MATH6010z: Project 3

ZHAO JUNDA
MSc of Financial Mathematics
The Hong Kong University of Science and Technology
jzhaobr@connect.ust.hk

LI MINGLUO
MSc of Financial Mathematics
The Hong Kong University of Science and Technology
mlicv@connect.ust.hk

HE HAOKAI
MSc of Financial Mathematics
The Hong Kong University of Science and Technology
hheap@connect.ust.hk

HUANG WENJIN
MSc of Financial Mathematics
The Hong Kong University of Science and Technology
whuangbk@connect.ust.hk

Abstract

As we discussed in the project 1, Home Credit try to make use of a variety of alternative data-including tel-co and transactional information-to predict their clients' repayment abilities. We use statistical and machine learning methods to make the predictions. We used logistic regression with cubic polynomial features to train the algorithm which help recognize the clients capable of repayment. In this report, we try to use some new methods to improve the Logistic Regression model or find a better model. And we get about 0.04 point improvement compared to the project 1.

1 Introduction

As we talked about in the project 1 before, many people struggle to get loans due to insufficient or non-existent credit histories. It's a huge challenge for Home Credit to recognize the clients' repayment abilities. So we have to use statistical and machine learning methods to help Home Credit make the predictions base on a variety of alternative.
In project 1, we used a Logistic Regression with cubic polynomial features and we got a back testing score of 0.70703 in Kaggle. After learning different machine learning methods and data augmentation, we wonder if we could improve the model or find a better model to solve the problem.

2 Model Improvement

There are 16 categorical variables in the training data set. 3 of which have no more than 2 unique categories and the rest have more than 2 unique categories. We use numeric value 0 and 1 to label the

3rd Project MAFS 6010Z AI in Fintech (Worked by Team LI/HE/HUANG/ZHAO 2021), Hong Kong, China.

former and one hot encoding to label the latter. After merging the training data set with the testing data set, there are 239 independent variables left for us to filter.
In project 1, we used a Logistic Regression with cubic polynomial features. We calculate the correlation between 239 independent variables and the dependent variable namely TARGET so that we can select variables that have the highest correlation to the TARGET to build the logistic regression model. We select EXT_SOURCE_1, EXT_SOURCE_2 and EXT_SOURCE_3 who have the most negative correlation as well as DAYS_BIRTH who has the most positive correlation and construct cubic polynomial features as in Table 1.

Table 1: Cubic Polynomial Features Table

EXT_SOURCE_2
EXT_SOURCE_3
EXT_SOURCE_2*DAYS_BIRTH
EXT_SOURCE_2*EXT_SOURCE_3
EXT_SOURCE_1*EXT_SOURCE_3
EXT_SOURCE_2*EXT_SOURCE_3^2
EXT_SOURCE_1*EXT_SOURCE_2*EXT_SOURCE_3
EXT_SOURCE_1*EXT_SOURCE_2*DAYS_BIRTH
EXT_SOURCE_2*EXT_SOURCE_3*DAYS_BIRTH

2.1 Feature Selection

When we review the model in project 1, we find it may be not reliable enough to directly choose those features according to their correlations. In this case, we adjust the features choosing methods and use a forward-stepwise-selection-like method, similar to those in Linear Regression Model, to get those reliable features in Table 2. And with this adjustment, the score of the test set is 0.70886, which is greater than the model in project 1.

Table 2: Features Table by Forward Stepwise Selection

EXT_SOURCE_2
EXT_SOURCE_3
EXT_SOURCE_1
CODE_GENDER_M
FLAG_DOCUMENT_3
NAME_EDUCATION_TYPE_Higher education
DAYS_ID_PUBLISH
REG_CITY_NOT_LIVE_CITY
NAME_INCOME_TYPE_Working
DAYS_LAST_PHONE_CHANGE
REGION_RATING_CLIENT_W_CITY
EMERGENCYSTATE_MODE_No
OCCUPATION_TYPE_Laborers
DAYS_REGISTRATION
NAME_EDUCATION_TYPE_Secondary / secondary special
REGION_POPULATION_RELATIVE
FLOORSMAX_MODE
AMT_GOODS_PRICE
REGION_RATING_CLIENT

2.2 Data Augmentation by GAN

From the Table 3, we may find that in the training data set, most of the observations show 0 in the TARGET variable. In this case, we wonder if the Generative Adversarial Networks could help improve the model by doing the data augmentation.

For Generative Adversarial Networks by Creswell et al. (2018), we follow the flow chart (Figure 1) and augment the data set which are with TARGET 1. As there are too many features highly desecrated,

Table 3: TARGET Distribution

TARGET	Observation	Percentage
0	282686	0.91927
1	24825	0.08073

we need to reduce the dimensions and standardize them when applying GAN. Otherwise, it can hardly get well train among the messy data.To simplify the problem, we only use GAN to generate the TARGET 1 samples but not TARGET 0's, so we don't need to imply label information among GAN. With the white noise and the generator, we could get some new data set. And after testing by the discriminator, we could make the data augmentation reliable.

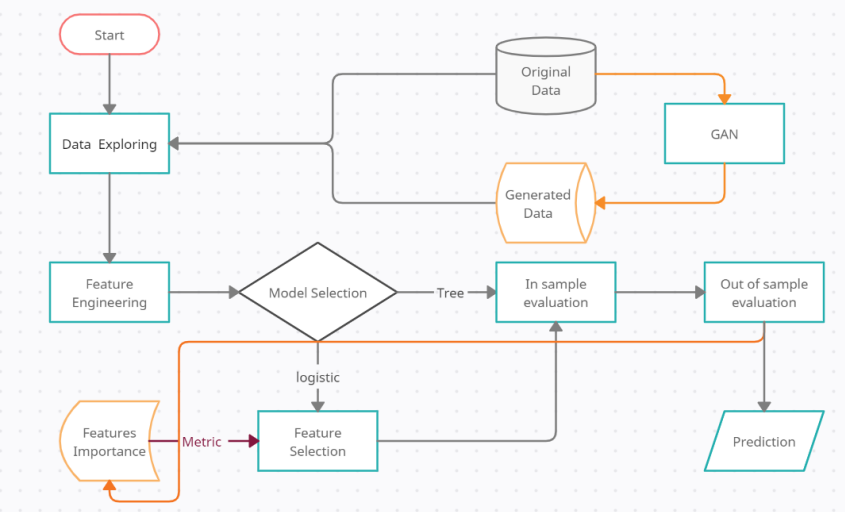

Figure 1: Flow Chart of GAN

Using different scales of data augmentation, we find that the score would reach 0.71169 when we choose to augment 2000 generated samples. It's clearly that GAN could help improve our original Logistic Regression, and we can get the following cubic polynomial features with the similar method as the original one.

Table 4: Cubic Polynomial Features Table with GAN

EXT_SOURCE_2*EXT_SOURCE_3*DAYS_ID_PUBLISH
EXT_SOURCE_3*DAYS_ID_PUBLISH
EXT_SOURCE_2*DAYS_ID_PUBLISH
EXT_SOURCE_2*EXT_SOURCE_3
EXT_SOURCE_1*EXT_SOURCE_2*EXT_SOURCE_3
EXT_SOURCE_2^2*DAYS_ID_PUBLISH
EXT_SOURCE_2^2*EXT_SOURCE_3
EXT_SOURCE_3^2*DAYS_ID_PUBLISH
EXT_SOURCE_2*EXT_SOURCE_3^2
AMT_GOODS_PRICE
DAYS_BIRTH*AMT_ANNUITY
AMT_CREDIT^2
AMT_CREDIT
AMT_GOODS_PRICE*AMT_ANNUITY
AMT_ANNUITY^2
AMT_CREDIT*AMT_ANNUITY
AMT_ANNUITY

3 Model Analysis

Not only trying to improve the Logistic Regression, we also try some other machine learning methods especially Gradient Boosting Decision Tree by Friedman (2001) to train a better model.

3.1 Extreme Gradient Boosting

Using the XGBoost (Extreme Gradient Boosting by Chen et al. (2016)), we got the best ROC curve (Figure 2) among all the models. However, the test set shows a score at only 0.64987. In this case, we believe the model is over-fitted and we need to either readjusted the model or find another way to avoid this situation.

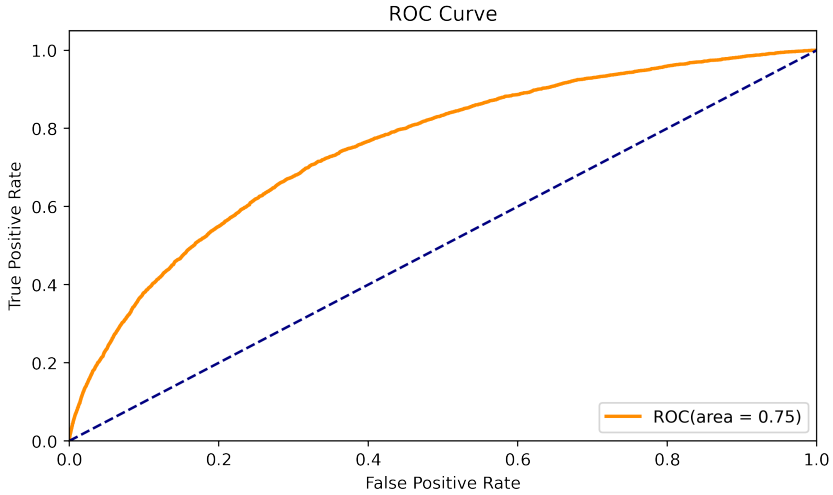

Figure 2: ROC Curve of XGBoost

3.2 Light Gradient Boosting Machine

Then, we choose the Light Gradient Boosting Machine(Ke et al. (2017)) because of its fast-speed as well as the reliable gradient boosting method. To show the performance comparison better, we first use the similar feature selection way as the previous Logistic Regression, ranking the correlation of each variables and getting the top 15 positively related variables and top 15 negatively related variables. In this way, with those 30 features and the original parameters, we get a score at 0.73590 for the LightGBM model with features selection and cross validation. It seems that LightGBM can really show better performance than the Logistic Regression model and the XGBoost model, so we decide to further improve this LightGBM.
When we adjust the gradient parameters as well as the selection of the original features in multiple times, we find that KFold and Learning Rate have nearly no effect on the performance of the model. In this case, we get our best LightGBM model with 0.05 learning rate, 3 kfold, using Median to fill NA, and using all the features. Furthermore, as we decide to use all features in LightGBM model, we cannot use GAN to do the data augmentation since that GAN is not fit for the data with large discrete features.

4 Conclusion

In total, comparing all performance of the models we trained, as shown in Table 6, the LightGBM model with parameter adjusted get the best score at 0.74578, whose score was only 0.70703 for our project 1. It really shows that the Gradient Boosting methods work well in this Home Credit prediction problem, although, in some way, it is more difficult for us to figure out the economic meaning between the TARGET and each features.
However, compared to the top performance of the Leaderboard, there is still a gap of 0.05 point for

Table 5: LightGBM Performance

Fill_NA	Features	KFold	Learning Rate	Score
Median	30_Selection	10	0.01	0.73590
Median	30_Selection	10	0.05	0.73609
Median	ALL_Features	10	0.01	0.74473
Median	ALL_Features	10	0.05	0.74463
Median	ALL_Features	6	0.05	0.74440
Median	ALL_Features	3	0.05	0.74578
Mean	ALL_Features	3	0.05	0.74345
Zero	ALL_Features	3	0.05	0.74285

us to improve. We think that it is because we did not figure out how to use the other features from other data sets to improve the model. In our opinion, there may be both large potential improvement and high risk of training a over-fitted model because of the scale of the features in the training set.

Table 6: Score Table

Model	Score
Logistic Regression + Cubic Polynomial Features	0.70703
Logistic Regression + Forward Stepwise Selection	0.70886
Logistic Regression + GAN	0.71169
XGBoost	0.64987
LightGBM + Features Selection	0.73590
LightGBM + Parameter Adjusted	0.74578

5 Contribution

Python Coding: ZHAO JUNDA, HUANG WENJIN, HE HAOKAI
Report Writing: LI MINGLUO

References

Creswell A, White T, Dumoulin V, et al. Generative adversarial networks: An overview[J]. IEEE Signal Processing Magazine, 2018, 35(1):53-65

Friedman J H. Greedy function approximation: A gradient boosting machine[J/OL]. The Annals of Statistics, 2001, 29(5):1189-1232. http://www.jstor.org/stable/2699986

Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C/OL]. KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: Association for Computing Machinery, 2016: 785–794. https://doi.org/10.1145/2939672.2939785

Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree [C/OL]. Guyon I, Luxburg U V, Bengio S, et al. Advances in Neural Information Processing Systems: volume 30. Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf