Warmup Project Report:
Home Credit Default Risk

Cao Bokai, Luo Zhuang, Shi Jie, Lai Fujie

September 19, 2021

Abstract

The tree model is widely used in machine learning problems, especially in the classification problem because of its strong interpretive ability. In this paper, our group mainly uses the random forest model and lightgbm model to predict home credit problems. Before construct tree models we collect and preprocess data from Kaggle, methods including outlier handling, label encoder, one hot, and so on. Our final model gets a 0.7714 score in Kaggle and can be improved in the future.

Workload: Cao Bokai almost did the whole coding work and he has a strong ability in data analysis. The other does the paperwork. Lai Fujie is trying his best to catch up with the work by reading Bokai'code and try to build a three from 0 to 1. He has done jobs in model selection. Luo Zhuang prepared the feature engineering and model results sections of the report. Shi jie prepared the data process portion of the report.

Part I: Data Selection

Our model used application_test, application_ train, bureau, previous_ application, POS_ CASH_ balance, credit_ card_ balance, installments_payments data sets, make full use of the existing information. We checked the dataframe.dtypes at the beginning of the data processing, employed labelEncode and onehot encode to the object variable. First, as the basic data set, we select most of the columns in application_ train and convert some of the data to dummy variable.

At the same time, there are some outliers in the data set, the values in OBS_30_CNT_SOCIAL_CIRCL which greater than 300 is obviously abnormal, so replaced by nan. We also analyze each data table in detail by calculating and drawing the mean square error, maximum value, box plot and other statistical methods. The column of DAYS_EMPLOYED, DAYS_FIRST_DRAWING, DAYS_FIRST_DUE, DAYS_LAST_DUE_1ST_VERSION, DAYS_LAST_DUE and DAYS_TERMINATION in the application_train, application_test and previous_application data tables has some abnormal values of 365243, which is significantly larger than the normal data, so we Replace it with np.nan.

The Bureau data set is a very important data set, because it records the past credit records of many residents, which directly reflect the residents' past lending habits and repayment ability, we have carried out the statistics of mean, sum, max, and min on the various data of each person, and achieved full use of the data. We dropped SK_ID_BUREAU, CREDIT_CURRENCY, CREDIT_Type and other columns that we think are of little significance, some important columns, such as whether borrowers have been borrowed loans or overdue in the past, have been included in the model. previous_application data set and the bureau data set have similar meanings, and both reflect residents' borrowing habits and repayment ability.

Our model also included previous_application dataset, and because the string is difficult to handle, we remove SK_ID_PREV, WEEKDAY_APPR_PROCESS_S--TART, HOUR_ APPR_ PROCESS_ START and other columns, while AMT_AN--NUITY is selected, we also selected some other representative columns such as POS_CASH _balance, credit_card_balance, installments_ payments, on which the number of economic behaviors of each resident in each data set had been mainly calculated, we used these results for the final regression. The missing values in the final train and test data sets are replaced by the median, at the same time, there are some missing values in the column of SK_ID_CURR of some data tables, we replaced them with 0 instead of the median.

Part II: Feature Engineering

Feature engineering, refers to a series of engineering approaches to filter better data features from raw data to enhance the training of models. Feature engineering usually includes data pre-processing, feature selection, dimensionality reduction, etc. In our model, in addition to the conventional outlier and missing value processing, we generate One-Hot Encoder for categorical features using

Label Encoding and add Polynomial Features for several important features.

Label Encoder and One-Hot Encoder

For categorical features in the dataset, including NAME_CONTRACT_TYPE (an identifier of whether the loan is cash or revolving), FLAG_OWN_CAR (an identifier of whether the customer owns a car) and FLAG_OWN_REALTY (an identifier of whether the customer owns a house or apartment), machine learning algorithms can not directly processed and need to be converted to numerical features.

Using LabelEncoder to assign the various labels a countable consecutive number. However this may lead to fallacies, because we just want the machine to distinguish them, no size comparison is intended. So at this point the label encoding is not enough and further conversion is needed.

Using pandas.get_dummies to generate dummy variables, giving each label a One-Hot Encoder, which will not be biased and will basically not affect the effectiveness of the vector space based metric algorithm. After the One-Hot Encoding, it will make the distance calculation between features more reasonable and, by converting the features to 0, 1, it will help to improve the calculation speed.

Polynomial Features

In machine learning, it is often effective to increase the complexity of a model by adding some nonlinear features to the input data. A simple and general approach is to use polynomial features, which allows to obtain higher dimensionality of features and terms that are related to each other. The construction of features can be done using the class PolynomialFeatur es, which is constructed by multiplying features with features.

We selected features EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3 and DAYS_BIRTH that have the highest correlation with TARGET to construct polynomial features. We specify up to 3 degrees, higher degrees are more likely to lead to overfitting.

Dimensionality Reduction

After the above processing, we ended up with more than three hundred features. Checking the correlation between features and wanting to remove the features with particularly high correlation, we finally found no such problems.

It should be noted that when the number of categorical features is large, the feature space will become very large after the One-Hot Encoding. In this case, PCA is usually used to reduce the dimensionality. But in our model, the number of categorical features is only three, so no dimensionality reduction is performed.

	TARGET
TARGET	1.000000
EXT_SOURCE_3	-0.178919
EXT_SOURCE_2	-0.160472
EXT_SOURCE_1	-0.155317
DAYS_BIRTH	0.078239
...	...
NAME_HOUSING_TYPE_Co-op apartment	-0.000312
ORGANIZATION_TYPE_Legal Services	-0.000236
FLAG_DOCUMENT_20	0.000215
ORGANIZATION_TYPE_Advertising	0.000117
ORGANIZATION_TYPE_Industry: type 7	-0.000094

240 rows × 1 columns

Figure 1: Correlation of features with TARGET

Normalization

For numeric features, it is common practice to normalize the data. We normalized all features after the above processing.

$$x^* = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Normalization is the interval scaling of a data set to an interval of [0,1], transforming data with units into data without units, i.e., unifying the data measure and eliminating the effect of units. This facilitates data processing and makes data processing faster and more agile. The most commonly used method of normalization is MinMaxScaler.

It is important to note that the decision tree model is based on information gain, information gain rate, and Gini index as the feature space partitioning method. Any monotonic transformation of features (which does not affect the ranking result) will not affect the model, so the tree model (including GBDT and Lightgbm) actually does not need to be normalized.

Part III: Model Selection

Logistic Regression

In Home Credit Default Risk we have to analyse two kinds of people:who may have the ability to repay their loan and who don't.So intuitivly we can code two kinds of people into 0 and 1 and apply linear regression produce the same result as linear discriminant anlaysis.Technically speaking we are building a logistic regression model.It's a process of modeling the probability of a discrete outcome given an input variable. It is a classification model, which is very easy to realize and achieves very good performance with linearly separable classes.

Tree Based Model

When considering a classification problem,it's intuitive that we divide things into two parts based on their different patterns like short or long, young and old,etc,and after dividing one branch we can consider another pattern,just like building a tree.Particularly, in machine learning the idea becomes what so called tree models.They are the most popular models because of their intelligibility and simplicity.But we have to build a far more sophiscated tree model facing Home Credit Default Risk data set,which cotains thousands of data, so that we can make better prediction.Among them we choose random forest and lightgbm.

Random Forest

A random forest is one of the ensemble methods, a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Random forests provide an improvement over bagged(boostraped aggregation) trees by way of a random forest small tweak that decorrelates the trees. When building these decision trees, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors. A fresh sample of m predictors is taken at each split, and typically we choose m$\approx\sqrt{p}$—that is, the number of predictors considered at each split is approximately equal to the square root of the total number of predictors.By randomly choosing samples in this way we can build a more robust model with less variance.

Normally speaking,in building random forest we can use Gini Index as the object function.That is to say in every node,the features possessing the least value of the Gini Index would get preferred.The Gini Index is determined by deducting the sum of squared of probabilities of each class from one, mathematically, Gini Index can be expressed as:

$$GINI = 1 - \sum_{k=0}^{N} p_i^2$$

Where p_i denotes the probability of an element being classified for a distinct class.

We could also use random forest to do feature selecting,which is a crucial step in preprocessing data.Here is an example we use to find feature importances in one of Home Credit Default Risk data set:

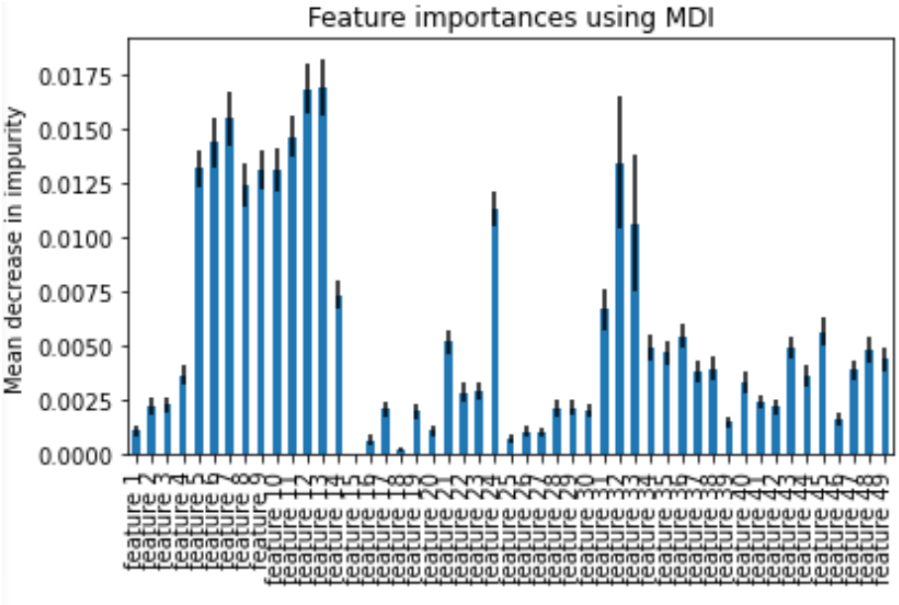

Figure 2: Feature importances using MDI by random forest

The mean decrease in impurity importance of a feature is computed by measuring how effective the feature is at reducing uncertainty (classifiers) or variance (regressors).It's the most basic but popular way of explaining a tree-based model and its ensembles.

Lightgbm

LightGBM is a gradient boosting tree based learning algorithms.Boosting can be interpreted as an optimization algorithm on a suitable cost function that we define on the tree,to be more specific ,optimize a cost function over function space by iteratively choosing a function (weak hypothesis) that points in the negative gradient direction.Other famous gradient boosting tree like Xgboost is slightly different from lightGBM in some algorithm details,but overall they share the same idea of gradient boosting and minimize the loss function.

In the Kaggle competition we have tried lots of model including logistic regression,xgbosst,etc.But among them lightgbm give us the highest score.It may be relative to its innovation in traditional boosting algorithm:

1.Appliy leaf-wise tree growth.

2.Implement a highly optimized histogram-based decision tree learning algorithm.

3.Propose gradient-based one-side sampling.

4.Use exclusive feature bundling algorithm.

In the end we blend random forest model and lightgbm model to achieve a better model performance.

Part IV: Model Results

At the beginning of the experiment, we only chose two files, application_train.csv and application_test.csv, to analyze using the lr and rf models. Both of them were trained quickly, but the scores obtained were not very high, 0.70 and 0.72, respectively.

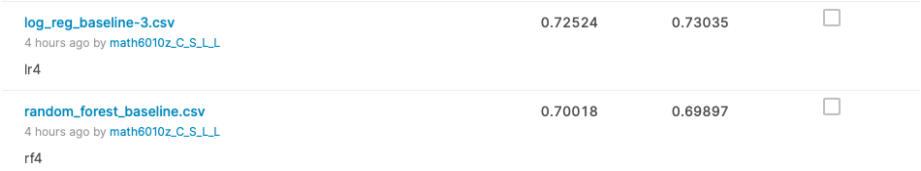

Figure 3: Results of lr & rf

Through this process we became familiar with the data structure and the model and tried to use lgbm to get better results. The result before tuning was 0.73, and after tuning it improved to 0.75. At first we set the tree depth deeper, which brought overfitting, and although it performed well on the training set with an auc of 0.77, the result on the test set was only 0.73.

By familiarizing ourselves with the previous stage, we added more features from more files and used the bureau.csv file additionally. The final lgbm results reached 0.77345 & 0.77138 for the auc.

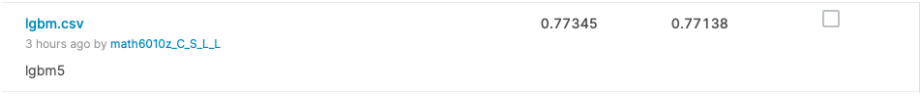

Figure 4: Final result of lgbm

Finally we tried model fusion, using model fusion such as majority voting fusion and weighted average fusion, but unfortunately the results were not improved.

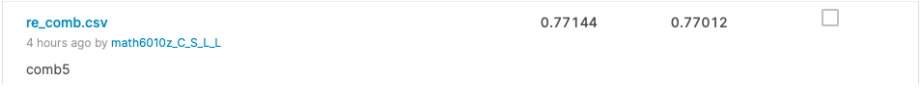

Figure 5: Result of combined model