Report of my methodology and findings in small loans

Ziyu Zong School of Computer Science and Cybersecurity
Communication University of China Email: ziyuzong@cuc.edu.cn

Background and Purpose

- There is a lending company that provides small loans to individual borrowers for 3 or 5 years.
- To conduct analysis on the data and uncover insights about the loan's performance.

Experimental Environment

Operating System: Ubuntu 16.04
Program Language: Python 3.5.2
Pandas Version: 0.17.1

Process and Findings

Step 1. Data Processing

Based on the dataset, my data preprocessing is as follows.

Projects	Operation
first column	drop the column
funded_amnt	drop the column
mths_since_last_delinq	replace the N/A with number zero
emp_length	transform string to integer
term	transform string to integer
int_rate	transform string to float
N/A in rows	drop the row that has more than 5 columns with missing values
profit_or_loss	create a new column

Table 1: Data Preprocessing

According to the table 1, there are some details.

- Drop the first column and 'funded_amnt' (same as loan_amnt).
- According to the interpretation of 'mth_since_last_delinq' and the number of missing values, the borrower should don't have last delinquency if the value is N/A, so I replace N/A with 0.
- For the same reason, the 'emp_length' is replaced by 0 if the value is N/A. In my method, the value is regarded as 1 if the value is '< 1 years'.
- Transform the string type into integer type. (term, int_rate)
- Create a new column named 'profit_or_loss' by 'total_rec_prncp' plus 'total_rec_int' minus 'loan_amnt'.

Step 2. Plot Diagram and Analyze

After data processing, I analyze features by SPSS and plot diagrams by Python. Lack of statistics and data mining theory, I tried a lot but can only find some normal phenomenon. There are some conclusions as follow.

To analyze the correlation about the features, I plot a heatmap.

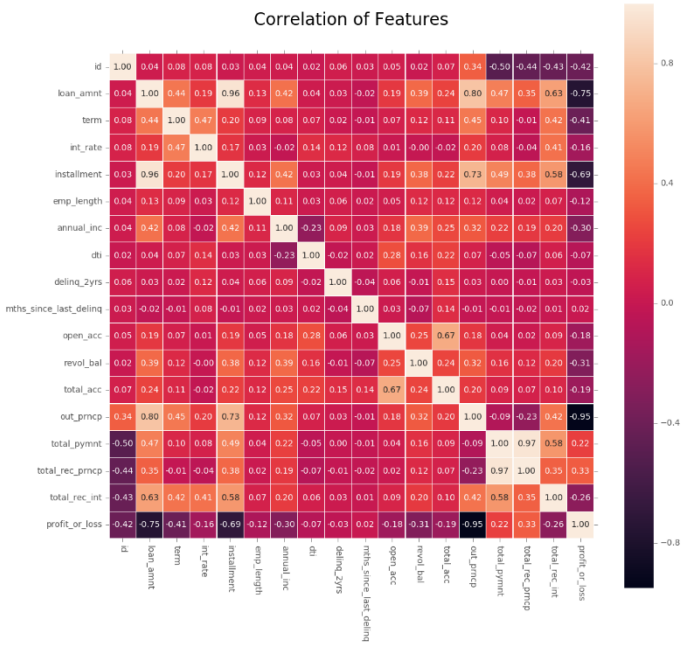

Figure 1: Heatmap about Features

In this figure, squares that are too dark or too light show the features of row and column are collinear. I pick up some features to plot a scatter figure and analyze the correlation between them.

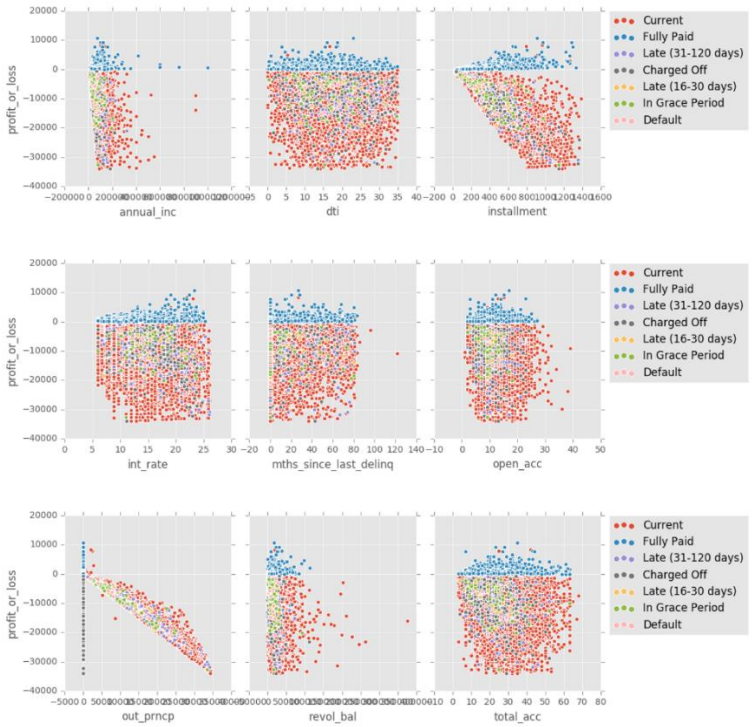

Figure 2: Scatter of Features and 'profit_os_loss'

In the scatter figure, the 'Current' dots are most quantity and the 'Fully Paid' dots are always profitable. Besides some normal phenomenon, I find a very interesting thing. The borrowers who have more than 20 credit accounts usually bring good loan, but people who have less credit accounts bring bad loan. Maybe some company have their own economic strategies through loan working capital.

Step 3. Using Model to Predict

According to the figure 2, I pick up some features to make further analysis. The features that I selected are as bellow.

'int_rate', 'installment', 'emp_length', 'annual_inc', 'dti', 'delinq_2yrs', 'open_acc', 'revol_bal', 'total_pymnt', 'profit_or_loss', 'loan_status'

After observing the data, I don't think default should be used as bad label, so I filter out the data of the default label as the dataset to be predicted. And divide the data without default into training set(80% of the data) and testing set(20% of the data).

In the beginning, I used Logistic Regression and MLP to predict the default data set. Both of their accuracy is more than 90%, and their predicted result are all good loan. And I try to use some multiclassification models for predict, but the predicted results are all 'Current'.

After changing the model, I got the same result, so I look up some reference and find this due to class-imbalance. There are 9073 good labels and 451 bad labels. The number of the good labels is 20 times than bad labels. So I used three undersampling methods to solve this problem.

My first method is using EasyEnsemble algorithm, this algorithm use Ensemble Learning to divide the positive sample into many groups for different classifier. Therefor it doesn't lose information on the whole. The accuracy rate of the EasyEnsemble is 58.0%. It is not outstanding, so I abandon this method.

My second method is extracting some positive sample randomly to make classes balance. Here is the number of samples I take for each classification. The number of positive samples is about twice that of negative samples.

Fully Paid	Current	Charged Off	Late (31-120 days)	In Grace Period	Late (16-30 days)
551	540	218	148	48	21

Table 2: The Number of Samples for each Classification

I use three base classifiers. There are Logistic Regression, Decision Tree, MLP. Their optimizers are all quasi-Newton method. I chose 1 hidden layer with 11 neurons and 2 hidden layers with 11 neurons in first layer and 2 neurons in second layer. The activation of MLP is relu, the rectified linear unit function (return $f(x) = \max(0, x)$). The initial learning rate I used is 0.001. Besides, the criterion of Decision Tree is Gini impurity, it have not bad accuracy and precision, so I used XGBoost to ensemble learning based on Decision Tree.

I use 10-fold cross validation to estimate these models and chose two evaluation criteria, accuracy and precision. I calculate the mean and std of the 10-fold cross validation. My result is as follow.

	Logistic Regression	Decision Tree	MLP (one layer)	MLP (two layers)	XGBoost
Accuracy (mean)	0.7792	0.7722	0.8317	0.8415	0.7956
Accuracy (std)	0.0866	0.1198	0.0815	0.0911	0.0781
Precision (mean)	0.8127	0.8463	0.8753	0.8816	0.8224
Precision (std)	0.0219	0.0273	0.0139	0.0199	0.0385

Table 3: 10-fold cross validation result of the model after random undersampling

Usually, we can't have both of good accuracy and good precision, but we can find the accuracy(mean) of MLP over 80% and the precision(mean) of MLP over 85%, besides both of them have less than 0.1 std according to the table 3. Therefore, it can be initially judged that the MLP fitting is better.

Because randomly undersampling has low reliability. I use the third method. I compress the data by K-Means clustering. The positive samples are grouped into 800 clusters, the negative samples are grouped into 400 clusters. The center points of each cluster are selected as the new data set. 80% of them are selected as training set, 20% of them are selected as testing set. I used the same 10-fold cross validation and the same models in my second method. The result is as follow.

	Logistic Regression	Decision Tree	MLP (one layer)	MLP (two layers)	XGBoost
Accuracy (mean)	0.6616	0.9866	0.8616	0.7866	0.9616
Accuracy (std)	0.0201	0.0125	0.036	0.041	0.0215
Precision (mean)	0.6718	0.99	0.8801	0.7757	0.9517
Precision (std)	0.0083	0.0097	0.0402	0.043	0.0293

Table 4: 10-fold cross validation result of the model after data compression

Due to time, this report cannot solve the problem of overfitting in model Decision Tree and XGBoost. I will fix that as soon as possible.

According to the table 4, we can see that the model after data compression should be more authoritative. Without considering the overfitting problem of classification tree, we can see that the fitting effect of one layer MLP is better than other obvious ones, and both the accuracy and precision are very considerable. After clustering, the average and variance of accuracy and precision obviously reach an equilibrium. Therefore, one hidden layer MLP is proposed to solve the problem of small loans.

The prediction of the 'Default' data set with one layer MLP model after data compression is as follow.

[1. 1. -1. -1. -1. -1. 1. -1. -1. 1. 1. -1. -1. -1. -1. -1.] (sorted by id)

There are 5 good labels and 11 bad labels.

Thanks for reading. If you have any questions, please contact me. I will keep learning and try my best in research.