Applying Machine Learning Algorithms to Predicting Loan Default Risk

DATS 6202 Group 5 Final Project-Individual Report
Summer 2018
Bijiao Shen

1. Introduction

Our project comes from a Kaggle competition and uses data from a company called Home Credit. Home Credit is a company that operates in several different countries including several countries in Europe and Asia in addition to the United States. The company's business model focuses on making loans to people with little or no credit. Their loan products primarily include point of sale loans (POS), cash loans, credit cards, debit cards, revolving loans, or car loans.

Our project uses supervised learning to predict a client's ability to repay a loan. Given the company's loan products, business model, and regions of operation, there are several unique challenges with this task. First, given that the company operates primarily by making loans to people with little or no credit, the ability to which we can use default on previous loans or credit history is limited. Moreover, since there is variation in which the company operates in, there is variation in measurement in various credit bureaus in each country. For example, lenders in the United States often use FICO scores as a large predictor in a borrower's ability to repay a loan. Given these challenges, we make use of a loan dataset provided by Home Credit.

The dataset covers a large number of loan features such as the loan type (i.e. cash or revolving loan), the gender of the client, the number of children the client has, and the annual income of the client. The dataset also has the target—whether the client had payment difficulties for the loan (i.e. default or not default). We employed three supervised machine learning algorithms to estimate the relationship between the target and key loan features, including the Support Vector Machine (SVM), Naïve Bayes, and Random Forest.

I contributed to the problem selection, variable selection and Naïve Bayes classification. I come from mortgage finance background and I do financial model validation at work, which will provide exposure of traditional statistical credit risk modeling experiences to me.

I proposed this "Home Credit Default Risk" problem because we find this problem both challenging and interesting. Non-banking alternative lending is a new field for funding and the customers it serves are not usual clients that traditional banks will have. Those borrowers generally don't have the standard loan applications that big banks use because of their unique financial condition. It is also hard to verify those data and the data field definition might be altered during real life practice due to the lack of the regulation. Usually traditional statistical models fail to make a good prediction so we want to utilize machine learning techniques to try to solve this classification problem.

2. Description of the data set

I worked on data set description for our project. Below is the part of my work

There is one dataset in CSV format, which is "application_train.csv", that is involved in the Home Credit borrower default analysis. The data is sourced from Kaggle "Home Credit Default Risk" competition[1]. The dataset used for SVM and Naive Bayes is slightly different from the one for Random Forest.

(1) Data Set Information

The dataset is originally from Home Credit, which is a non-banking financial institution, founded in 1997 in Czech Republic. The objective of the dataset is to diagnostically predict whether or not the borrowers repay the loan, based on certain measurements described in the data. The datasets consists of two parts: loan application information collected by Home Credit and previous loan perforation information collected by credit bureau. The dataset consists of several financial predictor variables and one target variable, Target.

<u>Support Vector Machine and Naïve Bayes</u>

(2) Number of Instances: 316
(3) Number of Attributes: 16
(4) Attribute Information:

Target

- Target Class Variable (0 or 1) - Categorical

Predictor

- Loan Type - Categorical
- Gender - Categorical
- Car Ownership - Categorical
- Realty Ownership - Categorical
- Number of Children - Numerical
- Education Type - Categorical
- Age - Numerical
- Income - Numerical
- Regional Rating based on Credit Home - Categorical
- Number of Requires to Credit Bureau about the Client - Numerical
- Normalized Score from External Data Source 1 - Numerical
- Normalized Score from External Data Source 2 - Numerical
- Normalized Score from External Data Source 3 - Numerical
- Address Mismatch Indicator - Categorical

<u>Random Forest</u>

[1] Detailed information can be accessed through the link: https://www.kaggle.com/c/home-credit-default-risk

(2) Number of Instances: 2581
(3) Number of Attributes: 17
(4) Attribute Information:

Target

- Target Class Variable (0 or 1) - Categorical

Predictor

- Loan Type - Categorical
- Gender - Categorical
- Car Ownership - Categorical
- Realty Ownership - Categorical
- Companion during Loan Application - Categorical
- Clients Income Type - Categorical
- Level of Highest Education - Categorical
- Family Status - Categorical
- Housing Situation of the Client - Categorical
- Occupation Type - Categorical
- Application Weekday - Categorical
- Occupation Type - Categorical
- Residential Apartment Building Type - Categorical
- Residential House Type - Categorical
- Residential Wall Type - Categorical
- Residential Emergency Type - Categorical

3. Machine Learning Networks

In this section, we describe the fundamental principle of the three training algorithms we use in the project: Support Vector Machine, Naïve Bayes, and Random Forest. We briefly go through some background information and the mathematical concepts of these algorithms.

Naïve Bayes

Naïve Bayes is a simple classification technique that relies on conditional probability, and predicts the most probable class given a set of inputs. It is also a simple technique for predicting the most probable class/label given a set of features/inputs. Naïve Bayes Classifiers are extremely fast and surprisingly accurate given their "naïve assumptions". It can be thought of as a "Purely Statistical" model involving joint density function, marginal density function, Bayes rule and maximum likelihood.

The Naïve Bayes equation is below with Y denotes class and X denotes features:

$$\underset{Y}{\operatorname{argmax}}\, P(Y|X) = \underset{Y}{\operatorname{argmax}}\, P(x_1|Y)P(x_2|Y)\cdots P(x_n|Y)|$$

The Naïve Bayes classifier relies on three things:

1) Independence assumption;
2) The notion of conditional probability;
3) Bayesian Inference - a method of statistical inference in which Bayes Theorem is used to update the probability for a hypothesis as more evidence becomes available.

Bayes rule is merely the mathematical relation between the prior allocations of credibility and the posterior reallocation of credibility (conditional on data).

Solving a classification problem, we need three basic steps for Naïve Bayes classifier:

1) Convert the data set into frequency tables and Use the frequency tables to calculate likelihood tables;
2) Use the product rule to obtain a joint conditional probability for the attributes;
3) Use Bayes Rule to calculate the posterior probability for each class variable them once this has been done for all classes, output the class with the highest probability.

4. Experimental Setup

In this section, we describe how we use the data to train and test the networks mentioned above. Specifically, we explain the data preprocessing, training and testing, and performance metrics we have performed. Given the different natures of the algorithms, the data preprocessing and training follow slightly different approaches.

Naïve Bayes

In order to applying Naïve Bayes classifier, we start with variable summary statistical analysis on key variables selected based on the opinion of industry experts. Then we do the pre-processing on the dataset, which includes treatment of missing value, data normalization and treatment of imbalanced data.

Next, we split the dataset between training set and test set with 70/30 ratio and then we set the Naïve Bayes classifier to be GaussianNB. We fit the model using training set and test the model performance using test set.

Finally, we conduct the performance analysis on the model using confusion matrix, ROC and classification report.

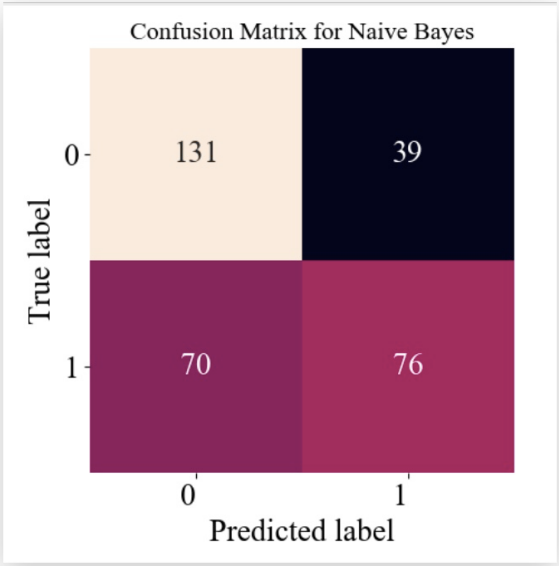
5. Results

We present and interpret the results of the three training algorithms mentioned above in this section.

Naïve Bayes

In the section, we examine the three common performance assessment methods. We have two classes in the analysis: 1 represents client with payment difficulties while 0 represents all other cases. The Confusion Matrix shows that we correctly classify 131 loans and 76 loans about its repayment performance. We classify 39 actual performing loans as loans with repayment difficulties and classify 70 actual loans which have repayment difficulties as performing loans.

Figure 5: **Naïve Bayes - Confusion Matrix**



The ROC and classification report show a similar result. We have a 65.51% (rounding to two decimal places) accuracy ratio and it is reasonable because of the credit risk estimation difficulty of alternative lending in developing countries. According to the classification report, we have 0.65 precision for estimating class 0 which is clients without payment difficulties and 0.66 precision for classifying class 1 which is clients with payment difficulties. The credit risk classification used in our analysis is more conservative towards default and it matches the purpose of the analysis.

```
Classification Report:
             precision    recall  f1-score   support

        0.0       0.65      0.77      0.71       170
        1.0       0.66      0.52      0.58       146

avg / total       0.66      0.66      0.65       316


Accuracy :  65.50632911392405


ROC_AUC :  71.55519742143433
```

In general, the algorithm performance assessment using the methods discussed above shows the difficulty of estimating credit risk for alternative lending in developing countries.

6. Summary and Conclusions

The analysis could be improved if I could better manage my time. I would like to spend more time doing data-preprocessing for missing values because we currently deleted all the missing values.

Bagging or Boosting could also be used to enhanced the model performance.

7. Percentage of the Code

All the codes that I used for the analysis is from GitHub Machine Learning 1 and I didn't use any external codes that are from other sources.