A Study on Credit Default Risk

Bo Liu

1. Introduction

Whether it be credit cards, home loans, or business loans, lending companies need to evaluate a customer's credit profile and income information to determine the willingness and ability to pay back the loans before making the decision of approving or declining the application. The challenges are usually that there may not be much data or relevant information available for the lender to make a good prediction if the customer will default. The fact that true default rate is generally less than 10% of the total portfolio also increases the difficulty for the true default accounts to be accurately predicted.

In this article we build a classification model to predict the probability of default on a new loan, based on customer's bureau information, income, previous loan status, credit card balance, install payment and so on. This model will provide lenders a predicted outcome of default or non-default to guide them making decisions to approve or decline the loan application.

The data of this project comes from Home Credit, which is a non-banking financial institution founded in 1997 in the Czech Republic. The company operates in 14 countries and focuses on lending primarily to people with little or no credit history, which will either fail to be approved for loans or became victims of untrustworthy lenders. The data comes from a variety of sources with more than 200 variables in total, and in some instances goes back to as far as 8 years of monthly balances, posing great challenges in data aggregation, wrangling, missing value imputation and feature engineering. In addition, due to the imbalanced nature of the target variable, we will use down sampling and compare the performances of 6 classification models evaluated on the training set, and finally apply the best selected model on an independently held out test set and report out the test AUC score, top 10 features and so on.

2. Data Wrangling

In this section, we will be implementing the basic data aggregation for each of the 5 data sources provided by Home Credit, by keeping only one or two of the summary statistics such as average, sum, max or min of each group having the same previous application ID. Categorical variables will be treated using one hot encoding. For variables with high cardinality of categories, some categories with low frequencies will be grouped before one hot encoding, and aggregation will be done after one hot encoding.

Current application data is the main table with static information for all current applications. No data cleaning is done at this point for this dataset. Merge this table with all of the 5 aggregated tables created above to get one combined lead file. Further data explorations and processing such as correlation analysis and missing value treatment will be performed on numerical variables, and one hot encoding will be applied to categorical variables after high cardinality categorical variables are treated.

3. Exploratory Data Analysis

The target variable is default vs. non default. As shown in Figure 3.1, overall default percentage is 8.07% in the entire dataset.

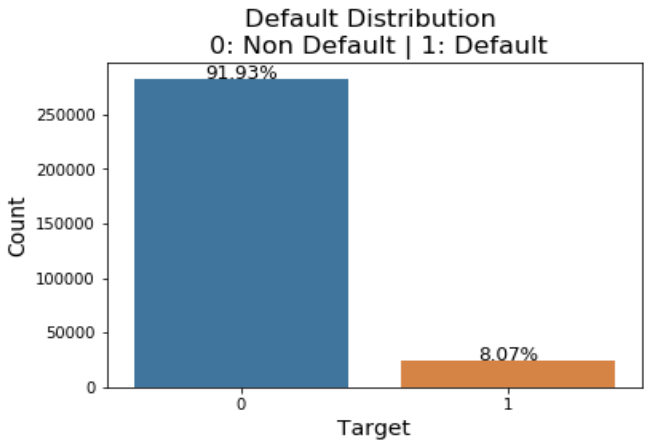


Figure 3.1

Next we explore distributions of some categorical variables.

Distribution of Contract type is shown in Figure 3.2. Revolving loans group appears to have a lower default rate than the Cash loans group. A hypothesis t-test can be done to test if the difference in default rate is statistically significant.

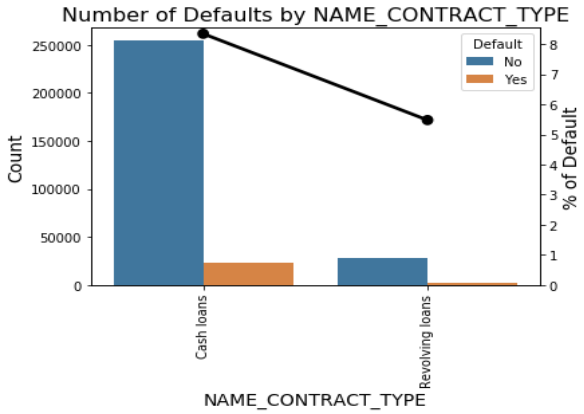


Figure 3.2

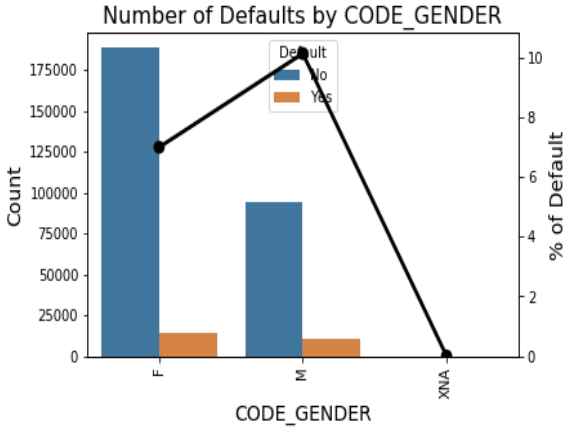


Figure 3.3

Notice from Figure 3.3 that there are 3 gender groups, we will remove the 4 records having gender = XNA. It can be seen from the graph that majority of the applicants are female, and the default rate within female applicants is lower than that in the male group.

Figure 3.4 shows some rank ordering in default rate by Education level. In general, applicants with low secondary education has the highest default rate. As we can imagine education level is

correlated with type of job the applicant can take and highly correlated with income, which is essential to repayment of a loan.

Occupation type variable shown in Figure 3.5 has over 15 categories, we will group the categories having <= 10000 counts into one group.

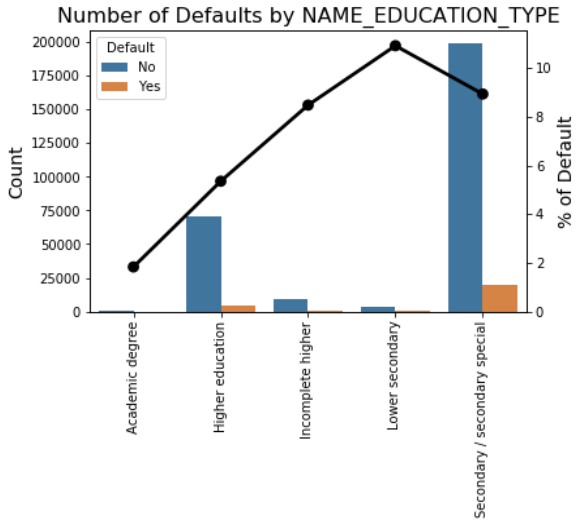


Figure 3.4

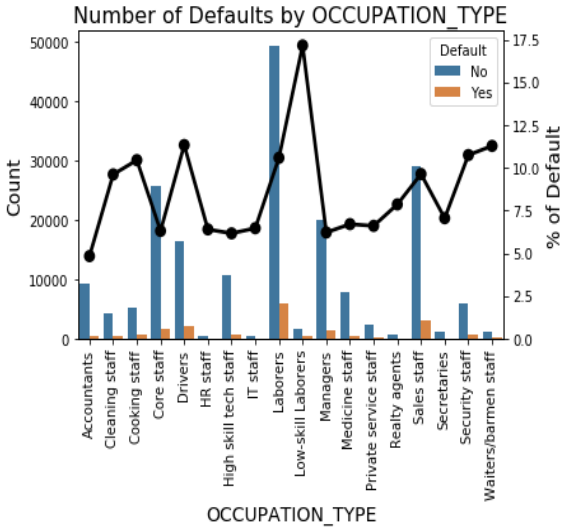


Figure 3.5

Next we explore some numeric variables by creating the correlation heatmap or KDE plots for each variable and split by default vs. non-default segments.

Figure 3.6 indicates there is a high positive linear correlation between observations of client's social surroundings with observable 30 days past due and 60 days past due. As a result, we remove one of them from the dataset.

The dataset contains many features related to the living area or conditions of the applicants. Their correlation heat map is shown in Figure 3.7. Some groups of variables are highly correlated, for example, 'APARTMENTS' variables and 'LIVINGAREA' variables.

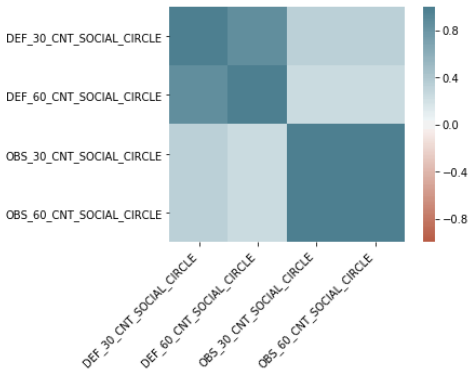


Figure 3.6

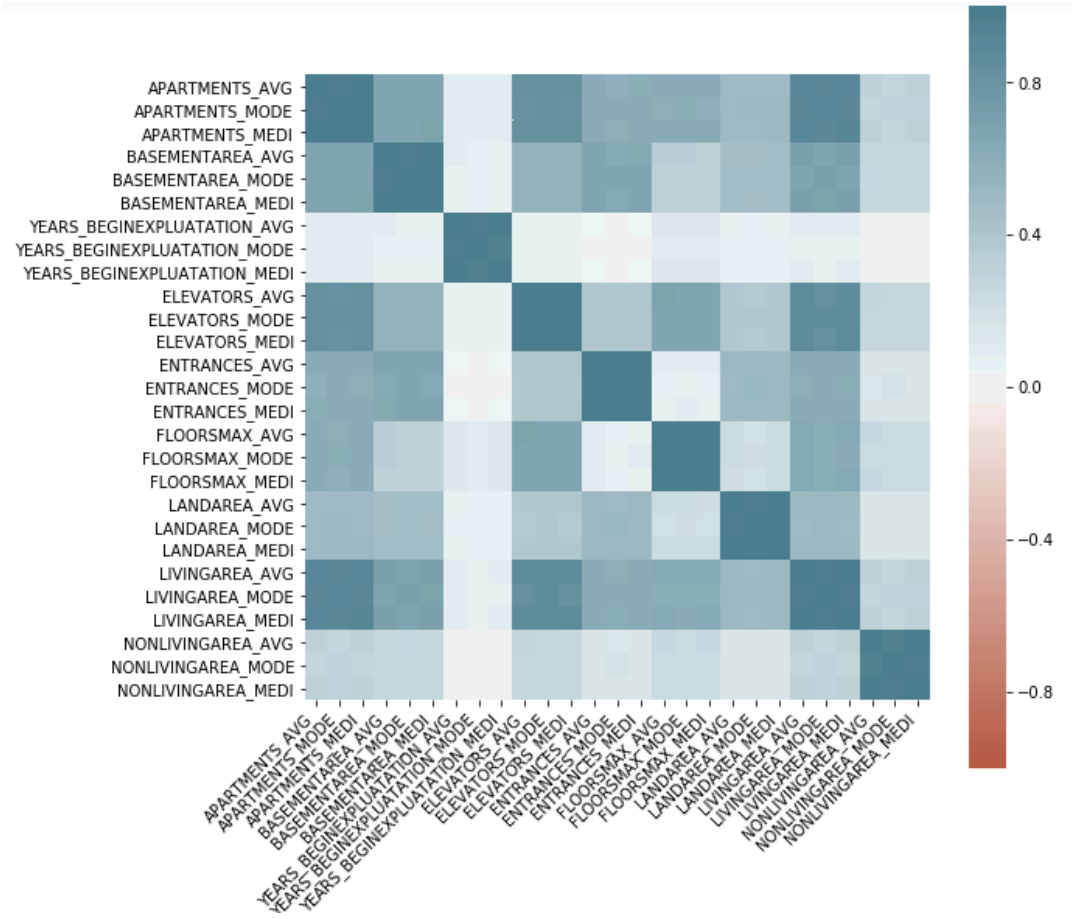
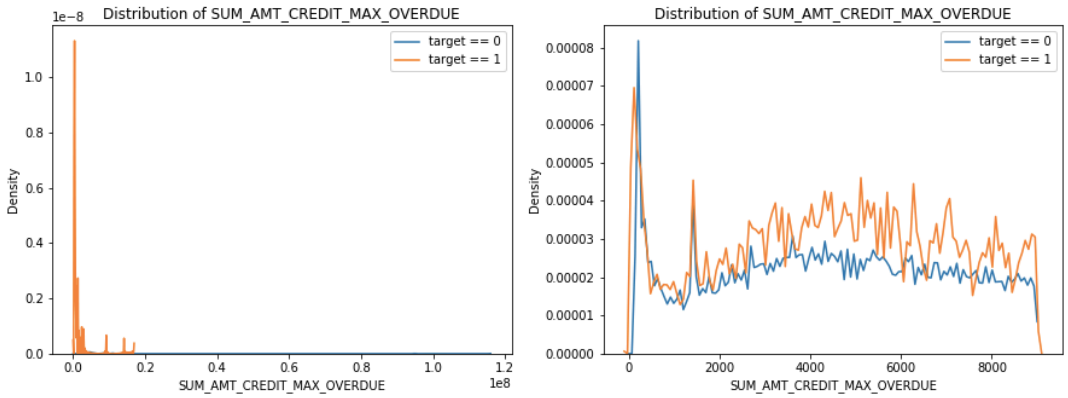


Figure 3.7

Figure 3.8 shows the KDE plots for some numeric variables related to the amount of credit or days of credit overdue. Left panel shows the original range of the variable. Since most of these variables are highly skewed to the right, we zoom in the head of the distribution and plot it on the right panel. A few variables show distinction between the default and non-default groups, while many others are quite similar in these 2 target groups.



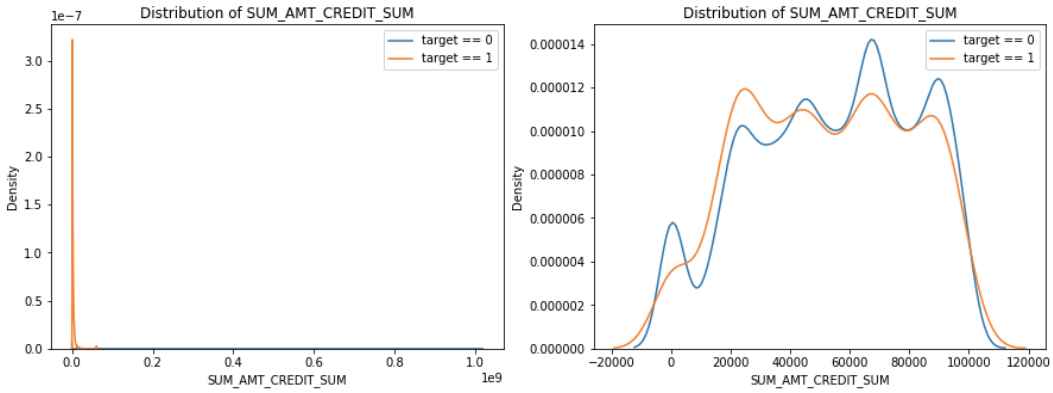


Figure 3.8

The following 3 variables are normalized scores from external data source. It turns out that several of the models we fit selected these variables to be the top important variables.

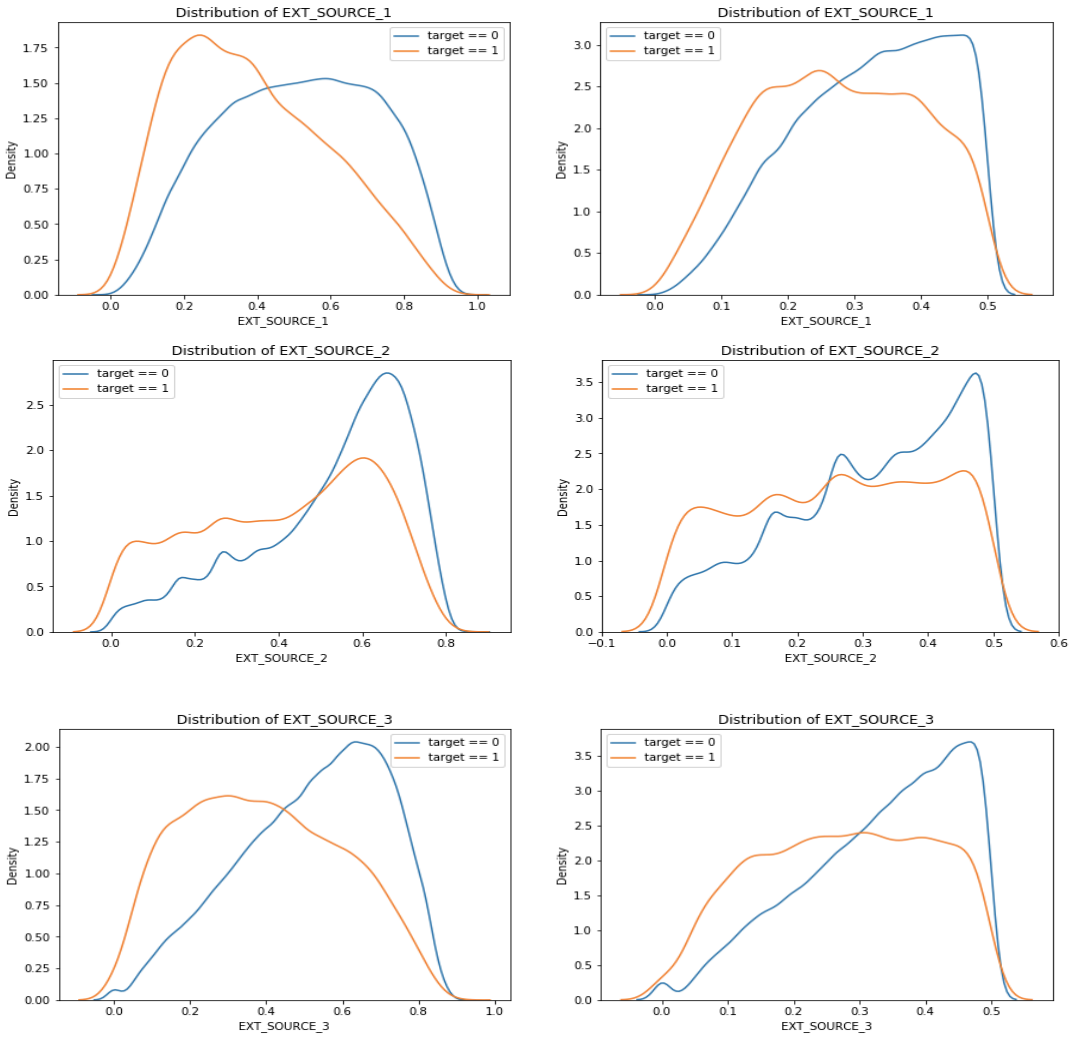


Figure 3.9

Variables in Figure 3.10 are originally presented in days in the dataset, convert them into years to better visualize the ranges. Here are some observations:

- "DAYS_EMPLOYED" has values over 1000 years, which are apparently errors.
- "MAX_DAYS_CREDIT_ENDDATE" is an aggregated field summarizing all previous applications within the same current application ID using the max aggregation function. It means over all the previous applications within the same current application ID, the maximum remaining duration of Bureau credit at the time of application. This variable should be a positive number, but we found negative values in the column. As a result, we floor the variable at 0. For missing values, impute using the median.

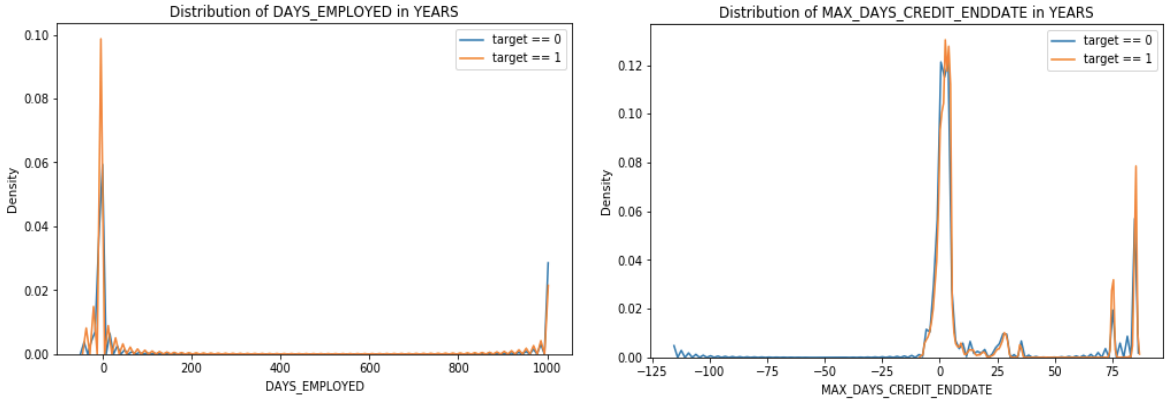


Figure 3.10

In addition, we also examined the following age-related variables in years (Figure 3.11).

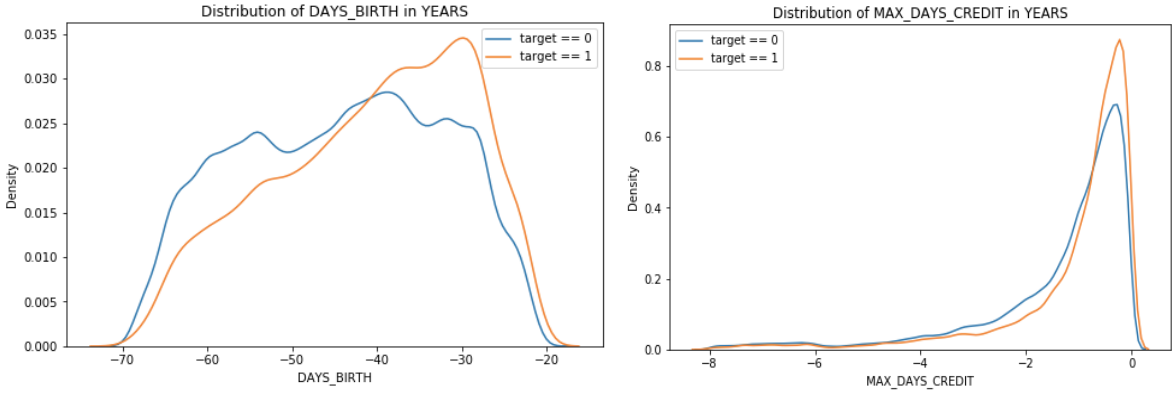


Figure 3.11

Distributions of variables such as "AMT_ANNUITY", "AMT_CREDIT" and "AMT_GOODS_PRICE" are skewed to the right, we recommend to take the log transformation to normalize the data. To save space, we only show the distribution of "AMT_ANNUITY" in Figure 3.12.

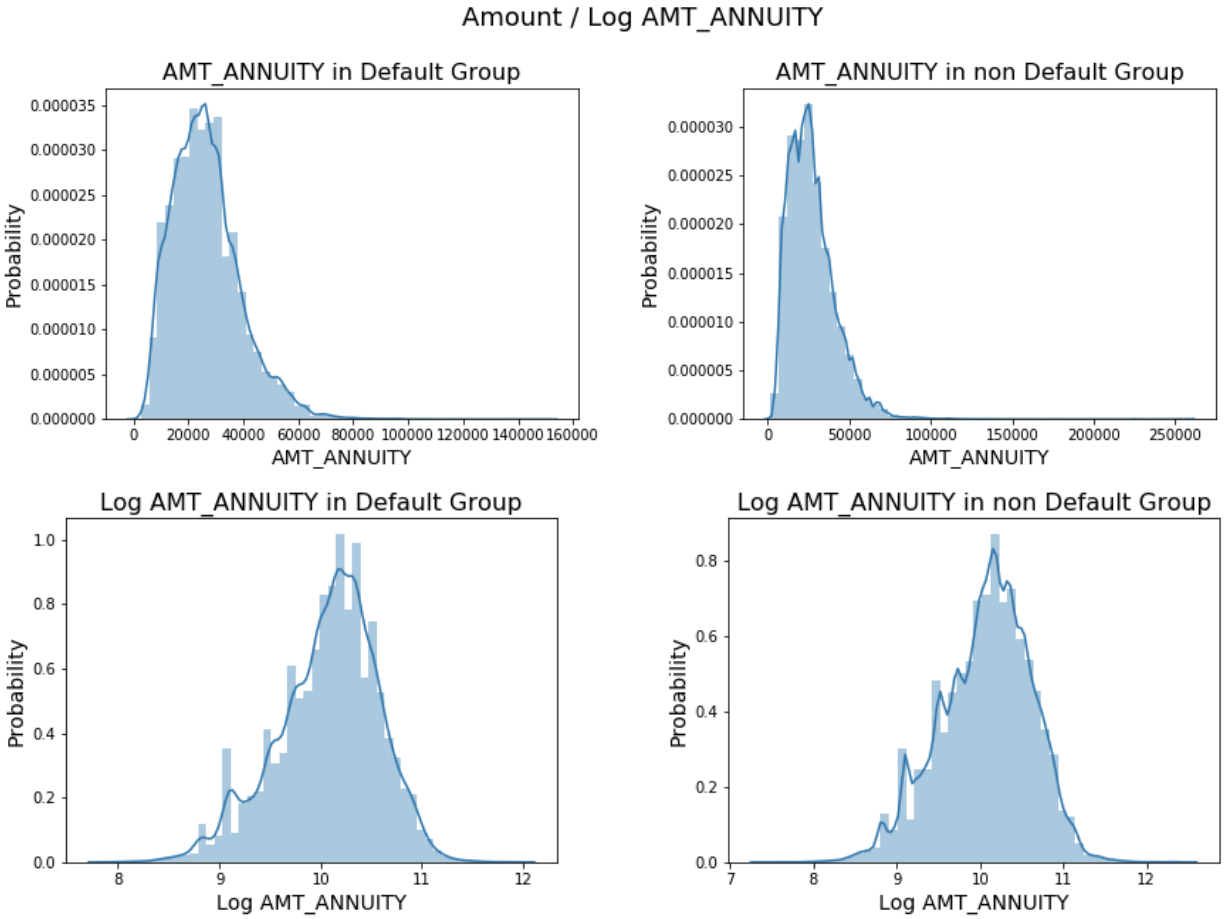
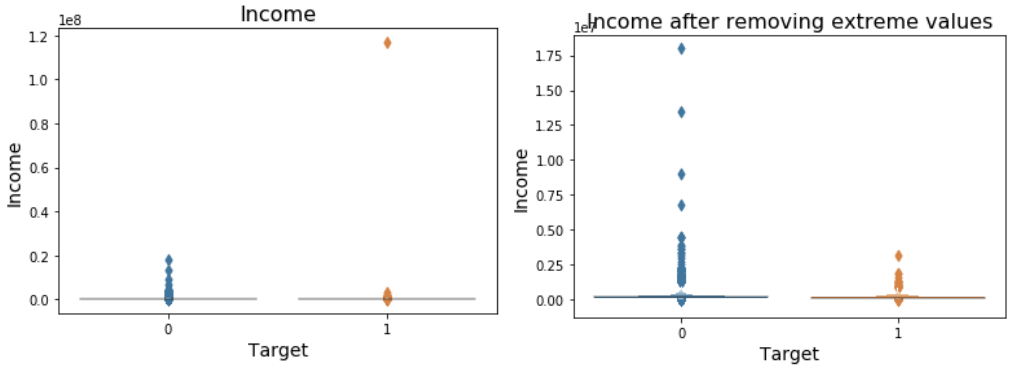


Figure 3.12

There is one extremely large income value in the default group ($120 million), and 4 very large values (> $7.5 million) in the non-default group. Recommend to remove the outliers and use log transformation to normalize the income data (Figure 3.13).



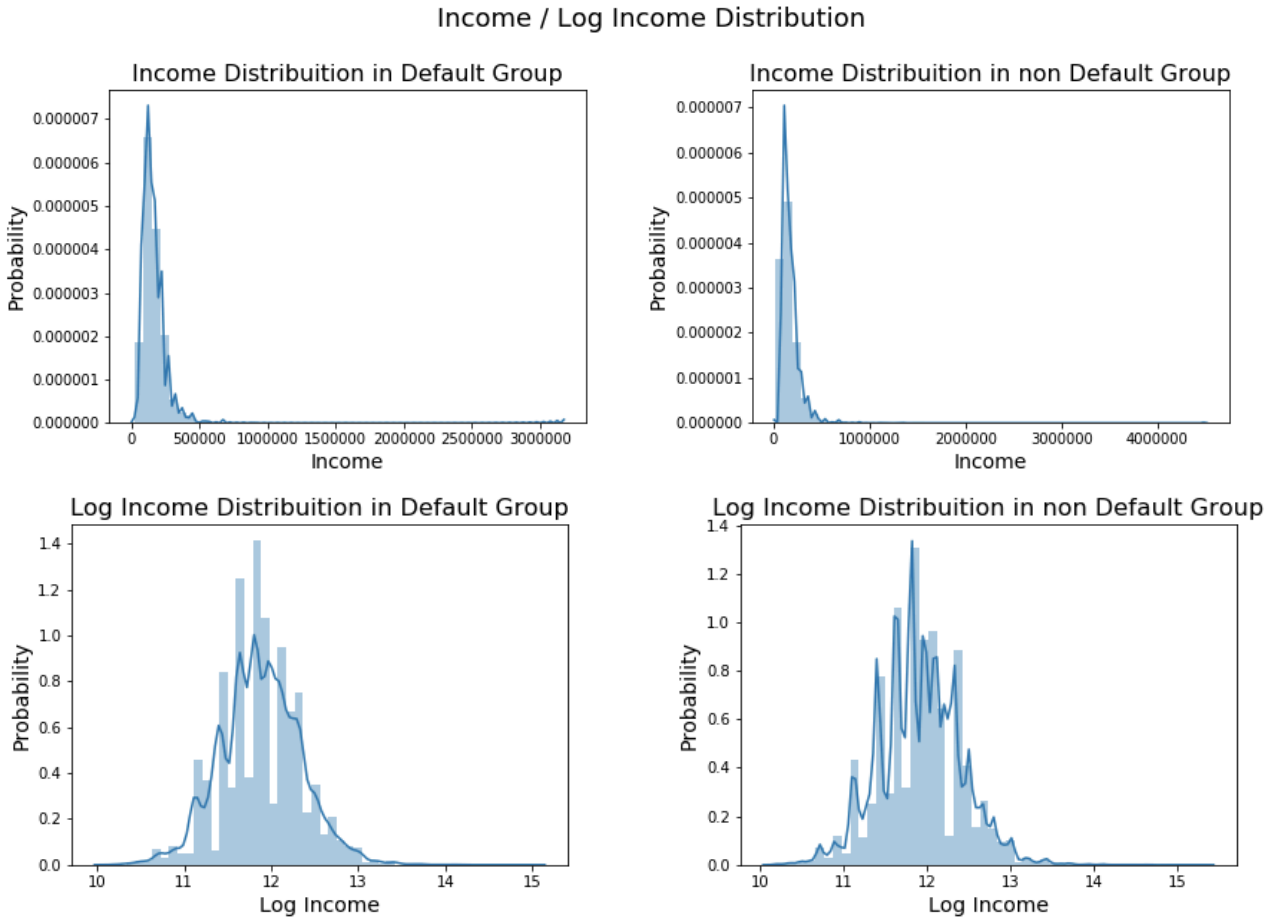


Figure 3.13

4. Building ML Models

In this section, we explore 6 commonly used classification models including tree-based methods such as Random Forest, XGboost and LightGBM, as well as classical models such as Logistic regression with regularization, K-nearest neighbors (KNN) and support vector machine / classifier (SVM). Due to the imbalanced nature of the data, we down sampled the non-default counts to be approximately the same as the default counts in the training data, while the test data is still a randomly held out 20% imbalanced dataset. Note that we also tried using the original imbalanced data to build a Random Forest model, and every data point was classified as non-default loan, which is not useful.

For each model, first a K-fold CV with RandomizedSearchCV in Scikit-learn is run on the training set to select the best combination of parameters using accuracy as criteria. Next, each model is fit on the entire balanced training set using the chosen best parameters, and performance metrics such as accuracy, confusion matrix, F1 score and AUC are reported.

Note that all metrics, for example accuracy and AUC, are reported using 0.5 decision threshold. Lenders can also choose their own threshold to come up with the tradeoff between precision and

recall rate which lie within the business risk appetite and tolerance, and eventually maximize the financial benefit.

4.1 Random Forest

Random Forest is a well-known bagging algorithm which builds hundreds or thousands of trees on bootstrap samples of data and a random subset of features to reduce overfitting. The model uses majority votes to determine the final classification labels.

We asked the model to select 5 combinations of parameters from n = 100, 200, 300, 400 and max depth ranging from 1 to 9 using 5-fold CV. Based on the CV scores, we empirically learned that the model will seriously overfit when max depth is greater than 10. The optimal parameter combinations chosen by randomized search cross validation is n = 200 and max depth = 6.

Top 10 features selected by the model and the ROC curve on the training set are shown in Figure 4.1.1 and 4.1.2. The top 3 features are normalized scores from external data source, which Home Credit did not reveal the sources and meaning. The rest important variables are age, maximum (aggregated out of all the Bureau IDs belonging to the same current application ID) number of days before current application that client apply for Credit Bureau credit, length of employment, and so on. The ROC curve can help analysts find a tradeoff point between the true and false positive rate that Home Credit's risk policy can undertake. Using some financial assumptions such as how much Home Credit would lose if one default case is mistakenly predicted as non-default case, or how much profit Home Credit would miss if one non-default case is incorrectly classified as default case, the company can get a profit and loss forecast financial sheet and make the most appropriate decisions on what threshold to choose.

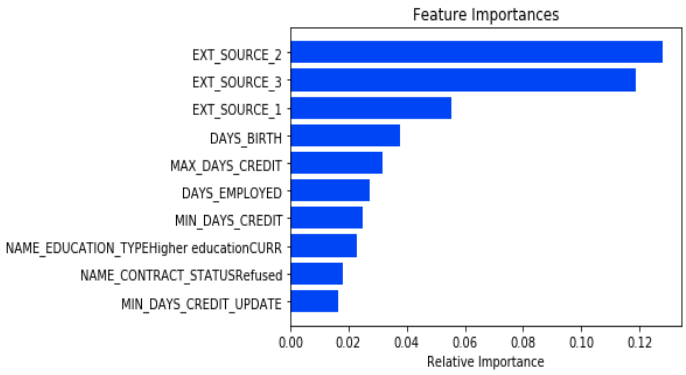


Figure 4.1.1

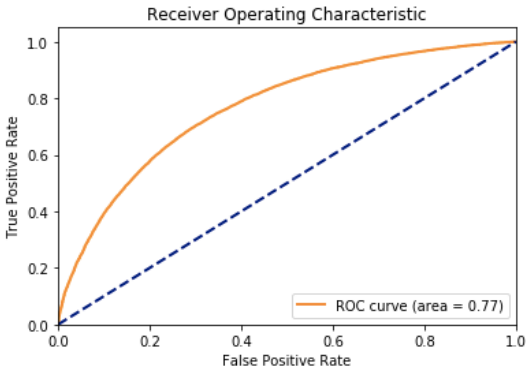


Figure 4.1.2

4.2 XGboost

XGboost is a scalable and accurate implementation of gradient boosting machines and it has proven to push the limits of computing power for boosted trees algorithms as it was built and developed for the sole purpose of model performance and computational speed.

In this model we tuned 4 parameters, number of trees, learning rate, max depth and minimum loss reduction required to make a further partition on a leaf node of the tree - gamma. The optimal n = 300, learning rate = 0.1, max depth = 3 and gamma = 0.001. Confusion matrix, top 10 features and ROC curve on the training data are shown in Figure 4.2.1, 4.2.2 and 4.2.3.

Confusion Matrix	**Predicted**	
Actual	Non-Default	Default
Non-Default	15,236	5,117
Default	5,304	14,552

Figure 4.2.1

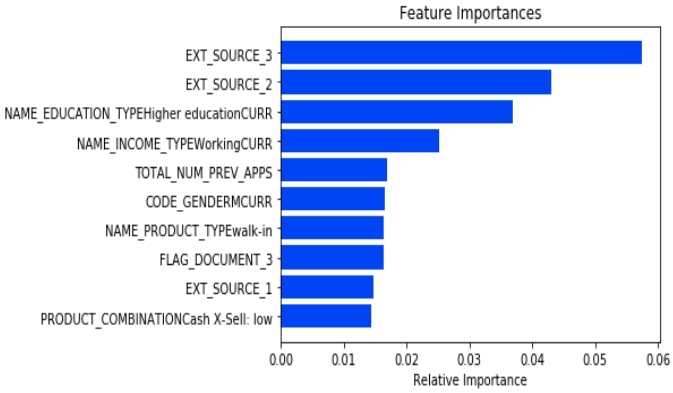


Figure 4.2.2

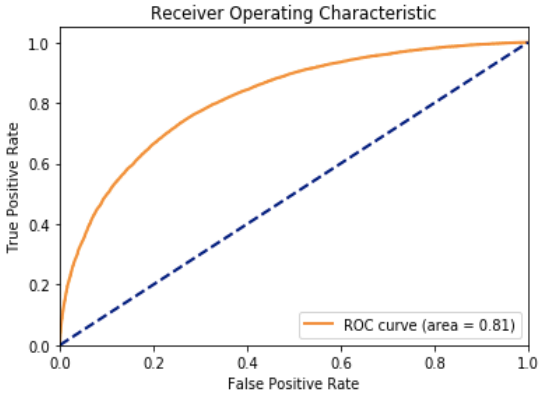


Figure 4.2.3

4.3 LightGBM

LightGBM uses XGboost as a baseline and outperforms it in training speed and dataset sizes it can handle. Some advantages of LightGBM include: faster training speed and higher efficiency, lower memory usage, better accuracy, support of parallel and GPU learning and capable of handling large-scale data.

We tuned number of trees, learning rate and max depth and LightGBM produces the best results among all of the model we trained. The optimal n = 400, learning rate = 0.1, max depth = 3. Confusion matrix, top 10 features and ROC curve on the training data are shown in Figure 4.3.1, 4.3.2 and 4.3.3.

Confusion Matrix	**Predicted**	
Actual	Non-Default	Default
Non-Default	15,327	5,026
Default	5,167	14,689

Figure 4.3.1

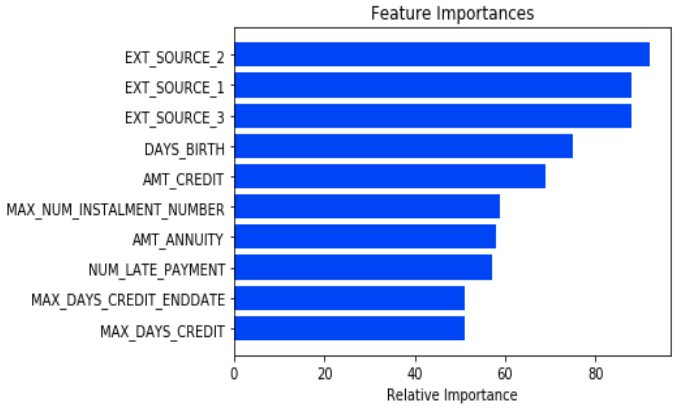


Figure 4.3.2

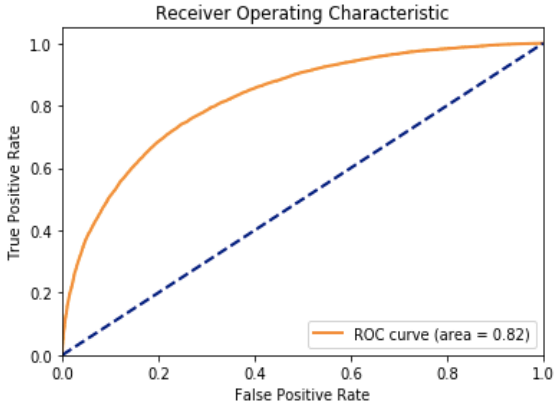


Figure 4.3.3

4.4 Logistic Regression with Regularizations

In this model, we searched over L1, L2 penalty, and regularization strength parameter C in the list of [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1, 1, 10, 100]. It turned out L1 penalty with C = 100 is the best combination. The AUC is comparable to the Random Forest model, and confusion matrix and ROC curve on the training data are shown in Figure 4.4.1 and 4.4.2.

Confusion Matrix	**Predicted**	
Actual	Non-Default	Default
Non-Default	14,531	5,822
Default	6,050	13,806

Figure 4.4.1

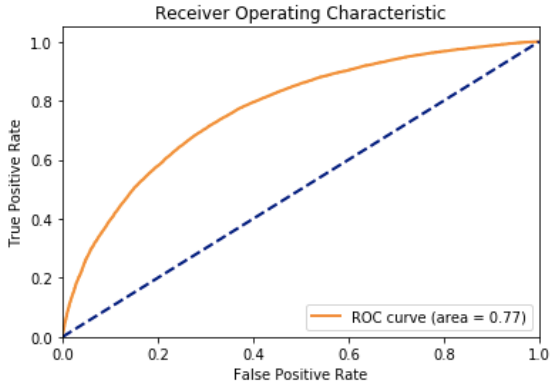


Figure 4.4.2

4.5 K-Nearest Neighbors (KNN)

In this model, we searched the best K from 1 to 19, and it turned out 15 is the best choice. The model performance is not as good as Random Forest and Logistic Regression, possibly due to the high dimension of the feature space, as it is known that algorithms that use distance as measure to classify “close by” points suffers badly from curse of dimensionality.

Confusion Matrix	Predicted	
Actual	Non-Default	Default
Non-Default	13,202	7,151
Default	7,346	12,510

Figure 4.5.1

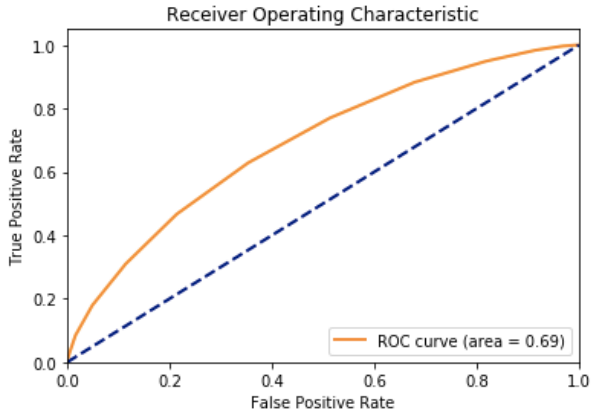


Figure 4.5.2

4.6 Support Vector Machine (SVM)

Kernel SVM is computationally intensive. To help convergence, we used MinMaxScaler in Scikit-learn to scale the features before training the model. Regularization parameter C is selected from range 1 to 100, kernel functions are chosen from RBF, Polynomial and Linear, and kernel coefficient gamma is chosen from the list [0.001, 0.0001]. Due to the long computation time, we used 2-fold CV and in the RandomizedSearchCV we only choose 2 parameter combinations, making total number of fit = 4. The best parameters selected are C = 52, RBF kernel and gamma = 0.001.

Performance of Kernel SVM is between Random Forest, Logistic regression and XGboost, LightGBM. Taking into account the extensive computation time, the later 2 models are preferable.

Confusion Matrix	Predicted	
Actual	Non-Default	Default
Non-Default	14,870	5,483
Default	5,626	14,230

Figure 4.6.1

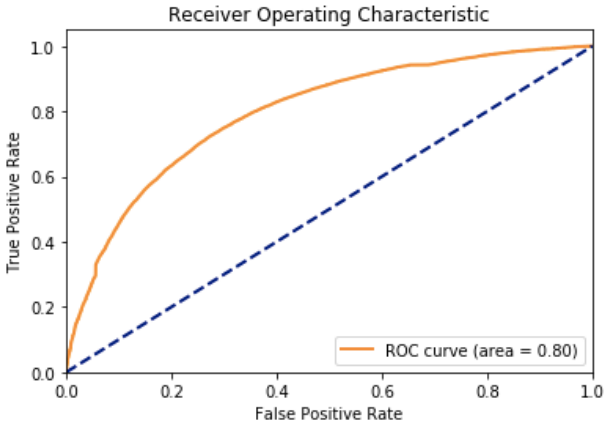


Figure 4.6.2

4.7 Summary

Table 4.7 shows the accuracy and AUC score for all of the models we trained. With balanced training data, LightGBM has the highest AUC.

Model	Accuracy	AUC
Random Forest	0.701733	0.769773
XGboost	0.740829	0.816779
LightGBM	0.746500	0.824474
Logistic Regression	0.704743	0.772114
KNN	0.639459	0.690565
Kernel SVM - RBF	0.723719	0.796507

Table 4.7

4.8 Evaluation on Test set

Finally, we evaluated LightGBM model on the independent test set, which is not balanced. The test accuracy and AUC are 0.7120 and 0.7768, respectively. Confusion matrix, classification report and ROC curve are shown in the following Figures.

Confusion Matrix	**Predicted**	
Actual	Non-Default	Default
Non-Default	40,303	16,230
Default	1,480	3,488

Figure 4.8.1

	precision	recall	f1-score	support
0	0.96	0.71	0.82	56533
1	0.18	0.70	0.28	4968
micro avg	0.71	0.71	0.71	61501
macro avg	0.57	0.71	0.55	61501
weighted avg	0.90	0.71	0.78	61501

Figure 4.8.2

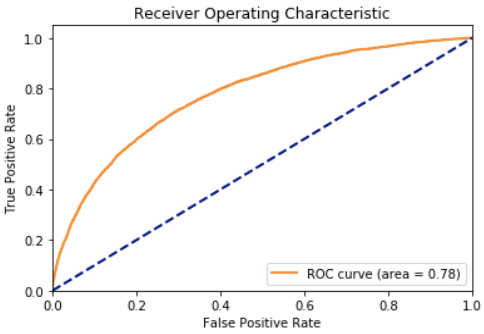


Figure 4.8.3

5. Conclusions and Future work

5.1 Conclusions

In this project we predicted the probability of default for each current loan for Home Credit, based on aggregated information from previous loan histories that belong to the same current loan ID, Bureau credit information, demographics and social network attributes of applicants, income and education level, living area conditions and so on.

Our model reports are based on 0.5 decision threshold when predicting if a future loan will go default or not. However, analysts in Home Credit can leverage the model predicted probabilities and adjust the threshold in their decisions to achieve better tradeoff between precision and recall rate, to fit in their risk appetite and eventually making an appropriate decision to decline or approve a new loan. In addition to the F1 score etc. mentioned above, we also provide top 10 features for the analysts to understand feature importance and ROC curves to help choosing the threshold.

To achieve all of these, we started by aggregating each of the 5 data sources provided by Home Credit, and conducted exploratory data analysis on both categorical and numeric variables to visualize correlations between the target and features. Some features show better separation of default vs. non-default and others not so much. We compared the performances of 6 classification models and below are our findings:

1. The model with the highest AUC on the training data is LightGBM. XGboost comes second and kernel SVM ranks the 3rd place, followed by Logistic regression with L1 penalty, random forest and KNN.
2. In terms of computation times, tree-based methods are generally faster than other methods. Logistic regression and KNN take slightly longer than tree-based methods. Kernel SVM is the slowest to train and predict.

5.2 Future work

We realized that the precision for all of the models are not ideal, somewhere around 0.17. Low precision indicates a high number of false positives, which means we are predicting a non-default customer as default. This will hurt the revenue of Home Credit as we are excluding many profitable good loans. On the other hand, our recall rate is about 70%, indicating we are correctly identifying 70% of the default loans within the truly default ones. There may be room to improve both the precision and recall rates. Specifically, below are some thoughts for future work:

1. Feature engineering can be further explored by looking into combinations of various columns, quadratic terms or using different aggregation methods at sub-ID levels.
2. In the final dataset used in modeling, Bureau balance dataset was not included in the features due to limited time and computation power. In future work, it can be aggregated and incorporated into the training data.
3. More missing value imputation techniques can be used to better impute the missing values based on overall shape of the distribution, rather than using the median for every numeric variable.
4. More sampling techniques such as up sampling of the minority class, or SMOTE can be used to compare with the performances with down sampling.
5. PCA or other dimension reduction methods can be leveraged to reduce the dimension of the feature space, however, model interpretability may be lost.
6. In this project we only tried to use scaled features in Kernel SVM model, to speed up the convergence of the algorithm. For all other 5 models we used the original scale of the variables, as tree-based methods can deal pretty well with different scales of variables. In future work we can try using scaled / normalized features in Logistic Regression and tree models as well, to see if there is any performance gain.
7. In all of the models we trained, we throw in all of the over 300 features without picking any subset of features. In Logistic regression, L1 penalty was selected, so we automatically got some feature selection benefits. Tree-based models naturally produce the rankings of features, so we can try to use the top 100 or so to train the model again to see if there are any performance improvements. In future work, more variable selection techniques can be considered to fit a more parsimonious model rather than using the full set of features.
8. In the SVM model cross validation step to select the optimal hyperparameters, due to the prohibitive training time of SVM algorithms, we only tried random search 2-fold cross validation with 2 random parameter combinations. More combinations can be tested to achieve possibly higher AUC provided with GPU or a high computation power CPU. Similar with tree-based models, more combinations can be tried when searching for best hyperparameters.
9. Other classification models can be tested such as neural networks, linear and quadratic discriminant analysis, other tree-based methods such as Adaptive Boosting, or other bagging and ensemble methods.

Reference

[1] https://www.kaggle.com/c/home-credit-default-risk/data

[2] https://github.com/lisalb168/Bo_project/tree/master/capstone%20project%201/notebook