Loan Default Prediction Challenge

Description of Solution

Junchao Lyu

Shenzhen, China

ljc0711@163.com

1. Summary

The framework of my solution consists of two parts:

1) Use binary classifier to identify whether the loan default or not

2) Use part of training data, which is the default loan, and part of test data, which has been predicted as the default loan in the first phase, then apply to the regression techniques.

2. Feature Selection/Extraction

Owing to the missing description of attribution information, the traditional approaches, which focus on personal information, become useless. Therefore, with respect to this competition, how to generate features is one of the key points.

In my solution, I Use the operators +,-,*,/ between two features, and the operator (a-b) *c among three features to generate new features, and get the top features based on the pearson correlation with the loss, then eliminate those similar features.

3. Modeling Techniques and Training

3.1 How to do the validation

The training data is ordered by the time, but the test data is random ordered. Therefore, to eliminate the time effect, we need to shuffle the training data randomly before the validation process.

3.2 Loss distribution and feature distribution

Owing to the long tail distribution of the loss, it is a proper way to use log(loss).

Otherwise, some features are also long-tail distributed, it is also useful to use the log operator. In my implementation, I only use the log operator on the feature f527-f528 after the validation.

3.3 Choose the probability cutoff of being predicted as the default loan

For this competition, It is always not the best choice to choose the probability 0.5 as the cutoff of being predicted as the default loan.

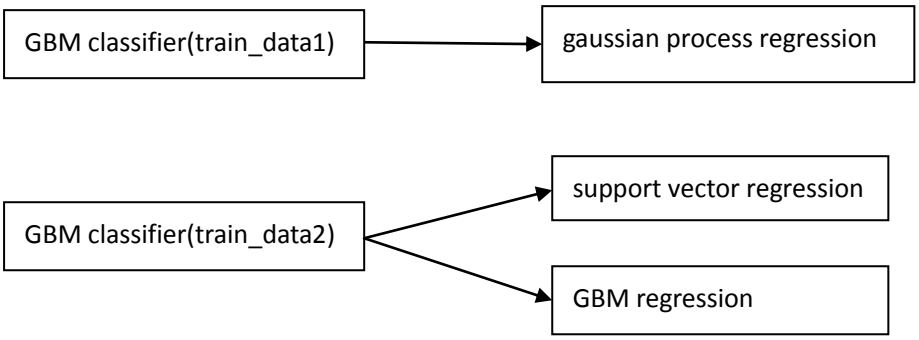
After checking the validation score and leaderboard score, I choose the probability 0.55 to be the cutoff. Namely, when the default probability is less than 0.55, the loan is regarded as no default. Otherwise, the loan is treated as the default loan, and would be processed by the regression phase.

3.4 Details of model

The solution refers to four models: GBM classifier, GBM regression,

Gaussian process regression, Support vector regression.

The overall framework of my solution is as followed:



It is always useful to blend several models. To make the models variant, the solution apply these strategies:

1) Use GBM classifier to train on the different training data

2) Use different regression techniques. For the Gaussian process regression, the correlation model (kernel function) is set as 'absolute_exponential', and for support vector machine, kernel function is set as 'rbf', for gbm regression, the loss function is set as 'ls'.

For the ensemble method, currently I linearly blended the prediction results from GBM regression, Gaussian process regression, support vector regression.

4. Code Description

The solution is written is python, and consists of two files: **features.py, predict.py**,

features.py tells which features are used in this solution, and how to generate the features.

predict.py is the main file for training and predicting. It consists of several functions:

load_train_fs, load_test_fs: load train file and test file into memory

train_type, test_type: extract features and losses from train data, and test data

toLabels: transform the loss into binary variables

output_preds: generate the output file based on the predictions

getTopFeatures: get the top feature indexes by invoking f_regression

get_data: generate the new data, based on which features are generated, and selected.

gbc_classify: use gbm classifier to predict whether the loan defaults or not

gbc_svr_predict_part: use support vector regression to predict the loss, based on the result of gbm classifier.

gbc_gbr_predict_part: use gbm regression to predict the loss , based on the result of gbm classifier

gp_predict: predict the loss based on the Gaussian process regressor, which has been trained.

gbc_gp_predict_part: train the guassian process regression.

gbc_gp_predict: use gbm classifier to predict whether the loan defaults or not, and invoke the function **gbc_gp_predict_part**.

gbc_svr_predict: to invoke the function **gbc_svr_predict_part**.

gbc_gbr_predict: to invoke the function **gbc_gbr_predict_part**.

5. Dependencies

sklearn

6. How to Generate the Solution (aka README file)

1) Download data from http://www.kaggle.com/c/loan-default-prediction/data
2) run predict.py

7. Additional Comments and Observations

Owing to the long tail distribution of the loss, it is a proper way to use log(loss). And we can find that the actual goal of regressors is minimizing the loss between log(y) and log(y'). Otherwise, as we know, the evaluation goal is minimizing the MAE. With respect to the gbm regression. the gbm regressor with the loss 'ls' actually minimizes the loss (log(y)-log(y'))^2, and the loss 'lad' actually minimize the loss |log(y)-log(y')|. So I tried to add the new loss functions to the gbm regressor, to minimizing the actual loss |y-y'|, but failed. I found that the gbm regressor with the loss 'ls' achieves the best performance in my experiments. However, I guess that there exists another loss function better than the loss 'ls'.

8. Reference:

"Why transform the dependent variable", http://cooldata.wordpress.com/2010/03/04/why-transform-the-dependent-variable/