Overdue Rate Prediction based on Personal and Behavior Information of Clients

6202 Project individual report
Professor: Amir Hossein Jafari
Name: Xueqi Zhou
Due Date: 2020/06/30

Introduction

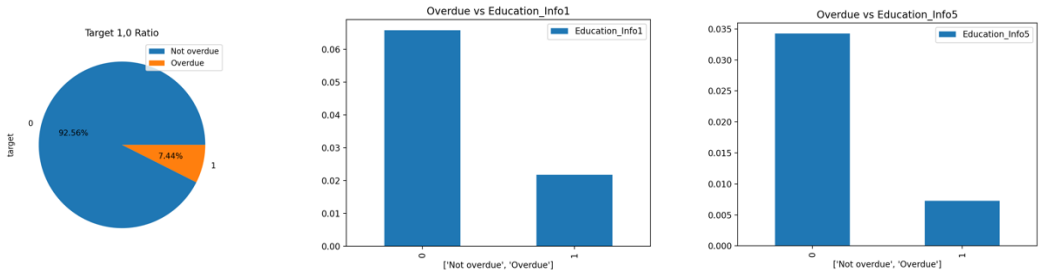
Based on user's historical behavior data of the Chinese Credit Platform, we are aiming to predict the probability of users overdue in the next 6 months. There are six major sections in our shared work. They are data visualization, data cleaning, feature engineering, feature selection, Model Building, and Accuracy calculation.

Methodology and Result

I participated in all six sections. All my works are achieved by Python.

(1) data visualization:

I drew three graphs to find the relationship between the target and some specific features

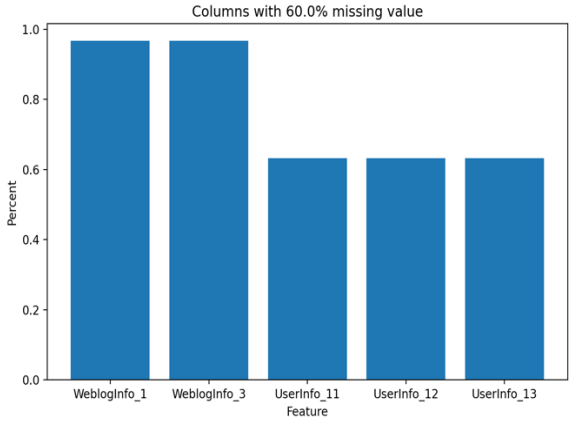


Graph 1

(2) data cleaning

The first thing I dealt with is missing value. We set 60% as our cutoff and delete the features which the ratio of missing value is larger than 60%. From the graph 2 below, we know that five features have large ratio of missing value. The ratio of some features like

WeblogInfo_1 and WeblogInfo_3 is even as high as 95%. We should definitely consider deleting these features.



Graph 2

The second thing I dealt with is finding the standard deviation of every feature and delete the features which have standard deviation near 0.

Thirdly, I find that in some features, the city's names are the same but in different format, such as "重庆" equals to "重庆市". Therefore, I renamed some cities and maintain the same format for same cities.

(3) feature engineering

I find the provinces which have the overdue ratio higher than 85% and create new features to determine whether the provinces names in features are these provinces. 1 means it is among the high overdue ratio provinces and 0 means not. We find 7 provinces satisfying the condition and create 7 new features based on them.

There are four columns showing city's information. They are UserInfo_2, UserInfo_4, UserInfo_8, and UserInfo_20. I tried to figure out whether there are differences in city's name between every two columns and create new features- "diff_". For example, diff_24 means difference between UserInfo_2 and UserInfo_4. 1 means two city's names are the same and 0 means not.

For the features containing cities' name, we give them rank based on their prosperity. 1 means first-tier cities, 2 means second-tier cities, and 3 means third-tier cities.

(4) feature selection

We are trying to select the first 200 most important features by the combination of five models. The models are Pearson correlation selector, chi-squared selector, random forest selector, logistic selector, and RFE. For every selector, it will create a new column showing whether to select the feature. We rank the feature importance based on the sum of the five new columns.

(5) Model Building

I used a single model MLPClassifier to make prediction.

(6) Accuracy calculation

I used AUC, to determine the accuracy of the model. The highest AUC is about 0.56, which means the accuracy is not ideal.

Conclusion

I have learned a lot from this project, except python coding skill. When we only get some numbers, the first thing we need to do is to visualize it and try to find the pattern behind the scene. Then, we need to clean the data to make sure that we will not be distracted by some bad formats and maintain the consistency of the dataset. Feature engineering is the most difficult one to do. According to the data, we need to create new features or combine some information in different features.

In such short time and due to the desensitization, we are unable to attain some realistic meanings from the dataset. Otherwise, we could put more efforts on feature engineering in the future. About 30% code, I copied from internet.