Delinquency Prediction Using Machine Learning

Zhaowei Liang
ME397 Engineering Data Analysis, Optimization and Visualization
May 11, 2018

1. Background and Motivation

Peer-to-peer (P2P) mode has to trend for many years, and there's no difference for loan business. Starting with Prosper Marketplace in 2006, many P2P loaning companies have enjoyed prospering and growing. However, because of the "crowdfunding" nature of the P2P lending, the risk that individual investors and companies took is much higher compared to traditional loaning company.

To predict risky loans, I will introduce a model to predict the whether a loan will be completed at the end. Data cleaning, exploratory data analysis, and other related steps will also be covered in this report. The goal of this project is to avoid investment in delinquent loans. Based on the data, Nearly 35% of the loans in the market ends up in delinquency.

The dataset I used is from Prosper Marketplace Llc. This dataset contains 81 features and more than 100,000 line of loan data. The data dictionary is attached in the reference section. Due to the page limit, I can't cover all the detail of this project. Please refer to the attached RMD file for detailed workflow.

2. Data Cleaning (RMD file section 1)

Before analyzing this dataset, I cleaned the dataset using various methods. Firstly, columns with information that may not help our analysis are deleted. Some columns are highly correlated with other columns and lead to removal. For example, The Credit Score Upper Range is always 19 more than the Credit Score Lower Range. It does not make any sense for me to include both features in the model. In such case, only one of the highly correlated columns will be kept.

Then, for remaining features, three remedies are implemented. Firstly, if the feature's standard deviation is not extensive, missing values will be replaced by the mean value of the whole column. Secondly, the numerical value is missing because it does not exist, the missing value will be replaced by 0. Finally, if the column with missing value has a relationship with other features, the missing value will be calculated based on the relationship with other features. The table 3-2 listed actions I took to resolve the rest of missing values.

Finally, within the "loan status" column, all stage for past due loans are grouped as one single past due to the stage. All stage except "completed" will be categorized into "Delinquency" stage. Later in the logistic regression phase, these two stages will be grouped into numerical values.

3. Exploratory Data Analysis (RMD file section 2)

In this section, I did the bivariate and multivariate analysis of the dataset to select features. Firstly, the credit score is analyzed. Plots will be quite simple because there are only two kinds of loans to be analyzed: Completed and Delinquency.

Based on box plot shown in Fig 3-1, different status of loans has different credit score range. Another good thing to point out is not many black dots are seen. Which means outliers are limited

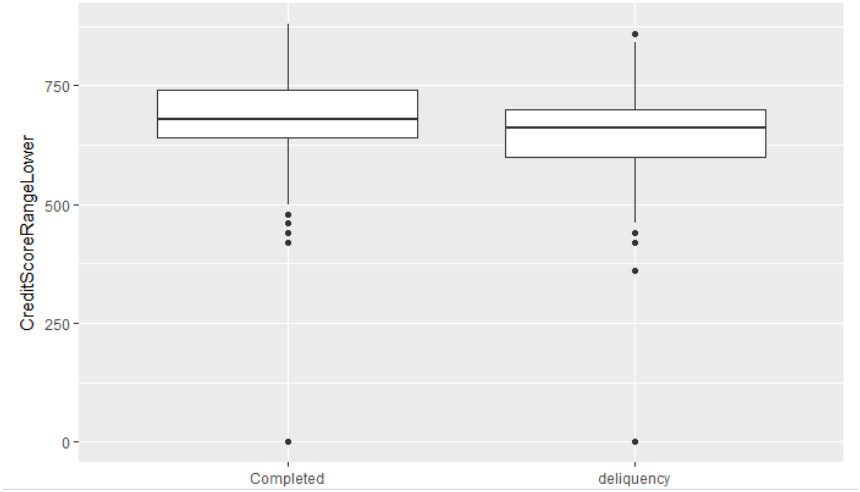

Fig 3-1 Boxplot Credit Score Range for Completed and Delinquency loans

Another plot worth to show is the multivariate plots with delinquency, credit range and monthly income. As show in fig 3-2, delinquent loans tend to crowd in low monthly income and low credit score area.

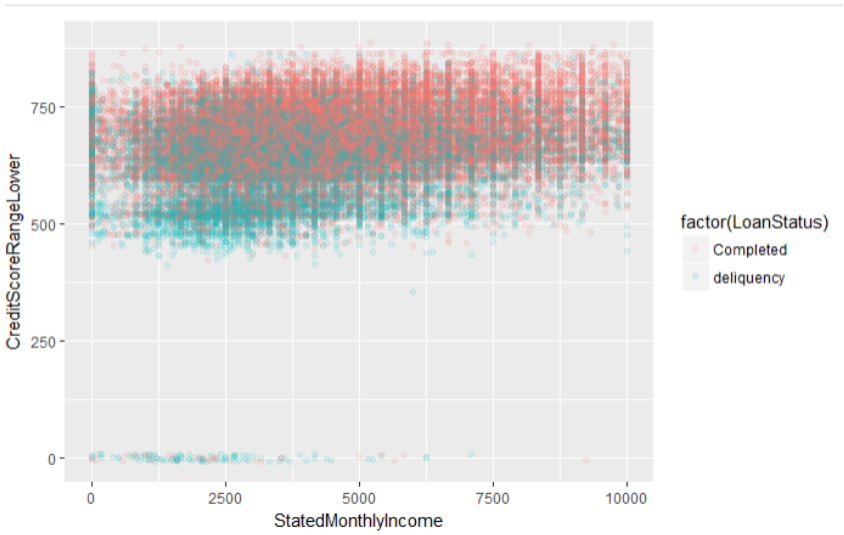

Fig 3-2 Credit score range vs Monthly income vs Loan Status

4. Machine Learning Models (Rmd file section 3)

4.1 Preprocessing

Firstly, I numerically encoded the loan status which makes the completed loan as 1 and delinquency to 0. Then, the dataset is divided into two parts: the training set and testing set. The ratio between this two sets is 2:1. The modeling phase involves two modeling methods: Logistic Regression and boosted Decision Tree. This model will be designed to be conservative which means I care more about false positive rather than false negative.

Categorical variables with different levels cannot be included in the model unless they are encoded or parse into dummy variables. For factors with only two levels, such as "IsBorrowerHomeowner." the True/False Boolean variables will be transferred to 1 and 0. However, if the features contain multiple levels, it will be parsed into dummy variables such as employment status. Fig 4.1.1 showing the parsed employment status feature, each dummy variable will be a binary variable to indicate whether a loan falls in the case. The full rank option is turned one to avoid dummy variable trap. This method is generally called one-hot encoding.

```
[25] "ListingCategory..numeric..20"      "EmploymentStatus."
[27] "EmploymentStatus.Employed"         "EmploymentStatus.Full.time"
[29] "EmploymentStatus.Not.available"    "EmploymentStatus.Not.employed"
[31] "EmploymentStatus.Other"            "EmploymentStatus.Part.time"
[33] "EmploymentStatus.Retired"          "EmploymentStatus.Self.employed"
```

fig 4.1.1 An example of One-Hot encoding

4.2 Feature selection

I my features using intuitive knowledge and Recursive Feature Elimination (RFE) from caret package.

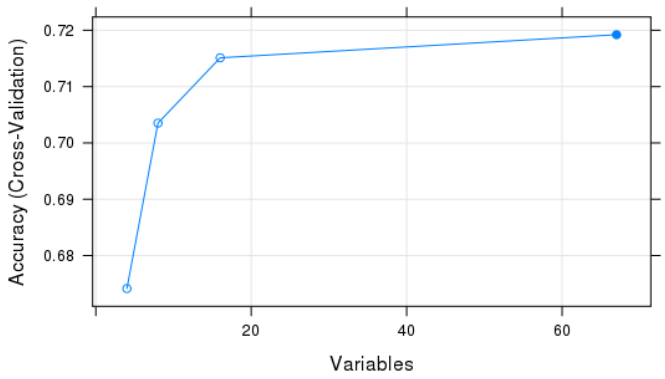

Fig 4.2.1 Auto variable selection with cross-validation

Plot shows the model's accuracy improved from 0.67 to 0.72 as the number of variables increased from around 10 to 68. However, it looks like the top 20 features is sufficient to cover most of the intel. As the result, I decided to only use the top 20 variables in the automatic selected features. Also, we need to keep in mind that accuracy is not the only performance index of this model. The true negative rate is what we are optimizing. Now, I have a set of features that need to be trained and tested.

4.3 Model training

The first model I selected called penalized logistic regression (PLR). Using this model requires the stepPlr package. This model is specifically good on predicting binary classification problem.
The second model is the general boosted model, and the third model is xgboosted tree model. These two models are very popular recently and considered as must-tries for all classification problems. Table 4.3.1 summarized their performance. The parallel package was used during the training phase to reduce training time.

Model Name	Accuracy	True Negative rate	Confusion Table
PLR (Penalized)	0.6926	0.6012	Reference Prediction 0 1 0 1679 1114 1 4692 11403
GBR model	0.7132	0.6353	Reference Prediction 0 1 0 2238 1285 1 4133 11232
Xgboosted Tree	0.7132	0.6294	Reference Prediction 0 1 0 2317 1364 1 4054 11153

Table 4.3.1 Summary table of model performance

The accuracy is specifically misleading because its simply just the underlying behavior of the data sets. If we create a prediction column and mark every loan completed, we are very likely to have similar accuracy rate. As the result, the true negative rate is what we are looking. Please notice that the R code has Pos Pred Value and Neg Pred Value flipped. As the result, I choose to use GBR as the final model.

5. Model implication and conclusion (Rmd File section 4)

To test the practicability, I randomly picked 20 rows of the data from the test datasets and assume a person is about to invest in these 20 borrowers. Assume this person is the sole loaner,

I predicted the potential loss and return before fees and taxes. Table 5.1 is the comparison between investing blindly and using the ML model.

profit_blind <dbl>	profit_ML <dbl>	return_blind <dbl>	return_ML <dbl>	profit_increase <dbl>	return_increase <dbl>
-59648.8718	-31558.722	-0.46767656	-0.377754232	28090.1500	0.089922330
-10847.4500	7853.950	-0.08626203	0.078894525	18701.4000	0.165156553
-24233.0610	2013.825	-0.20679320	0.026068932	26246.8860	0.232862131
-34503.6250	-26783.450	-0.34306363	-0.308565092	7720.1750	0.034498542
-16287.9460	-3589.946	-0.13026605	-0.038339378	12698.0000	0.091926673
-57871.7596	-27545.200	-0.45053218	-0.330278177	30326.5596	0.120254003
-20794.8100	-6466.600	-0.18947262	-0.089133012	14328.2100	0.100339613
-10454.6800	-6259.650	-0.09543208	-0.060188942	4195.0300	0.035243140
-4565.3282	-4003.320	-0.03599281	-0.042606191	562.0079	-0.006613379
-16908.7680	-3992.318	-0.15432636	-0.043600917	12916.4500	0.110725446

1-10 of 30 rows Previous 1 2 3 Next

```
mean(Return_table$profit_increase, na.rm=TRUE)
mean(Return_table$return_increase, na.rm=TRUE)
```

```
[1] 12374.82
[1] 0.07570227
```

Table 5.1 first 10 rows of the profit and return rate comparison table

As shown in the comparison table 5.1, by using the GBR machine learning model to do binary classification, we expect $12374.82 increase in profit and 7.5% increase in return rate on average. It is expected that both machine learning model and blindly investing generate a negative profit at this stage because we assumed each loan had only one investor and neglected all taxes and fees.

Overall, our model generates a positive result and has its practicality. This model can still be improved by trying more machine learning method and tuning the parameter of trained models.

Note:

Package caret is easy to run but hard to install, please feel free to let me know if any package dependencies needs to be resolved.

Reference

James Carson, RStudio at TACC, (2014), GitHub repository, https://github.com/charlespwd/project-title

Manuel Amunategui, Modeling 101, (2014), GitHub repository, https://amunategui.github.io/binary-outcome-modeling/

gappy (https://stats.stackexchange.com/users/30/gappy), Variable selection procedure for binary classification, URL (version: 2010-07-25): https://stats.stackexchange.com/q/606

Max Kuhn, Classification and Regression Training, GitHub repository, https://github.com/topepo/caret

Appendix I: Content table for the Rmd file