(x7)

x8 = subset(data,data$int_rate>=bin_int_rate[9] & data$int_rate<bin_int_rate[10])
x8_prop = 1-length(which(x8$loan_status=='Fully Paid'))/nrow(x8)

z7 = cbind(x0_prop,x1_prop,x2_prop,x3_prop,x4_prop,x5_prop,x6_prop,x7_prop,x8_prop)
z7 = apply(z7, 2,function(x) log(x/(1-x)))

bin_int_rate_mid = (bin_int_rate[1:9] + bin_int_rate[2:10])/2
plot(bin_int_rate_mid,z7,main="Interest Rate vs Log Default-Paid Ratio",
     col="red",type="l",xlab="Mid number of Interest Rate",ylab="Log Default-Paid Ratio")


## categorical data
plot(grade, as.factor(loan_status), col=c("gray","red"))
plot(term, as.factor(loan_status), col=c("gray","red"))
plot(home_ownership, as.factor(loan_status), col=c("gray","red"))
plot(emp_length, as.factor(loan_status), col=c("gray","red"))
plot(purpose, as.factor(loan_status), col=c("gray","red"))
plot(subset(data, purpose!='debt_consolidation')$purpose,
     as.factor(subset(data, purpose!='debt_consolidation')$loan_status), col=c("gray","red"))



############ Modeling ###############

###### preprocessing ######
# delete ownship "any"
data = data[-12485,]

# change grade
temp = rep(0,length(data[,1]))
temp[which(data[,4] == "A")] = 1
temp[which(data[,4] == "B")] = 2
temp[which(data[,4] == "C")] = 3
temp[which(data[,4] == "D")] = 4
temp[which(data[,4] == "E")] = 5
temp[which(data[,4] == "F")] = 6
temp[which(data[,4] == "G")] = 7
data[,4] = temp

# change paid or not
temp = rep(0,length(data[,1]))
temp[which(data[,9] == "Fully Paid")] = 0
temp[which(data[,9] != "Fully Paid")] = 1
data[,9] = temp

# delete length
data = data[-which(data[,3] == "n/a"),]

# change emp_length
temp = rep(0,length(data[,1]))
temp[which(data[,3] == " < 1 year")] = 0
temp[which(data[,3] == "1 year")] = 1
temp[which(data[,3] == "2 years")] = 2
temp[which(data[,3] == "3 years")] = 3
temp[which(data[,3] == "4 years")] = 4
temp[which(data[,3] == "5 years")] = 5
temp[which(data[,3] == "6 years")] = 6
temp[which(data[,3] == "7 years")] = 7
temp[which(data[,3] == "8 years")] = 8
temp[which(data[,3] == "9 years")] = 9
temp[which(data[,3] == "10+ years")] = 10
data[,3] = temp

# purpose column
temp = rep(0,length(data[,1]))
temp[which(data[,11] == "debt_consolidation")] = 1
temp[which(data[,11] != "debt_consolidation")] = 0
data[,11] = temp

# ownership
temp = matrix(0,length(data[,1]),3)
temp[which(data[,5] == "RENT"),1] = 1
temp[which(data[,5] == "MORTGAGE"),2] = 1
temp[which(data[,5] == "OWN"),3] = 1
data[,14] = temp[,1]
data[,15] = temp[,2]
data[,16] = temp[,3]
names(data)[14:16]<-c("rent","mortage","home")

# term
temp = as.numeric(gsub('months', '', data$term))
data[,13] = temp

data = data[,-c(5, 6, 12)]
```

```
y = data[,7]
y = as.factor(y)
x = data[,-7]
set.seed(123)
index = sample(1:length(y), length(y), replace = F)
index_test = index[1:(length(y)*0.2)]
y_test = y[index_test]
x_test = x[index_test, ]
y_train = y[-index_test]
x_train = x[-index_test, ]


###### Logistic Regression ######
# for training data
library(glmnet)
log_train = glm(y_train~. , data = x_train, family = binomial("logit"))
summary(log_train)
train_fitted = log_train$fitted.values
train_fitted = as.numeric(train_fitted)

# goodness of fit
sum(residuals(log_train, type = "pearson")^2)
1 - pchisq(deviance(log_train), df.residual(log_train))

false_neg_train = rep(0,100)
error_train = rep(0,100)
for (i in c(1:100)){
        temp = rep(0,length(train_fitted))
        temp[which(train_fitted>=i/100)] = 1
        temp[which(train_fitted<i/100)] = 0
        false_neg_train[i] = sum(y_train[which(temp==0)]==1)/sum(y_train==1)
        error_train[i] = 1 - (sum(y_train[which(temp==0)]==0)+sum(y_train[which(temp==1)]==1))/length(y_train)
}

# for test data
x_test = data.frame(x_test)
colnames(x_test) = colnames(x_train)
test_pred_log = predict(log_train, newdata=x_test, type="response")

false_neg_test = rep(0,100)
error_test = rep(0,100)
for (i in c(1:100)){
        temp = rep(0,length(test_pred_log))
        temp[which(test_pred_log>=i/100)] = 1
        temp[which(test_pred_log<i/100)] = 0
        false_neg_test[i] = sum(y_test[which(temp==0)]==1)/sum(y_test==1)
        error_test[i] = 1 - (sum(y_test[which(temp==0)]==0)+sum(y_test[which(temp==1)]==1))/length(y_test)
}

par(mfrow=c(1,2))
plot(c(1:100)/100, false_neg_train, type="l", col="red", lwd=2, xlab="threshold",
     ylab="rate", main="training data", bg=5)
lines(c(1:100)/100, error_train, type="l", lwd=2)
text(0.7, 0.9, labels="false negative rate", col="red", cex=1.2)
text(0.7, 0.1, labels="error rate", col="black", cex=1.2)
plot(c(1:100)/100, false_neg_test, type="l", col="red", lwd=2, xlab="threshold", ylab="rate", main="test data")
lines(c(1:100)/100, error_test, type="l", lwd=2)
text(0.7, 0.9, labels="false negative rate", col="red", cex=1.2)
text(0.7, 0.1, labels="error rate", col="black", cex=1.2)
logi_rate_table = cbind(false_neg_train, error_train, false_neg_test, error_test)

par(mfrow=c(1,1))

# model selection
library(leaps)
model_sel = regsubsets(y~., data = x, method="exhaustive")
summary = summary(model_sel, matrix=T)
which_var = as.data.frame(summary$outmat)
which_var
summary$bic
plot(model_sel, scale = "bic", main = "model selection with BIC", col="red")

# L1 Logistic Regression
library(glmnet)
lasso_train <- glmnet(as.matrix(x_train), y_train, alpha=1, family='binomial')
plot(lasso_train)


###### random forest ######
# RF without Down-sampling
library(randomForest)
rf<-randomForest(y_train~., data=x_train)
plot(rf,log="y")
table_rf_train = table(rf$predicted, y_train)
train_error_rate_without = (table_rf_train[1,2] + table_rf_train[2,1]) / sum(table_rf_train)
train_false_negative_rate_without = table_rf_train[1,2] / sum(table_rf_train[,2])

rf_pred = predict(rf, x_test)
table_rf_test = table(rf_pred, y_test)
test_error_rate_without = (table_rf_test[1,2] + table_rf_test[2,1]) / sum(table_rf_test)
test_false_negative_rate_without = table_rf_test[1,2] / sum(table_rf_test[,2])

# RF with DS
library(randomForest)
rf_ds<-randomForest(y_train~., data=x_train, sampsize=c(7000,7000), strada=y_train)
varImpPlot(rf_ds, col="red", main="feature importance")
rf_ds_train = rf_ds$predicted
table_rf_ds_train = table(rf_ds_train, y_train)
train_error_rate_ds = (table_rf_ds_train[1,2] + table_rf_ds_train[2,1]) / sum(table_rf_ds_train)
train_false_negative_rate_ds = table_rf_ds_train[1,2] / sum(table_rf_ds_train[,2])

rf_ds_pred = predict(rf_ds, x_test)
table_rf_ds_test = table(rf_ds_pred, y_test)
test_error_rate_ds = (table_rf_ds_test[1,2] + table_rf_ds_test[2,1]) / sum(table_rf_ds_test)
test_false_negative_rate_ds = table_rf_ds_test[1,2] / sum(table_rf_ds_test[,2])

table_errorrate = rbind(train_error_rate_without, test_error_rate_without,
                        train_error_rate_ds, test_error_rate_ds)
table_false_neg = rbind(train_false_negative_rate_without, test_false_negative_rate_without,
                        train_false_negative_rate_ds, test_false_negative_rate_ds)
table = cbind(table_errorrate, table_false_neg)
colnames(table) = c("error rate", "false negative rate")
rownames(table) = c("training data without Down-sampling", "test data without Down-sampling",
                    "training data with Down-sampling", "test data with Down-sampling")
```

References

[1] Fred L. Ramsey and Daniel W. Schafer. (2013). *The Statistical Sleuth: A Course in Methods of Data Analysis, Third Edition* . Boston: Brooks/Cole, Cengage Learning.

[2] Michael H. Kutner, Christopher J. Nachtsheim, John Neter and William Li. (2004). *Applied Linear Statistical Models, Fifth Edition*. New York: McGraw-Hill/Irwin

[3] Morris H. DeGroot and Mark J. Schervish . (2012). *Probability and Statistics, Fourth Edition*. Boston: Pearson Education

[4] Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani. (2013). *An Introduction to Statistical Learning with Applications in R*. New York: Springer Science+Business Media

[5] C. Kenrick Hunte. (1996).Controlling Loan Default and Improving the Lending Technology in Credit Institutions. *Savings and Development*. Vol. 20, No.1, 45-59