Kaggle Contest: Home Credit Default Risk

Hao WU
hwubx@connect.ust.hk

Yiming ZHU
yzhucd@connect.ust.hk

Jihong TANG
jtangbd@connect.ust.hk

Xian WANG
xian.wang@connect.ust.hk

1 Introduction

Some people need to borrow from lenders when they are pursuing education or starting a business, but may not be able to apply for a loan because they do not have sufficient credit history, such as the student population. And as the number of applicants increases, it can be very time-consuming for lending organizations to look at an applicant's historical credit history. If the borrower's repayment ability could be predicted in some way, it would not only help customers who already have a regular income but lack a credit history, but also help lending organizations estimate safe credit lines to lend to customers who do not have sufficient credit history, as well as save lenders labor and time costs.

Home Credit, an international consumer finance provider, provided a large dataset and motivated machine learning engineers and researchers to come up with techniques to build predictive models through a Kaggle competition.

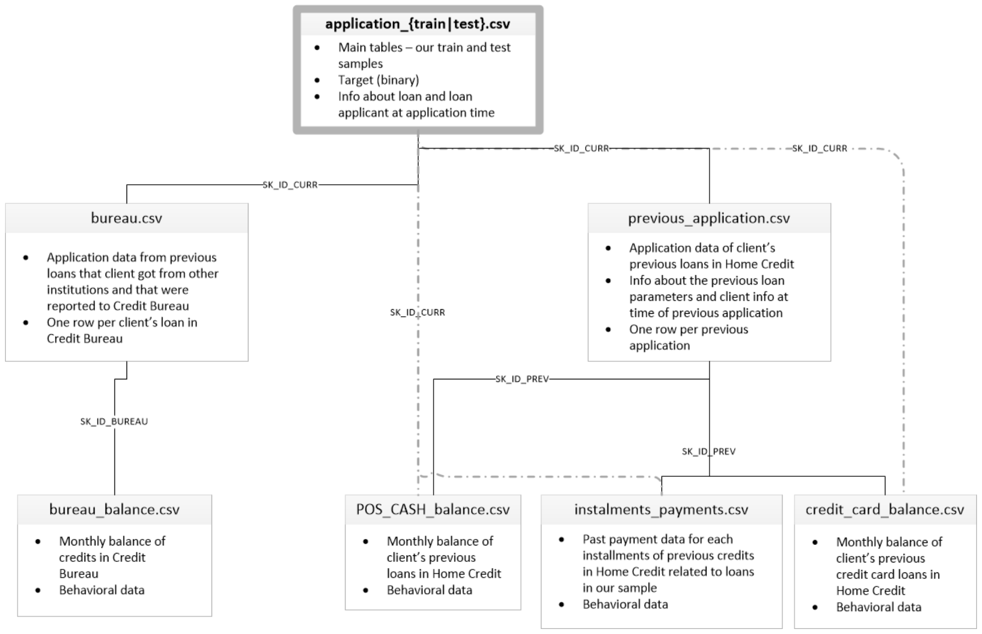
1.1 Dataset

The provided dataset is divided into multiple relational tables containing details of many borrowers (see Figure 1), which is a standard supervised classification task (0: non-defaulters, 1: defaulters) and is an imbalanced dataset, since defaulters are in the minority of applicants.

- **application_{train|test}.csv** The main training and test data, containing information about each loan application, with one row representing one loan. Identified by the characteristic SK_ID_CURR. TARGET indicates 0:loan repaid or 1:loan not repaid.
- **bureau.csv** Customer's credit history. Each previous credit has its own line in bureau.
- **bureau_balance.csv** Monthly data of previous credits in bureau.
- **POS_CASH_balance.csv** Monthly customer data from previous points of sale or cash loans at Home Credit.
- **credit_card_balance.csv** Monthly data of customers' previous credit cards used at Home Credit.
- **installments_payments.csv** The customer's previous loan payment history with the Home Credit.
- **HomeCredit_columns_description.csv** Provides definitions for all columns.

1.2 Machine Learning Problem Description and Analysis

This is a supervised learning classification problem that contains training data points and relevant category labels. For a given user data, we need to predict the category labels associated with that client. And this is a binary classification with only positive (1) and negative (0). In addition to that, we mentioned in the previous section that this is an imbalanced dataset, so we use the ROC AUC



application_{train|test}.csv
Main tables – our train and test samples
Target (binary)
Info about loan and loan applicant at application time
SK_ID_CURR
bureau.csv
Application data from previous loans that client got from other institutions and that were reported to Credit Bureau
One row per client's loan in Credit Bureau
previous_application.csv
Application data of client's previous loans in Home Credit
Info about the previous loan parameters and client info at time of previous application
One row per previous application
SK_ID_BUREAU
SK_ID_PREV
bureau_balance.csv
Monthly balance of credits in Credit Bureau
Behavioral data
POS_CASH_balance.csv
Monthly balance of client's previous loans in Home Credit
Behavioral data
instalments_payments.csv
Past payment data for each installments of previous credits in Home Credit related to loans in our sample
Behavioral data
credit_card_balance.csv
Monthly balance of client's previous credit card loans in Home Credit
Behavioral data

Figure 1: Structure of Relational Tables.

metric. The metric is insensitive to category imbalance. It ranks the probability of predicting a positive category label and calculates the area under the ROC curve, the better the model the higher the score, with the metric ranging from 0 to 1. The simple random guess model has a ROC AUC of 0.5.

2 Exploratory Data Analysis (EDA)

Exploratory data analysis is first performed to examine the data for trends, dependencies, and anomalies, etc., to generate a summary of the data. And modeling choices are made based on the EDA results.

2.1 Basic Statistics

First load all 8 tables, check their shape and notice that the size of the dataset is very huge, containing hundreds of original features. The statistics on TARGET also verified that this is an imbalanced class problem. The main train and test tables using the unique ID (SK_ID_CURR) to refer to all other tables.

Target	Count	Rate
0	282686	0.919
1	24825	0.081

Table 1: Distribution of the Target Value.

2.2 Examine Missing Values

All tables contain a large number of missing values that need to be filled in when building a machine learning model. We counted the missing values for each column of each table. In future work, we can choose to use models such as XGBoost, LightGBM and Boosting Models which can handle missing values, or drop columns with a high percentage of missing values. After a missing value analysis, we found that training dataframe has 121 features and 67 of them have missing values.

Column	Number of missing	Missing rate
COMMONAREA_MEDI	214865	0.698723
COMMONAREA_AVG	214865	0.698723
COMMONAREA_MODE	214865	0.698723
NONLIVINGAPARTMENTS_MEDI	213514	0.694330
NONLIVINGAPARTMENTS_MODE	213514	0.694330

Table 2: Top5 features with most missing values.

Column	Number of unique categories
NAME_CONTRACT_TYPE	2
FLAG_OWN_CAR	2
FLAG_OWN_REALTY	2
EMERGENCYSTATE_MODE	2
CODE_GENDER	3

Table 3: 5 examples object-type columns (16 columns in total)

2.3 Categorical Variables and Encoding

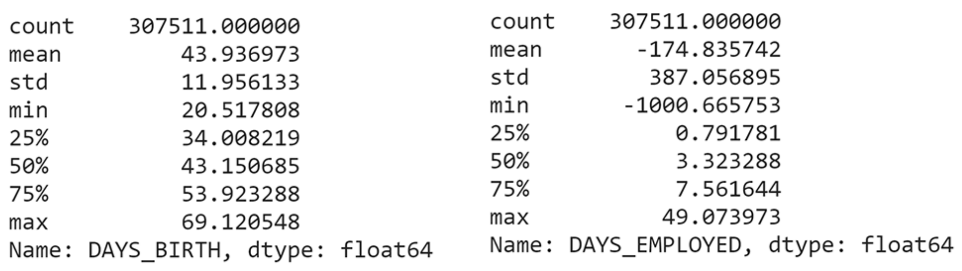
The object column contains strings, which are categorical features that count the number of unique entries in each object column (see Table 3) Most categorical variables have only a relatively small number of unique entries, these variables need to be encoded as numbers. The training data set has 16 object columns in total, and we also analyze the number of unique categories for each of them.

There are two encoding methods, Label encoding and One-hot encoding. Since the value of label encoding to each category is random, the same processing again for more than two unique category labels may be completely different, but One-hot encoding is the safe choice at this case. However, the number of One-hot encoding features will increase drastically due to the increase in the number of categorical variable categories. So we finally choose to use Label encoding for categorical variables with only two categories and One-hot encoding for categorical variables with more than two categories.

Hot coding creates more columns in the training data and requires aligning the data frame and removing more columns to ensure that the same columns need to be present in the training and test data.

2.4 Anomalies

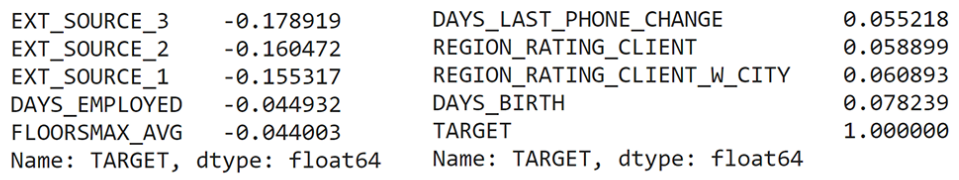
There may be exceptions in the data due to input errors or equipment errors. We use the description method to view the statistics of the columns to detect anomalous values. First we process the DAYS_BIRTH column (divide by -365) to convert the data into units of years (see Figure 2). There appears to be no anomalies, and all the customer ages look reasonable.



```
count    307511.000000
mean         43.936973
std          11.956133
min          20.517808
25%          34.008219
50%          43.150685
75%          53.923288
max          69.120548
Name: DAYS_BIRTH, dtype: float64
```

```
count    307511.000000
mean       -174.835742
std         387.056895
min       -1000.665753
25%           0.791781
50%           3.323288
75%           7.561644
max          49.073973
Name: DAYS_EMPLOYED, dtype: float64
```

Figure 2: Statistics of DAYS_BIRTH and DAYS_EMPLOYED column.



```
EXT_SOURCE_3    -0.178919      DAYS_LAST_PHONE_CHANGE        0.055218
EXT_SOURCE_2    -0.160472      REGION_RATING_CLIENT          0.058899
EXT_SOURCE_1    -0.155317      REGION_RATING_CLIENT_W_CITY   0.060893
DAYS_EMPLOYED   -0.044932      DAYS_BIRTH                    0.078239
FLOORSMAX_AVG   -0.044003      TARGET                        1.000000
Name: TARGET, dtype: float64   Name: TARGET, dtype: float64
```

Figure 3: Top5 negative and positive correlated columns to TARGET.

Looking at the Days Employed DAYS_EMPLOYED column in the same way reveals that there are anomalies in the data. The maximum value ("min" in the figure after transformation) is about 1000 years (see Figure 2), which does not correspond to the actual situation. Besides, after an extra analysis for the default rate, we found that it seems that anomalous customers have a lower result than normal customers (see Table 4).

Group type	Default rate
Anomalies	5.40 %
Non_anomalies	8.66 %

Table 4: Default rate of group with(without) anomalies.

We use np.nan to fill the exceptions and create a flag column to indicate the value was anomalous or not. The processed data looks as expected. Note that the operation of creating new columns and populating them is done for both training data and test data.

2.5 Correlation Analysis

In this section we calculate the **Pearson** correlation coefficient between each variable and the target using the **corr()** method in DataFrame. Figure 3 shows the relevance to target and sort. The results show that DAYS_BIRTH is the most positively correlated, which indicates that the older the customer is, the less likely he or she is to default.

To visualize the effect of age on the target, Figure 4 is a kernel density estimate (KDE) plot with TARGET as the variable. KDE shows the distribution of individual variables. the TARGET== 1 curve is skewed towards the younger end. Although this is not significant (-0.07 correlation coefficient), it does have an effect on TARGET. In the next step we grouped the ages in intervals of 5 years and calculated the mean of TARGET. The results obtained are shown in Figure 5. The results show a clear trend that younger applicants are more likely to not repay their loans. So it is necessary for lenders to take some precautions.

3 Feature Engineering

Feature engineering is the pipeline aiming at improve the following machine learning modeling by constructing or selecting features that are more important or related to the TARGET value. Combined with the DEA analysis part, it will provide suggestions for the features chosen in the following modeling part. In this work, we tried to create some new features based on the most significant or finance-related features. The methods we tried here are polynomial methods and prior knowledge based methods.

3.1 Polynomial Methods

We created polynomial features with EXT_SOURCE variable and DAYS_BIRTH variable and applied a degree of 3 to see the results. These features are previously identified as most correlated with the target values. The new features were named using get_feature_names. The correlation between the top correlated new features and TARGET(see Table5) have shown that the feature engineering increased the correlation of the features a little, however, we do not know whether such kind of increasing will bring some benefits to our modeling work.

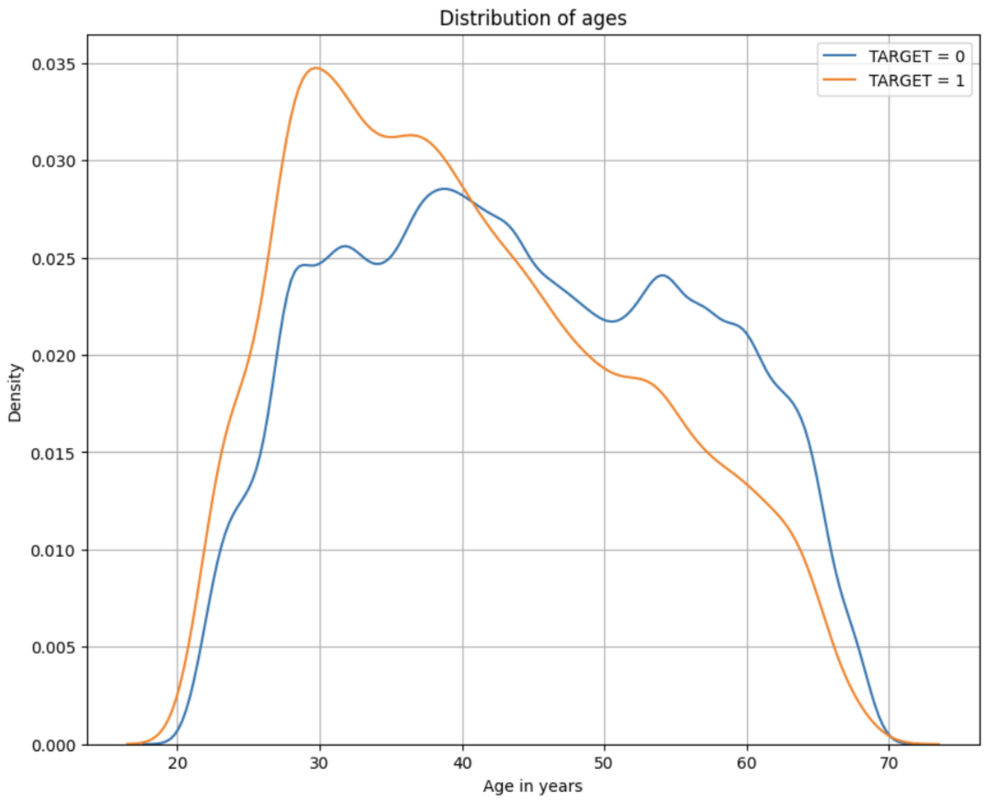


Figure 4: KDE plot of DAYS_BIRTH column.

Feature	**Correlation value**
EXT_SOURCE_2 EXT_SOURCE_3	-0.193939
EXT_SOURCE_1 EXT_SOURCE_2 EXT_SOURCE_3	-0.189605
EXT_SOURCE_2^2 EXT_SOURCE_3	-0.176428
EXT_SOURCE_2 EXT_SOURCE_3^2	-0.172282
EXT_SOURCE_1 EXT_SOURCE_2	-0.166625
EXT_SOURCE_1 EXT_SOURCE_3	-0.164065
EXT_SOURCE_2	-0.160295
EXT_SOURCE_1 EXT_SOURCE_2^2	-0.156867
EXT_SOURCE_3	-0.155892
EXT_SOURCE_1 EXT_SOURCE_3^2	-0.150822
EXT_SOURCE_1 EXT_SOURCE_2 DAYS_BIRTH	0.155891
EXT_SOURCE_2 DAYS_BIRTH	0.156873
EXT_SOURCE_2 EXT_SOURCE_3 DAYS_BIRTH	0.181283

Table 5: Correlation value of top correlated features

3.2 Prior Knowledge Based Methods

Referring to Aguiar's work, we apply some knowledge in the field of finance to determine whether a customer will default on a loan using some characteristics. In this part, we have tried to create several new features based on the known knowledge, and the features we would like to create are: CREDIT_INCOME_PERCENT(see Equation 1), ANNUITY_INCOME_PERCENT(see Equation 2), CREDIT_TERM(see Equation 3), DAYS_EMPLOYED_PERCENT(see Equation 4).

$$CREDIT\_INCOME\_PERCENT = AMT\_CREDIT/AMT\_INCOME\_TOTAL \tag{1}$$

$$ANNUITY\_INCOME\_PERCENT = AMT\_ANNUITY/AMT\_INCOME\_TOTAL \tag{2}$$

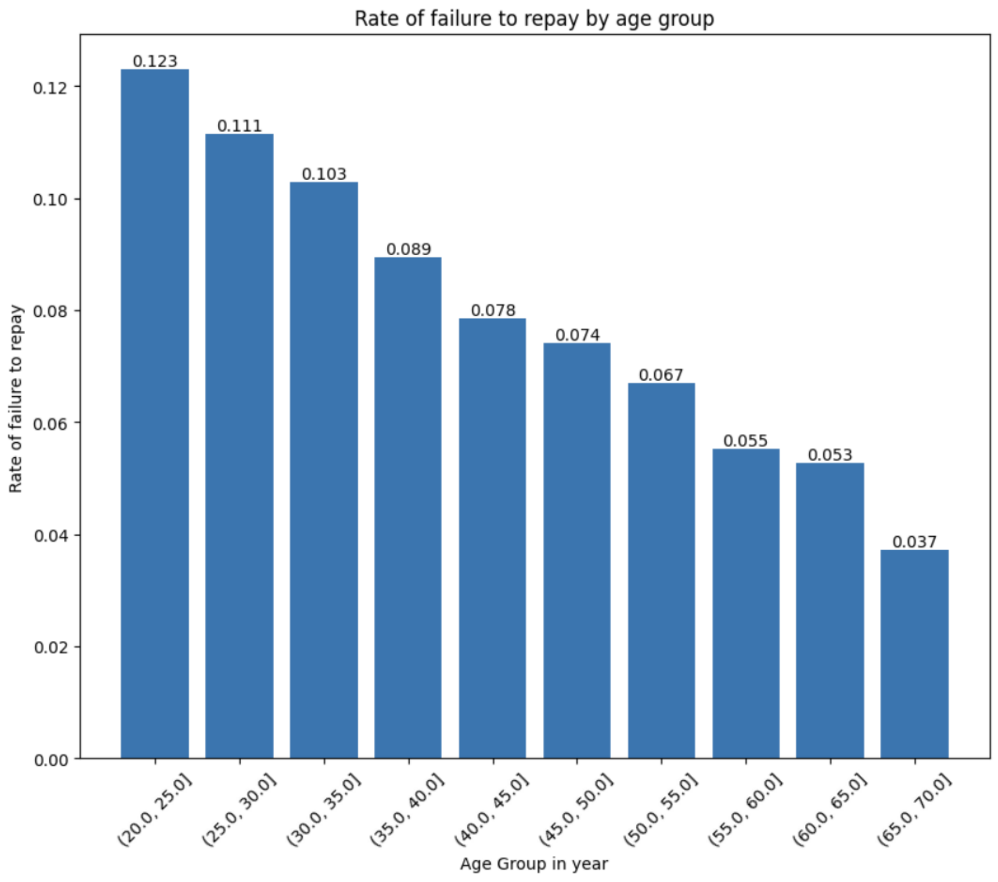


Figure 5: Bar plot of failure rate to pay in different age group.

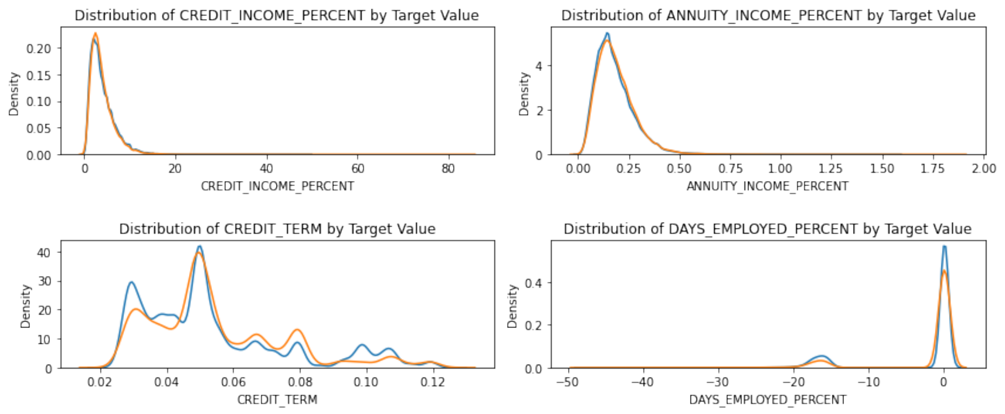


Figure 6: KDE plot of all the prior knowledge-based features.

$$CREDIT\_TERM = AMT\_ANNUITY/AMT\_CREDIT \tag{3}$$

$$DAYS\_EMPLOYED\_PERCENT = DAYS\_EMPLOYED/DAYS\_BIRTH \tag{4}$$

We have also investigated the KDE plots of all the above created features to see if there are any significant differences for different TARGET groups. Only CREDIT_TERM show some differences between the two groups and such kind of features maybe useful for further machine learning models(see Figure 6).

4 Machine Learning Modelling

4.1 Baseline - Logistic Regression

After pre-processing the missing values in the data and normalizing the range of some features, we can use the machine learning methods to learn the distribution of the data set. As for the baseline model, we choose the logistic regression, which is a simple methods to estimate the linear correlation between the various input feature and the output result. To achieve this baseline, we will just use all of the features after pre-processing.

For this baseline, we use **LogisticRegression** from **Scikit-Learn** to achieve it. To avoid the overfitting problem in our model, we choose a low regularization parameter $C = 0.001$. Other parameters we only use the default setting to get the experiment result.

The private score of the logistic regression is 0.68107, public score is 0.67715. It is not a bad result in such a complex data set for the baseline. The types of the input feature is too much. But the logistic regression only consider all the factor in the same focus. It means that logistic regression does not correctly value the most important factors and does not exclude the influence of some useless factors. In order to solve this problem, we have tried different algorithms based on the process in the baseline.

4.2 Random Forest Classifier

To improve the poor performance in our baseline logistic regression, Random forest can be used to train a better model. The random forest can handle many factors in using hundreds of trees in the model, which may be better suited to this kind of multivariate data. But after we got the result, the private score of the random forest is 0.68009, public score is 0.67877. Its performance is similar to the logistic regression, which is not good as the theory. The reason is that the importance of each feature is not correctly analyzed in random forest. It means that this kind of method can not be perform well in this data set.

Based on the feature engineering, we try to use partly features in the data to train the random forest. Althought it can not perform good as the next two algorithms, it improves the baseline random forest model. It means that our feature engineering is meaningful in this project.

4.3 XGBoost

Extreme Gradient Boosting(XGBBoost) is an integrated machine learning algorithm which based on Gradient Boosting Decision Tree(GBDT). XGBoost introduces tree regularization method to alleviate the problem of model overfitting which is quite useful in many data. It performs a second-order Taylor expansion on the loss function, using both first and second-order gradients. With these setting, XGBoost is generalization method in various data set. XGBoost itself cannot handle categorical variables, and instead accepts only numerical data, like a random forest.

Therefore, before classifying data can be fed into XGBoost, it must be processed through various encoding methods, such as marker encoding, mean encoding, or unique thermal encoding. Thus, we just reuse the process in the baseline. After training, we can get the private score is 0.72356, public score is 0.72278. The result shows that the XGBoost has strong ability to fit our data ranther than the previous methods.

4.4 LightGBM

Lightgbm uses the decision tree algorithm based on histogram, which has advantages in computational cost. XGBoost adopts the Level-Wise tree growth strategy, while LightGBM adopts the Leaf-Wise growth strategy, which is oriented by maximum information gain. In this kind of strategy, the model can convergence easily but the risk of overfitting increases.

As shown in Table 6, we have tried different parameters of the LightGBM. The result shows that in the parameter whose learning rate is 0.06 and reg alpha is 0.1 we can get the best result, which is 0.74579 in private Score and 0.74427 in public score.

Methods	learning rate	reg_alpha	Private Score	Public Score
LightGBM	0.05	0.1	0.73734	0.73397
LightGBM	0.06	0.1	0.74579	0.74427
LightGBM	0.065	0.1	0.74618	0.74362
LightGBM	0.06	0.2	0.73784	0.73594

Table 6: Comparison of LightGBM with different parameters.

5 Conclusions

In this project, we utilize the various machine learning methods to solve the home credit default risk problem. In the first stage, we do the exploratory data analysis to generate a clean data set which can be directly used in various machine learning algorithms. Moreover, we analyze the correlation among different features. In the second stage, we use both Polynomial methods and Prior knowledge based methods to construct the important feature in our data. In the last stage, we do the experiments in various algorithms based on above analysis. The result is shown in Table 7, LightGBM achieve the State-of-the-Art in this data set.

Methods	Private Score	Public Score
Logistic Regression	0.68107	0.67715
Random Forest	0.68009	0.67877
XGBoost	0.72356	0.72278
LightGBM	0.74579	0.74427

Table 7: Comparison of different method.

Contribution

Hao WU was responsible for the modeling part. **Yiming ZHU** was responsible for the EDA part. **Jihong TANG** was responsible for the Feature Engineering section. **Xian WANG** was responsible for writing the Report outline, leaving the data section empty. All team members participated in making the slides and attended the video presentation. The group members communicated and cooperated well throughout the collaboration period, and tasks were well distributed.