Machine Learning System for Home Credit Risk Competition

Haoran Li, Jiaxin Bai, Qi Hu, Ying Su [*]

Abstract

Home Credit Default Risk is a Kaggle Competition that aims to predict whether a client will repay its loans or not. It is helpful for both financial services and clients for healthy and safe borrowing experiences. Given the data about clients' loan applications, previous credits with monthly balances (previous point of sales are also included), credit card balances, previous loan applications and corresponding repayment history, the competition is designed to demonstrate the potential of machine learning engineering. Such complicated and heterogeneous datasets not only require powerful models that is able to generalize well, but also need dedicated data pre-processing and feature engineering. In this project, we explore various pre-processing, engineering and model tricks to see what actually helps improve the performance. We conduct extensive experiments to show our implementation can achieve competitive results with straightforward intuitions.

1 Introduction

Credit system is a foundation of modern financial system. Financial institutions decide whether to provide help and get benefits by assessing the credit level of borrowers. However, insufficient or non-existent credit histories make many people difficult to get loans, which unfortunately is often exploited by untrustworthy lenders.

To solve the problem, Home Credit tries strives to provide financial services for unbanked population. The repayment abilities is essential for positive loan experience. Home credit makes use of a variety of alternative data–including telecommunications and transactional information– to evaluate the abilities. Though Home Credit has already been using machine learning methods in their bussiness, to fully tap the potential of their data, they announce the Kaggle Challenge Competition, hoping to provide better experience for their customers.

2 Problem Formulation

Home Credit Default Risk is a competition organized on Home Credit and realesed on kaggle platform. The completion is designed to help Home Credit design a positive and safe borrowing experience.

2.1 Data

There are seven different sources of data, namely, application_train(test).csv, bureau.csv, bureau_balance.csv, POS_CASH_balance.csv, credit_card_balance.csv, previous_application.csv, installments_payments.csv. The descriptions for each column in different data file are included in HomeCredit_columns_description.csv.

[*] Names are listed alphabetically.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Application_train(test).csv is the main training and testing data with information about each loan application. Training data contains a TARGET column indicating whether the loan was paid. For each loan, a feature id SK_ID_CURR is used as identification.

Bureau.csv and bureau_balance.csv describes the clients' previous credits from other financial institutions and monthly data about the previous credits in bureau respectively. Each row is identified by feature SK_ID_CURR or SK_ID_BUREAU.

Prevoius_application.csv contains the information about previous loan applications in Home Credit. Details of related data are in **POS_CASH_balance.csv, credit_card_balance.csv, and installments_payments.csv**. Each row is identified by feature SK_ID_CURR or SK_ID_PREV.

2.2 Task formulation

For the Home Credit Risk task, the objective is to build an robust system by using machine learning methods with the historical financial and heterogeneous data to predict the probability of an application repaying a loan.

To build the system, normally three stages are involved. First, a feature pre-processing module creates the features from tables by commonly used operations such as groupby and aggregation. Second, a feature engineering module extracts or refines the raw feature so as to provide engineered features as the input to machine learning models. Finally, given the input feature, a model learns the features and makes final predictions for the task.

3 Feature Pre-processing

In this section, we briefly talk about feature pre-processing techniques we used for the project. After manual inspect of the Home Credit Default Risk dataset, we notice that the data is highly heterogeneous and several columns are incomplete (missing values). Moreover, the data types are not unified, some of columns use categorical values while other columns are based on numeric values. Thus, it is crucial to pre-process the dataset before feature selection and model training.

3.1 Missing Values

For the Home Credit Default Risk dataset, there are 217 columns in total and around 47% columns that contain missing values. What's worse, 10% columns have 60% empty rows. Such features with high volume of missing data may not help the model learning. Therefore, we consider using columns dropping and imputation to handle the missing values.

Columns dropping: dropping the whole column when the column has too many missing values is the direct way for model training. Hence, we perform columns dropping for both original features and aggregated features when the number of missing values exceeds the given threshold (in our case, the threshold is set to be 0.6).

Imputation: after columns dropping, there are still many columns with no more than 60% missing values that may be beneficial for model learning and we cannot just discard these columns. So we consider using Imputation by adding the median values to missing values so that our model can exploit those features to improve its learning.

3.2 Data Aggregation

Features with individual records like "MONTHS_BALANCE" in "bureau_balance" dataset can be summarized when modelling a single user. Intuitively, we are more interested in the summary statistics of a user's behavior than this user's individual records. For rows with the same "SK_ID_CURR", we group these records together to obtain the summary statistics like max,min,mean and sum.

3.3 Categorical Values

Besides numerical values, there are categorical values we need to care about. We perform one-hot encoding on those categorical values by appending the feature sizes on a single column.

4 Feature Engineering

Automated feature engineering is important in the system. Specifically, feature selection or dimensionality reduction can help improve estimator's accuracy scores or to boost the performance on high-dimensional datasets.

4.1 Correlation selection

Correlations measures are widely used for select the useful features. In this method, we first compute the correlations between the input and output variables, and remove those variables that have a low correlations. Since the variable data types of our dataset are all numerical, we measure the correlation between the input and output by using Person's correlation.

4.2 Backward Elimination

The idea of backward elimination is to remove the features that do not have significant effects on the model output for a fitted model. In a backward elimination process, we first train the model with all features, then compute the feature importance of all input features. After this, the features that do not have significant effects on the output are eliminated. Finally we use the remaining features to train the model again. This process can also be conducted multiple times until all the parameters are statistically significant.

4.3 PCA based

Principal component analysis (PCA). Linear dimensionality reduction using Singular Value Decomposition (SVD) of the data to project it to a lower dimensional space Abdi H et al. [2]. The input features are firstly projected into low dimensional by an unsupervised feature extraction module PCA before learned by estimators. We use the PCA tool from scikit-learn package.

4.4 Polynominal Features

The input features are interacted in nonlinear ways. Though these interactions can be modeled by a learning algorithm, we try to expose these features by simple modeling algorithms in feature engineering stage. we generate features that are powers of existing features as well as interaction terms between existing features like features normalized score from external data source. We use PolynomialFeatures funtion from scikit-learn package to generate the power of normalized external scores and their interaction.

4.5 Domain Knowledge Features

Some features cannot reflect clients' financial conditions directly, and are influenced by other features. In feature processing stage, we can handcraft some domain knowledge feature by our understanding of credit system. For example, the income and the credit cannot refelct the risk of bad debt, as it has different risk to lend same amount of money to different people, therefore, using our commonsense knowledge, we can generate a new features about the percentage of credit within income. By our understanding of credit, we can generate a bunch of features, like the percentage of credit within asset, the percentage of children within families etc.

5 Model and Ensemble

In this section, we introduce the details of models we use to learn the home credit data. We choose ensemble-based methods, combining predictions of several base estimators built with a given learning algorithm in order to improve generalizability and robustness over a single estimator. Specifically, averaging methods aim to reduce the variance (e.g., random forest) and boosting methods aim to reduce the bias (e.g., light GBM).

Model	Private Score	Public Score
RandomForest	70.88	71.10
Extra-Trees	71.95	71.61
Light GBM	77.78	77.41
Ensemble	77.76	78.00

Table 1: Result of different models and ensemble model from multiple light GBM.

5.1 Random Forest

The RandomForest algorithm and the Extra-Trees method. A diverse set of classifiers with randomness are created and the final prediction is given based on the avarage prediction of the individual classifiers Breiman L et al. [1].

5.2 Light GBM

LightGBM is a gradient boosting framework that uses tree based learning algorithms Ke G et al.[3]. It is of high efficiency, accuracy and interpretability. It achieves the speeding up of training process by sampling larger gradients and exclusive feature bundling.

5.3 Cross Validation

As there is no officially released development dataset for model selection, we use cross validation [2] to select the model for evaluation. In the commonly used k-fold cross validation, the training set is split into k splits and follows the the training procedure as follows:

- A model is trained with $k - 1$ splits of the training data;
- The test data and the remaining split of training dataset are evaluated on the resulting model.

Average of the results of the test data is used as the final prediction for the model.

5.4 Ensemble

We average the results from individual estimators as the final ensemble result. We mainly consider averaging the prediction results from 5 models with various feature engineering techniques as mentioned in Section 4. All 5 models are based on light GBM and can be summarized as follows:

- Vanilla GBM trained with all features after feature pre-processing.
- Poly-feature and domain knowledge GBM: GBM trained with feature importance filtering and generated polynominal features and domain knowledge features.
- Corr-GBM: The GBM methods using the selected features obtained from correlation filtering.
- Backward-GBM: The GBM methods using the selected features obtained from backward elimination.
- PCA-GBM: The GBM method using the selected features obtained from PCA dimensionality reduction.

6 Analysis

6.1 Averaging methods and boosting methods

The results of different models is shown in 1. As we can see, among the results, boosting method Light GBM is better than averaging method RandomForest or Extra-Trees. It shows the effectiveness of gradient boosting framework.

[2] https://en.wikipedia.org/wiki/Cross-validation_(statistics)

Data Constitution	Private Score	Public Score	Feature Dim
Train	74.61	74.45	241
Train+Bureau	75.86	75.75	452
Train+Bureau+Previous	77.96	78.02	1411

Table 2: Ablation study on data merging.

Filter threshold	Num Features	CV Train	Private Score	Public Score
0.000	243	0.763	0.753	0.754
0.003	182	0.763	0.753	0.754
0.006	149	0.762	0.753	0.753
0.009	128	0.762	0.753	0.753
0.012	104	0.762	0.753	0.754
0.015	92	0.753	0.734	0.731

Table 3: Correlation-based feature selection results

As Light GBM achieves decent performance by training a single model, we further performe different feature selection methods on Light GBM based model and do an ensemble. For ensemble method, we simply use the averaged scores, which achieves further preformance gain compared to individual models. Details of individual results are illustrated in section 5.4.

6.2 Advantanges of Pre-processing and Data Merging

Feature importance of aggregated statistics. Here we study the correlation between extracted features and ground truth labels to see if our pre-processed features are helpful or not. We find that most significant correlations come from columns EXT_SOURCE_3, EXT_SOURCE_2 and EXT_SOURCE_1 in the original application_train dataset. This indicates that EXT_SOURCEs are useful features. Still, we can find 16 aggregated features during pre-processing out of top 20 features with largest magnitude of correlation. This result of correlations show that our pre-processed features for statistics aggregation indeed related to the labels.

Ablation study on data merging. Here we perform the ablation study to see if we can gain improvement after merging more data. We use the same pre-processing techniques and models for a fair comparison. We mainly consider all dataset as three data types: current applications used for training set (**Train**), credits provided by bureau (**Bureau**) and previous relevant information (**Previous**). The results are shown in Table 2, and consistent improvement can be observed after including more and more data.

6.3 Feature selection

In this part, we separately analyze the effects of different strategies of feature selection.

6.3.1 Correlation-based filtering

In this method, we used the correlation based filtering to remove redundant features in our model. It is observed that when we use the correlation-based method, we can remove one-hundred and forty features and the final results does not change too much. However, it is observed that if we remove additional twelve features, the final performance starts to drop.

6.3.2 Backward Elimination

On the other hand, we also used the backward elimination to eliminate the redundant features. The conclusion is similar to the correlation-based method that we can remove more than half of the feature without affecting the final performance.

6.3.3 PCA dimensionality reduction

In this part, we analyze the result of ablation study on dimension of PCA based feature selection method. From the result shown in Table 5], we can see that PCA based methods decrease the

Eliminated Features	Num Features	CV Train	Private Score	Public Score
0	243	0.763	0.753	0.754
30	213	0.763	0.753	0.754
60	183	0.763	0.753	0.754
90	153	0.762	0.753	0.754
120	123	0.763	0.753	0.754
150	93	0.763	0.753	0.754

Table 4: Backward elimination filtering results

Exp #	Private Score	Public Score	Feature Dim
original	77.78	77.41	1539
PCA#1	73.78	75.16	128
PCA#2	74.68	75.36	256
PCA#3	75.22	76.19	512
PCA#4	75.64	76.66	1024

Table 5: Ablation study on PCA based feature selection.

performance both on the private score and the public score as the dimension decreases. All the features from 7 csv files are used as original input. From feature dimension 1539 to feature dim as 128, the performance on the both splits drops about 3%, while the feature dimension is reduced to 8.3% of the original. This shows that a small portion of the features can be used to model the system with a decent performance, which is quite important and useful when computation resource is limited.

6.4 Feature Generation

In this part, we separately analyze the effects of different strategies of feature generation – polynominal features and domain knowledge features. In polynominal features generation, we generated 30 features about the power of externel data and their interaction. In domain knowledge features generation, we generated 18 domain features using our expert knowledge.

Ablation study on feature generation Here we perform the ablation study to see the effectiveness of the generated data. The results are shown in Table 6, the domain knowledge generation can significantly improve the model performance.

7 Contribution

- **Jiaxin Bai**. Jiaxin Bai is responsible for the feature selection methods, including the correlation selection and backward eliminations (Section 4.1 and Section 4.2). Also he conducted analysis on the experimantal results (Section 6.3.1 and 6.3.2).
- **Qi Hu**. Qi Hu is responsible for the feature generation, including the polynominal features and domain knowledge features (Section 4.4 and Section 4.5). Also conducted analysis on experimental reuslts (Section 6.4)
- **Haoran Li**. Haoran Li is responsible for the feature pre-processing (Section 3), feature importance and ablation study on data merging (Section 6.2).
- **Ying Su**. Ying Su is responsible for the problem formulation (Section 2), modeling part (Section 5.1-5.3), and ablation study on PCA-based feature selection (Section 6.3.3)

Data Constitution	Private Score	Public Score	Feature Dim
Original	77.61	77.59	285
Original + Polynominal	77.65	77.65	315
Original + Polynominal + Domain	78.25	78.43	333

Table 6: Ablation study on data data generation.

References

[1] Breiman L. Random forests[J]. Machine learning, 2001, 45(1): 5-32.

[2] Abdi H, Williams L J. Principal component analysis[J]. Wiley interdisciplinary reviews: computational statistics, 2010, 2(4): 433-459.

[3] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree[J]. Advances in neural information processing systems, 2017, 30.