Credit Default Risk Prediction

Bo Liu

Table of Contents

Problem Statement

- Home Credit is a lending company that strives to provide positive and safe borrowing experience for its customers.
- Like all other lending companies, Home Credit leverages a wide variety of data sources from its customers such as bureau information, payment history of previous loans, and income information to predict the customer's credit worthiness and ability to repay the loans.
- Given the collected attributes and default / non-default labels for each customer, we can build classification models to predict the probability of loan default and obtain the top features that separate default from non-default loans. The model output can be used as guidance to approve or decline a new loan application.

Source Data Information

Source: https://www.kaggle.com/c/home-credit-default-risk/data

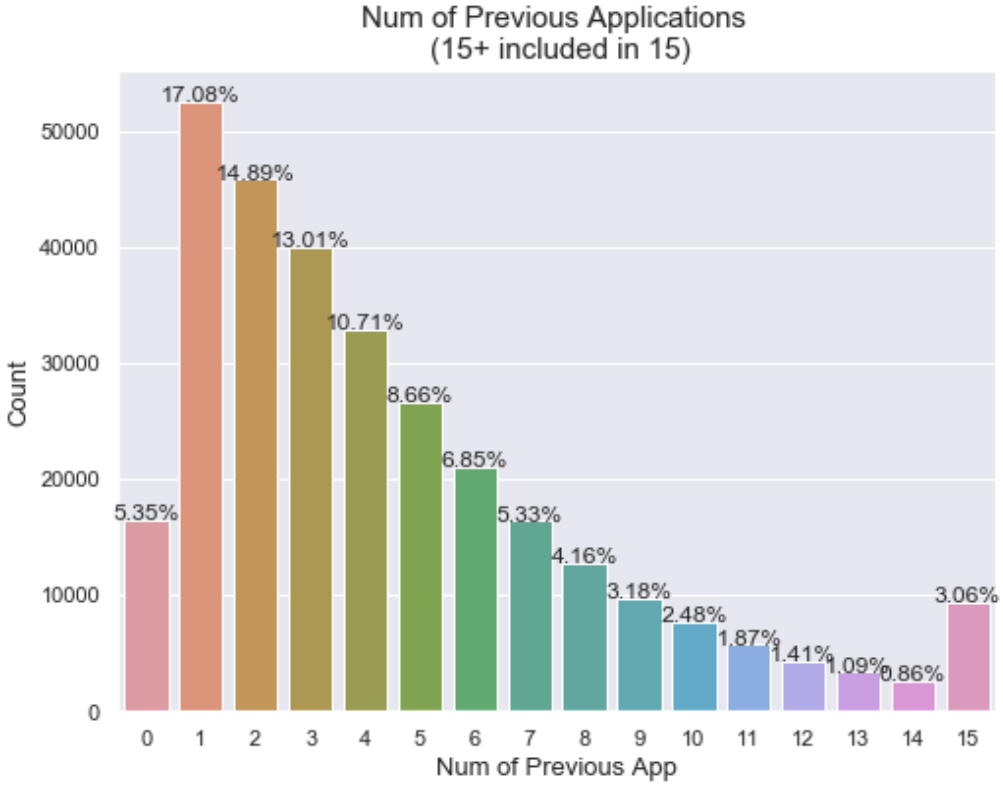
- *Application:* 305K unique loan numbers identified by SK_ID_CURR. 121 customer attributes such as income, occupation, total living area, and so on. TARGET = 0: the loan was repaid or 1: the loan was not repaid.
- *Bureau:* Client's previous credits from other financial institutions. Each previous credit has its own row in bureau identified by SK_ID_BUREAU, but one SK_ID_CURR can have multiple SK_ID_BUREAU.
- *Bureau Balance*: Monthly data about the previous credits in bureau. Each row is one month of a previous credit, and a single previous credit (SK_ID_BUREAU) can have multiple rows (months). 27 million rows.
- *Previous Application*: Previous applications for loans at Home Credit of clients who have loans in the current application data. Each current loan in the application data can have 0,1,2,...previous loans. Each previous application has one row and is identified by SK_ID_PREV. 1.7 million unique SK_ID_PREV.
- *Pos Cash Balance*: Monthly data about previous point of sale or cash loans clients have had with Home Credit. Each row is one month of a previous point of sale or cash loan, and a single previous loan can have many rows.
- *Credit Card Balance*: Monthly data about previous credit cards clients have had with Home Credit. Each row is one month of a credit card balance, and a single credit card can have many rows.
- *Installments Payment*: Payment history for previous loans at Home Credit. There is one row for every made payment and one row for every missed payment.

Note: Due to CPU processing power limitations, "Bureau Balance" file was not used in this study.

Data Aggregation

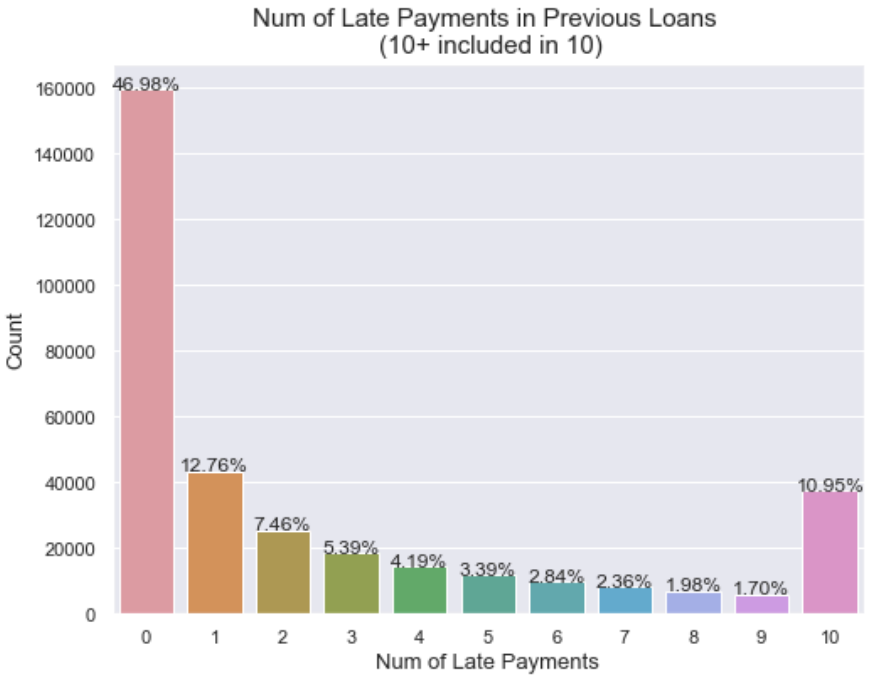
Numerical variables	Keep one or two of the summary values such as average, sum, max or min within each sub ID, in this case could be previous application ID or bureau ID.
Categorical variables	If with high cardinality, some categories with low frequencies are grouped before one hot encoding, and aggregation is done after one hot encoding using sum or max aggregation.
Data Merge	Use current application ID as merging key to combine all sources of data. Final data set contains 300+ features.

Feature Engineering – Adding New Features



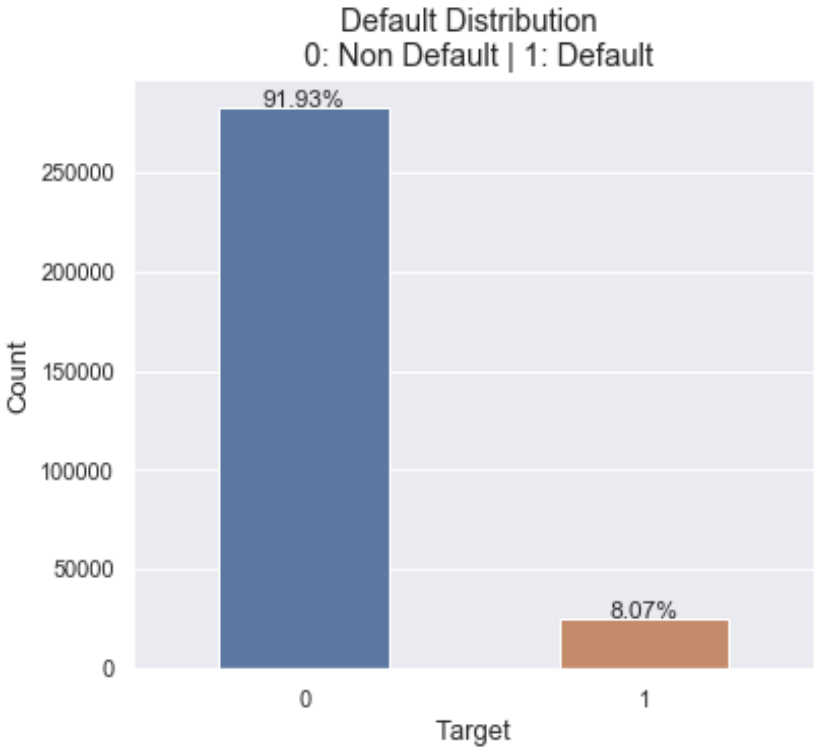
- Add a numeric column that counts how many previous applications the customer has with Home Credit

Feature Engineering – Creating New Features



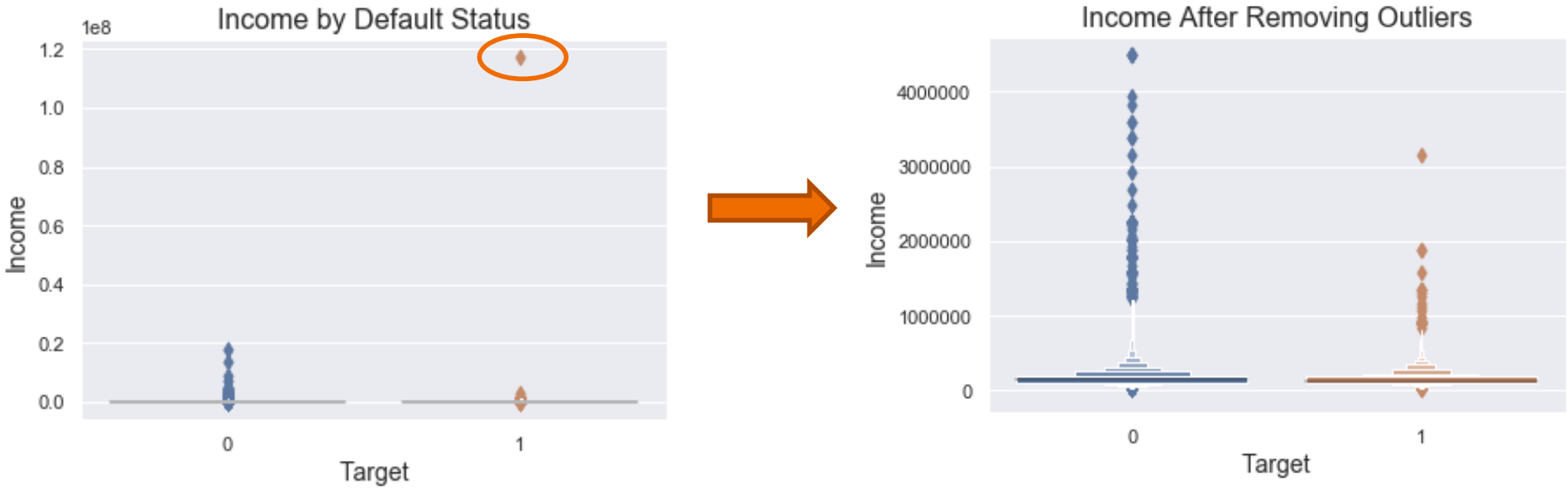
- DAYS_INSTALMENT: when the installment of previous credit was supposed to be paid (relative to application date of current loan)
- DAYS_ENTRY_PAYMENT: when was the installments of previous credit paid actually (relative to application date of current loan)
- DAYS_INSTALMENT >= DAYS_ENTRY_PAYMENT meaning the payment is made prior to the due date, which is on time payments, otherwise it would be a late payment
- Aggregate by counting total number of late payments within previous and current application ID

EDA – Default Rate



- Default Rate is 8.07%, indicating an imbalanced data set

EDA – Outlier

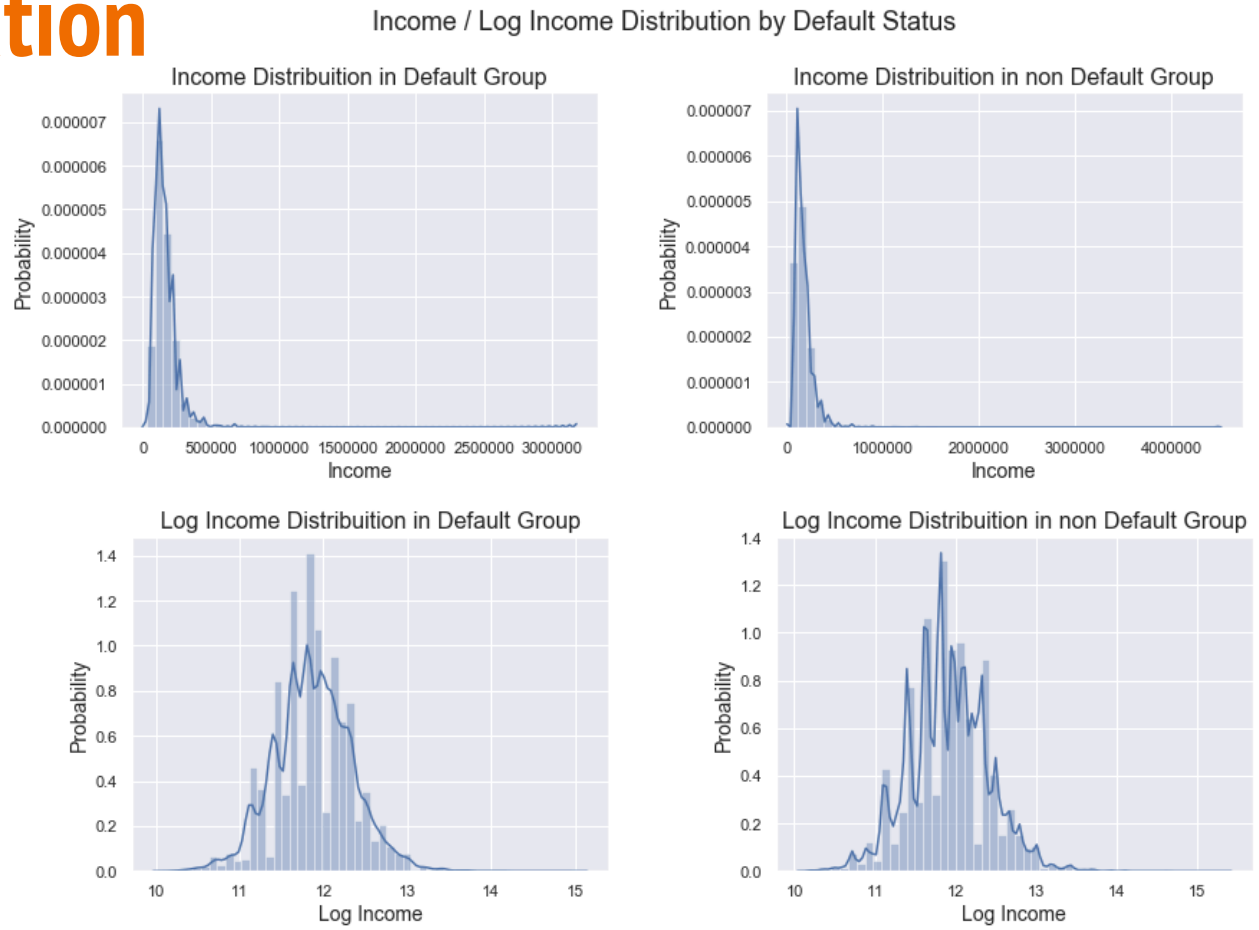


- Extreme income value exists in the default group (1.2*10^8). The occupation of this data point shows “Laborers”
- Further investigation of the extreme values warrants removing rows with income > 5,000,000

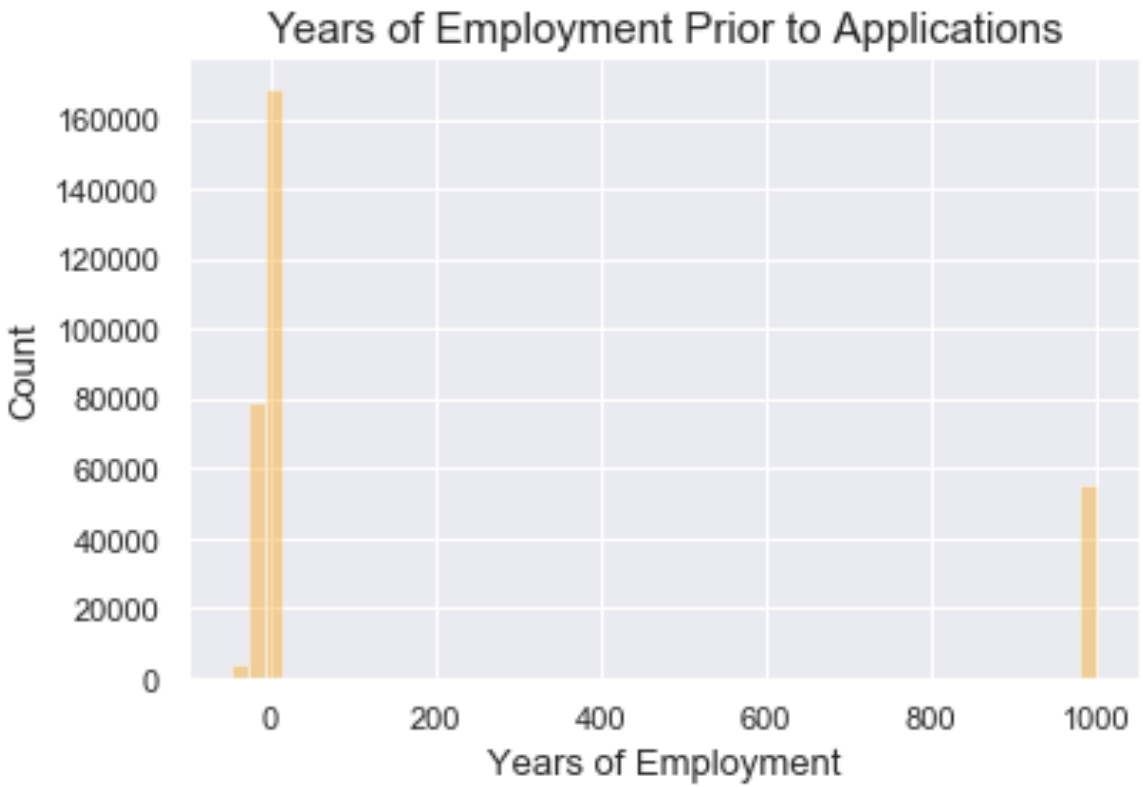
EDA – Transformation

- Income distribution is highly skewed to the right. Use Log transformation to normalize the data

- 2 sample Kolmogorov–Smirnov Test and Wilcox Rank Test on the distribution of log(income) for default and non default group shows significant difference when sample size is relatively large (in thousands). But no significance is detected when randomly sample a few hundreds from both target groups

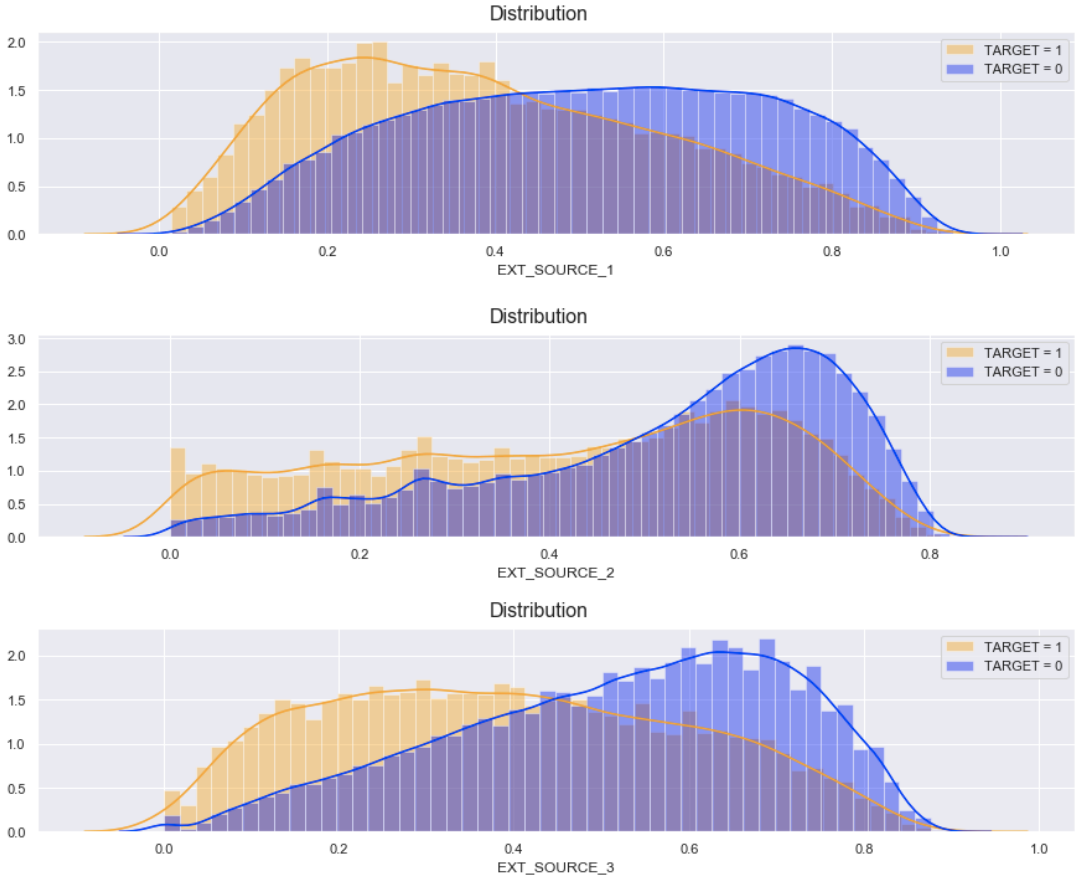


EDA – Data Errors



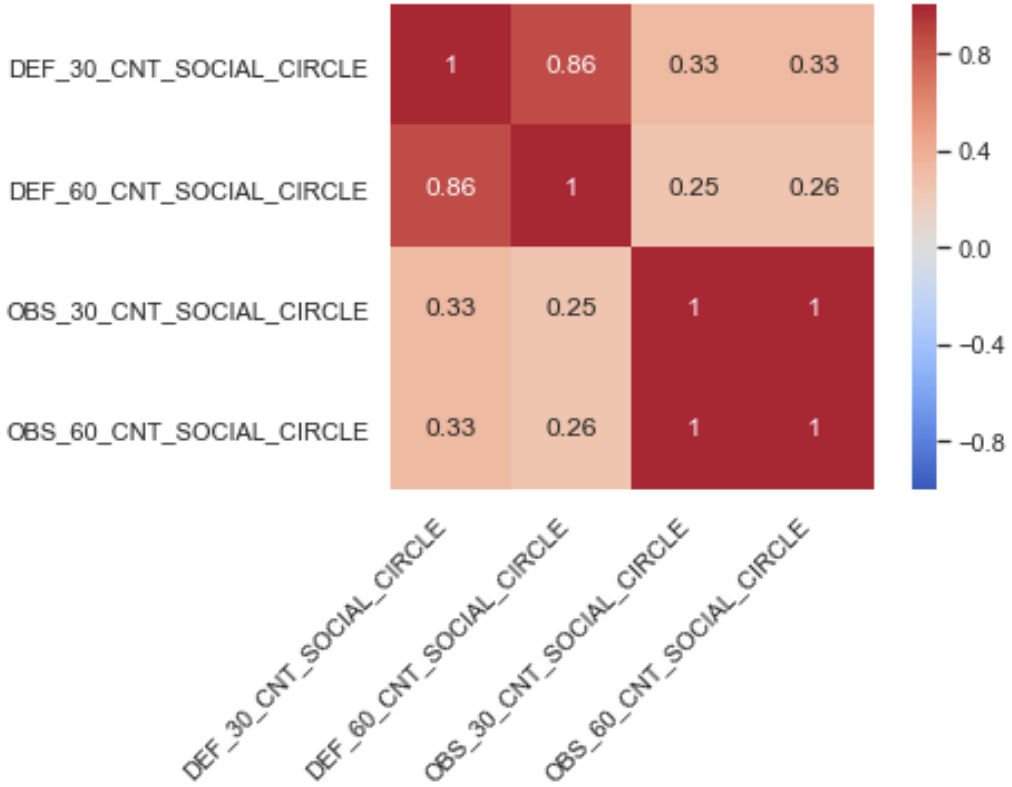
- Employment variable is defined as number of years prior to current application. Valid numbers are negative, cutting off at time 0 (time of application)

EDA – Distribution by Default Status



- EXT_SOURCE_1 – EXT_SOURCE_3 are normalized scores from external source
- Visualization shows these 3 variables have different distributions (shapes) in default group vs. non default group, which may make them good differentiators of the 2 target groups

EDA – Correlations Between Variables



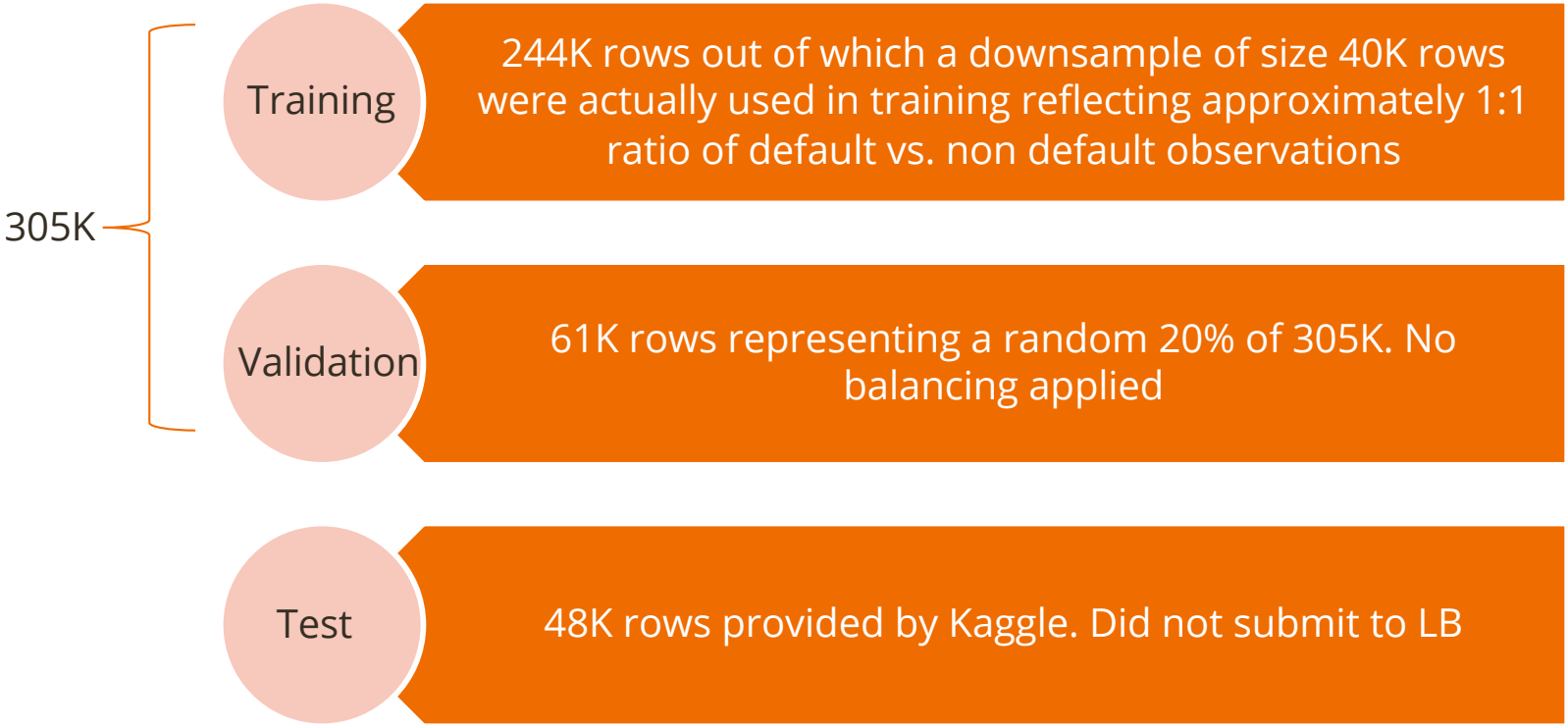
- OBS_30_CNT_SOCIAL_CIRCLE is perfectly linearly correlated with OBS_60_CNT_SOCIAL_CIRCLE. Remove one of them can stabilize model parameter estimation

EDA – Categorical Variables

ORGANIZATION_TYPE	Non Default	Default
Advertising	0.918415	0.081585
Agriculture	0.895273	0.104727
Bank	0.948145	0.051855
Business Entity Type 1	0.918616	0.081384
Business Entity Type 2	0.914716	0.085284
Business Entity Type 3	0.907004	0.092996
……	……	……

- Total 57 categories of organization type. Missing organization is defined as an extra category XNA
- Generate a new column using the % of default counts within each category to replace the variable organization type

Modeling – Data Partition



Modeling – Choice of Classification Methods

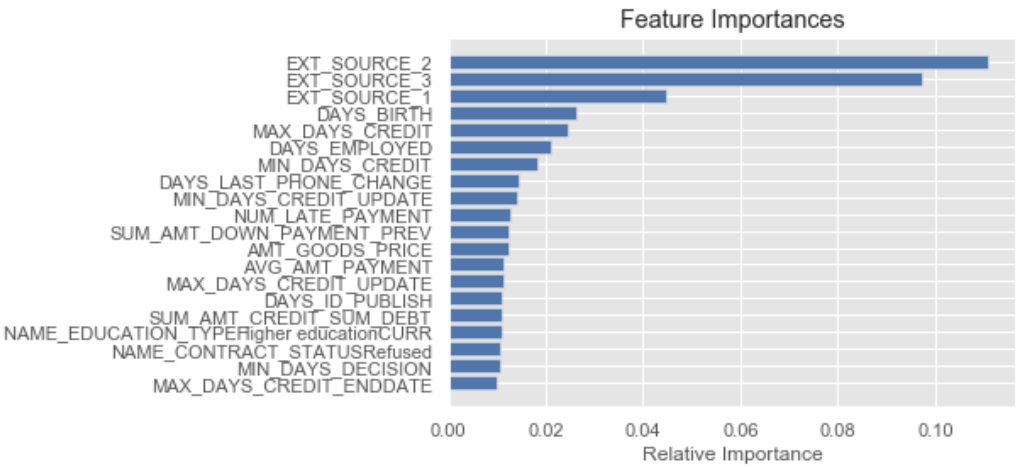
5-Fold Randomized Search CV is applied to tune the following hyperparameters, selecting the parameter combination that results in the highest average Area Under the ROC curve score on the five 1-fold validation sets:

- Logistic Regression – Penalty type (L1 or L2); regularization parameter (C). Highly corrected Inputs are removed and features are scaled.
- Random Forest – Number of trees (n), maximum depth (d)
- XGBoost – Number of trees (n), maximum depth (d), learning rate (r), minimum loss reduction required to make a further partition on a leaf node of the tree (gamma)
- LightGBM – Number of trees (n), maximum depth (d), learning rate (r)
- KNN – K
- SVM – Kernel function (linear, rbf, or poly), kernel coefficient (C), regularization parameter for L2 (gamma). Features are scaled for faster convergence.

Modeling – Performance

Random Forest

- Best hyperparameters: Number of trees = 400; maximum depth = 9
- Best average AUC score on validation sets: 0.750



AUC Score	hyper_1	hyper_2	hyper_3	hyper_4	hyper_5
validation_fold_1	0.746	0.744	0.760	0.757	0.763
validation_fold_2	0.724	0.722	0.738	0.736	0.740
validation_fold_3	0.728	0.727	0.744	0.742	0.746
validation_fold_4	0.736	0.732	0.749	0.748	0.751
validation_fold_5	0.737	0.735	0.748	0.746	0.750
Avg of Score	0.734	0.732	0.748	0.746	0.750
Std of Score	0.009	0.008	0.008	0.008	0.008

Modeling – Performance

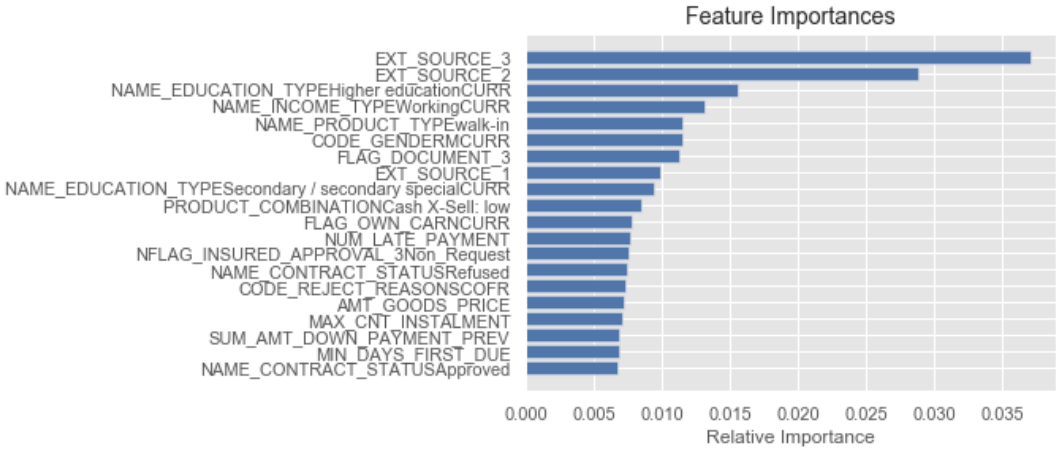
Logistic Regression

- Best hyperparameters:
  Penalty = L1;
  regularization parameter = 1
- Best average AUC score on validation sets: 0.764

Column	Coefficient Estimate
CLOSED_CT	4.685918
Active_CT	4.291237
SUM_AMT_CREDIT_PREV	3.592521
AMT_CREDIT	2.777601
SUM_AMT_CREDIT_SUM_OVERDUE	2.009202
NAME_YIELD_GROUPhigh	1.378000
FLAG_DOCUMENT_2	1.117748

XGBoost

- Best hyperparameters:
  Number of trees = 200; max depth = 5;
  learning rate = 0.1; gamma = 0.001
- Best average AUC score on validation sets: 0.772



Modeling – Performance

KNN

- Best hyperparameters: K = 18
- Best average AUC score on validation sets: 0.586

SVM

- Best hyperparameters: Kernel function = poly; C = 87; gamma = 0.001
- Best average AUC score on validation sets: 0.757

LightGBM

- Best hyperparameters: Number of trees = 400; max depth = 1; learning rate = 0.5;
- Best average AUC score on validation sets: 0.770

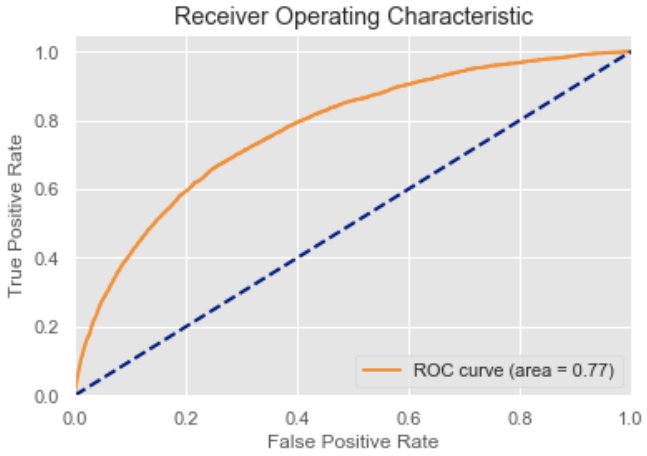
Model – Final Model and Evaluation

Summary:

XGBoost model has the highest average AUC score on the validation folds, and therefore selected to be the final model recommendation.

Model	Average AUC	STD AUC
Random Forest	0.7500	0.0076
XGBoost	**0.7722**	**0.0074**
LightGBM	0.7704	0.0065
Logistic Regression	0.7644	0.0068
KNN	0.5860	0.0053
SVM	0.7565	0.0013

	Precision	Recall	F1-Score	Support
Non-Default	0.96	0.71	0.82	56533
Default	0.18	0.69	0.28	4968



- Apply XGBoost model on the independent testing set, which is a random 20% of the original imbalanced dataset. AUC on the test set shows 0.7738
- Using 0.5 as cutoff threshold, precision and recall for each target group are shown in the report. Adjusting the cutoff threshold will lead to different precision and recall values and this will be left as a judgmental call from the credit analyst.

Potential Uses of the Model

- Important features of the model can help decision makers understand the variables that separate default loans from non default loans
- Probability of default for each loan can be leveraged jointly with other variables to design appropriate approval / decline / referral strategy that balances risk and reward
- This model can be used as one component of a suite of models that answers different business questions such as predicting the profit that a loan generates over a period of time.

Future Work

- Feature engineering can be further explored by looking into combinations of various columns, interaction terms or using different aggregation methods at sub-ID levels.
- 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.
- More missing value imputation techniques can be used to better impute the missing values based on overall shape of the distribution. Methods such as MICE, Datawig can be explored.
- More sampling techniques such as up sampling of the minority class, or SMOTE can be used to compare with the performances of down sampling.
- PCA or other dimension reduction methods can be leveraged to reduce the dimension of the feature space, however, model interpretability may be lost.
- 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.
- Other classification models such as neural networks, linear or quadratic discriminant analysis, AdaBoost, CatBoost, or other bagging and ensemble methods, can be experimented with this data in the future.

Q & A