Home Credit Analysis Report

Shiying Wang, Qing Gao, Ruochen Zhong, Zhixin Zheng, Tianrun Zhu

ABSTRACT

An approach to identify what factors contribute to clients' default rate and evaluation on lending risk to a specific client by using past data is described in this paper. We use feature engineering to extract and select useful features in order to determine the ability of clients to repay the loan on time. Through the investigation of some trending classification algorithms and the help of the ensemble learning, we hope to improve the performance of our final model. The result shows that our best model is LightGBM with combined data and upsampling method. To go beyond the current model performance, we need more information about clients and more estimators in the LightGBM model to identify clients' default risk.

I. INTRODUCTION

Home Credit is an international non-bank, consumer finance group. It provides loans to people who need financial support and to those who have insufficient or non-existent credit histories due to lack of bank accounts. In order to provide loan services to a wider range of population such those are disqualified due to less or no bank histories, there will exist a higher risk for providing loan to this kind of clients which they may have a weaker ability to handle financial difficulties and default their loans due to multiple conditions or struggles. Our purpose is to evaluate each client by different models is trained with Home credit provided data, and try to determine what types of clients most likely default.

II. EXPLORATORY DATA ANALYSIS

We have in total of seven datasets downloaded from Kaggle, which the main dataset "application_{train|test}.csv" (122 features) includes information about loans and loan applicants at application time.[1] In order to have a more comprehensive profile of clients, we decide to combine features all seven datasets. Therefore, the final dataset we are using contains 339 features.

Data Overview. To better understand the recorded clients in the Home Credit datasets, we visualize the distributions of their family status, educational background, income type, and house type. From Figure 1, despite the client's ability to repay the loan, there is a majority group of clients in each feature signed contracts. According to these distribution graphs, our first instinct to conclude the majority group of people who need financial support are married working people who also have a high school diploma and own a house/apartment.

Data Preprocessing. Throughout data exploration and a trail with some algorithms, we notice there are several issues that we need to deal with before formal modeling: 1) The data is highly unbalanced: approximately 8.07% of Target is in group 1 (unable to repay the loan) while around 91.93% of Target is in group 0 (other cases), and we will use Upsampling and Downsampling to resolve the problem. 2) In the other six datasets, each client has several records, so we decide to deal with the numeric features by merging each of them by their mean, minimum, maximum, and sum and the categorical features by building several new columns to present each classes' value count in one categorical feature or by choosing the most frequent class of each client. 3) A high proportion of the missing

value for some features. We dropped features with more than 40% of missing data. After that, we used a heatmap to observe missing value correlation and determine to drop features which are highly uncorrelated with the TARGET feature. For those correlated features, we fill missing values with the median for numeric features and use 'XNA' (not available) to replace the missing value.

Feature Selection. We first manually select some features based on our domain knowledge, and we combine some categories in certain features based on their distributions in target 0 and target 1. We also use Random Forest Feature Importance to select the feature where we remove the features with 0 importance. The higher frequency of using some features to split the decision tree means these features will cause a smaller entropy or impurity. It indicates that each decision tree may have a better performance on classifying new observation with proper hyperparameter tuning. We build 1000 estimators as trees in the forest to see the feature importance. We also try to use hierarchical clustering to split the features into 4, 6 and 12 groups, and we remove those that were not in the same group with our Target; however, the results do not improve at all, so we decide to keep those features.

	feature	importance
241	AMT_CREDIT_SUM_DEBT_MAX	1.902438e-02
7	DAYS_EMPLOYED	1.520280e-02
3	AMT_ANNUITY	1.467738e-02
94	CNT_PAYMENT_Pre_sum	1.390115e-02
264	NAME_EDUCATION_TYPE	1.268074e-02
116	AMT_PAYMENT_Ins_sum	1.241854e-02
32	DAYS_LAST_PHONE_CHANGE	1.241790e-02
234	DAYS_CREDIT_MEAN	1.231633e-02
99	Own_payment_Ins_mean	1.224691e-02
115	AMT_INSTALMENT_Ins_sum	1.115256e-02

Table 1

Table 1 is the top-ten result of importance based ranking, which implying these features are more frequently chosen when splitting dataset and produce smaller entropies. It indicates these features contain more useful information and are more capable to train a higher performance model than other lower importance score features.

In Figure 2, we visualize some important features chosen from the feature selection of random forest. The distributions of these features are slightly different comparing Target 1 and Target 0 classes. For the 'Age' feature, Target 1 group is relatively younger than Target 0, which indicates that young people are more likely to be evaluated as an unqualified client by Home Credit. For the 'Last Phone Changed' feature, it suggests that clients who have consistent phone are more likely to repay the loan on time. For the 'Days Credited' feature and the 'Days Employed' feature, they imply for those who have longer credit time and days of employed record are more likely to repay the loan on time. The different distribution of Target 1 and Target 0 tell us those young people who don't have enough income, working experience or reachable contact information are more likely to be classified as Target 1.

After selecting features and dealing with missing values, we have in total of 53 feature for base data and 263 features for the full data.

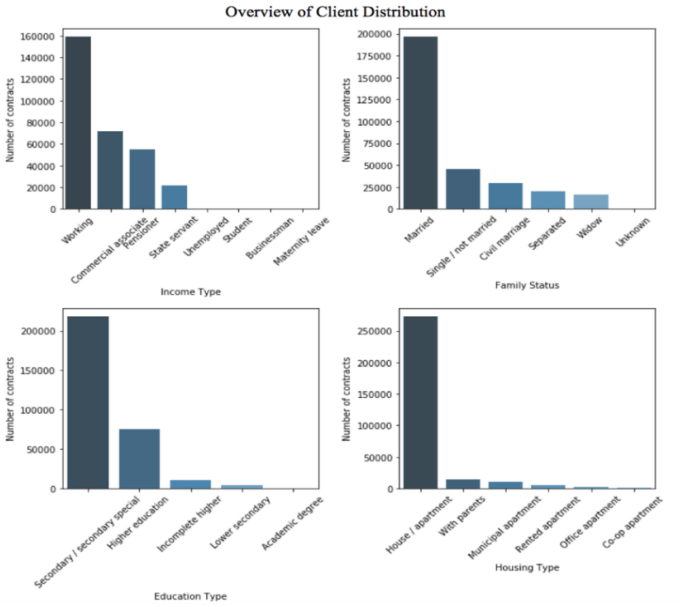

Figure 1

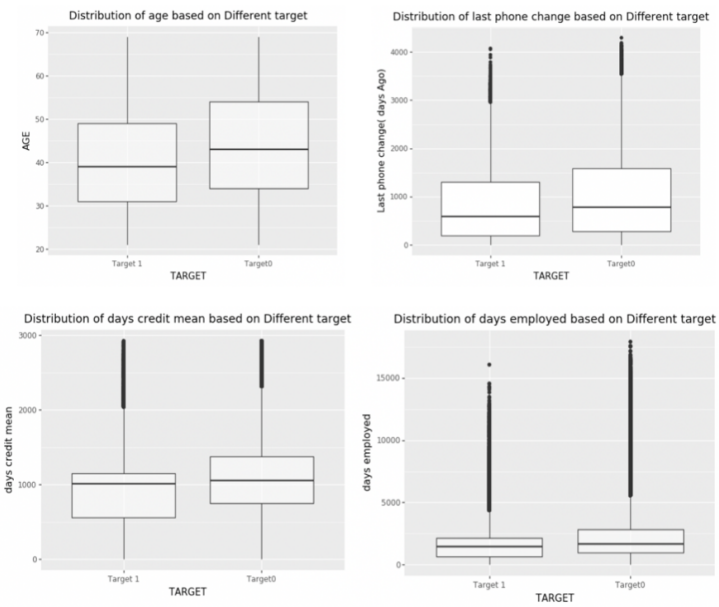

Figure 2

III. METHOD

A. Modeling

In order to achieve higher accuracy or AUC (Area under Curve) for the final model, here are some strong candidates for classification problem in traditional and modern machine learning algorithms:

Support Vector Machine (SVM). SVM is one of the most powerful traditional supervised learning algorithms for classification. It tries to find the maximum distance margin between two or multiple classes to create a more robust classification rule than perceptron or logistic regression. Since we are facing the classical binary classification problem, we attempt to use SVM with linear kernel and rbf Gaussian kernel to classify whether the target belongs to class 0 or 1.

Artificial Neural Network (ANN). ANN is one of the most versatile and powerful deep learning algorithms when we need to deal with traditional regression and classification problems. Each node in ANN solves a simple regression or classification problem, and these nodes function together and structure as a neural network. We will have a huge hypothesis set[1] given its complex structure. For instance, when we deal with a cat image classification problem, given enough high-quality dataset after a decent number of back propagation, each node in the ANN will find its different role such that some nodes will identify the color pattern and some nodes will identify the eye, the whisker, or etc. Thus, ANN can approximate some very complex target functions (solve some complex problems).

[1] Hopythesis set/ Function set: The complexities or the number of possible model for certain algorithms. Such that, Linear regression can only depict some simple linear problems which has sample complexity. ANN can approximate a lot of complex Target model which has a high complexity.[2]

K-Nearest Neighbor (KNN). KNN is a non-parametric method which can be used for both classification and regression. In the classification case, KNN can classify classes by non-linear and disjoint boundaries without the limitation of linearity. Since other features are not linearly correlated with our Target, we tried KNN to classify the Target. The classification boundary based on a majority vote of its k nearest neighbor, where k is the hyperparameter which is the number of the nearest neighbor. KNN uses the distance function to measure the nearest neighbors, and here we use Minkowski as the distance function.

Random Forest. Random Forest is a popular machine learning algorithm for classification by bagging method with multiple weak decision trees. In this ensemble, Random Forest constructs multiple decision trees with downsampling of the train data, make conclusions about our target value by the majority of the weak classifier. By the design of decision tree, random forest can automatically select a useful feature for reaching the lowest entropy when it splits dataset.

LightGBM. LightGBM is a trending machine learning algorithm due to its high efficiency and state of arts performance. It is a modified and more efficient version of Gradient boosting decision tree, where GBDT is a boosted decision tree by adding predictors to fit the residual errors from the previous predictor, compared to the original implementation of Gradient boosting tree which using all data points and all features. In order to speed up the training time without losing model accuracy, LightGBM modifies GBDT with downsampling method (Gradient-based One-Side Sampling) and feature merging (Exclusive Feature Bundling).

Gradient-based One-Side Sampling selects some datasets which have large gradients. These data points have large gradients can contribute more information for model training. Thus, lightGBM can get almost the same amount of information from the downsampling dataset which this smaller size of the dataset will speed up the training process.

Exclusive Feature Bundling combines or bundles some features by assuming most features in a large number of feature set are almost independent or mutually exclusive. Thus, these exclusive features can be combined together without losing a significant amount of information in an ideal situation.

By using GOSS and EBF, lightGBM can tremendously increasing training efficiency with the similar original GBDT's performance. These traits make lightGBM one of the most powerful boosting methods in modern algorithms.

B. Balancing Data

Upsampling. To solve the unbalancing problem, one of the methods is to use upsampling. The traditional way of upsampling is simply resampling the same percentage of each class with replacement until both classes are roughly the same number of observations. This may help the model to solve the highly unbalanced issue, but the new observation is duplicated from the original data. The model won't be able to gain more information to further understand or classify small percentage labels. It will only use the same data point to train the model. This may also cause overfitting in some cases.[3] Thus, We use an advanced up-sampling method called SMOTE (Synthetic Minority Over-sampling Technique). SMOTE doesn't generate duplicates from the minority class data. It produces 'synthetic' new data by random generate a data point within or between the chosen neighbors of each minority class data point.[3] This method will generate new information which may help some models' generalization ability.

Downsampling with Ensemble (Balanced Bagging). In order to solve the problem of unbalanced data, we also tried the method of downsampling to balance data. We hope to train a bunch of weak models with balanced

downsampling dataset. According to the ensemble method and the law of large numbers, the ensemble with a large number of weak but independent estimators will have a much higher accuracy in the hard voting classifier setting.

We split the data between Target 0 and 1 and selected 30% from both datasets as test data and the rest 70% as train data. We randomly select data from train data to ensure the proportion of target 1 is 50%, trained the balanced train data and predicted n times of randomly selected train data on the same test data.

IV. RESULT & DISCUSSION

Through investigation and exploration of some algorithms with the main dataset which have 53 features, we get the following result. By considering their performance without remedy and with remedy (resampling methods), we will pick some better performance models for further investigation given full dataset with 263 features. We hope to reach a higher performance in the AUC standard by the help of resampling methods and voting classifier.

TABLE 2

CONFUSION MATRIX INTRODUCTION[2]

	Actual Positive:Target 0	Actual Negative:Target 1
Predict Positive	TP: Actual Target 0 is predicted as Target 0	FP:Actual Target 1 is predicted as Target 0
Predict Negative	FN:Actual Target 0 is predicted as Target 1	TN:Actual Target 1 is predicted as Target 1

SVM. Although SVM can deal with unbalanced data, the model performance is inferior. The highly unbalanced distribution in the Target can affect SVM's ability to find a proper margin since there will be few data points in the minority class. For the linear kernel, the accuracy is 60.68%, and the confusion matrix as follows:

TABLE 3

CONFUSION MATRIX FOR SVM

	Actual Positive	Actual Negative
Predict Positive	56880	36482
Predict Negative	3415	4702

For rbf Gaussian Kernel, the accuracy is 63.38%, and the confusion matrix is similar to linear kernel.

As the confusion matrix shown, both Type I error and Type II error is high, and the accuracy is not high enough. On the other hand, the training time for both SVM models is more than 6 hours, which is relatively long. Due to the high cost of the training time and low model performance, we decided to stop using this model in our case.

ANN. With unbalanced data, we tried to use drop out method and L1 and L2 regularization methods to help the model to approximate the target function[3], but it still falls into the local minima which it predicts all new data point

[2] *Note that we use False Positive and Type I error exchangablely, and we use False Negative and Type II error exchangablely .*

[3] Target function: The model that generate the label or numeric value for regression or classification problem in the dataset. We try to use machine learning algorithm to approximate the target model for achieving a higher performance.[2]

as the class with a higher distribution. Therefore, we concluded that ANN is not capable to handle extreme unbalanced data well. With training model for 150 iterations in upsampling data, ANN results 48.01% accuracy and 0.6325 AUC score, which are worse than the random guessing. With the upsampling methods, ANN does avoid the problem which it guessing one class; however, its model performance was not good enough with around 400 total and more nodes in ANN. Thus, we also decided to stop using this model to seek further improvements.

KNN. Since KNN is based on voting in nearby neighbors, we do not need to use sampling methods in this model. We use voting classifier for 40 KNN with 5 neighbors in our based model to classify the class. The accuracy is 91.43%, but AUC score is only 0.5028. The confusion matrix as follows:

TABLE 4

CONFUSION MATRIX FOR KNN

	Actual Positive	Actual Negative
Predict Positive	92680	682
Predict Negative	8011	106

The model classify most class as target 0, which makes the accuracy is very high; however, it cannot represent our model performance since the false negative is very high and AUC score is around 0.5.

Random Forest. Since the original data was highly unbalanced, the AUC of the base Random Forest model was 0.5 which indicated that the model just randomly guessed the target value instead of making classifications. Also, for unbalanced data, the accuracy is more than 90%, since the model mostly guess the class with higher distribution. Thus, we use resampling method to evaluate the model. Our benchmark model with the main data after downsampling has an average AUC score of 0.63. The figure 3 is the precision-recall curve for our benchmark model in random forest with no sampling, upsampling and downsampling methods. The precision-recall curve for downsampling method is improved but it is decreased for our upsampling methods. Thus for Random Forest model, we use downsampling methods for further model improvement.

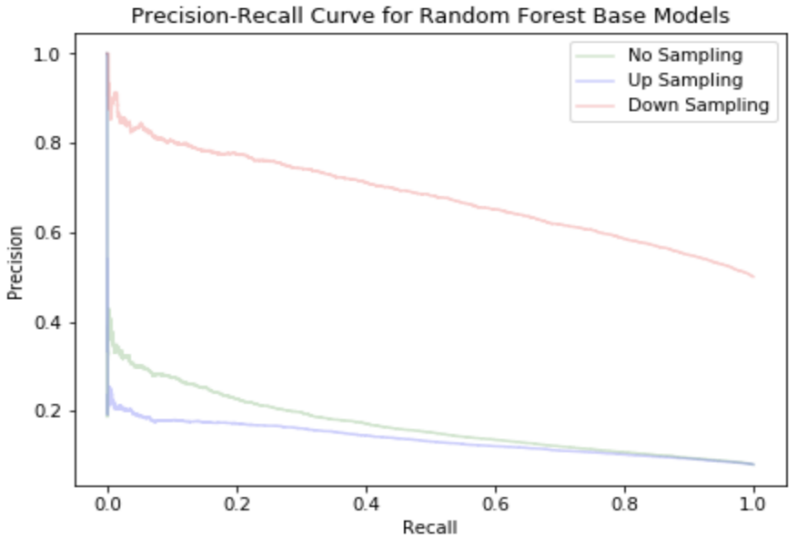

Figure 3. Precision-Recall Curve for Base Random Forest Models

In order to improve the benchmark model, we use the full data which combine the datasets with more representative features. We reduce and combine classes of several categorical features, and drop the features with zero feature importance score. The improved model has an average 0.65 AUC score. The confusion matrix is as follows:

TABLE 5

CONFUSION MATRIX FOR RANDOM FOREST WITH DOWNSAMPLING

	Actual Positive	Actual Negative
Predict Positive	62036	31251
Predict Negative	2890	5302

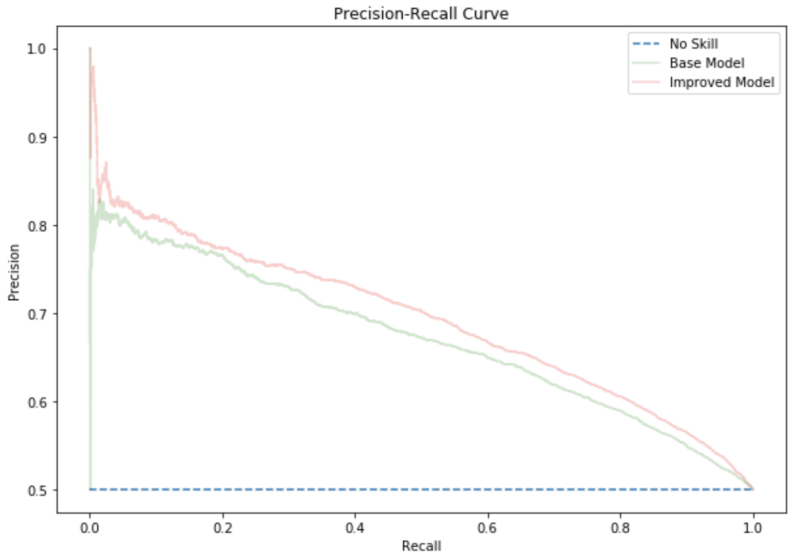

Figure 4. Precision-Recall Curve for Random Forest Models Comparison

Because the AUC is the area under the ROC curve, and ROC is likely to be influenced by the imbalanced dataset, we drew a precision and recall curves in figure 4 to double check whether our model really improved. Here, the precision means the proportion of true positive in the sum of true positive and false positive. The recall means the proportion of true positive in the sum of true positive and false negative. In this plot, the curve of our improved model has a higher precision at all levels of recall. Because Precision and Recall curve is insensitive to imbalanced data, this plot proves our improved model works.

LightGBM. Given the main dataset, we notice lightgbm has a decent performance. Its AUC is 0.7304234816362317. Its accuracy is 0.9191556873835967. Its false positive is high, but it still correctly predicts some actual negative as Target 1. Also, there is a small number of false negatives. It means this model only misclassified a small number of actual positive(Target 0) as negative. From applicants' viewpoint, they hope the model to have the smallest number of false negatives as possible. So, more applicants can get loans from Home credits. For the company's financial perspective, we want to reduce the financial loss due to the false positive. Thus, we hope to improve the model in the sense of smaller number of false positives with the trade of the same number of false negative or a larger number of false negatives. The confusion matrix is as following:

TABLE 6

CONFUSION MATRIX FOR LIGHTGBM

	Actual Positive	Actual Negative
Predicted Positive	93067	7909
Predicted Negative	295	208

Through the help of up sampling in the main data, LightGBM archives 0.7528817679153652 AUC score which is slightly better than the previous LightGBM model. Furthermore, we notice the number of false positives decreases, and the number of correctly predicted negative increases. It means the number of misclassified actual Target 1 decreases. In the ideal situation, we hope the model's false positive and false negative both decreases. Thus, the company and the applicants are both benefits from the model.

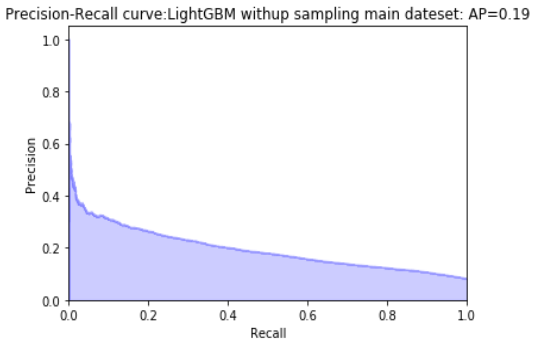

Figure 5. Precision-Recall Curve for Base LightGBM

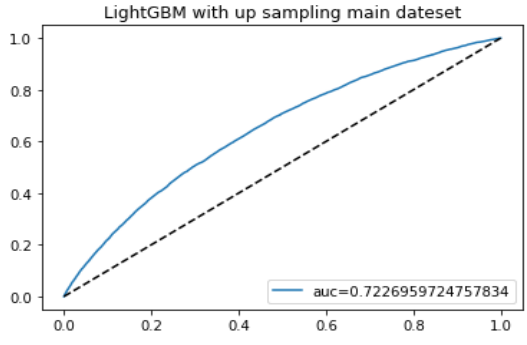

Figure 6. ROC Curve for Base LightGBM

From the exploration of these algorithms, we select those models such as Random forest and LightGBM. These can handle the local minima with or without the help of resampling methods; and also have a decent AUC number. Thus, we use them as the candidate of our final model given full data with 263 features.

With full dataset and up sampling method, we notice that the AUC and accuracy increase again. For the LightGBM without upsampling, the AUC increases to 0.7549517018537376 and accuracy increases to 0.9194513150504046. Also the number of false positive decreases to 7725. For LightGBM with upsampling, the AUC increases to 0.7528817679153652 and accuracy increases to 0.9190078735501926. The number of false positive decreases to 7688.

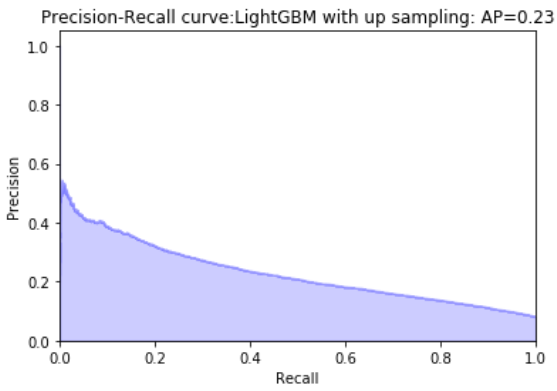

Figure 7. Precision-Recall Curve for LightGBM Up Sampling

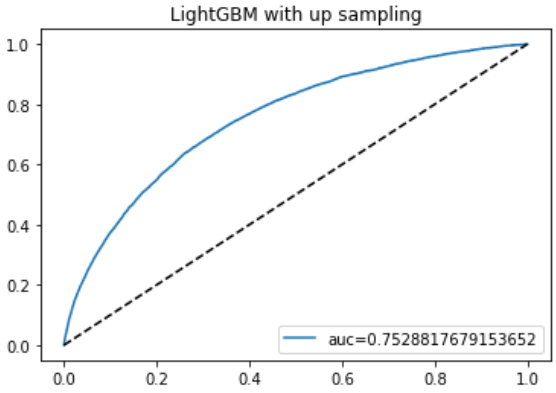

Figure 8. ROC Curve for LightGBM Up Sampling

Although LightGBM has distinct high AUC score than other models, its Precision-recall curve is not perform well as Random Forest did. This might be because we use upsampling methods in lightGBM and also the lightGBM model is trained and improved based on the AUC score.

Voting Classifier of 10 KNN, Voting Classifier with 10 LightGBM, Voting Classifier with Random Forest and LightGBM. For these three classifiers, we notice that they get similar or even worsen AUC score and accuracy. The AUC of voting classifier with KNN is only 0.506511. The AUC of Voting Classifier with LightGBM drops to 0.742841. The AUC of Voting Classifier with LightGBM is around 0.763947. All of these voting classifiers show increased numbers of false which is the least wanted thing from the company's perspective. The reason behind the decreasing of AUC and accuracy is the Voting Classifier performs best when all of its weak classifier are independent from each other. In our classifier , this is not the case. Thus, there are some degrade in terms of performance.

Overall, our best model is LightGBM with full data and the upsampling method. Since the lightGBM is based on boosted method, we can try to increase its AUC and accuracy by increasing the number of estimators in the classifier.

V. CONCLUSION

After our data exploration and model selection, we form different models such as Random Forest and LightGBM and make improvements from our base models. By thoroughly comparing all candidates of final models, our final model is lightGBM which has AUC score of approximately 0.75. Since the default risk is dependent on other information which is unobservable, subjective and unstable, it is hard for us to classify a client's default risk based on the current quantitative data. Our final accuracy rate is 0.919 and the credit company can use this prediction result as a reference to help them make decisions on whether lending loan to the client.

For future improvement, the Home Credit should work closely with Credits Bureau to collect more useful features and data in those clients who are unable to repay the loan on time. For the model, we can fine-tune the hyper-parameter of LightGBM such that increasing the number of estimators in classifiers and other decision tree related parameters for achieving a higher model performance.

VI. Reference

[1] Home Credit Default Risk. (n.d.). Retrieved from https://www.kaggle.com/c/home-credit-default-risk/data

[2] Lin, H. (n.d.). *Machine Learning Foundations*. Lecture. Retrieved from https://www.csie.ntu.edu.tw/~htlin/course/mlfound18fall/doc/01_handout.pdf.

[3] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (n.d.). *SMOTE: Synthetic Minority Over-sampling Technique*(Publication). doi:https://doi.org/10.1613/jair.953