ECO3080 Project Report

Data Set:

Lending Club

Group Members:

Chunlin SHI

Linge QI

Xinyue CUI

Yingxuan BIAN

1. Introduction

1.1 Background Information

Lending Club was founded in 2006 as a platform for peer-to-peer lending. When the lender applies for a loan from Lending Club, Lending Club will ask the customer to fill in the loan application form online or offline, collect the basic information of the customer, and at the same time make use of the information of the credit investigation institutions of the third-party platform.

"Lending Club" dataset contain complete loan data for all loans issued through the 2007-2015, including the current loan status (Charged Off, Fully Paid, etc.) and latest payment information. Additional features include credit scores, number of finance inquiries, address including zip codes, and state, and collections among others.

1.2 Motivation

Based on these information attributes, by generating models using various machine learning algorithm, we can forecast whether the lender will default, and then suggest Lending Club whether to issue loans to the applicant.

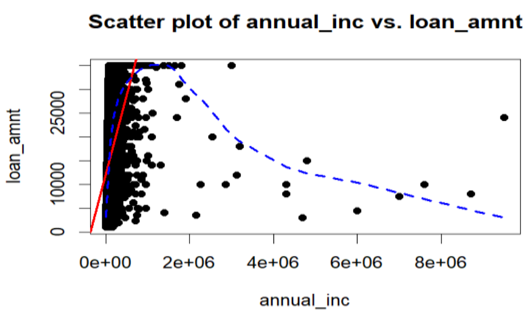
2. Research Questions

Since our aim for the project is to forecast whether the lender will default, the main focus are as follows:

1. Which variables significantly influence the status of loan?
2. Which model(s) perform well in qualitative prediction for this data set?

3. Eye-balling Data Analysis

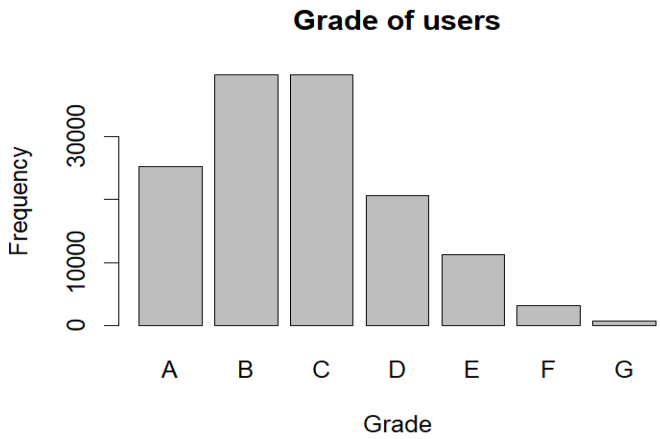
3.1 Annual Income & Loan Amount



Graph 1: Scatter Plot of Annual Income & Loan Amount

By scatter plot, we can find a positive linear relationship between annaul income and loan amount when annual income is smaller than the threshold. Most proportion of the data points lies on the left hand of the threshold. But then, when annual income is larger than the threshold, the relationship becomes negative.

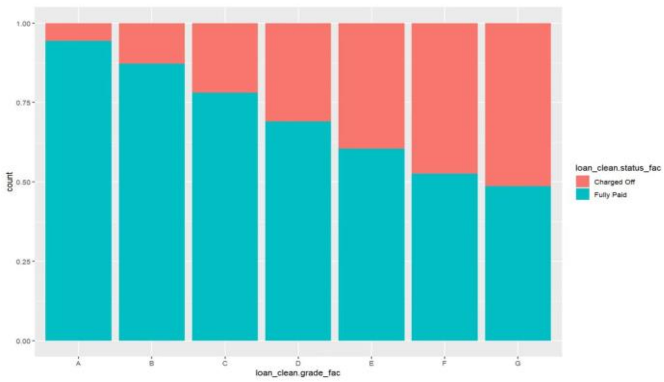
3.2 Distribution of Grade of Users



Graph 2: Histogram of Grades of Users

Grade of users range from A to G with skewness to right. That indicates that most of the users belong to high grades with good credit (i.e., A, B and C).

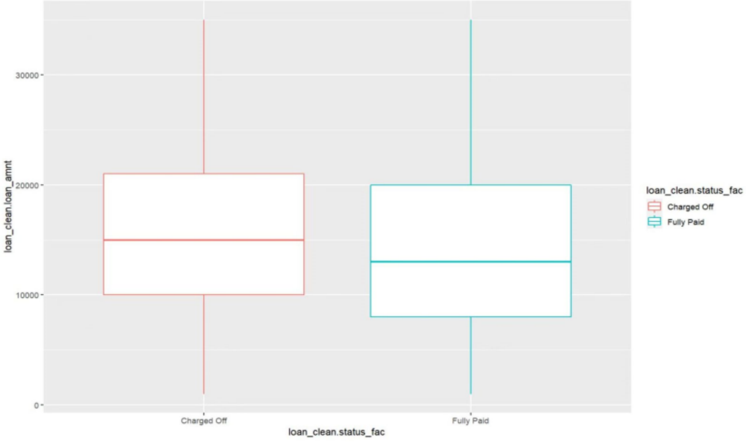
3.3 Loan Status for Users of Different Grades



Graph 3: Histogram of Loan Status for Users of Different Grades

Looking at the histogram, it is clear that higher grade users have lower proportion of "Charged Off" loan status and higher proportion of "Fully Paid" loan status, and lower grade users has higher proportion of "Charged Off" loan status and lower proportion of "Fully Paid" loan status. It shows that the grade brings about a reasonable evaluation of users' credit and loan payment status.

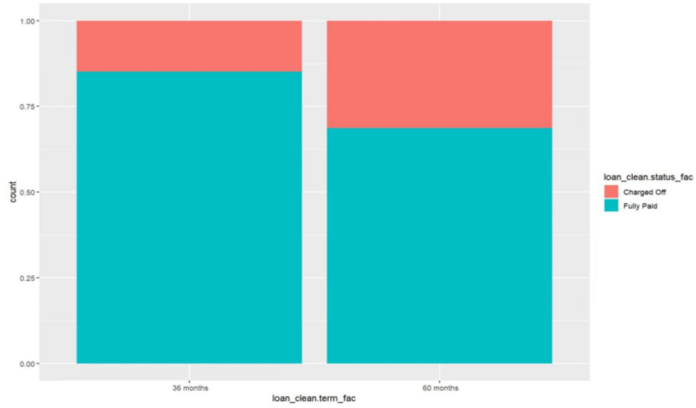
3.4 Loan Status for Users with Different Annual Income



Graph 4: Box Plots of Loan Status for Users with Different Annual Income

According to the box plot, the range between 1st quarter and 3rd quarter as well as the range between largest and the smallest annual income for "Charged Off" and "Fully Paid" users are quite similar. However, the median annual income for those "Charged Off" is higher than those "Fully Paid".

3.5 Loan Status for Users with Different Payment Time



Graph 5: Histogram of Loan Status for Users with Different Payment Time

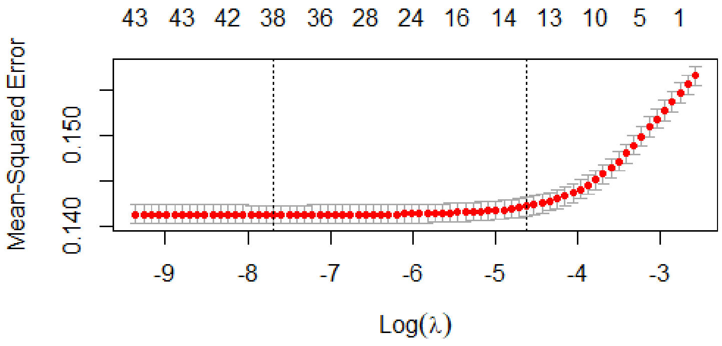
Considering the loan status for users with different payment time, those pay in 36 months has 20% higher rate of "Fully Paid" (i.e., 20% lower rate of "Charged Off") compare with those pay in 60 months. It indicates that payment time is influential to loan status.

4. Machine Learning Methods

Before moving to machine learning methods, we first accomplish data preprocessing and variable selection.

We dropped some variables (id, emp_title, and zip_code) irrelevant to what we want to study. Then we deleted missing values and empty values. After that, we created dummy variables for qualitative variables (term, grade, emp_length, home_ownership, and purpose). We set the training set and test set by random sampling 7-to-3. We chose predictors using Lasso, fico_range_high, delinq_amnt, grade_C, some emp_lengths, some home_ownerships, and some purposes were dropped.

We try to find the best λ using lasso regression.



Graph 6: MSE of Different Lambda

For this dataset, the best lambda is 0.0004604658, and the best k was 11 by train.kknn in the kknn library.

```
Call:
train.kknn(formula = loan_status ~ ., data = lendingclub_train)

Type of response variable: nominal
Minimal misclassification: 0.21987
Best kernel: optimal
Best k: 11
```

4.1 LDA Model

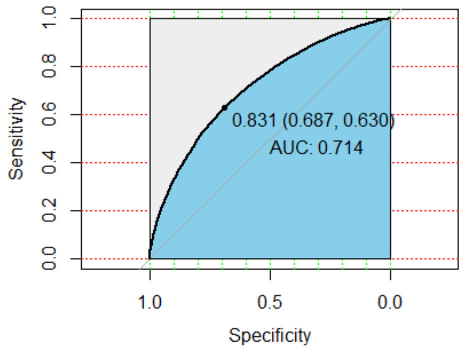
From Table 1 in Appendix, we can see that for the variable grade, as the grade gets lower, their correlations with loan_status decrease from positive to negative. Besides, loan_amnt, dti, delinq_2yrs, etc are negatively related with loan_status, while fico_range_low, total_acc, mort_acc are positively related with loan_status.

Next, we use the test set to see how well this training set performs with LDA.

Table 1 Confusion Matrix of LDA Model

```
      Pred.
True     0     1
   0  1012  6728
   1  1039 30636
```

```
[1] 0.802943
```



Graph 7: ROC Curve of LDA Model

From above, we can conclude that the training accuracy for this model is 0.8029.

4.2 KNN Model

K-Nearest Neighbors is a nonparametric method, which does not make any assumptions about the shape of decision boundary. Therefore, if the decision boundary is highly nonlinear, KNN would be a good approach. However, KNN does not give which predictors are important, so it is impossible to obtain the coefficient estimation table.

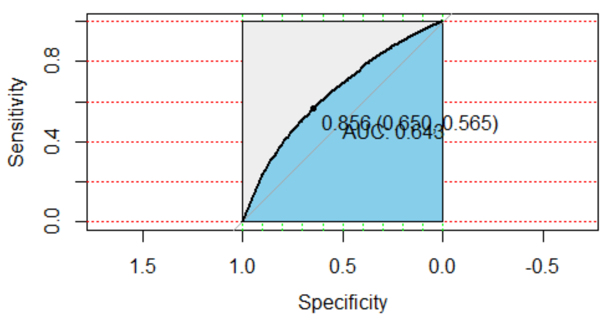
For this dataset, the best k was 11 by train.kknn in the kknn library. Under KNN model, the accuracy was about 77.50%. The confusion matrix was as below:

Table 2 Confusion Matrix of KNN Model

```
      Pred.
True     0     1
   0  1242  6498
   1  2371 29304

[1] 0.7749841
```

From the confusion matrix, false positive rate was too high. We may need to increase the threshold to decrease false positive rate. Under the ROC-AUC curve, it could be concluded that the best choice was to let the threshold be 0.856.



Graph 8: ROC Curve of KNN Model

4.3 Logistic Model

From Table 2 in Appendix, we can see that among those statistically significant terms, the correlation between variable grade loan_status shows a decreasing trend grade level gets lower. Besides, loan_amnt, dti, delinq_2yrs are negatively correlated with loan_status, while fico_range_low, total_acc, mort_acc are positively correlated with loan_status.

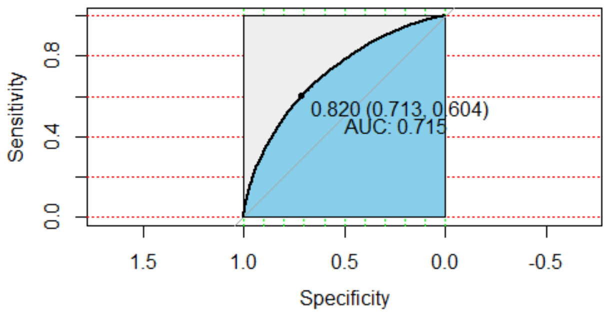
Under the logistic model, the accuracy is about 0.8059.

Table 3 Confusion Matrix of Logistic Model

```
        Pred.
True      0     1
   0    469  7271
   1    380 31295
```

```
[1] 0.8058861
```

From the confusion matrix, false positive rate was too high. We may need to increase the threshold to decrease false positive rate. Under the ROC-AUC curve, it could be seen that the best choice was to let the threshold be 0.820.



Graph 9: ROC Curve of Logistic Model

4.4 Probit Model

According to the Appendix Table 3 summary statistics of the Probit model, dti, fico_range_low, inq_last_6mths, open_acc, bc_util, mort_acc, term_ 36 months, grades, home_ownerships are the most statistically significant. Above all, term_ 36 months and the grades have economically significances. From which it could be seen that comparing with 60 months loans, the repayment rate of 36 months loans is higher. From grade A to G, the borrowers' default rate is getting higher and higher.

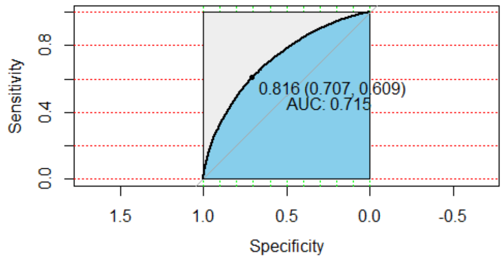
Under the probit model, the accuracy is about 0.8057.

Table 4 Confusion Matrix of Probit Model

```
        Pred.
True      0     1
   0    421  7319
   1    339 31336
```

```
[1] 0.8057085
```

From the confusion matrix, false positive rate was too high. We may need to increase the threshold to decrease false positive rate. Under the ROC-AUC curve, it could be seen that the best choice was to let the threshold be 0.816.



Graph 10: ROC Curve of Probit Model

4.5 Random Forest

Random forest gives an improvement from linear model by allowing interactions among predictors without suffering from the curse of dimensionality. Firstly we do a simple variable selection. 3 variables are excluded from our analysis. The variables “id” and “zip_code” are excluded because they are just randomly assigned identification codes and do not contain useful information for predicting the loan status. The variable “emp_title” is excluded because it contains a huge amount of employment categories which are hard to further classify.

The training and validation set are randomly selected using sample() function in R with the probability train: test = 7:3. After splitting the training and validation set, we conduct random forest method on the training data using randomForest package in R to classify loan status using the rest 19 explanatory variables. We set mtry = 5 (because 5 is close to $\sqrt{p}$), which means that 5 predictors are considered in each split of the tree.

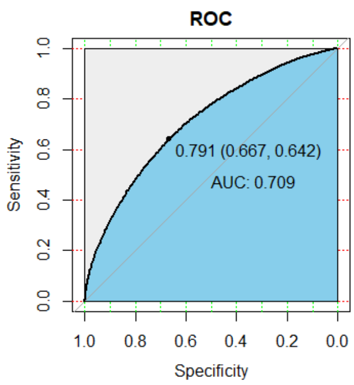
We use the obtained model to make prediction on the test set. The test classification error rate is 0.1940. And the accuracy rate is 0.8060 correspondingly.

Table 5 Confusion Matrix of Random Forest Model

(0: Charged Off; 1: Fully Paid)

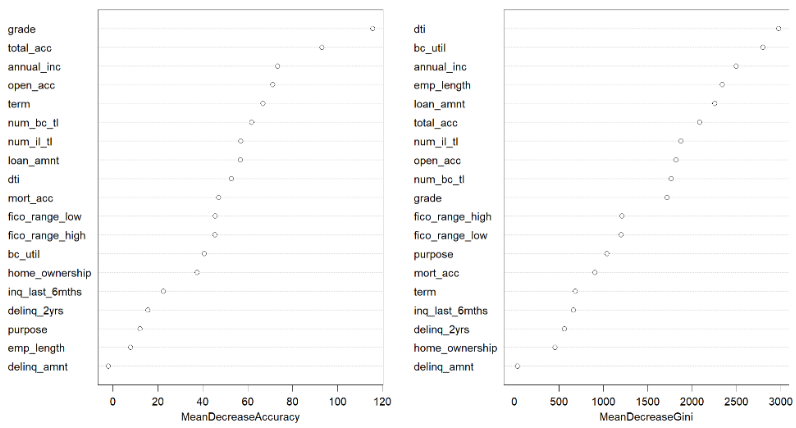
	rf.pred.response	0	1	Total
loan_status				
0		418 (5.5%)	7242 (94.5%)	7660 (100.0%)
1		398 (1.3%)	31331 (98.7%)	31729 (100.0%)
Total		816 (2.1%)	38573 (97.9%)	39389 (100.0%)

The above table shows that in the random forest prediction, Type I error rate (False negative) = 1.3%, Type II error rate (False positive) = 94.5%. Specificity is only 5.5%, indicating that the random forest model behaves poor in predicting the Charged Off cases and tends to misclassify them as Fully Paid ones.



Graph 11: ROC Curve of Random Forest Model

The above plot shows that AUC = 0.709, indicating that the model has enough predicting power. The highest prediction accuracy (0.791) appears at the top left of the curve, when sensitivity equals 0.667 and specificity equals 0.642.



Graph 12: Variable Importance of Random Forest Model

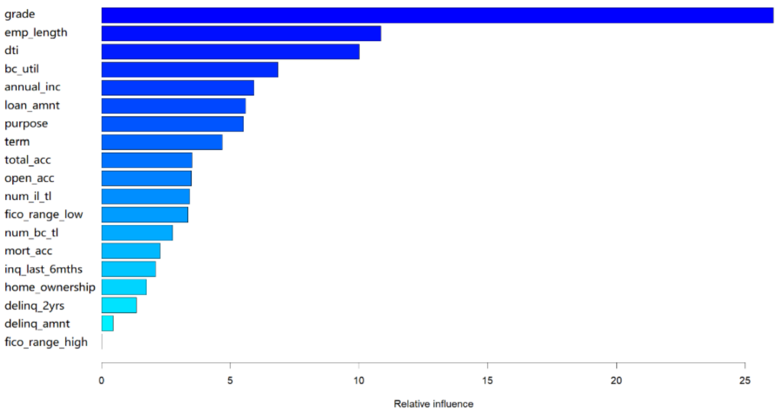
We obtain two different measurements of variable importance. Plot on the left side is based upon the mean decrease of accuracy in predictions on the out of bag samples when a given variable is excluded from the model. The right side is a measure of the total decrease in Gini index that results from splits over that variable averaged over all trees. We observe that loan grade and number of total accounts are the two most important variables according to mean decrease in prediction accuracy; debt to income ratio and ratio of total current balance to high credit/credit limit for all bankcard accounts are the two most important variables according to mean decrease in Gini index.

4.6 Boosting

Boosting method sequentially fits the tree to the residuals of the previously grown trees. It improves prediction accuracy over a simple decision tree by learning slowly. The data pre-processing, variable selection and the split of training and validation set is the same as what we did in the random forest part. Then we fit a boosting model to our data using package gbm in R. We set n.trees =

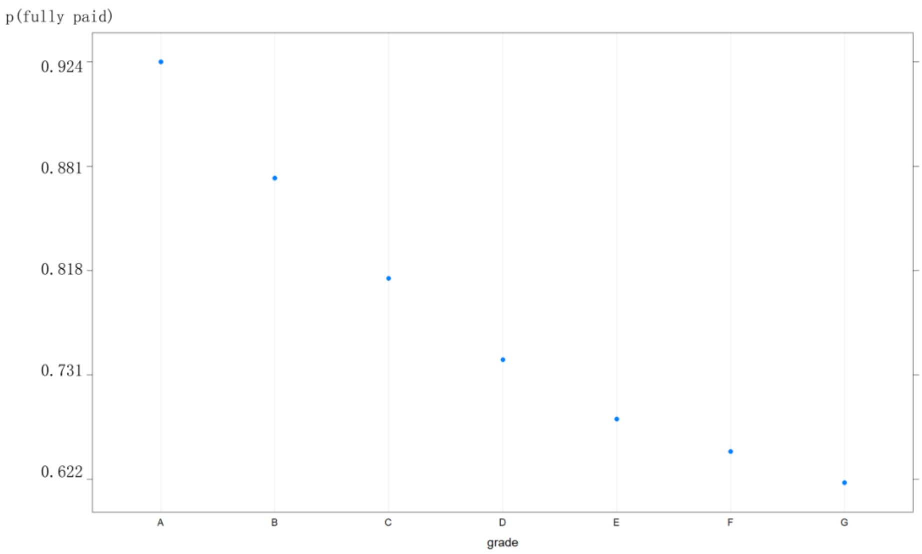
2000 indicating we want 2000 trees and interaction depth = 4 limiting the depth of each tree.

We use the obtained boosting model to make prediction on the test set. The test classification error rate is 0.1952, which is almost the same as that of random forest model (0.1940). The accuracy of prediction by boosting is 0.8048.



Graph 13 Relative influence of boosting model

From the above table and plot, we observe that the relative influence of grade dominates the others. The results indicate that loan grade is the most important variable for predicting loan status in our boosting model. Then we produce partial importance plot of the variable grade to illustrate the marginal effect of loan grade on loan status keeping the other variables fixed.



Graph 14: Partial importance of variable grade.

As the client's loan grade changes from A to G, his/her probability of fully pay the loan monotonically decreases from 0.924 to 0.622 on average, keeping other explanatory variables fixed.

5. Results and Findings

The project has two main results. One of them is the importance level of different independent variables on dependent variable. And another one is the performance of various machine learning methods.

Considering the independent variables that significantly influence the dependent variable (in our case is "loan_status"), the variable "grade" and "dti" are selected. The result can be supported by significant level of logistic and probit model, as well as the measurement of importance of random forest and boosting model. By referring to the tables of summary of coefficients, it is shown that "grade" and "dti" are at 0.001 significant level, which is quite low. And by referring to the graphs of importance of variables, "grade" and "dti" are always ranking at top.

Comparing the six machine learning methods we used, random forest has the largest prediction accuracy, while the KNN has lowest prediction accuracy. Also, we notice that the difference between random forest, logistic and probit model are within 0.0005, which is quite small. Therefore, to answer the second research question, for qualitative prediction with this data set, especially the loan status, it is suggested to consider using methods like random forest, logistic and probit.

Table 6: Test Accuracy of Different Machine Learning Methods (By Descending Order)

Machine Learning Method	Accuracy
Random Forest	0.8060
Logistic	0.8059
Probit	0.8057
Boosting	0.8048
LDA	0.8029
KNN	0.7750

The project has two main findings. Firstly, by using machine learning methods properly, the company can interpret the loan status (charged off or fully paid) about 8 times out of 10 on average. Secondly, "grade", also known as assigned loan grade, is an effective indicator for the company to refer to, as it shows the credibility of users.

6. Conclusion

Though machine learning methods are helpful, we have to do trade-off in many cases. In this project, random forest shows the highest accuracy but it is slow when computation power is limited, and also interpretability is weaker than logistic model.

It is crucial for people to choose suitable methods according to various scenarios to achieve the goals effectively and efficiently.

Appendix

Table 1: Summary Statistics of LDA Model

```
Call:
lda(loan_status ~ ., data = lendingclub_train)

Prior probabilities of groups:
        0         1 
0.1943828 0.8056172 

Group means:
  loan_amnt annual_inc      dti delinq_2yrs fico_range_low inq_last_6mths open_acc total_acc  bc_util mort_acc num_bc_tl num_il_tl
0  16266.67   72763.95 20.93691   0.3868658       686.0038      0.7281982 12.56906  25.90351 64.89945 1.411031  8.187671  9.149130
1  14993.44   79590.59 18.52300   0.3417419       695.5194      0.5396742 11.92636  25.49912 61.27277 1.736959  8.165364  8.658056
  `term_ 36 months` `term_ 60 months`    grade_A   grade_B   grade_D    grade_E    grade_F     grade_G `emp_length_1 year`
0         0.5099290         0.4900710 0.04844213 0.1792806 0.2264362 0.16177211 0.05521061 0.013089445          0.07400571
1         0.7363108         0.2636892 0.21383164 0.3081346 0.1245495 0.05934594 0.01491409 0.003009812          0.06788949
  `emp_length_2 years` `emp_length_4 years` `emp_length_8 years` `emp_length_9 years` `emp_length_10+ years`
0           0.09749958           0.06399284           0.05599373           0.04268054              0.3399340
1           0.09379007           0.06120851           0.05381220           0.04275823              0.3582216
  home_ownership_MORTGAGE home_ownership_RENT purpose_credit_card purpose_home_improvement purpose_house purpose_major_purchase
0               0.4273648           0.4709963           0.1953907               0.05610561   0.004027521             0.01700509
1               0.5082399           0.3896425           0.2505028               0.06003428   0.003279751             0.01734354
  purpose_moving purpose_other purpose_renewable_energy purpose_small_business purpose_vacation
0    0.006600660    0.04654025             0.0011187559            0.012026626      0.004642837
1    0.005331282    0.04539013             0.0003914106            0.007517782      0.005479748
```

```
Coefficients of linear discriminants:
                                   LD1
loan_amnt                -3.389106e-06
annual_inc                1.102198e-07
dti                      -2.087244e-02
delinq_2yrs              -2.890499e-02
fico_range_low            6.241348e-03
inq_last_6mths           -1.113595e-01
open_acc                 -2.169028e-02
total_acc                 8.542401e-03
bc_util                   2.485974e-03
mort_acc                  5.178587e-02
num_bc_tl                -1.182061e-02
num_il_tl                -2.422912e-03
`term_ 36 months`         3.851454e-01
`term_ 60 months`        -3.851454e-01
grade_A                   8.472397e-01
grade_B                   5.693206e-01
grade_D                  -5.691449e-01
grade_E                  -1.114513e+00
grade_F                  -1.620438e+00
grade_G                  -1.916094e+00
`emp_length_1 year`      -3.973401e-02
`emp_length_2 years`     -4.403688e-03
`emp_length_4 years`     -2.870904e-02
`emp_length_8 years`     -3.563930e-02
`emp_length_9 years`     -4.808005e-03
`emp_length_10+ years`    1.132750e-02
home_ownership_MORTGAGE   1.134311e-01
home_ownership_RENT      -2.354263e-01
purpose_credit_card       6.439362e-02
purpose_home_improvement -9.800315e-02
purpose_house             4.338874e-01
purpose_major_purchase   -6.939907e-02
purpose_moving            9.287527e-02
purpose_other             1.512949e-01
purpose_renewable_energy -1.531470e+00
purpose_small_business   -1.426542e-01
purpose_vacation          2.259183e-01
```

Table 2: Summary Statistics of Logistic Model

```
Call:
glm(formula = loan_status ~ ., family = binomial, data = lendingclub_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1079   0.2934   0.5009   0.6922   1.5452

Coefficients: (1 not defined because of singularities)
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)              -2.903e+00  3.039e-01  -9.552  < 2e-16 ***
loan_amnt                -4.079e-06  1.314e-06  -3.104  0.00191 **
annual_inc                3.260e-07  2.083e-07   1.565  0.11763
dti                      -1.648e-02  1.160e-03 -14.208  < 2e-16 ***
delinq_2yrs              -2.196e-02  9.143e-03  -2.402  0.01629 *
fico_range_low            6.147e-03  4.240e-04  14.499  < 2e-16 ***
inq_last_6mths           -8.271e-02  1.002e-02  -8.252  < 2e-16 ***
open_acc                 -1.760e-02  2.346e-03  -7.500 6.38e-14 ***
total_acc                 7.380e-03  2.259e-03   3.267  0.00109 **
bc_util                   2.051e-03  3.813e-04   5.378 7.55e-08 ***
mort_acc                  4.759e-02  6.380e-03   7.459 8.69e-14 ***
num_bc_tl                -1.057e-02  3.456e-03  -3.059  0.00222 **
num_il_tl                -2.651e-03  2.410e-03  -1.100  0.27145
`term_ 36 months`         5.575e-01  2.201e-02  25.328  < 2e-16 ***
`term_ 60 months`                NA         NA      NA       NA
grade_A                   1.172e+00  4.198e-02  27.913  < 2e-16 ***
grade_B                   5.000e-01  2.529e-02  19.770  < 2e-16 ***
grade_D                  -3.266e-01  2.498e-02 -13.075  < 2e-16 ***
grade_E                  -5.704e-01  3.048e-02 -18.712  < 2e-16 ***
grade_F                  -7.885e-01  4.921e-02 -16.025  < 2e-16 ***
grade_G                  -9.137e-01  9.777e-02  -9.345  < 2e-16 ***
`emp_length_1 year`      -3.505e-02  3.604e-02  -0.973  0.33070
`emp_length_2 years`     -2.366e-03  3.224e-02  -0.073  0.94149
`emp_length_4 years`     -2.072e-02  3.818e-02  -0.543  0.58736
`emp_length_8 years`     -2.561e-02  4.048e-02  -0.633  0.52689
`emp_length_9 years`     -5.990e-03  4.529e-02  -0.132  0.89477
`emp_length_10+ years`    9.783e-03  2.218e-02   0.441  0.65912
home_ownership_MORTGAGE   1.002e-01  3.137e-02   3.195  0.00140 **
home_ownership_RENT      -1.770e-01  3.099e-02  -5.712 1.12e-08 ***
purpose_credit_card       5.539e-02  2.288e-02   2.421  0.01549 *
purpose_home_improvement -8.953e-02  3.947e-02  -2.268  0.02332 *
purpose_house             2.332e-01  1.409e-01   1.655  0.09784 .
purpose_major_purchase   -6.497e-02  6.859e-02  -0.947  0.34353
purpose_moving            3.619e-02  1.100e-01   0.329  0.74224

purpose_other             1.062e-01  4.267e-02   2.490  0.01278 *
purpose_renewable_energy -9.650e-01  3.049e-01  -3.165  0.00155 **
purpose_small_business   -1.444e-01  8.594e-02  -1.680  0.09295 .
purpose_vacation          1.732e-01  1.251e-01   1.384  0.16625
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 90591  on 91967  degrees of freedom
Residual deviance: 81380  on 91931  degrees of freedom
AIC: 81454

Number of Fisher Scoring iterations: 5
```

Table 3: Summary Statistics of Probit Model

(Intercept)	-1.590e+00	1.683e-01	-9.448	< 2e-16	***
loan_amnt	-2.008e-06	7.354e-07	-2.731	0.006320	**
annual_inc	1.020e-07	1.046e-07	0.975	0.329726	
dti	-9.655e-03	6.611e-04	-14.604	< 2e-16	***
delinq_2yrs	-1.395e-02	5.275e-03	-2.644	0.008189	**
fico_range_low	3.476e-03	2.338e-04	14.869	< 2e-16	***
inq_last_6mths	-4.784e-02	5.838e-03	-8.195	2.51e-16	***
open_acc	-9.711e-03	1.346e-03	-7.215	5.40e-13	***
total_acc	4.234e-03	1.288e-03	3.287	0.001013	**
bc_util	1.174e-03	2.177e-04	5.391	7.00e-08	***
mort_acc	2.651e-02	3.569e-03	7.427	1.11e-13	***
num_bc_tl	-6.453e-03	1.968e-03	-3.279	0.001043	**
num_il_tl	-1.488e-03	1.376e-03	-1.081	0.279832	
`term_ 36 months`	3.241e-01	1.272e-02	25.470	< 2e-16	***
`term_ 60 months`	NA	NA	NA	NA	
grade_A	5.958e-01	2.098e-02	28.398	< 2e-16	***
grade_B	2.754e-01	1.393e-02	19.770	< 2e-16	***
grade_D	-1.960e-01	1.478e-02	-13.261	< 2e-16	***
grade_E	-3.506e-01	1.839e-02	-19.068	< 2e-16	***
grade_F	-4.888e-01	3.023e-02	-16.170	< 2e-16	***
grade_G	-5.693e-01	6.061e-02	-9.393	< 2e-16	***
`emp_length_1 year`	-1.774e-02	2.062e-02	-0.860	0.389554	
`emp_length_2 years`	-5.945e-04	1.837e-02	-0.032	0.974179	
`emp_length_4 years`	-1.366e-02	2.175e-02	-0.628	0.530140	
`emp_length_8 years`	-1.205e-02	2.309e-02	-0.522	0.601638	
`emp_length_9 years`	-5.238e-03	2.572e-02	-0.204	0.838590	
`emp_length_10+ years`	7.769e-03	1.258e-02	0.617	0.536931	
home_ownership_MORTGAGE	5.979e-02	1.776e-02	3.368	0.000758	***
home_ownership_RENT	-9.836e-02	1.765e-02	-5.573	2.50e-08	***
purpose_credit_card	3.212e-02	1.280e-02	2.510	0.012076	*
purpose_home_improvement	-4.903e-02	2.231e-02	-2.197	0.028000	*
purpose_house	1.343e-01	8.157e-02	1.647	0.099585	.
purpose_major_purchase	-3.096e-02	3.899e-02	-0.794	0.427172	
purpose_moving	2.076e-02	6.376e-02	0.326	0.744764	
purpose_other	6.543e-02	2.436e-02	2.686	0.007223	**
purpose_renewable_energy	-5.670e-01	1.863e-01	-3.044	0.002333	**
purpose_small_business	-7.652e-02	5.084e-02	-1.505	0.132266	
purpose_vacation	1.020e-01	7.011e-02	1.455	0.145750	

Table 4: Variable Importance of Random Forest Model
(0: Charged Off; 1: Fully Paid)

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
loan_amnt	-17.8155832	58.465643	56.620911	2254.49090
term	36.5076567	50.564270	66.773509	684.74425
grade	58.9255391	85.862582	115.573023	1716.92208
emp_length	-8.2536109	13.576409	7.854659	2340.45226
home_ownership	-9.3324355	41.937472	37.403221	455.49390
annual_inc	-39.3106032	87.155269	73.206722	2494.36719
purpose	-2.0157182	14.177151	12.037109	1039.59571
dti	2.4415013	55.876855	52.717942	2972.82358
delinq_2yrs	-0.5565630	17.087645	15.488559	559.53053
fico_range_low	-2.8369655	47.065346	45.511475	1202.23696
fico_range_high	-5.2208562	47.709003	45.280476	1209.42286
inq_last_6mths	6.0513664	21.113690	22.468492	661.08571
open_acc	-30.0723578	79.645990	71.046443	1817.50034
total_acc	-56.7344197	107.782212	93.088364	2084.01818
bc_util	-17.7700851	51.327220	40.650180	2793.81918
delinq_amnt	-0.6682237	-1.750276	-1.960017	33.67667
mort_acc	-20.1720139	54.549365	47.044894	903.94375
num_bc_tl	-39.5562012	75.626049	61.673428	1764.82510
num_il_tl	-33.8650694	72.724633	56.813922	1871.97779

Table 5: Variable Importance of Boosting

	var	rel.inf
grade	grade	26.0915058
emp_length	emp_length	10.8472085
dti	dti	10.0138571
bc_util	bc_util	6.8529953
annual_inc	annual_inc	5.9206912
loan_amnt	loan_amnt	5.5955601
purpose	purpose	5.5174031
term	term	4.6884604
total_acc	total_acc	3.5137305
open_acc	open_acc	3.4844572
num_il_tl	num_il_tl	3.4209070
fico_range_low	fico_range_low	3.3526404
num_bc_tl	num_bc_tl	2.7568446
mort_acc	mort_acc	2.2806624
inq_last_6mths	inq_last_6mths	2.1013971
home_ownership	home_ownership	1.7479902
delinq_2yrs	delinq_2yrs	1.3617458
delinq_amnt	delinq_amnt	0.4519434
fico_range_high	fico_range_high	0.0000000

Table 6: Summary Statistics of Probit

```
Call:
glm(formula = loan_status ~ ., family = binomial(link = probit), 
    data = lendingclub_train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.9680   0.2840   0.5010   0.6957   1.5189  

Coefficients: (1 not defined because of singularities)
                              Estimate Std. Error z value Pr(>|z|)
```

Codes

```
##############################Preprocess################################
#Drop useless columns
library(dplyr)
lendingclub=select(lendingclub,-c(id,emp_title,zip_code))

#Drop NA
sum(is.na(lendingclub))
lendingclub=na.omit(lendingclub)
lendingclub=subset(lendingclub,emp_length!='')

#Create dummy variables
library(fastDummies)
lendingclub=dummy_cols(lendingclub,select_columns=c('term','grade',
                                   'emp_length',
                                   'home_ownership','purpose'))
lendingclub=select(lendingclub,-c(term,grade,emp_length,home_ownership,purpose))

#Convert dependent variable to 0/1 classified variable
lendingclub$loan_status[lendingclub$loan_status=="Charged Off"]=0
lendingclub$loan_status[lendingclub$loan_status=="Fully Paid"]=1
lendingclub$loan_status=as.numeric(lendingclub$loan_status)

#Split the data set into training set and test set
set.seed(3080)
train=sample(1:nrow(lendingclub),round(0.7*nrow(lendingclub)),replace=FALSE)
lendingclub_train=lendingclub[train,]
lendingclub_test=lendingclub[-train,]

#Choose predictors using Lasso
X=model.matrix(loan_status~.,lendingclub)[,-1]
Y=lendingclub$loan_status
```

```
library(glmnet)
lasso.mod=glmnet(X,Y,alpha=1,lambda=5)
summary(lasso.mod)
plot(lasso.mod)
lasso.mod$beta
lasso.mod$a0

cv.out=cv.glmnet(X[train,],Y[train],alpha=1)
plot(cv.out)
bestlambda=cv.out$lambda.min
bestlambda

lasso.pred=predict(lasso.mod,s=bestlambda,newx=X[-train,])
mean((lasso.pred-Y[-train])^2)
out=glmnet(X,Y,alpha=1)
predict(out,type="coefficients",s=bestlambda)

#Drop irrelevant variables
lendingclub=select(lendingclub,-c(fico_range_high,delinq_amnt,grade_C,
                    `emp_length_< 1 year`,`emp_length_3 years`,
                    `emp_length_5 years`,`emp_length_6 years`,
                    `emp_length_7 years`,home_ownership_ANY,
                    home_ownership_OWN,purpose_car,
                    purpose_debt_consolidation,purpose_medical,
                    purpose_wedding))
lendingclub_train=select(lendingclub_train,-c(fico_range_high,
                          delinq_amnt,grade_C,
                          `emp_length_< 1 year`,
                          `emp_length_3 years`,
                          `emp_length_5 years`,
                          `emp_length_6 years`,
                          `emp_length_7 years`,
                          home_ownership_ANY,
```

```
                                    home_ownership_OWN,purpose_car,
                                    purpose_debt_consolidation,
                                    purpose_medical,purpose_wedding))
lendingclub_test=select(lendingclub_test,-c(fico_range_high,delinq_amnt,grade_C,
                                  `emp_length_< 1 year`,
                                  `emp_length_3 years`,
                                  `emp_length_5 years`,
                                  `emp_length_6 years`,
                                  `emp_length_7 years`,
                                  home_ownership_ANY,
                                  home_ownership_OWN,purpose_car,
                                  purpose_debt_consolidation,
                                  purpose_medical,purpose_wedding))
library(pROC)

################################LDA################################
library(MASS)
lda=lda(loan_status~., data=lendingclub_train)
lda
lda_pred=predict(lda, lendingclub_test)
names(lda_pred)
lda_class=lda_pred$class
table(lendingclub_test$loan_status,lda_class,dnn=c("True","Pred."))
mean(lda_class==lendingclub_test$loan_status)
plot(roc(lendingclub_test$loan_status,lda_pred$posterior[,2]),print.auc=TRUE,
    auc.polygon=TRUE,grid=c(0.1,0.2),grid.col=c("green","red"),
    max.auc.polygon=TRUE,auc.polygon.col="skyblue",print.thres=TRUE)

################################KNN################################
lendingclub$loan_status=factor(lendingclub$loan_status)
lendingclub_train$loan_status=factor(lendingclub_train$loan_status)
lendingclub_test$loan_status=factor(lendingclub_test$loan_status)
library(kknn)
```

```
knn=train.kknn(loan_status~.,lendingclub_train)
summary(knn)

knn.mod=kknn(loan_status~.,lendingclub_train,lendingclub_test,k=11)
knn_pred=fitted(knn.mod)
table(lendingclub_test$loan_status,knn_pred,dnn=c("True","Pred."))
mean(knn_pred==lendingclub_test$loan_status)
plot(roc(lendingclub_test$loan_status,knn.mod$prob[,2]),print.auc=TRUE,
    auc.polygon=TRUE,grid=c(0.1,0.2),grid.col=c("green","red"),
    max.auc.polygon=TRUE,auc.polygon.col="skyblue",print.thres=TRUE)

##########################Logit & Probit############################
logitreg=glm(loan_status~., data=lendingclub_train, family=binomial(link=logit))
summary(logitreg)
logit_probs=predict(logitreg,lendingclub_test,type="response")
logit_pred=rep(0,39415)
logit_pred[logit_probs>0.5]=1
table(lendingclub_test$loan_status,logit_pred,dnn=c("True","Pred."))
mean(logit_pred==lendingclub_test$loan_status)
plot(roc(lendingclub_test$loan_status,logit_probs),print.auc=TRUE,
    auc.polygon=TRUE,grid=c(0.1,0.2),grid.col=c("green","red"),
    max.auc.polygon=TRUE,auc.polygon.col="skyblue",print.thres=TRUE)

probitreg=glm(loan_status~., data=lendingclub_train, family=binomial(link=probit))
summary(probitreg)
probit_probs=predict(probitreg,lendingclub_test,type="response")
probit_pred=rep(0,39415)
probit_pred[probit_probs>0.5]=1
table(lendingclub_test$loan_status,probit_pred,dnn=c("True","Pred."))
mean(probit_pred==lendingclub_test$loan_status)
plot(roc(lendingclub_test$loan_status,probit_probs),print.auc=TRUE,
    auc.polygon=TRUE,grid=c(0.1,0.2),grid.col=c("green","red"),
    max.auc.polygon=TRUE,auc.polygon.col="skyblue",print.thres=TRUE)
```

```
########################RandomForest##############################
# delete missing value in the data:
install.packages("dplyr")
library(dplyr)
loan<-na_if(loan,"")
loan<-na.omit(loan)

# change data type:
loan$term<-as.factor(loan$term)
loan$grade<-as.factor(loan$grade)
loan$emp_length<-as.factor(loan$emp_length)
loan$home_ownership<-as.factor(loan$home_ownership)
loan$loan_status<-dplyr::recode(loan$loan_status,"Charged Off"=0,"Fully Paid"=1)
loan$loan_status<-as.factor(loan$loan_status)
loan$purpose<-as.factor(loan$purpose)
install.packages("randomForest")
library(randomForest)

# split training and test sample
set.seed(3080)
train <- sample(1:nrow(loan), 100000)
loan.test <- loan[-train,]

# random forest
set.seed(3080)
rf.loan <- randomForest(loan_status ~ loan_amnt + term + grade + emp_length +
home_ownership + annual_inc +
              purpose + dti + delinq_2yrs + fico_range_low + fico_range_high +
inq_last_6mths +
              open_acc + total_acc + bc_util + delinq_amnt + mort_acc +
num_bc_tl + num_il_tl,
             data = loan, subset = train,
             mtry = 5, importance = TRUE)
```

```
# confusion matrix:
install.packages("MASS")
library(MASS)
rf.pred.response <- predict(rf.loan, loan.test, type = "response")
table(rf.pred.response,loan.test$loan_status)
install.packages("summarytools")
library(summarytools)
summarytools::ctable(loan.test$loan_status,rf.pred.response)

# variable importance:
importance(rf.loan)
varImpPlot(rf.loan)

# classification error rate:
1 - mean(rf.pred.response == loan.test$loan_status)

# ROC Curve:
rf.pred.prob <- predict(rf.loan, loan.test, type = "prob")
library(pROC)
rf_roc <- roc(loan.test$loan_status, rf.pred.prob[,2])
plot(rf_roc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),
     grid.col = c("green", "red"), max.auc.polygon = TRUE,
     auc.polygon.col = "skyblue", print.thres = TRUE, main = 'ROC')

################################Boosting##############################
loan<-Data1_loan
library(dplyr)
loan<-na_if(loan,"")
loan<-na.omit(loan)
loan$term<-as.factor(loan$term)
loan$grade<-as.factor(loan$grade)
loan$emp_length<-as.factor(loan$emp_length)
```

```
loan$home_ownership<-as.factor(loan$home_ownership)
loan$purpose<-as.factor(loan$purpose)
loan$loan_status<-as.factor(loan$loan_status)
loan$loan_status<-as.logical(loan$loan_status)

# split the data set into training and test set
set.seed(3080)
train <- sample(1:nrow(loan), round(0.7*nrow(loan)))
test <- loan[-train,]

library(gbm)
set.seed(3080)
boost.loan <- gbm(loan_status ~ loan_amnt + term + grade + emp_length +
home_ownership + annual_inc +
          purpose + dti + delinq_2yrs + fico_range_low + fico_range_high +
inq_last_6mths +
          open_acc + total_acc + bc_util + delinq_amnt + mort_acc + num_bc_tl +
num_il_tl,
         data = loan[train,], distribution = "bernoulli",
         n.trees = 2000, interaction.depth = 4, verbose = T)
summary(boost.loan)
plot(boost.loan,i="grade")
plot(boost.loan,i="dti")
plot(boost.loan,i="emp_length")
boost.pred <- predict(boost.loan, newdata=loan[-train,])
boost.pred.logit<-exp(boost.pred)/(1+exp(boost.pred))
boost.pred.logit<-cut(boost.pred.logit,breaks=c(0,0.5,1))
boost.pred.logit<-recode(boost.pred.logit,"(0,0.5]"=FALSE,"(0.5,1]"=TRUE)
table(boost.pred.logit)
1 - mean(boost.pred.logit == test$loan_status)
```