2 Data Preparation

2.1 Obtain the data

We download the data for all loans issued through the 2007-2015 including the current loan status and latest payment information from Lending Club.

<https://www.kaggle.com/wendykan/lending-club-loan-data>

2.2 Preprocess the data

The original data’s dimension is 887383×75, and it desperately needs some cleaning before becoming ready to be further modeled. The R code we used to do data pre-processing is also attached in Appendix A!!!

The first thing we did is to redefine the loan status, our Y labels, into True (those who are under “Current”, “Fully Paid”, “In Grace Period”, “Does not meet the credit policy. Status: Fully Paid”) and False (those who are under “Charged off”, “Late(31-120 days)”, “Late(16-30 days)”, “Default”, “Does not meet the credit policy. Status: Charged Off”)classes so that we could simply the problem into a binary situation.

The second most important thing is to reduce the number of features. We start by deleting all the goal-irrelevant and redundant features. For example, information for administrative purpose like “member\_id” and “issuded\_date”, “next\_due\_time” are dropped. And we also delete columns like “grade”, since all the information is contained in subgrade. Secondly, the features have the same values across all cases are dropped, like “policy\_code” and “pymnt plan”.

Thirdly, we removed all the a little bit more than five hundreds joint applicants from the input and dropped the redundant “application\_type” feature.

Fourthly, we discarded all the features with more than 50% NA in them.

Fifthly, we convert the “desc” feature from string to the count of the string length.

Sixthly, we removed all rows with NA in, including those have abnormal 999 and 9999 values, which statistically meant to represent NA.

Seventhly, we remove the numeric features having high covariance (>0.95) as well as those who are responsible for off-the-roof (>100) VIF (Variance Inflation Factor) with caution. Because some of our models are relative not sensitive to the correlations between features while the others are. We’d like to compare them as well on this front.

Finally, we normalized all numeric data to 0-1 range for simplicity.

Additional treatment for logistic regression and its boosting, we expanded all the categorical to design matrix of each sublevels with dummy variable to better cope with the package.

In the end, we have a ready-to-use input data with dimension 877,860 × 32 (877,860 × 141 for logistic regression and its boosting), the list of kept features are listed in Appendix B!!!! The processed data is imbalanced, with approximately 95% true cases as dominating majority and 5% false cases as minority as the Figure. 1 Number shows.

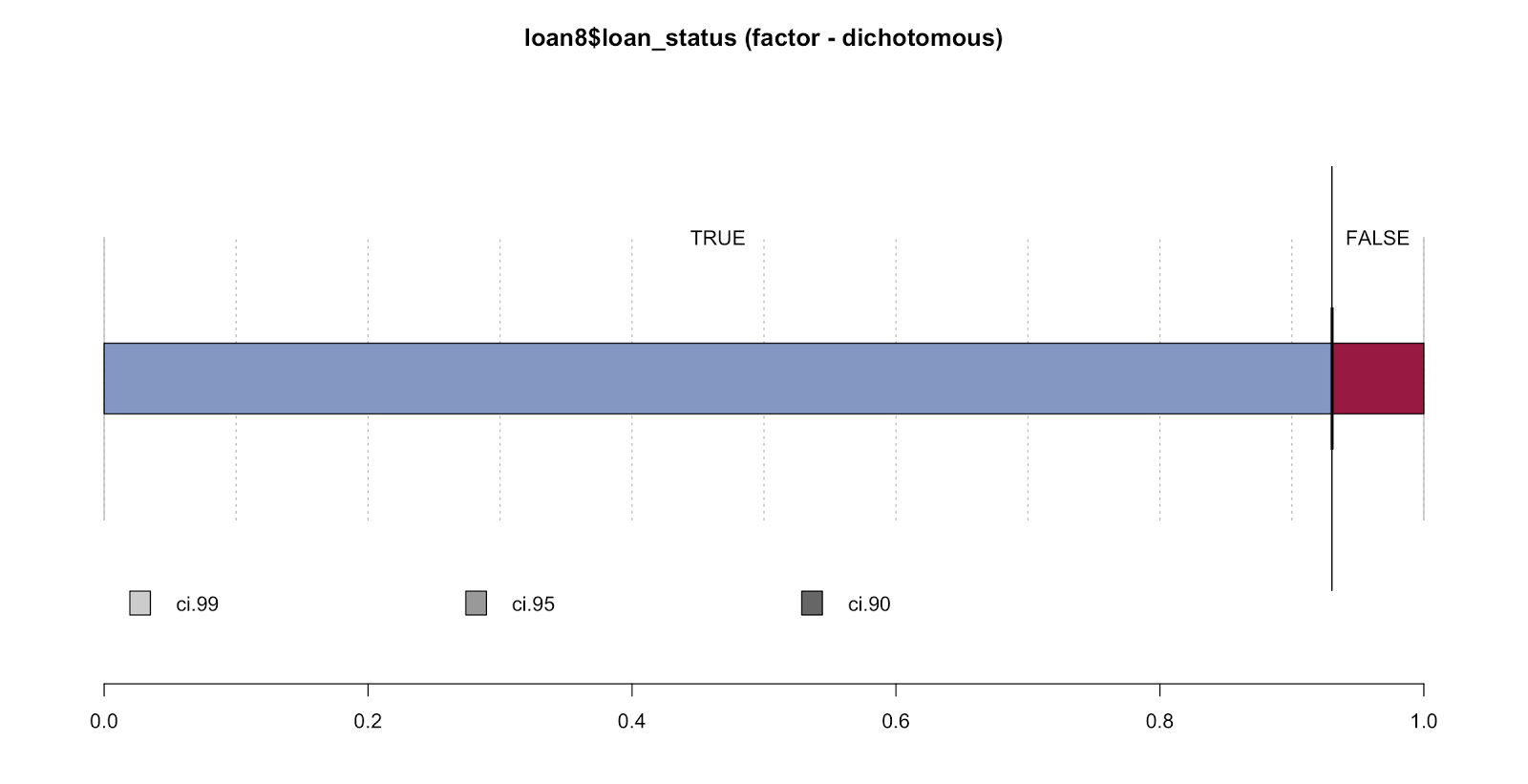


Figure 1. The distribution of true and false classes in the processed dataset.

Classification methods and results

Logistic Regression

As a classic categorical classifier, logistic regression is a good alternative to LDA (linear discriminant analysis). Binary logistic regression is clearly solid choice to deal with this specific dataset. It estimates the conditional probability that a characteristic is present given the values of explanatory random variables:

For this case, means the prediction that the borrower will pay back the loan, means the prediction that the borrower will default.

The data is split into 80% as training and 20% as testing, ensured by a 5-fold cross validation. We adopted logistic regression with L1 regularization (LASSO) on the dataset hoping to achieve better results. This is because when we use the expanded design matrix as input, there are in total 141 variables to train. And some of them are quite weakly associated with the predicted results, such as the “addr\_state”. So by L1 regularization, we could choose the features more smartly by avoiding overfitting on the training set. As we known, logistic regression is not very sensitive to the high correlations between features, which is also proved in this case. But it is affected greatly by the imbalanced data distribution. We didn’t notice this until we saw the result under default setting where cut-off threshold is 0.5, meaning a probability higher than 0.5 will be classified as positive, and a probability lower than 0.5 will be classified as negative. The confusion matrix and performance matrix where 0.5 is chosen as the discriminative cut-off values is shown:

Confusion Matrix when cut-off = 0.5

|  |  |  |
| --- | --- | --- |
|  | Reference: False | Reference: True |
| Prediction: Negative | 646 | 5181 |
| Prediction: Positive | 11707 | 158038 |

Performance Matrix when cut-off = 0.5

|  |  |  |
| --- | --- | --- |
|  | Reference: False | Reference: True |
| Precision | 0.111 | 0.931 |
| Recall | 0.052 | 0.968 |
| F-score | 0.071 | 0.949 |
| Accuracy | 0.904 | |

According to the above matrices, the models works beautifully for the true cases, regardless the precision, recall or F-score, the accuracy is also high. Nevertheless, the performance for the false case seems dreadful in comparison. Given our research goal, it is far from ideal. By theory, this discrepancy in prediction powers for two cases are caused by the imbalance of our data. Although a more popular way to address this issue is to over sample the minority groups until it is in balance with the previous majority group again, given the limited time and resources allocated, a tuning over the discriminative cut-off threshold is a more feasible strategy for us. Thus, we proceeded with the ROC (Receiver Operating Characteristic) curve plotting (shown in Figure 2). ROC is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. Ideally, when the false positive rate is 0, the true positive rate is already 1. From the curve, we took the closet point to (0,1) as our optimal performance point, which corresponds to a discrimination threshold equal to 0.924. Eventually, we adopt this optimal cutoff threshold acquired from ROC curve into our logistic regression with L1 regularization with 5-fold cross validation, and the result is listed in the following two matrices.

Confusion Matrix when cut-off = 0.924

|  |  |  |
| --- | --- | --- |
|  | Reference: False | Reference: True |
| Prediction: Negative | 10066 | 15306 |
| Prediction: Positive | 2390 | 147810 |

Performance Matrix when cut-off = 0.924

|  |  |  |
| --- | --- | --- |
|  | Reference: False | Reference: True |
| Precision | 0.397 | 0.984 |
| Recall | 0.808 | 0.906 |
| F-score | 0.532 | 0.943 |
| Accuracy | 0.899 | |

We could see without any loss on performances regarding the true cases and overall accuracy, there is a considerable improvement on the prediction regarding the false cases due to the tuning of cutoff threshold. In addition, we can check the relative importance of those features as a result of L1 regularization, as shown in Figure 3. Without surprises, the loan amount, payment, interest rate are vital when it comes to predict the reliability of a borrower, while some other seems-to-be important features like home ownership and so forth are not quite influential after L1 regularization. Meanwhile, one interesting feature is the not mandatorily required description seems to be more influential to the prediction than the length of employment and a slew of others.

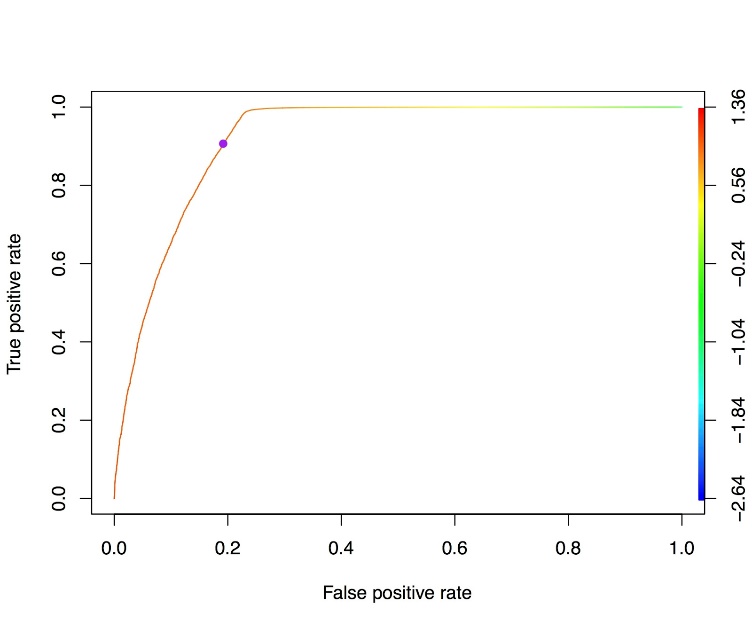


Figure 2. The ROC curve of tpr vs. fpr, the color scale represents the corresponding discriminative cut-off threshold value. The purple point shows the location of the closest point to (0,1) on the curve, which has a corresponding cut-off threshold value of 0.924.

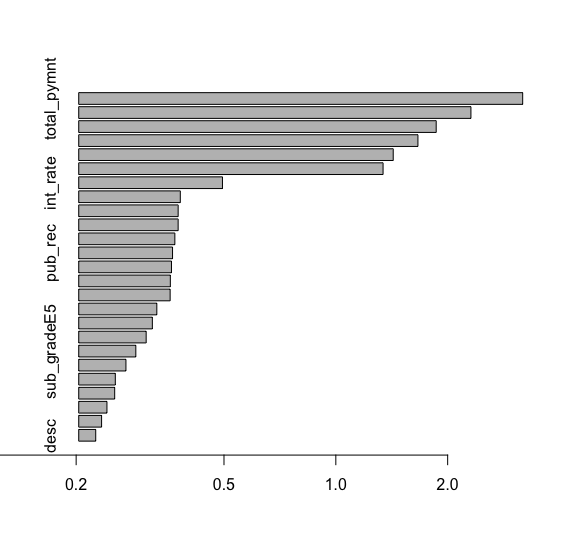


Figure 3. The barplot of features’ weights according to the L1 regularization under logistic regression.

Appendix A

#packages

library(datasets)

library(data.table)

library(stringr)

library(glmnet) # for LASSO

library(usdm) # for VIF and multicollinearity

library(caret) # for feature selection purpose

library(ROCR) # for ROC curve

loan0 <- read.csv("lending-club-loan-data/loan.csv")

# use data as is, then undersample majority/oversample minority and try again.

# row deduction

drops <- c("id","member\_id",

"emp\_title",

"earliest\_cr\_line",

"tot\_coll\_amt",

"tot\_cur\_bal",

"title","grade","issue\_d","url","pymnt\_plan",

"zip\_code","earlist\_cr\_line","mths\_since\_last\_delinq",

"mths\_since\_last\_record","last\_pymnt\_d","last\_pymnt\_amnt",

"next\_pymnt\_d","last\_credit\_pull\_d","mths\_since\_last\_major\_derog",

"policy\_code","annual\_inc\_joint",

"dti\_joint","verification\_status\_joint", "open\_acc\_6m",

"open\_il\_6m","open\_il\_12m","open\_il\_24m","mths\_since\_rcnt\_il",

"total\_bal\_il","il\_util","open\_rv\_12m","open\_rv\_24m","max\_bal\_bc",

"all\_util","total\_rev\_hi\_lim","inq\_fi","total\_cu\_tl","inq\_last\_12m")

loan1 <- loan0[,!(names(loan0) %in% drops)]

# remove instances with joint applicant or dti has a value 9999

# (meaning NA in survey), then remove the applicaiton\_type field

loan2 <- subset(loan1,application\_type =="INDIVIDUAL")

loan2 <- loan2[,!(names(loan2) == "application\_type")]

# move the y column to the last column of data frame

loan3 <- loan2

loan3 <- cbind(loan3[,!(names(loan3) %in% "loan\_status")],loan3$loan\_status)

colnames(loan3)[35] <- "loan\_status"

#Simplify Y lables

loan4<-loan3

YN <- c("Issued")

YT <- c("Current","Fully Paid","In Grace Period","Does not meet the credit policy. Status:Fully Paid")

YF <- c("Charged Off","Default", "Late (16-30 days)","Late (31-120 days)", "Does not meet the credit policy. Status:Charged Off")

loan4 <- loan4[!(loan4$loan\_status %in% YN),] # delete all status being "Issued"

loan4$loan\_status <- factor(loan4$loan\_status %in% YT) # Assigning the status in YT with True, the rest with FALSE

#convert desc to length of desc

loan4$desc <- str\_length(loan4$desc)

loan4$desc[is.na(loan4$desc)] <-0

# remove instances with NA

indx <- apply(loan4, 2, function(x) any(is.na(x))) # check if loan4 still contains NA or inf

colnames(indx) # show the column names of these missing data if there is any,

loan5 <- na.omit(loan4)

#re-order the data by numeric first, catagorical second, y to the last.

loan6 <- loan5

facvec <- which(sapply(loan6, class) == 'factor')

numvex <- which(sapply(loan6,class) == 'numeric')

loan6 <- cbind(loan6[numvex],loan6[facvec])

# Find the correlation matrix of all numeric columns in loan6

# Follow the method found at: http://machinelearningmastery.com/feature-selection-with-the-caret-r-package/

set.seed(127)

corr <- cor(loan6[,1:sum(table(numvex))]) # calc the correlation matrix between all the numeric values from data

#corr[lower.tri(corr,diag = TRUE)]<-NA # transform the correlation matrix to a upper triangular matrix

# print(corr) # summarize the correlation martrix

highlycorrelated <- findCorrelation(corr, cutoff = 0.95) # find attribute that are highly correlated (ideally >0.75)

#print(colnames(loan6[,highlycorrelated])) # print indexes of highly correlated attributes

##[1] "funded\_amnt" "funded\_amnt\_inv" "total\_pymnt" "total\_pymnt\_inv"

#[5] "out\_prncp\_inv"

### We found all the three columns with the suffix inv are almost pure colones of their non-suffix versions.

### According to the dictionary, they are just the non \_inv surffix amount invested by the investors on the platform, which has no cross interest with our research goal. Therefore,we decide to delete them all with confidence.

### funded\_amnt and total\_amnt are nearly exactly identical to loan\_amnt because by dictionary they are still investor exclusively related, so we can delete them for its redundancy.

as <- c("funded\_amnt\_inv", "total\_pymnt\_inv", "out\_prncp\_inv")

loan7 <- loan6[,!(names(loan6) %in% as)]

corr1 <- cor(loan7[,1:(sum(table(numvex))-sum(table(as)))]) # calc the correlation matrix between all the numeric values from data

highlycorrelated1 <- findCorrelation(corr1, cutoff = 0.9) # find attribute that are highly correlated (ideally >0.75)

#print(colnames(loan7[,highlycorrelated1])) # print indexes of highly correlated attributes

## Here we get a NULL, meanning no more redundant attributes after screening by the correlation matrix

## note: the correlation matrix should be corr1

# rescale all the numeric data into (0,1) interval

loan8 <- loan7

for (ic in 1:(sum(table(numvex))-sum(table(as)))) {

loan8[,ic] <- loan8[,ic] - min(loan8[,ic])

loan8[,ic] <- loan8[,ic]/max(loan8[,ic])

}

# expand the factor columns as design matrix without the intercept

loan9 <- cbind(loan8[,1:23],model.matrix(~ term + sub\_grade + emp\_length + home\_ownership + verification\_status + purpose + addr\_state + initial\_list\_status + loan\_status, loan8)[,-1])

# check if loan9 still contains NA or inf

#indx <- apply(loan9, 2, function(x) any(is.na(x)| is.infinite(x)))

#colnames(indx) # show the column names of these missing data if there is any

Appendix B

|  |  |
| --- | --- |
| Feature names | Description |
| addr\_state | The state provided by the borrower in the loan application |
| annual\_inc | The self-reported annual income provided by the borrower during registration. |
| collection\_recovery\_fee | post charge off collection fee |
| collections\_12\_mths\_ex\_med | Number of collections in 12 months excluding medical collections |
| delinq\_2yrs | The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years |
| desc | Loan description provided by the borrower |
| dti | A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower’s self-reported monthly income. |
| emp\_length | Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years. |
| funded\_amnt | The total amount committed to that loan at that point in time. |
| home\_ownership | The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER. |
| initial\_list\_status | The initial listing status of the loan. Possible values are – W, F |
| inq\_last\_6mths | The number of inquiries in past 6 months (excluding auto and mortgage inquiries) |
| installment | The monthly payment owed by the borrower if the loan originates. |
| int\_rate | Interest Rate on the loan |
| loan\_amnt | The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value. |
| open\_acc | The number of open credit lines in the borrower's credit file. |
| out\_prncp | Remaining outstanding principal for total amount funded |
| pub\_rec | Number of derogatory public records |
| purpose | A category provided by the borrower for the loan request. |
| recoveries | post charge off gross recovery |
| revol\_bal | Total credit revolving balance |
| revol\_util | Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit. |
| sub\_grade | LC assigned loan subgrade |
| term | The number of payments on the loan. Values are in months and can be either 36 or 60. |
| total\_acc | The total number of credit lines currently in the borrower's credit file |
| total\_pymnt | Payments received to date for total amount funded |
| total\_rec\_int | Interest received to date |
| total\_rec\_late\_fee | Late fees received to date |
| total\_rec\_prncp | Principal received to date |
| verified\_status\_joint | Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified |