Predict Student Loan Repayment

PROJECT REPORT

Microsoft Professional Capstone in Data Science [DAT102x]
Tinh Huynh, July 2017

Executive Summary

The goal of this Capstone project, in a competition format, is to predict the United State students' loan repayment rate (in percentage), i.e. this is a supervised learning Regression problem in Data Science.

Student loans, given to students at US institutions of higher education, and their repayment rates play an important role in deciding which segments of schools, programs, regions or students' demographics (family income, scores, graduation status, etc.) are a good investment, which can be simply understood as the students will get good jobs and repay their loan.

The competition evaluation metric is the Root-mean-square-error (RMSE, the smaller the better), explained as the square root of the average of the squared differences of the model's predicted value and the actual value, which are tested on a holding-out private dataset.

Initial Data Exploration

There are 03 datasets provided in the competition: Training, Training label and Testing. Besides, there is also a Submission format sample file for illustrating purpose to eliminate the wrong data format of submitted file to the scoring system.

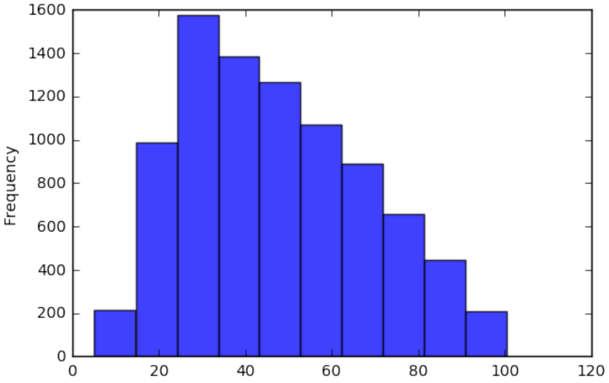
The Training data is mapped 1:1 with the Training label data based on *row_id* field, and both datasets have 8705 observations. Each observation is the total percent of the school's repayment status of its student loan, so a *repayment_rate* = 20 means there are 20 percent of the students at that specific school who repay their loan.

The Testing dataset has 6391 records, and both Training and Testing datasets have 443 features (together with the *row_id)*.

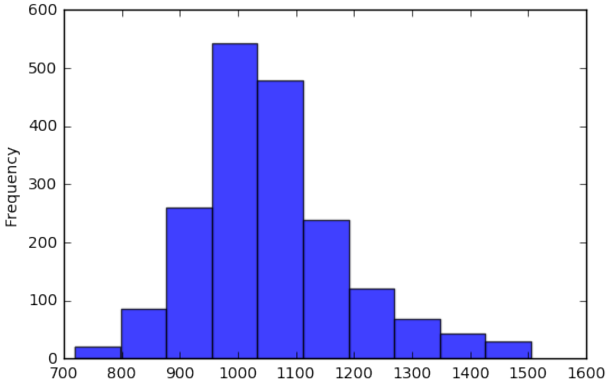
Based on the histogram plot, the *repayment_rate* is right-skewed with median=44.86, mean=47.37, and mode=33.24.

Using the feature importance of LightGBM model, we can select out some of the most correlated features to the *repayment_rate*.

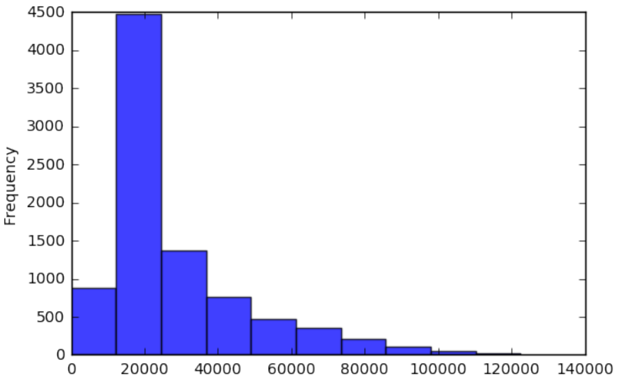
repayment_rate



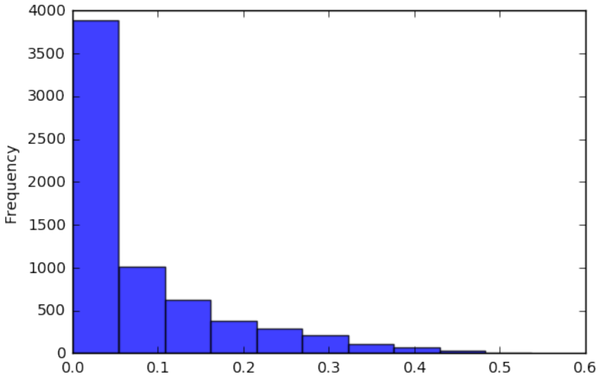
admissions__sat_scores_average_overall



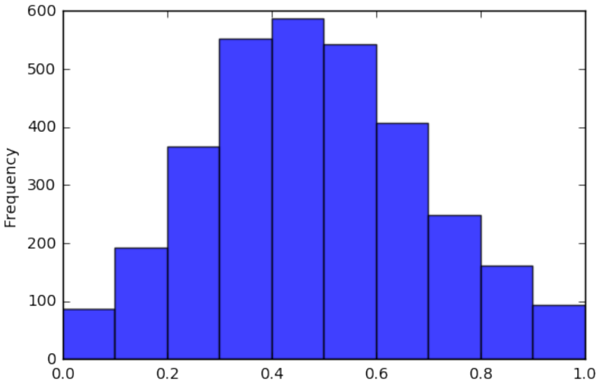
student__demographics_median_family_income



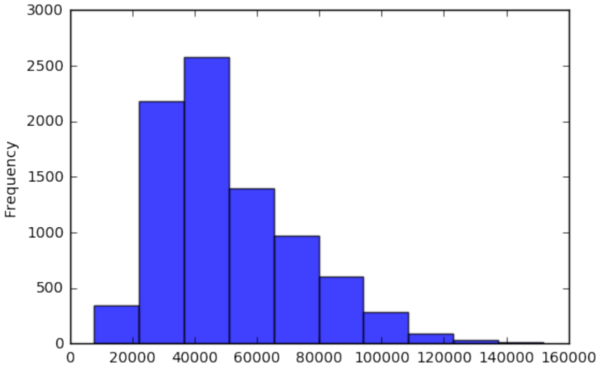
student__share_highincome_110001plus



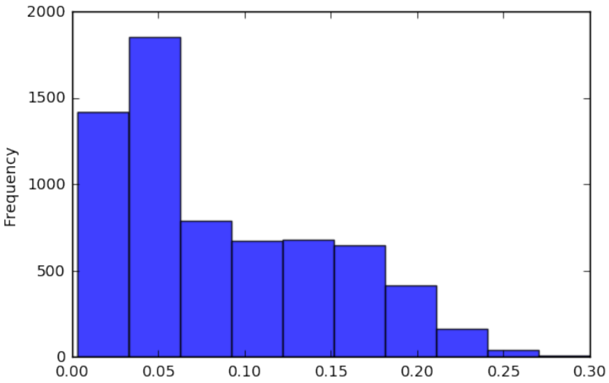
completion__completion_rate_4yr_150nt



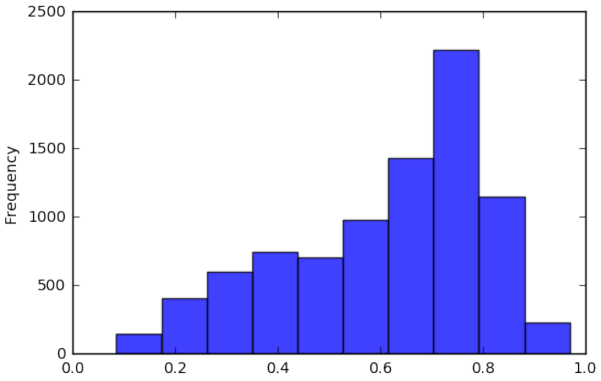
student__avg_dependent_income_2014dollars



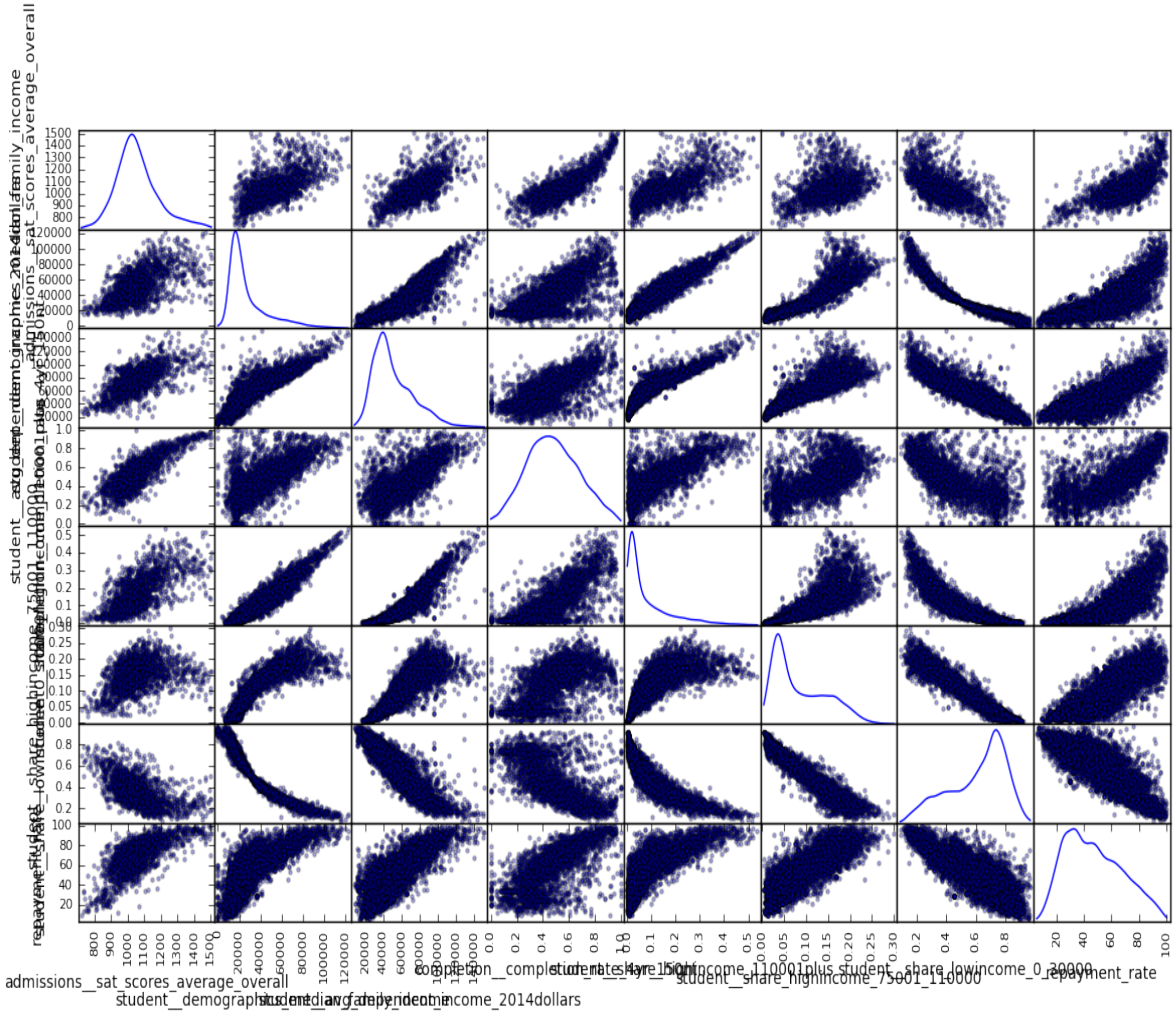
student__share_highincome_75001_110000



student__share_lowincome_0_30000



The correlation matrix plot of these features:



Data Preparation

Since released by Microsoft in June 2017, the LightGBM package has gained a lot of attraction of the data science community and practitioners; it has also been used a lot in many machine learning competitions such as Kaggle's. That is the reason I would like to try it together with XGBoost and Keras multi-perceptron deep learning model with Theano and TensorFlow backend to compare their prediction performance, speed of model training etc.

Below are the data preparation steps performed before feeding the data to these models:

1. Combined Training and Testing dataset into 1 DataFrame because there is no need (and also risky) to do double work (data imputing, cleaning, transforming, etc.) for separate 2 of them.
2. Imputed numeric features with mean or median for missing value.
3. Imputed categorical features with *Unknown* label for missing value.
4. One hot encoded for some categorical features.
5. Dropped some (actually, a lot of) unnecessary features.

The final dataset (with 376 features) is split back for training (joined with the Training label dataset based on *row_id*) and testing.

Model Training and Testing

The first model to train is LightGBM, and with all available features plus hot-encoding added ones, the testing score (based on RMSE metric) is 8.2111, this result was not too bad for the first try.

After dropping a lot of (weak correlated) features, but without cross-validation, the submitted scoring results were not quite stable, and the best RMSE was only increased to 7.2045.

Then the cross-validation of 5-fold was applied as well as a grid searching for the best set of tuning parameters, the RMSE of LightGBM model was improved to 6.3957 at best.

XGBoost was the next model to use simply because of its efficiency and effective predicting power in a large number of machine learning competitions. However, the XGBoost training times were much slower than LightGBM (at least 6 times), and the performance was only increased a little to the best RMSE of 6.3853.

Therefore, LightGBM should be the first model to try if we want to search for the best parameters such as boosting type, learning rate, max depth, number of estimators, sub sample, etc. after that we can apply these parameters to train XGBoost model and test to see if the performance is improved or not.

The Keras multi-layer perceptron models with Theano (and then Tensorflow) backend were the last ones to try. However, the testing results were not exciting, only 7.5 was the best RMSE although in training this metric was dropped to 4.5 (about loss of 20 for MSE), so the model were somehow overfit, this may be because our training dataset is too small to apply these deep learning models.

Conclusion

This is a very interesting project and predicting the student loan repayment is a quite challenging regression problem, but in my opinion, Microsoft has helped us a lot by providing the very complete datasets for training and testing, i.e. we do not need to do so much of the hard work related to feature engineering.

Below are some of my lessons learned from this capstone project:

- ✓ Cross validation is always necessary and the best practice to avoid overfitting as well as ensure the model testing performance.
- ✓ LightGBM model training speed is amazing fast, and much faster than XGBoost (at least 6 times) with almost the same prediction power.
- ✓ Always try to have some actions with your data, do not just throw everything to the models.