Report of TC1 Project

Theme: Otto competition on kaggle

PENG Qixiang
LI Zizhao

January 31, 2018

1 Introduction to data-set and metric

Otto competition is a multi classification problem. The data-set is like: 61878 train samples; 9 labels; 93 features.

For metric, we use log-loss here, the formulation is like:

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} y_{ij} log(p_{ij})$$

2 Feature engineering

Firstly, we draw the histograms of each feature, and find out their distribution are similar, as shown in Fig. (a). Most of values are 0. So our experiments showed centering and scaling don't work in this case.

Afterwards, we use random forest to do find the most important features and find all features are approximately same important, as shown in Fig.b.

And we also tried PCA, and it didn't work also. So we decided use total original data.

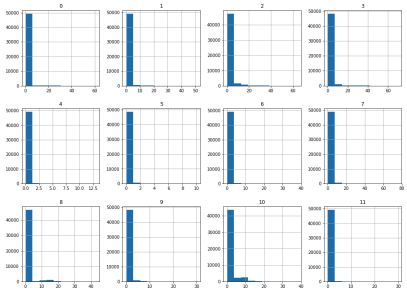

Figure 1: Hist of first 12 features

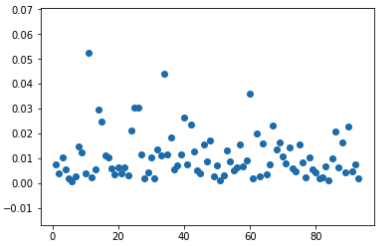

Figure 2: Importances of features

3 Algorithms

Here, we used three basic algorithms: Naive Bayes, Xgboost, Random Forest, Extra-trees. The fist one is the standard method, and performed worst.

The last three ones are the ensemble methods. In fact, if we took the default settings for their hyper-parameters, they played worse than Naive Bayes.

But after fine-tune, they showed good result. But pay attention here, it may take so much time. For example, we took three days to fine-tune the n-estimators, max-depth and max-child-weight of xgboost by using grid-search with 3-fold cross-validation. So afterwards when fine-tuning hyper-parameters of random forest and extratrees, we gave up the cross-validation in consideration of computation power.

Note here that we also applied calibrated probability technique on random forest and extratrees.

Here are comparison between the different results

	default setting	after fine-tune	calibrated probability
random forest	1.40	0.53	0.48
extratrees	1.44	0.51	0.47
xgboost	0.49	0.44	None

Table 1: Results comparison

Finally, we found out we can bagging these three models to get the stronger generalization ability.

4 Results

Here are our results judged by kaggle:

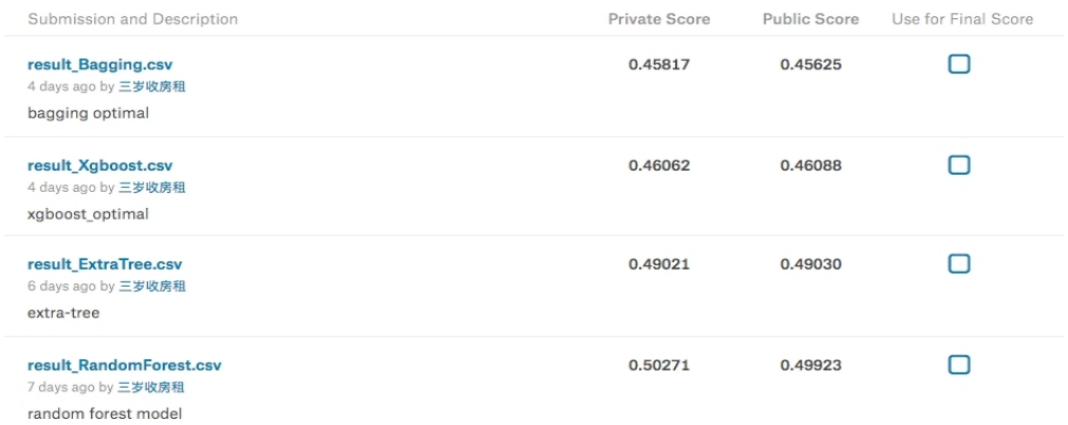

Figure 3: Final results