Machine Learning Engineer Nanodegree

Capstone Proposal

Ka Pang (Sammy) Yu
October 19th, 2017

Proposal

Credit Card Default Prediction in Taiwan

Domain Background

In the mid of 2000, the credit card issuers in Taiwan faced the cash and credit card debt crisis; and the delinquency got to the peak in the third quarter of 2006. In order to increase market share, card-issuing banks in Taiwan over-issued cash and credit cards to unqualified applicants. At the same time, most cardholders, irrespective of their repayment ability, overused credit card for consumption and accumulated heavy credit and cash card debts. The crisis caused the blow to consumer finance confidence and it is a big challenge for both banks and cardholders (I-Cheng Yeh, Che-hui Lien, 2009).

In order to reduce the risk of default, we can develop machine learning models to predict whether an individual would default based on the clients' information such as the clients' personal information, their financial statements and repayment records, etc.

This is a very interesting topic because credit card default happens in almost every country. And I believe many credit card issuers, such as the banks are working hard to manage this type of risk. The project helps me to learn how machine learning could solve this kind of issue.

Problem Statement

The problem I want to solve in this project is predicting whether a credit card holder would default of repaying his/her credit card balance. Solving this problem can help the credit card issuers to consider if they should issue their credit cards to this type of customers or consider if they should suspend some existing cards before the clients start accumulating a huge amount of unpaid balance in their accounts and become unable to repay. This is a *binary classification* problem. The predicting result is either "Default" (1) or "Not Default" (0).

Datasets and Inputs

The dataset is obtained from UCI Machine Learning Repository (*https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients*) and the author is I-Cheng Yeh. According to professor Yeh in his paper "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients", the data was took in October, 2005, from an important bank (a cash and credit card issuer) in Taiwan and the targets were credit card holders of the bank. Among the total

30,000 observations, 6636 observations (22.12%) are the cardholders with default payment. The research employed a binary variable – default payment (Yes = 1, No = 0), as the response variable.
There are totally 23 input features or variables in the dataset. Their details are listed below:

X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
X2: Gender (1 = male; 2 = female).
X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
X4: Marital status (1 = married; 2 = single; 3 = others).
X5: Age (year).
X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.
X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005.
X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005.

Not all the data and features are necessarily to be used in this project. It really depends on how relevant they are during the data exploration phase in this project. Some features may be aggregated, or they may be decomposed into new features.

Solution Statement

The solution to the problem would be a machine learning classifier which can predict if a credit card client would default or not. I will try four different algorithms, namely *Decision Tree*, *SVM*, *Gradient Boosting* and *Neural Networks*; and find the one with the best predicting results by comparing their performance in the evaluation metrics. The dataset will be preprocessed and cleaned as described in the project design section below before it's used to train and test those classifiers. The final selected classifier will be serialized (convert the model into a stream of bytes) and saved in a file. And when using the saved classifier to predict new outcome, we just need to de-serialize it (convert the bytes back into a model) and run it against the new input data.

Benchmark Model

The benchmark model I am going to use is a *naive predictor*. Since most individual are "Not Default" (~77%) in the dataset, we could simply point to a credit card client and say "this client is not default" and generally be right without ever looking at the data. Such a statement is little naive, since we have not considered any information to substantiate the claim. But it is simple, so it is still a good benchmark model I can use to compare with all the classifiers I am going to train, including the simple *Decision Tree* model.

Evaluation Metrics

- *Accuracy* – the percentage of correct default cases those were correctly predicted by the classifier.

- *F-beta score* – which is a harmonic mean of *precision* and *recall.* The value is between 0 and 1. A good model would have a higher score.
- *Area under the ROC (Receiver operating characteristic) curve* – a ROC curve is to be plotted for each classifier and the area under it will be calculated. 100% of AUC-ROC is the perfect score, while under 50% is considered a failure.

Project Design

Here is the workflow of the project:

1. *Data Exploration* – in this step, the raw data (in .xls format) is loaded to an IDE for viewing. The structure and characteristics of the data will be examined, including checking the missing values and the outliers. Graphs or charts are to be plotted for some stand-alone features or a group of features so that we can visualize the distribution of the values and the correlation among the feature variables.
2. *Data Prepossessing* –
   a) *Missing values* – once the data is well understood, the next step would be dealing with the missing values in some feature variables. Some features are to be deleted or imputed depending on how many values are missing and the type of the features.
   b) *Outliers – logarithmic transformation* may apply to the data to minimize the impact of the outliers. Some may be simply removed if they are just errors.
   c) *Imbalanced data* – in this particular dataset, only 22.12% of the data are already classified as "Default". In this case, the predictions of the classifier could tend to "Not Default". So some processing steps are needed to make the data be more balanced so that the classifiers can be trained to be more general.
   d) *Data scaling or normalization* – the values of some feature variables may need scaling in order to minimize the impact of the very large or very small values in some variables.
   e) *Feature selection and transformation – w*hen some feature variables are not relevant to the output variable; or they are redundant or co-linear with other variables. Those features may be removed. On the other hands, some variables may need to be decomposed or aggregated, such as using *Principal Component Analysis* to make them work well for the classifiers.
3. *Model Training and Testing* – in this step, the original dataset is split into a training dataset and a test dataset. The training dataset is used to train those classifiers mentioned above; and the test dataset is used to validate the trained classifiers. Cross validation will be used during the training phase.
4. *Reporting* – All the project details and workflow will be reported in a formal document, including all the programming code, graphs, discussion and conclusion.