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. For more details, interested readers can refer to Appendix A. Data Aggregation.

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.

In the next section we will show some EDA of some of the variables. For more detailed variable handling, please refer to Appendix B. Exploratory Data Analysis and Further Processing.

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.

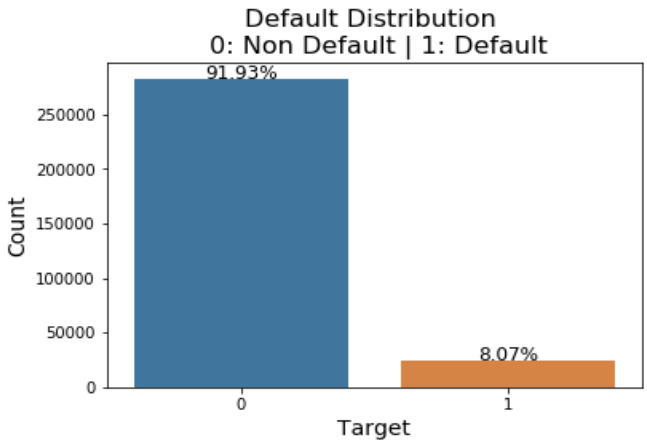


Figure 3.1

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.

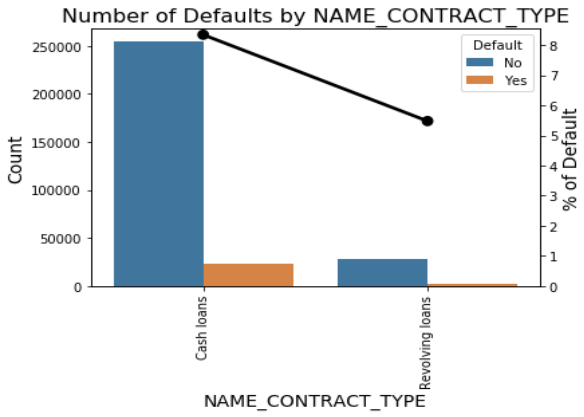


Figure 3.2

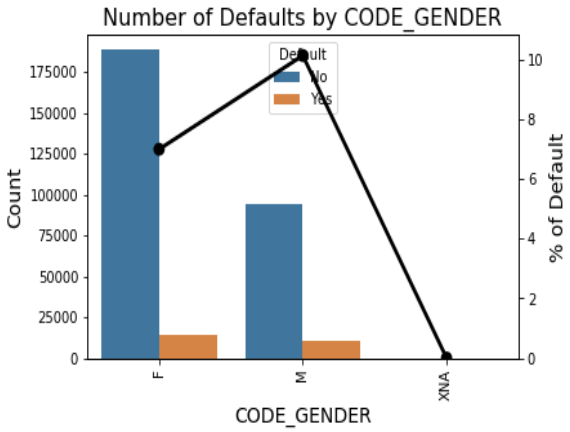


Figure 3.3

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.

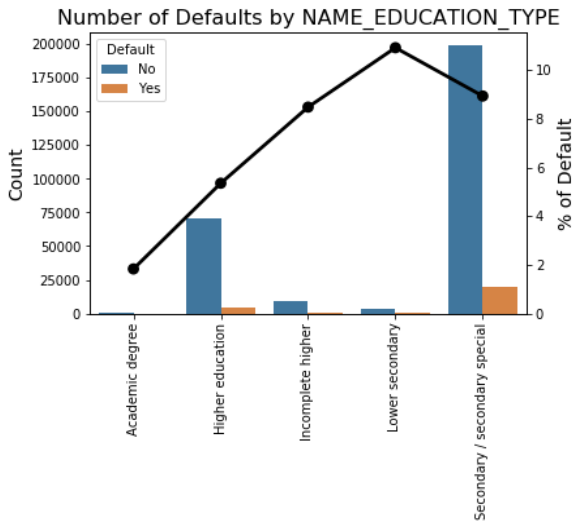


Figure 3.4

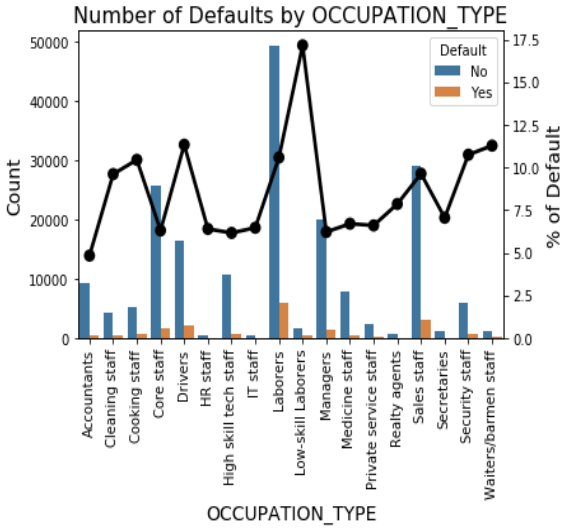


Figure 3.5

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. There is also high linear correlation between observation of client's social surroundings defaulted on 30 days past due and defaulted on 60 days past due, and we also remove one of them.

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 are highly correlated with 'LIVINGAREA' variables.

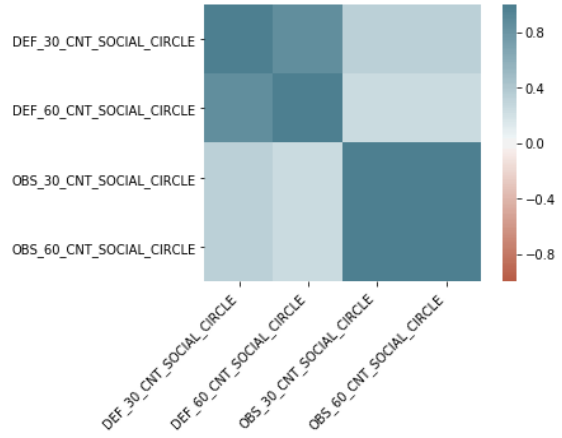


Figure 3.6

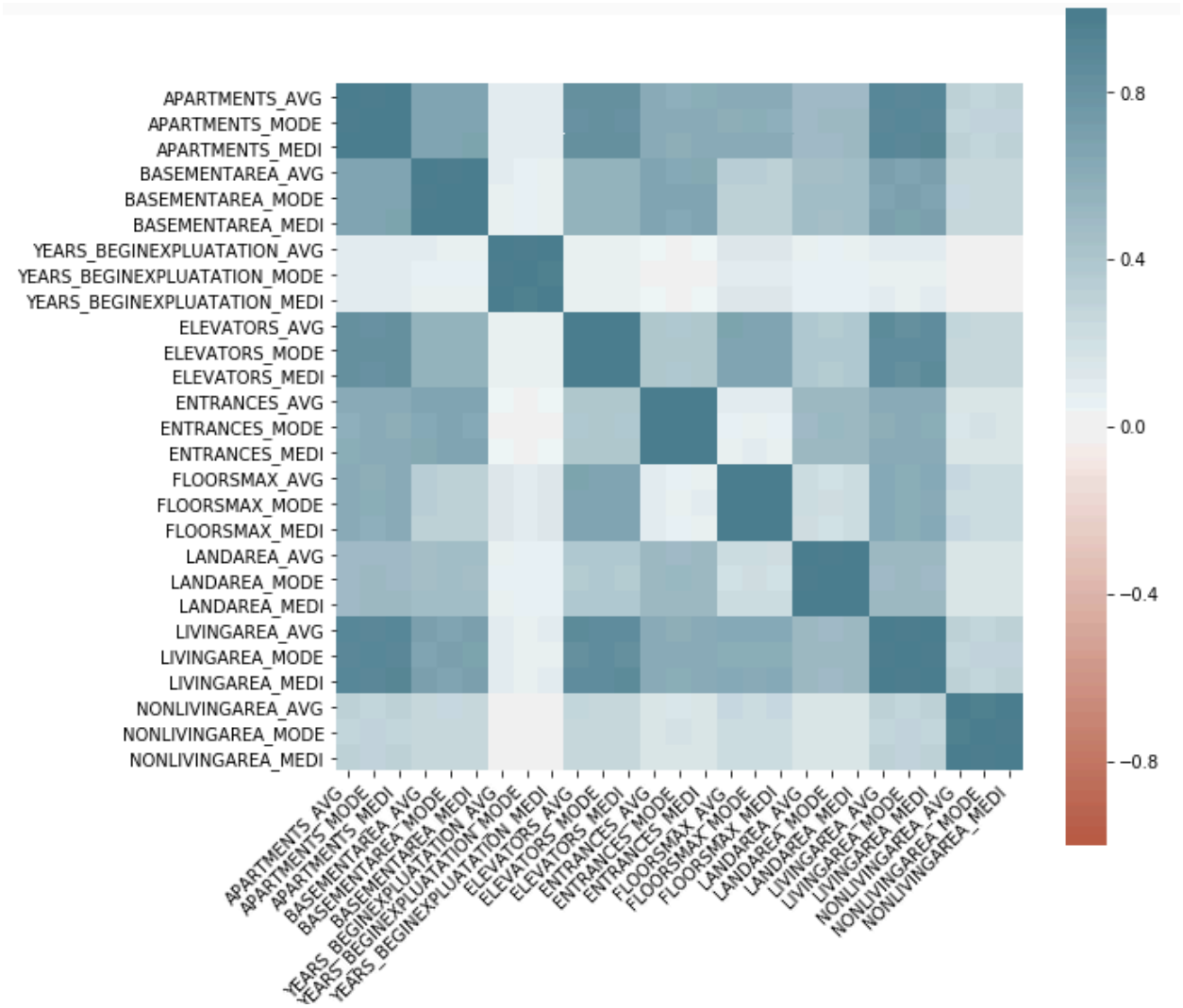


Figure 3.7

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.

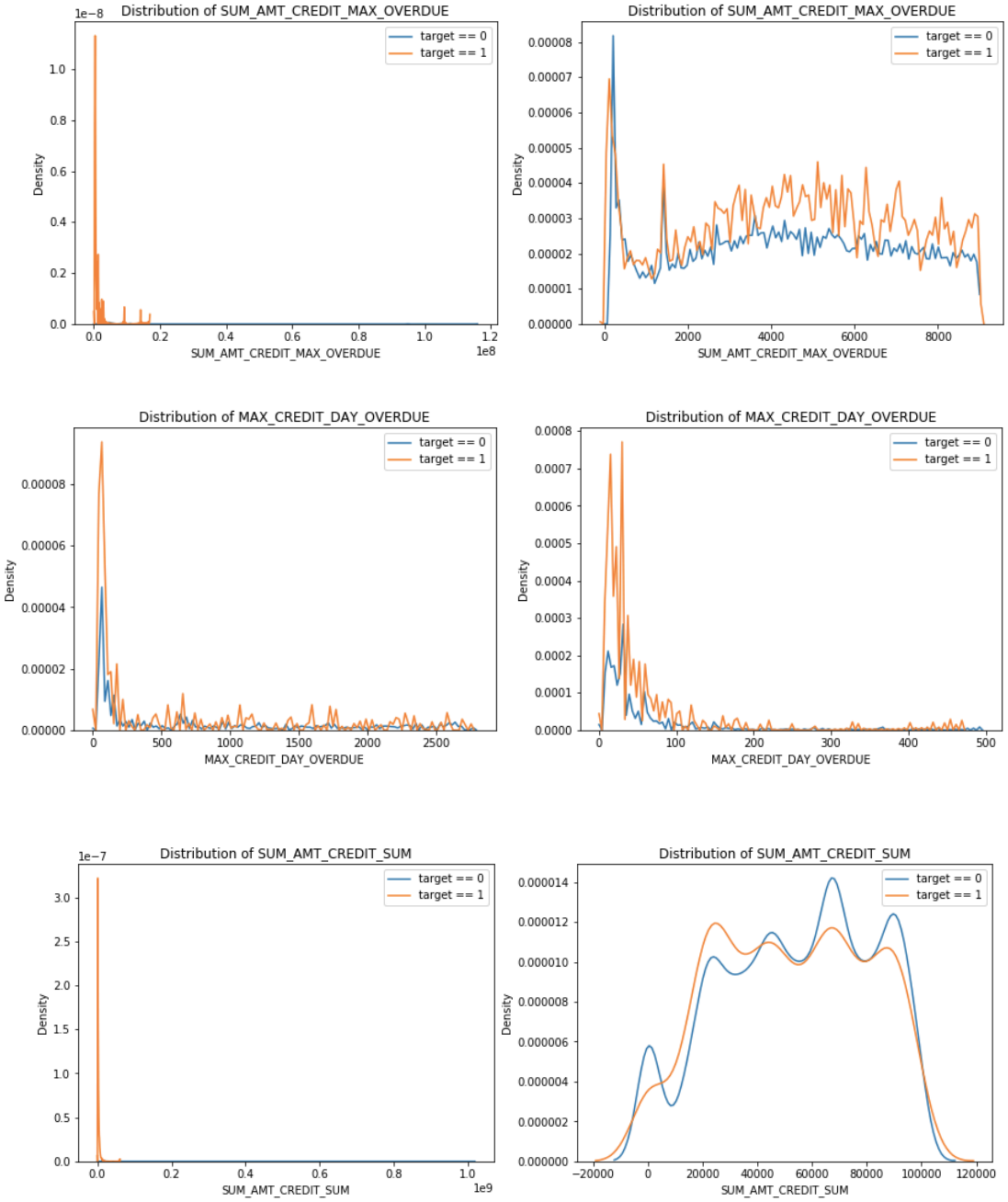


Figure 3.8

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

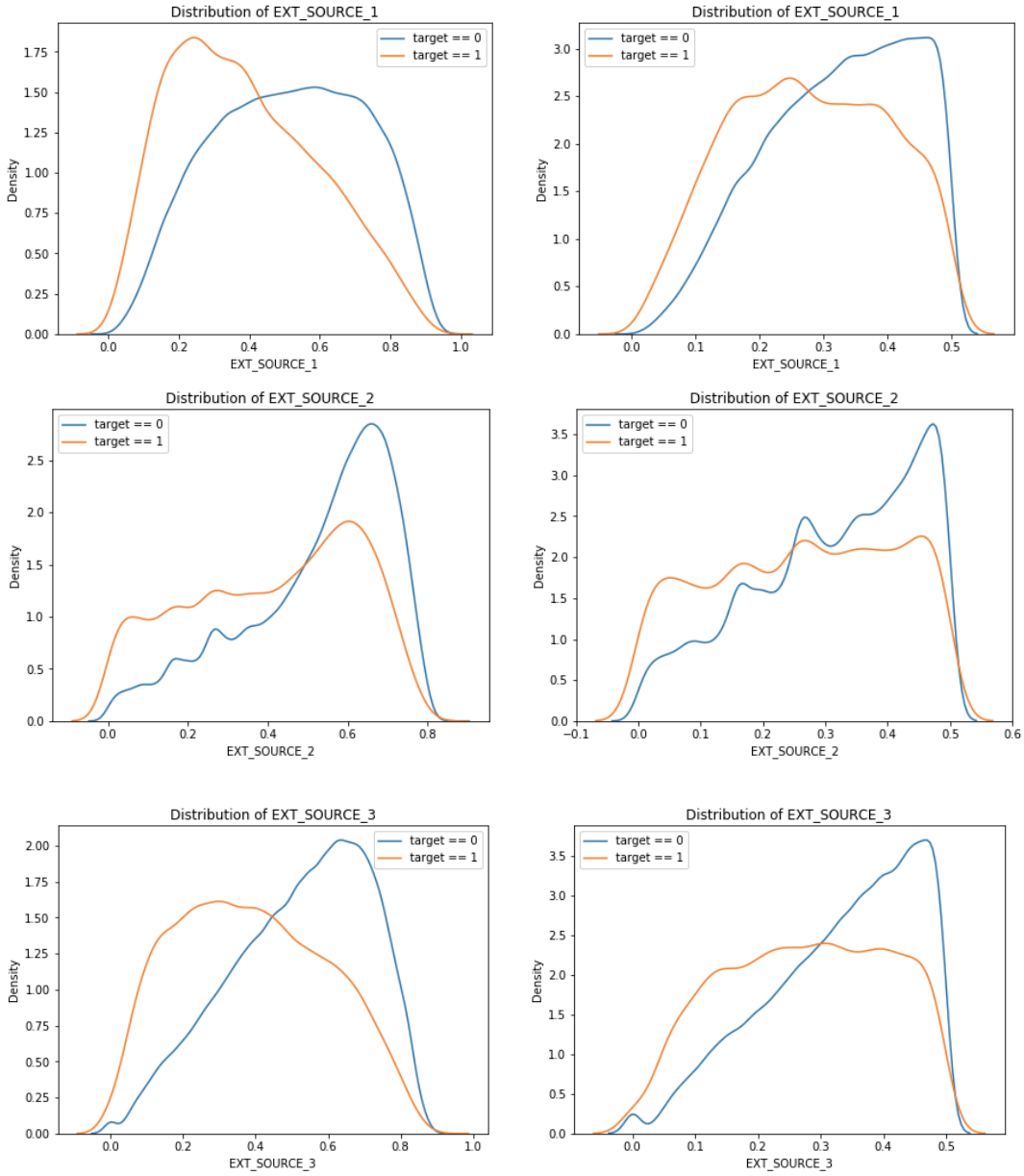


Figure 3.9

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.

- "MAX_DAYS_CREDIT_UPDATE" is an aggregated field similar to the variable above. It means over all the previous applications within the same current application ID, the maximum days before current application when the last information about the Credit Bureau credit come. This should be a negative number as it's counting backwards from the current application date. For all the positive numbers, define them as 0.

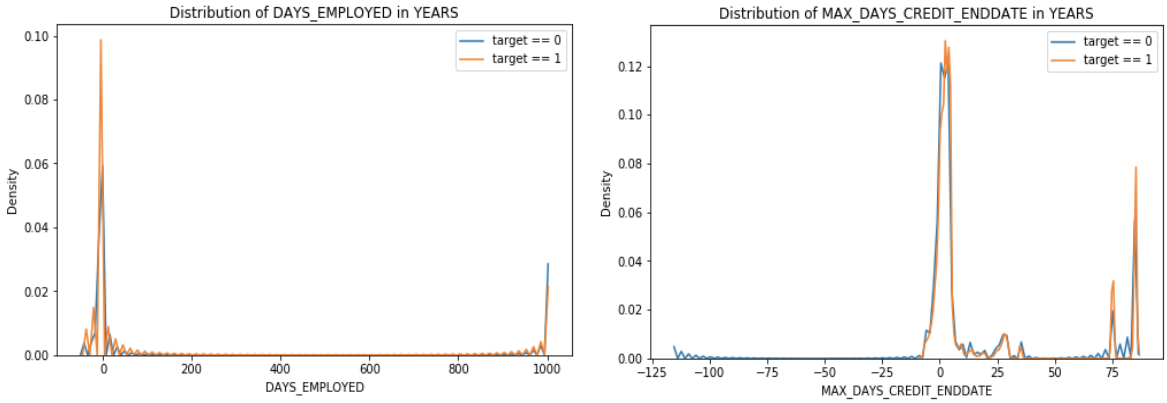


Figure 3.10

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

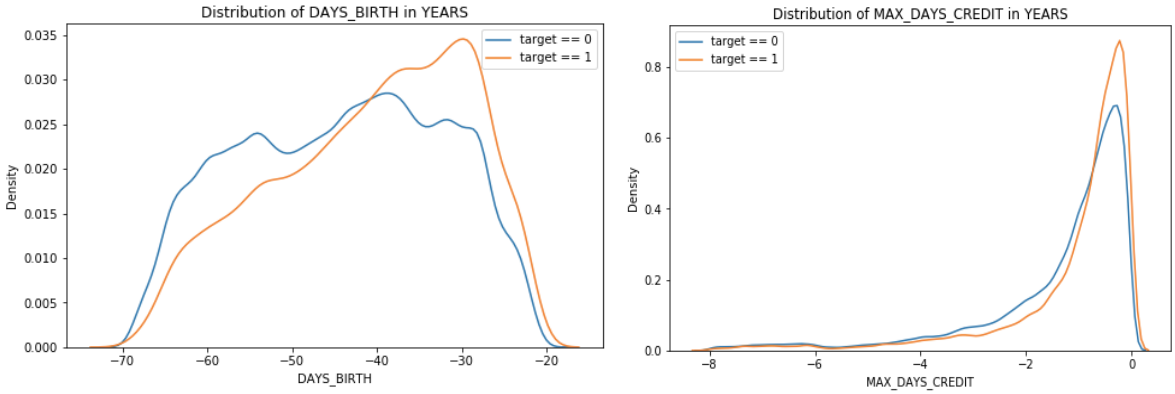
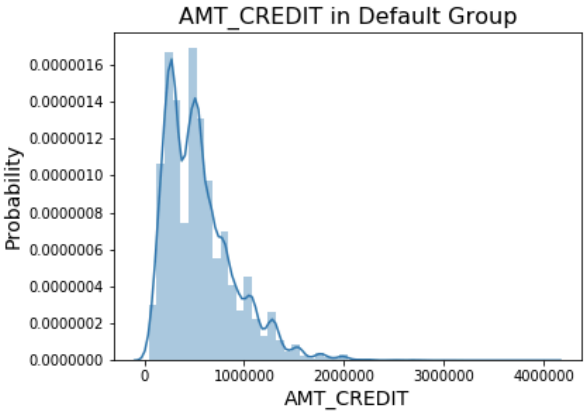
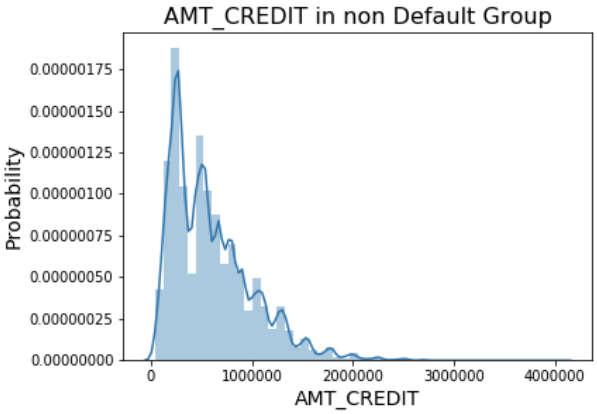


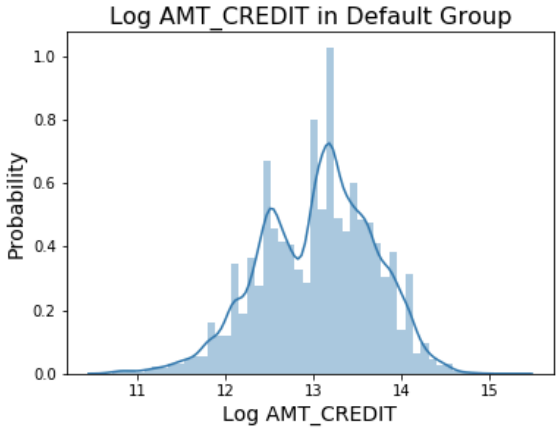
Figure 3.11

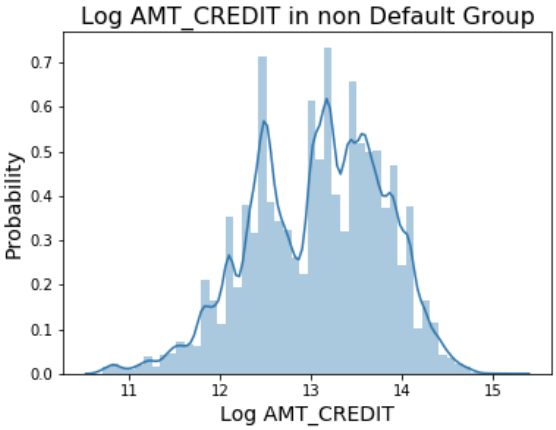
Distributions of "AMT_ANNUITY", "AMT_CREDIT" and "AMT_GOODS_PRICE" are skewed to the right, we recommend to take the log transformation to normalize the data (Figure 3.12).

Amount / Log AMT_CREDIT

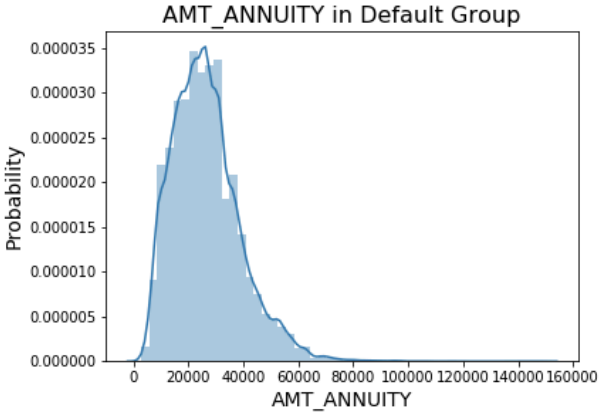


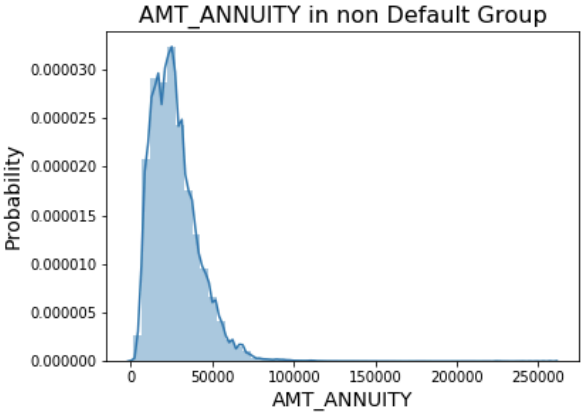






Amount / Log AMT_ANNUITY





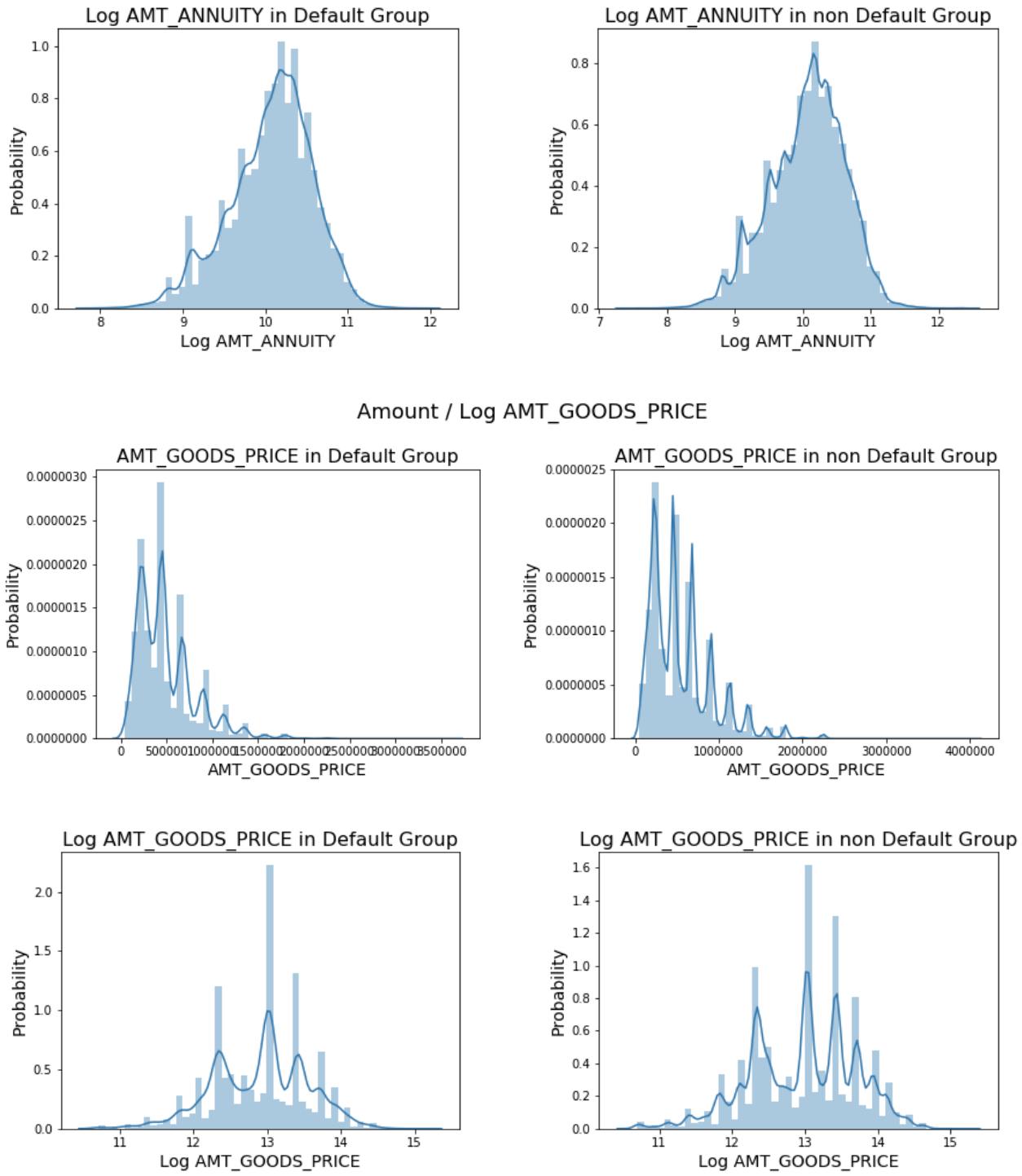


Figure 3.12

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).

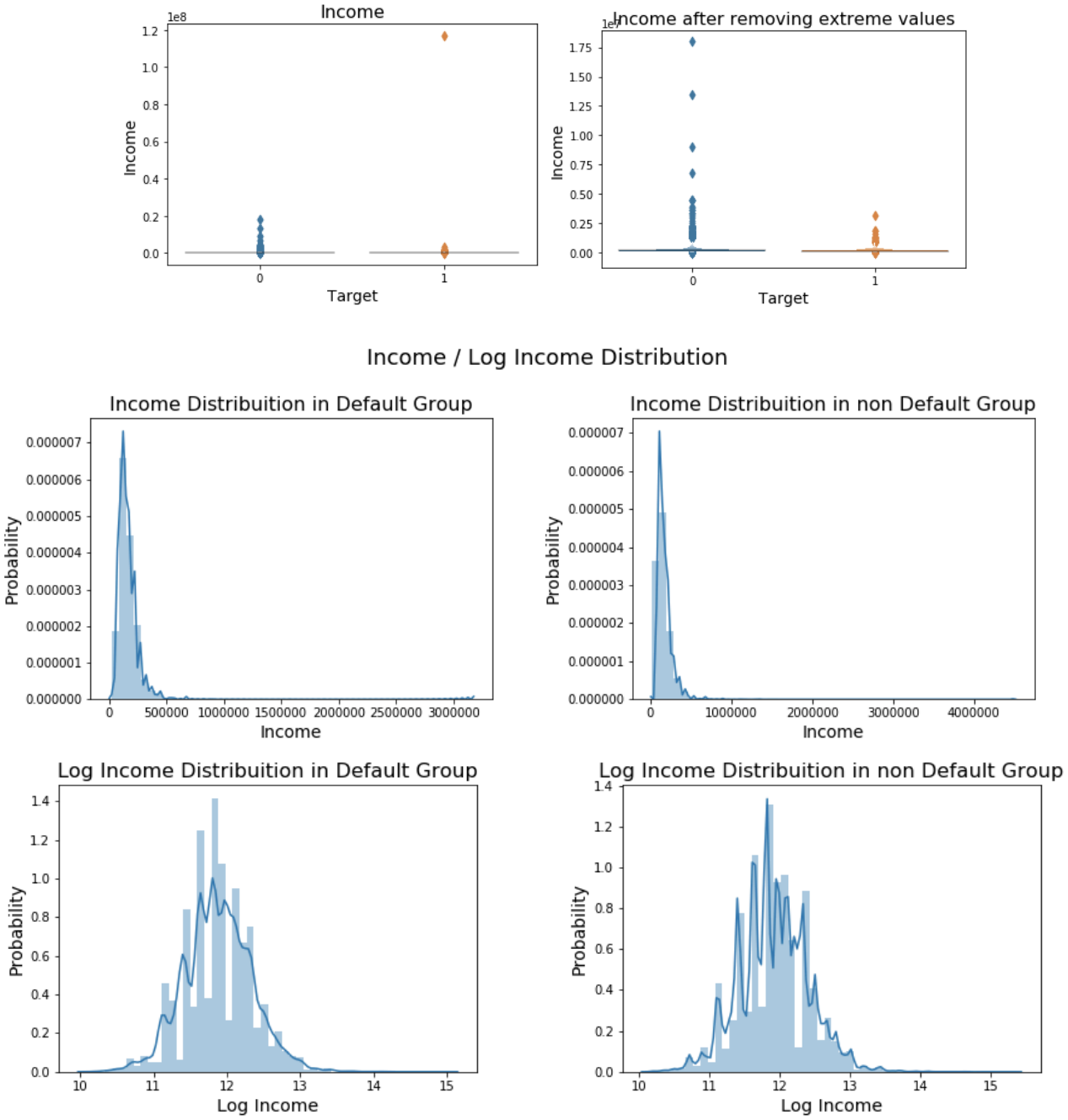


Figure 3.13

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. Using accuracy as criteria makes sense as we have balanced our training data. Next, each model will be 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.

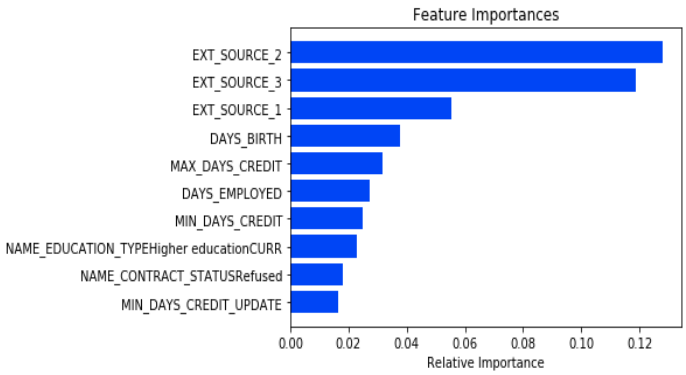


Figure 4.1.1

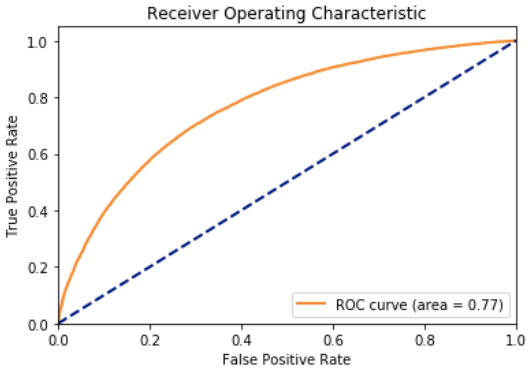


Figure 4.1.2

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

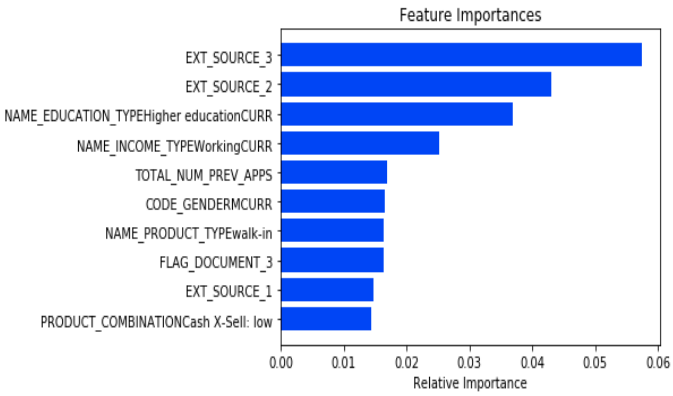


Figure 4.2.2

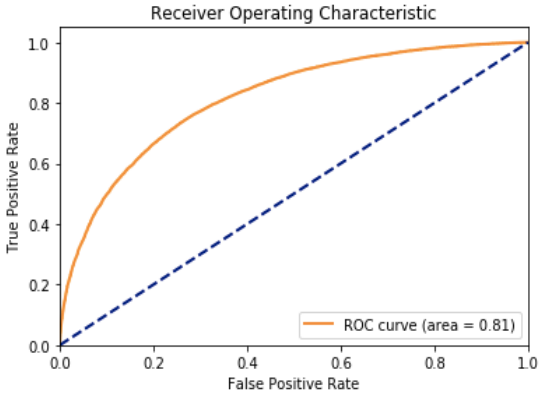


Figure 4.2.3

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

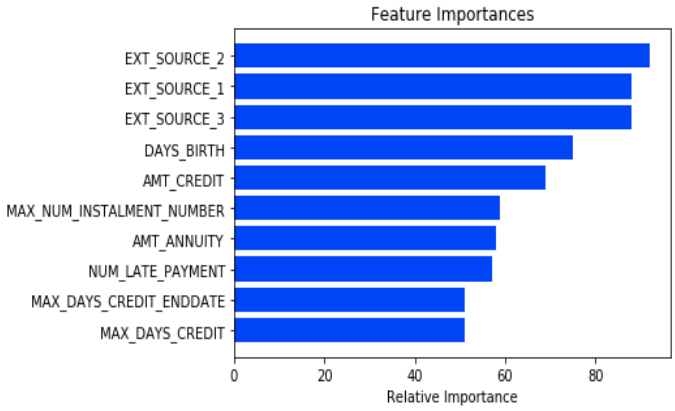


Figure 4.3.2

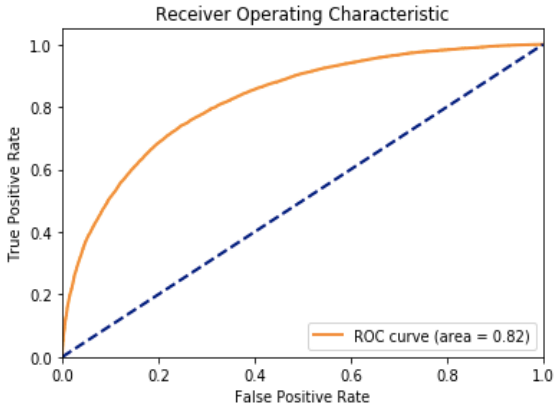


Figure 4.3.3

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

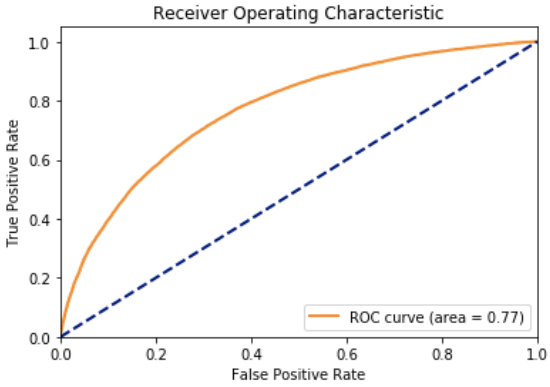


Figure 4.4.2

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 will suffer 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

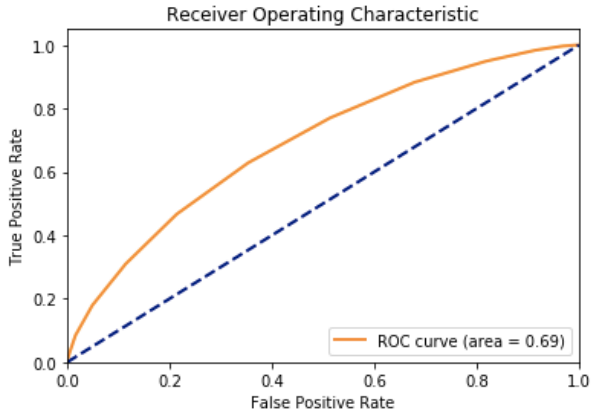


Figure 4.5.2

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

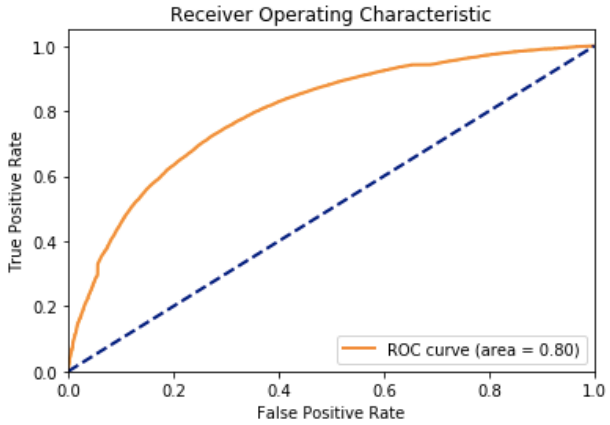


Figure 4.6.2

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

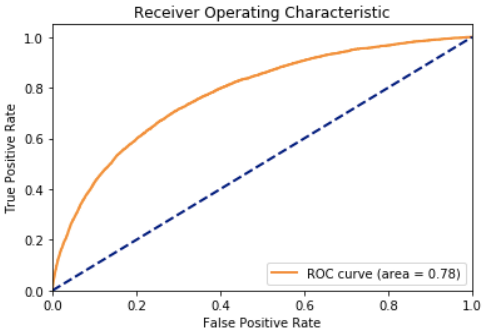


Figure 4.8.3

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 is our findings.

1. The model with the highest AUC (0.7465) 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. We believe more thoughtful feature engineering work can to be done and more models can be tried. 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

Appendix A. Data Aggregation

In this section, we discuss more detailed treatment of variables in each of the 6 datasets provided by Home Credit.

1. Bureau data: This dataset contains all of the client's previous credits provided by other financial institutions that were reported to Credit Bureau (CB), such as number of days past due on CB credit at the time of application. For each current application ID, there are as many rows as the number of credits the client had in Credit Bureau before the application date. That means each current application ID corresponds to multiple rows, and each row represents a different Bureau ID. This dataset has a total of 17 columns.

   At each row (Bureau ID) level:

   1) Drop CREDIT_CURRENCY column as the variable is not informative, 99% are values of 1.
   2) CREDIT_TYPE has 15 categories and many have very few counts, consolidate any type that are not in the largest 2 categories into a new category called "Loan", as these are all related to some type of loans such as car loans, Microloan and so on. Only keep 3 final categories: Consumer Credit, Credit Card, Loan. Then use one hot encoding to create 3 dummy variables corresponding to each credit types.
   3) Consolidate any non-active status in column CREDIT_ACTIVE into 'Closed' status. Then use one hot encoding to create 2 dummy variables corresponding to each status, active or closed.

   At each current application ID level:

   1) Count total number of Bureau ID's for each current application ID.
   2) Sum over each of the credit type dummy variables to get the total number of consumer credit, credit card and loan for each current application ID.
   3) Sum over each of the status dummy variables to get the total number of closed and active statuses for each current application ID.
   4) Variables named starting with "AMT" represent current amount of credits, debts, limits of credit cards, or max / sum of amount overdue shown on the Bureau, aggregate these variables using sum to capture the total amount for each current application ID.
   5) Variables named starting with "DAYS" represent at the time of application, how many days since Bureau credit ended, or remaining days of Bureau credit, or how many days before current application did client apply for Bureau credit, or days overdue and so on. Using max or min or both summary statistics to aggregate these columns is reasonable.
   6) Use max to aggregate CREDIT_DAY_OVERDUE to capture the maximum days overdue on CB for the same current application ID.
   7) Use max CNT_CREDIT_PROLONG to capture the maximum times was the Credit Bureau credit prolonged for the same current application ID.

2. Credit card balance data: This dataset provides the credit card balance information for previous application IDs in each month prior to current application. Variables include credit card limit during the month, amount drawing at ATM during the month, etc. for each previous application ID. Each current application ID corresponds to multiple previous IDs, and each previous ID corresponds to multiple rows and each row represents the month relative to current application, where -1 meaning the month prior to current application. This dataset has a total of 23 columns.

   At each row (MONTHS_BALANCE) level:

   1) Variable NAME_CONTRACT_STATUS has 7 categories and very few counts in many categories. To reduce the number of categories, create a new dummy variable "STATUS_ACTIVE" with only 2 categories, value 1 meaning active, 0 meaning inactive.
   2) Variable SK_DPD_DEF means number of days past due during each month in the past before the current application. Since over 97.5% of the values are 0, and only less than 0.5% of the values are >= 1, create a new dummy variable to group all the > 0 values into 1 category. Similar treatment is used on variable SK_DPD.

   At each unique current and previous application ID combination level:

   1) Variables named starting with “AMT” are amount of credit card balances, credit limit, ATM drawings, payments etc. in each of the past months before the current application. Aggregate each “AMT” variable using either average, maximum or sum over the past months.
   2) Variables named starting with “CT” are number of drawings in each of past months before the current application. Aggregate each “CT” variable by sum to get the total number of drawings over the past months.
   3) Use max function to aggregate the dummy variable created from STATUS_ACTIVE, to represent whether the record is active within each unique combination of current application ID and previous application ID.
   4) Sum over the dummy variables created from SK_DPD_DEF to get the total number of records having SK_DPD_DEF > 0 within each unique combination of current application ID and previous application ID. Similar treatment for variable SK_DPD.

   At each current application ID level:

   All variables are aggregated again using either the sum or average function, to get the summarized value within each unique current application ID, over all previous application IDs.

3. Pos Cash balance data: This dataset provides the monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit. Similar to the credit card balance dataset, each current application ID corresponds to multiple previous IDs, and each previous ID corresponds to multiple rows and each row represents

the month relative to current application, where -1 meaning the month prior to current application. This dataset has a total of 8 columns.

At each row (MONTHS_BALANCE) level:

1) Since NAME_CONTRACT_STATUS has 9 categories and very few counts in some categories, to reduce the number of categories, only keep Active and Completed categories, all other statuses are consolidated into 1 category called 'Other'.

At each unique current and previous application ID combination level:

1) Further reduce the number of categories in NAME_CONTRACT_STATUS by creating a new dummy column to represent completed contracts (value 1). All others will be considered as non-completed (value 0).
2) Variable CNT_INSTALMENT means term of previous credit. Create 2 columns to keep the max and min term of previous credit within each unique combination of current application ID and previous application ID.
3) Variable CNT_INSTALMENT_FUTURE represents Installments left to pay on the previous credit. Create 2 columns to keep the max and min.
4) Treatment for variable SK_DPD and SK_DPD_DEF are similar to the credit card balance dataset above.

At each current application ID level:

All variables are aggregated again using either the sum or min, max function, to get the summarized value within each unique current application ID, over all previous application IDs.

4. Installment payment data: This dataset provides past payment data for the previously disbursed credits in Home Credit. Each current application ID corresponds to multiple previous application IDs and each previous application ID contains multiple rows, where one row for every payment that was made plus one row each for missed payment. One row is equivalent to one payment of one installment OR one installment corresponding to one payment. This dataset has a total of 8 columns.

   At each unique current and previous application ID combination level:

   1) Keep the min and max of variable NUM_INSTALMENT_VERSION and NUM_INSTALMENT_NUMBER. The version number signifies payment parameter changes; however, no additional information is available to further engineer this feature.
   2) 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 number of late payments.
   3) Keep average AMT_INSTALMENT and average AMT_PAYMENT across different rows.

At each current application ID level:

Logic of aggregation at SK_ID_CURR level are the same as those used in aggregating at the SK_ID_PREV level described above.

5. Previous application data: This dataset contains information about all previous applications for Home Credit loans of clients who have loans in the current application. Each current application ID corresponds to multiple rows which represent one previous application ID. This dataset has a total of 38 columns.

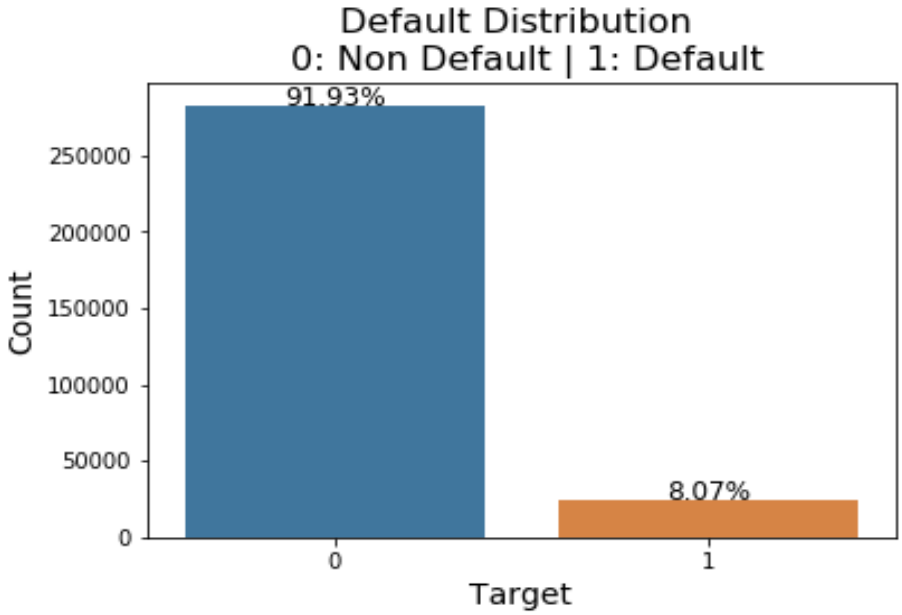
    1) For variables named starting with “AMT” or “RATE”, such as AMT_ANNUITY, AMT_CREDIT and so on, aggregate by summing up the total annuity, total amount of requests, total credit, down payment etc. information over all previous applications. Get the maximum down payment and interest rate etc. from all of the previous applications.
    2) For categorical variables named starting with “NAME” and a few others, first do one hot encoding at each row level, then sum at SK_ID_CURR level to get the total number of instances over all previous application IDs.
    3) For variables named starting with “DAYS” such as the first and last due days of each previous application, get the min and max of days (relative to current application date) over all previous applications.
    4) For columns with high cardinality, first group some categories with fewer frequencies, then do one hot encoding at each row level, and finally, sum at SK_ID_CURR level to get the total counts that previous application appeared in that category, within the same current application ID. As an example, variable "NAME_CASH_LOAN_PURPOSE" has over 20 categories, group the categories whose frequency are <= 20000 to simplify the variable into 4 categories. Then use one hot encoding to create 4 dummy variables representing the presence of each of the 4 categories. Finally, sum up at current application ID level to get the total number of counts that belong to each of the 4 categories.

6. Current application training data: This 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 dataset.

Appendix B. Exploratory Data Analysis and Further Processing

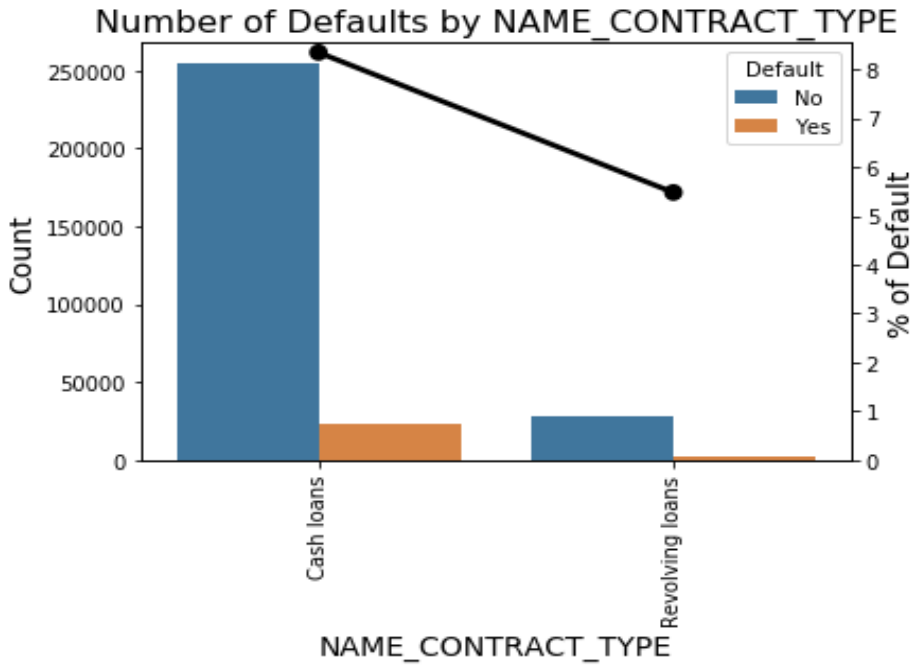
At the end of the last section, all datasets coming out of the 6 sources are combined into one table, and categorical variables only exist in the main lead application table, as the categorical variables in all 5 other tables are all aggregated and summarized into numeric variables using one hot encoding and summary functions. Further preprocessing of the combined table will be done in this section, mainly treating and exploring the categorical and numeric variables in the lead table. First of all, columns with over 60% missing values are deleted.

1. The nature of default data is highly imbalanced, as shown in the following graph, overall default percentage is 8.07% in the data set.

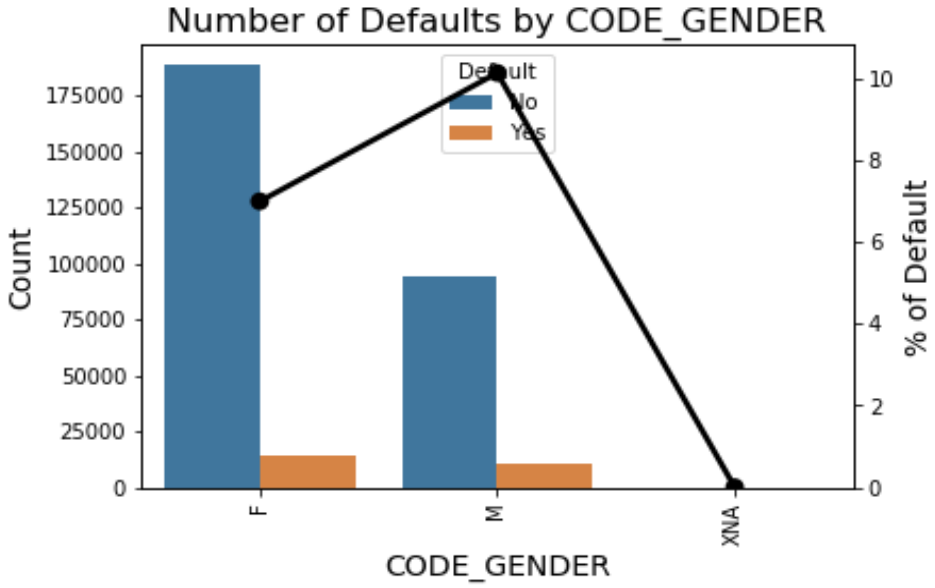


2. Exploration of categorical variables. Looking at the default rate within each category in a categorical variable helps visualizing which variables distinguish default vs. non default records better. Due to the vast number of categorical variables in the dataset, here we only show a few.

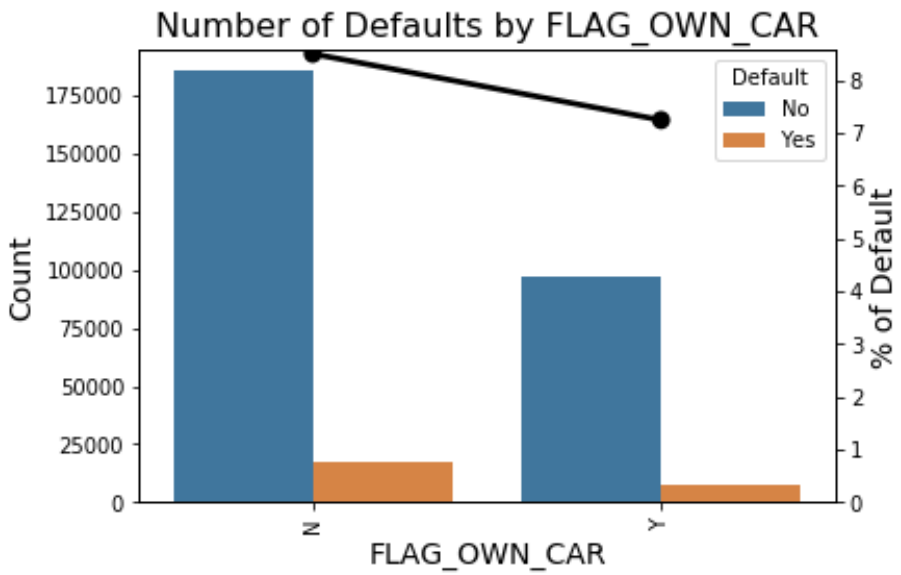
1) Contract type. Revolving loans group appears to have a lower default rate than the Cash loans group. A hypothesis t-test can be done to test if the difference in default rate is statistically significant.



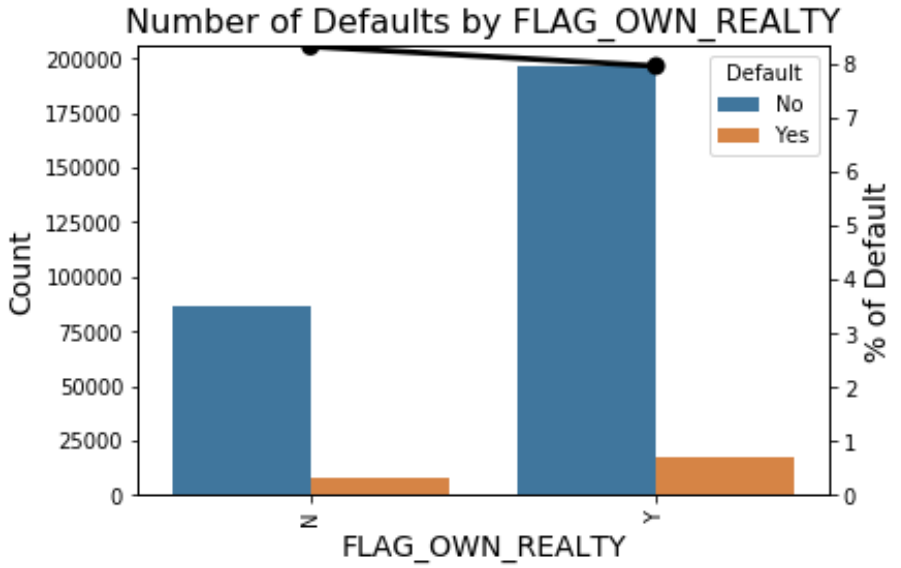
2) Gender. Notice that there are 3 gender groups, we will remove the 4 records having gender = XNA. It can be seen from the graph that majority of the applicants are female, and the default rate within female applicants is lower than that in the male group.



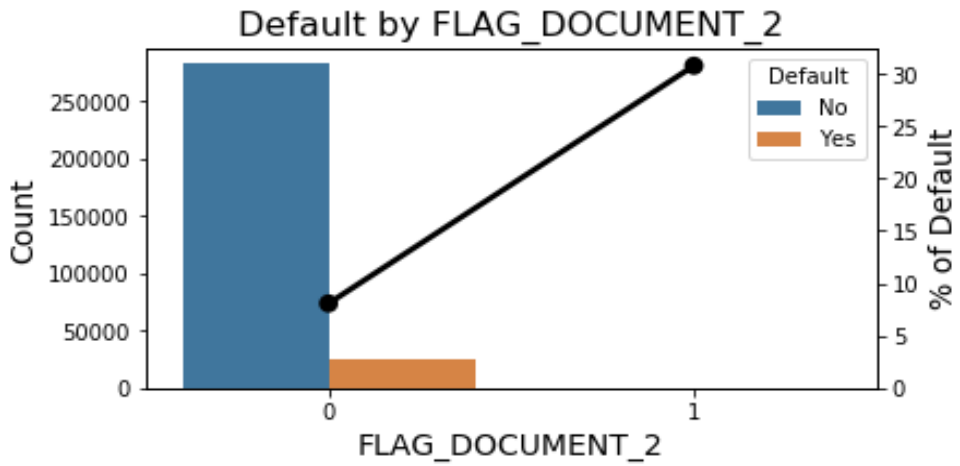
3) Own a car or not. Majority of the applicants do not own a car, default rate within applicants who own a car is 7.24%, whereas the default rate in the other group is 8.5%. A hypothesis t-test can be done to conclude if the difference is statistically significant or not.

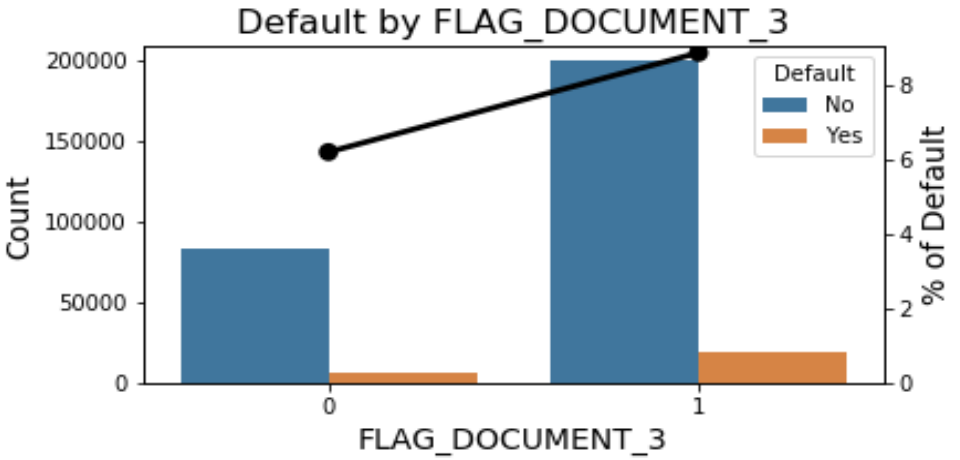


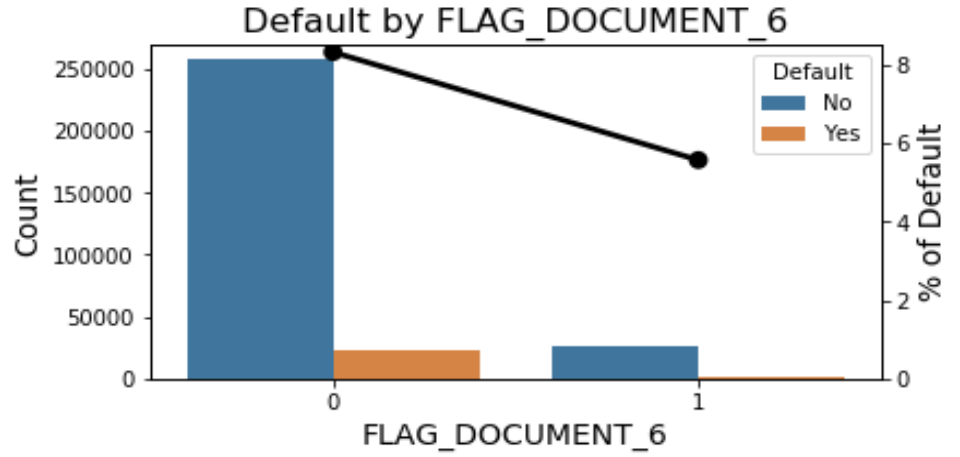
4) Own realty or not. Majority of the applicants own realty; however, the default rate is pretty close in the 2 groups, which may indicate the variable does not have strong power in separating default loans from non-default loans.

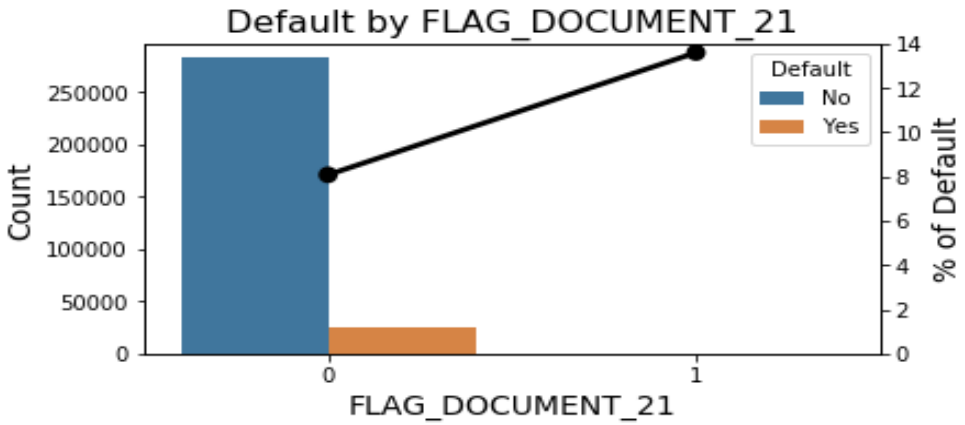


5) If documents 2 ~ 21 are provided by the applicant during the application. There are no detailed descriptions of what each document is, we simply check the distribution of each document variable to see how they distinguish default from non-default applications. Here we only display the graphs of a few document variables. Except for document 3 where a greater number of applicants provided the document, for all the rest documents, majority of the applicants did not provide. For most of the documents, the default rate in the group that provided the document is lower than the group that didn't provide. But there are a few documents that show the opposite behavior.

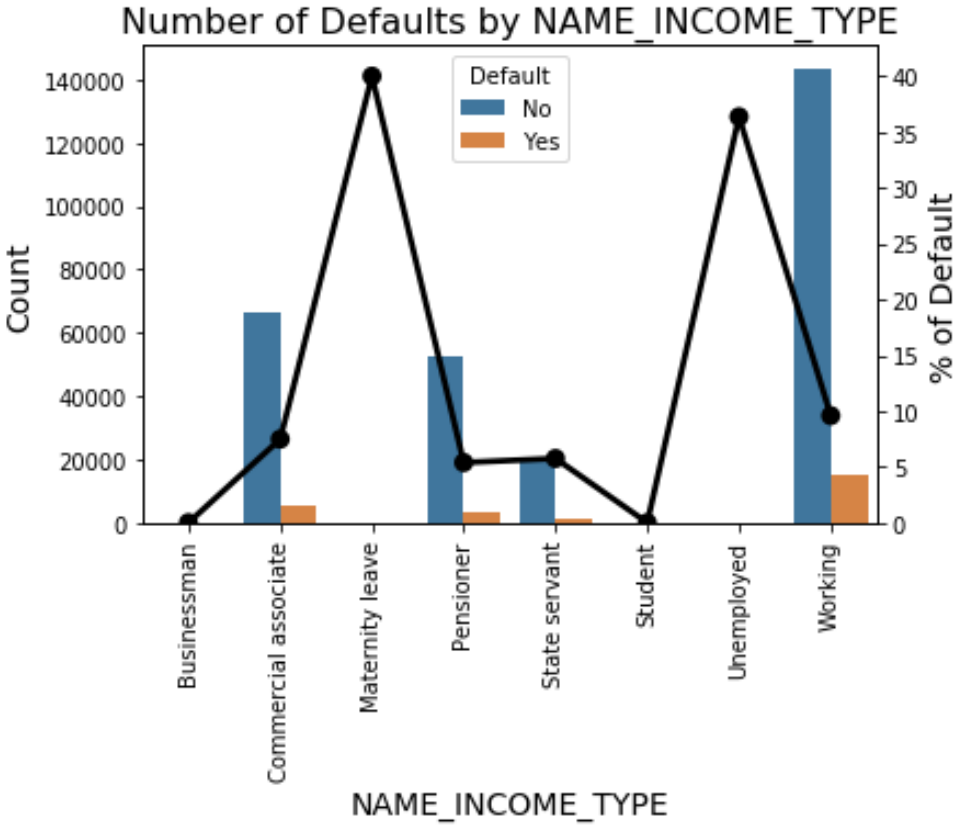

Default by FLAG_DOCUMENT_2
Default
No
Yes
Count
% of Default
FLAG_DOCUMENT_2
0
1
0
50000
100000
150000
200000
250000
0
5
10
15
20
25
30


Default by FLAG_DOCUMENT_3
Default
No
Yes
Count
% of Default
FLAG_DOCUMENT_3
0
1
0
50000
100000
150000
200000
0
2
4
6
8

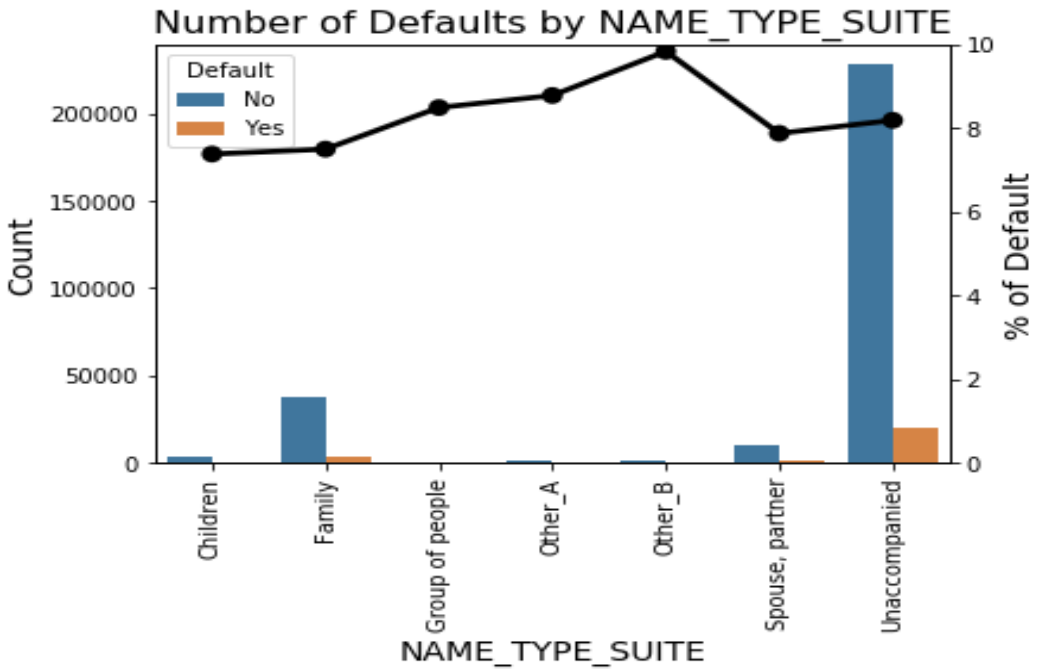

Default by FLAG_DOCUMENT_6
Default
No
Yes
Count
% of Default
FLAG_DOCUMENT_6
0
1
0
50000
100000
150000
200000
250000
0
2
4
6
8

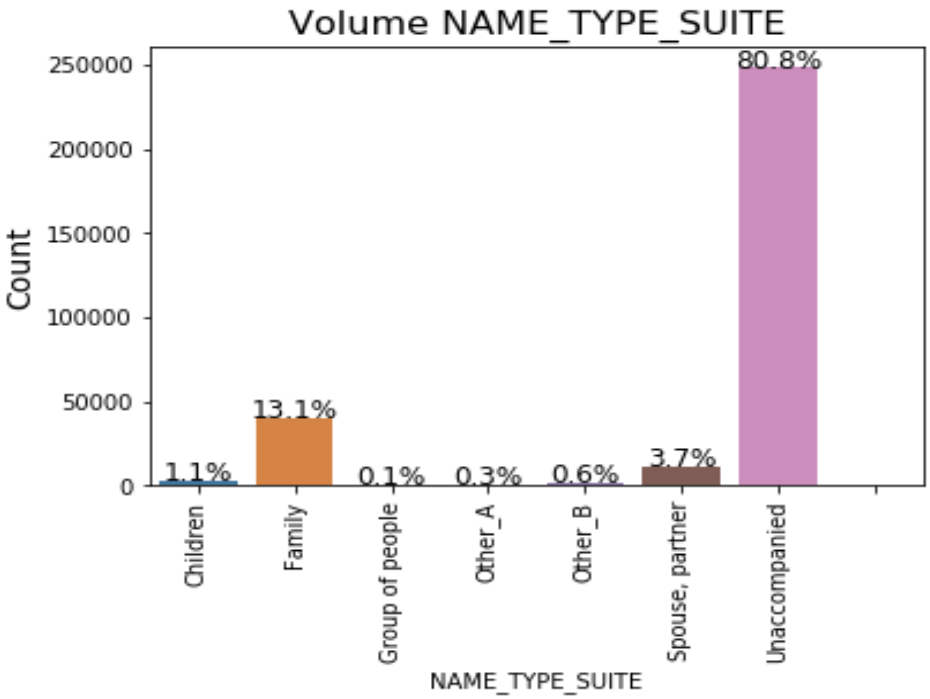


6) It can be seen from below that applicants who are on maternity leave or who are unemployed have the highest default rate. This make sense as these applicants may not have consistent regular income to pay the loan.

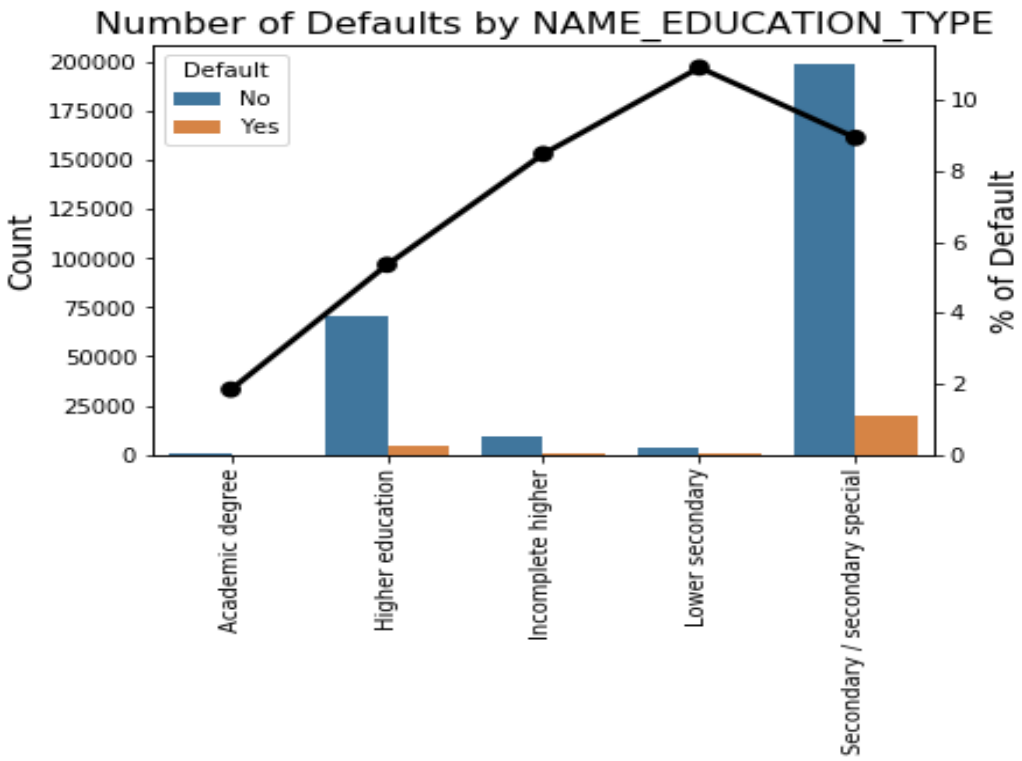


7) Accompaniment of applicant at the time of application. In general, this variable does not provide much separation power. Majority of the applicants came alone without any accompaniment. Notice that there are ‘nan’ values in the variable that need to be coded as a new category ‘UKN’ indicating unknown.

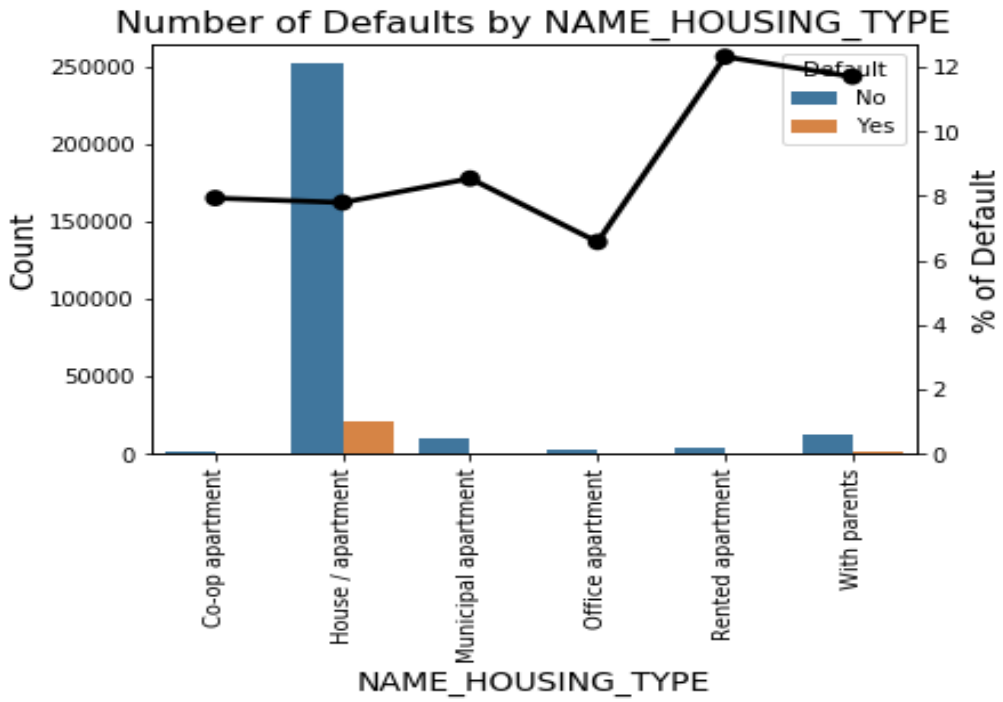




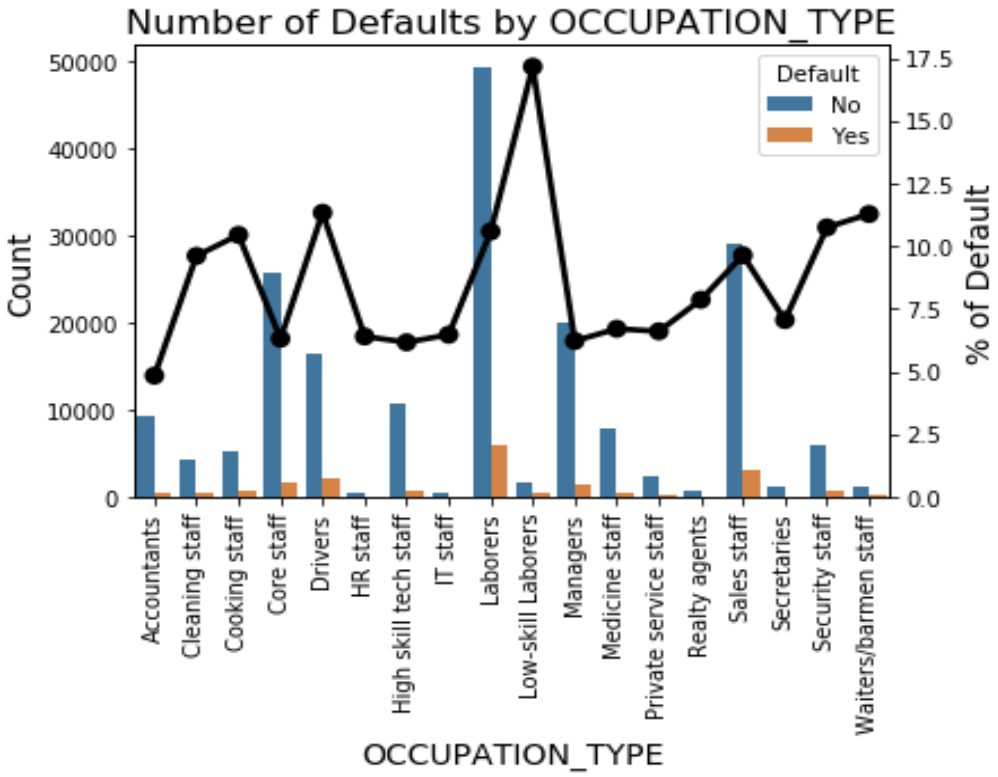
8) Education level. There is a rank ordering in default rate by education level. In general, applicants with low secondary education has the highest default rate. As we can imagine education level is correlated with type of job the applicant can take and highly correlated with income, which is essential to repayment of a loan.

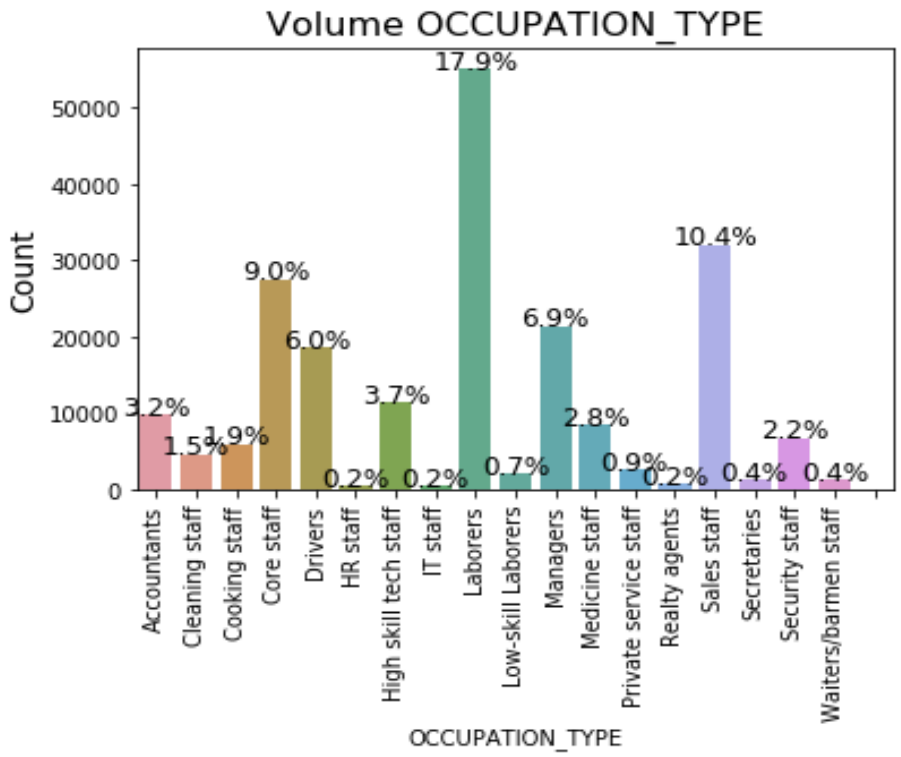


9) This variable again is highly correlated with the financial stability of the applicant. People who rent a house or live with parents may not have a regular income, thus having difficulty repaying the loan.

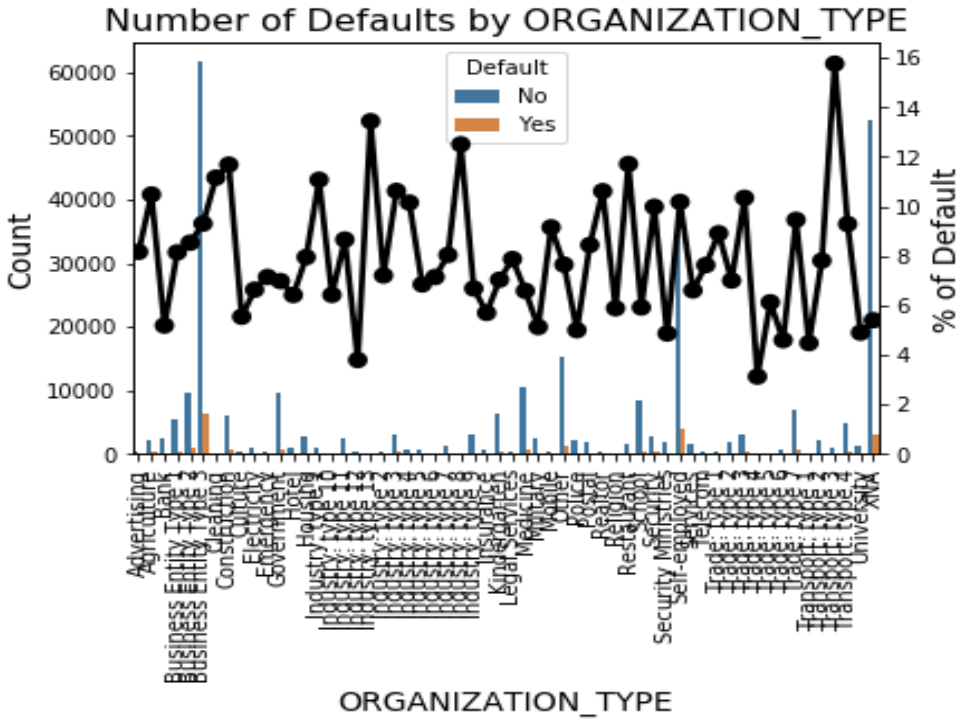


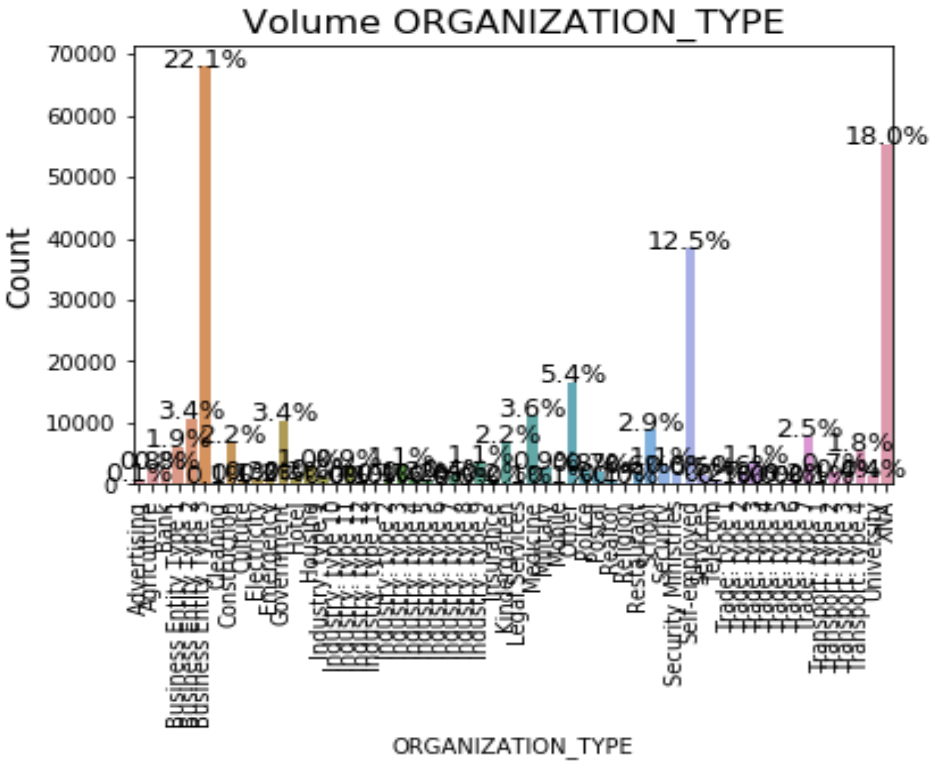
10) Occupation type. This variable has over 15 categories, we group the categories having <= 10000 counts into 1 group. Also notice ‘nan’ values that needs to be coded into a new category.



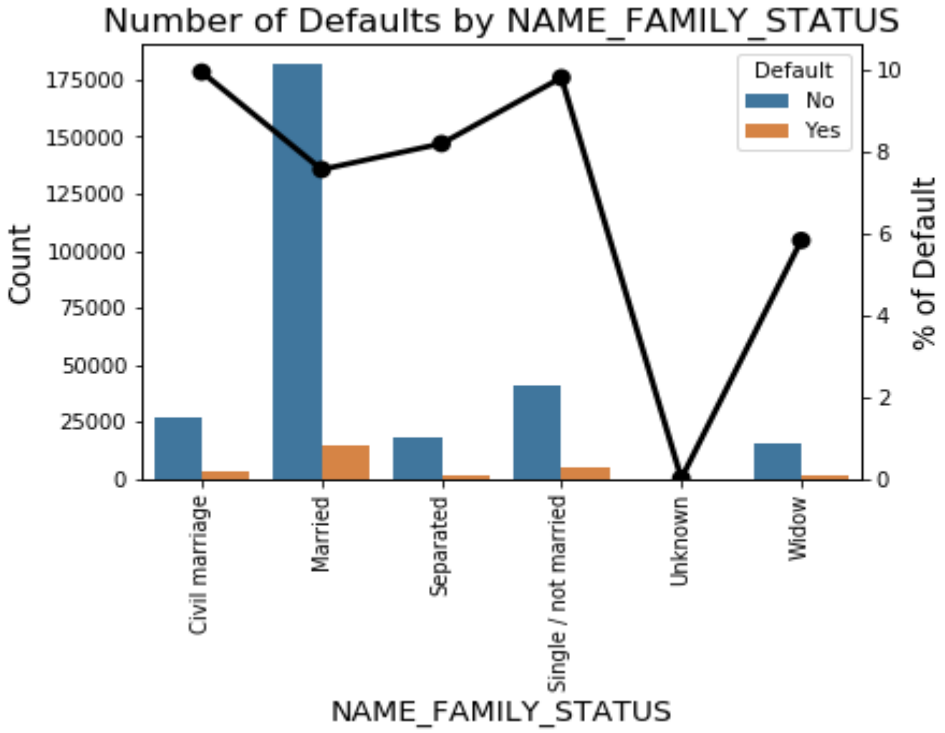


11) Variable organization type contains more than 20 categories and ‘nan’ values in the column, group the categories with < 30000 counts into 1 group and code ‘nan’ values into a new category.

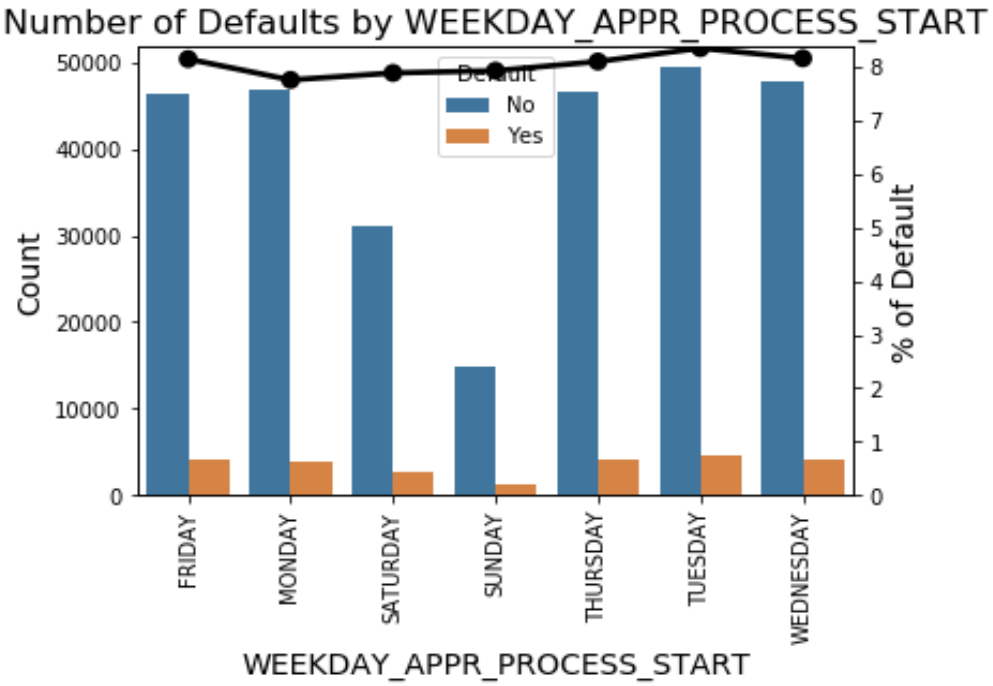




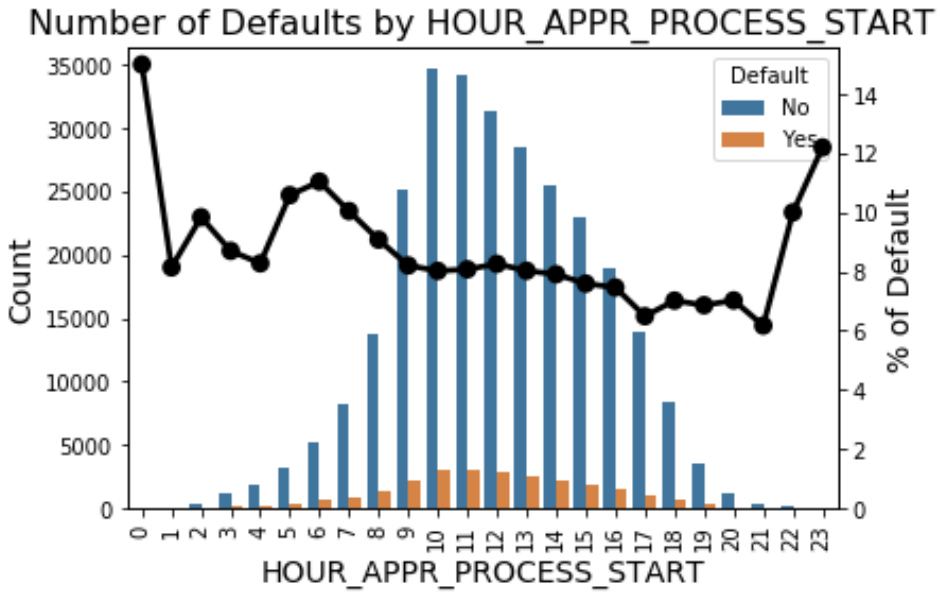
12) Family status.



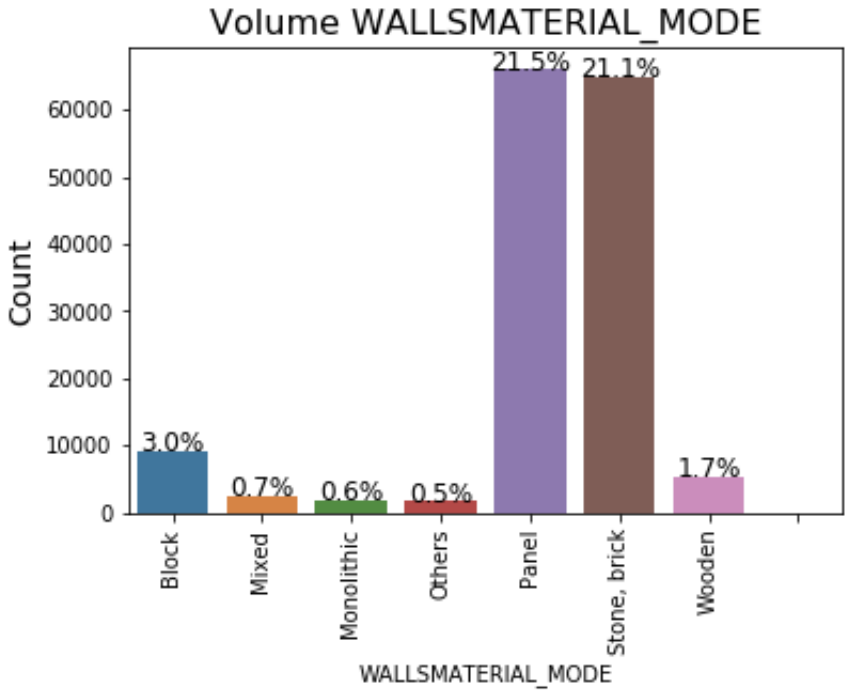
13) Application start time by weekday. It is expected that application volumes are higher during weekdays compared to weekends.



14) Application starting time by hour of the day. Clearly volumes are higher during working hours between 9am and 4pm, and much lower before 9am or after 4pm.

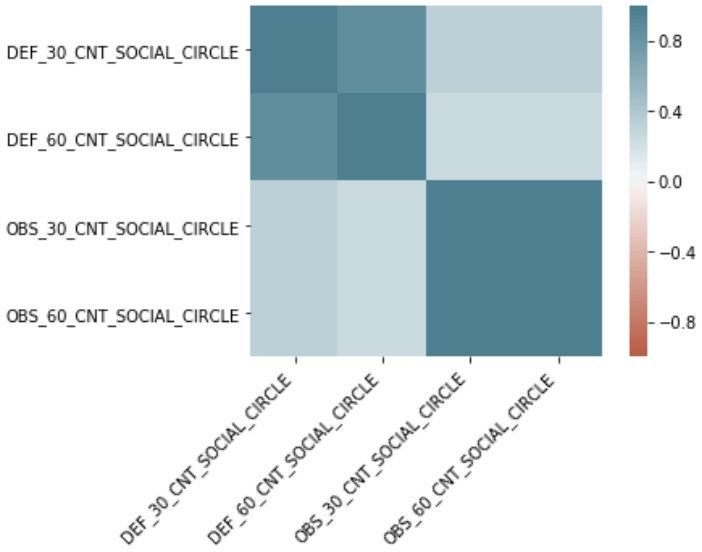


15) Wall materials. This variable tells the wall material of the house that the applicant lives in. Note there are 'nan' values that need to be treated.

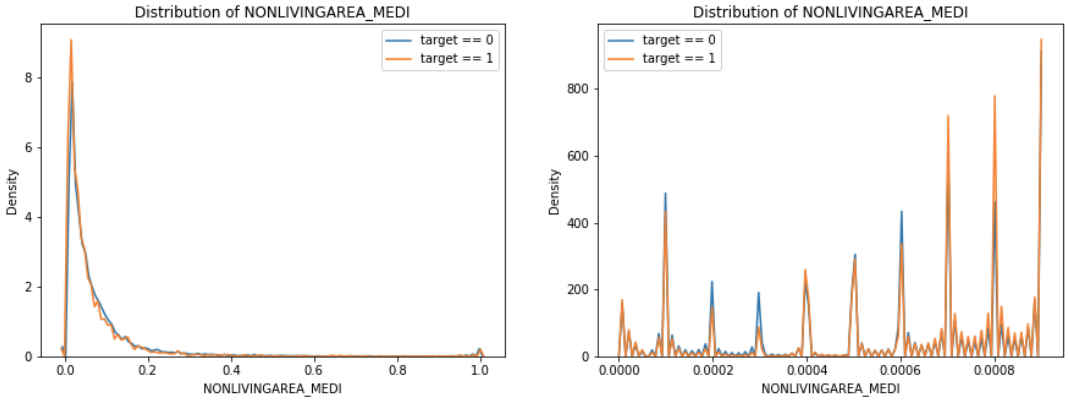


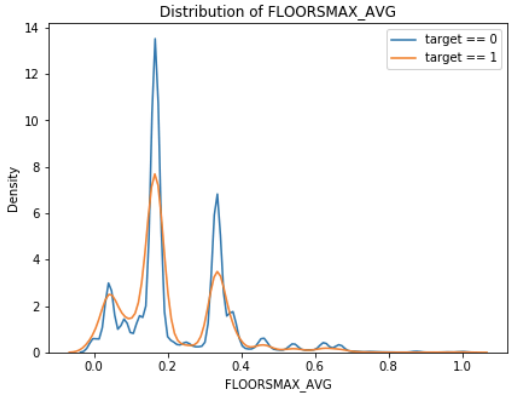
3. Exploration of numeric variables. Here we explore some numeric variables by creating the KDE plots for each variable and split by default vs. non-default.

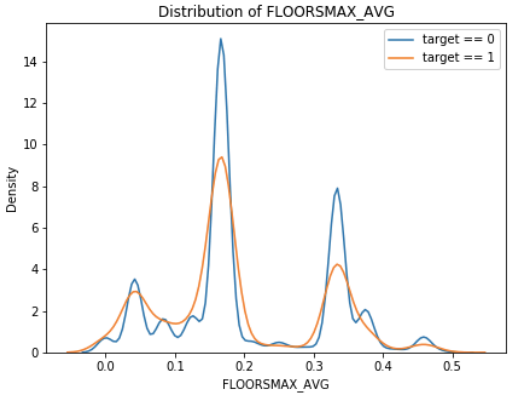
1) There is perfect positive linear correlation between observations of client's social surroundings with observable 30 days past due and 60 days past due. As a result, remove one of them from the dataset. There is also high linear correlation between observation of client's social surroundings defaulted on 30 days past due and defaulted on 60 days past due, so remove one of them.

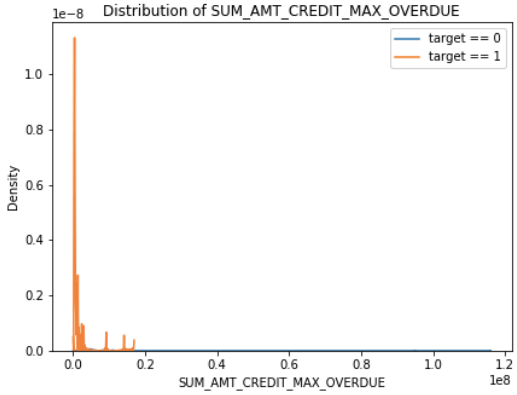


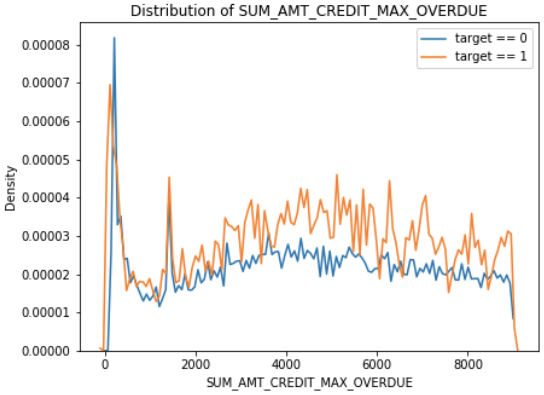
2) Below are KDE plots for some numeric variables. Left panel shows the original range of the variable. Since most of the variables are highly skewed to the right, we zoom in the head of the distribution and plot it on the right panel. A few variables show distinction between the default and non-default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

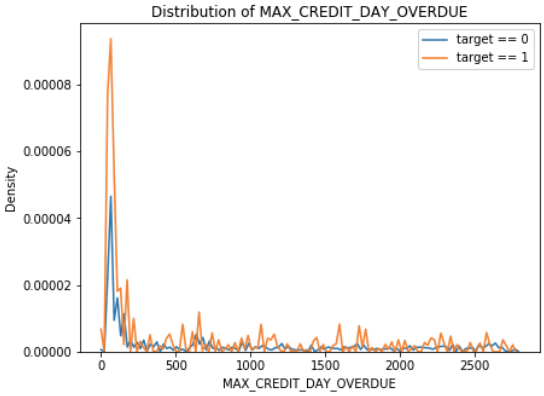


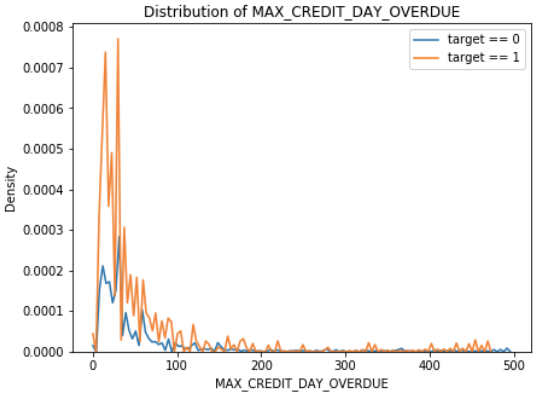

Distribution of FLOORSMAX_AVG
target == 0
target == 1
Density
FLOORSMAX_AVG

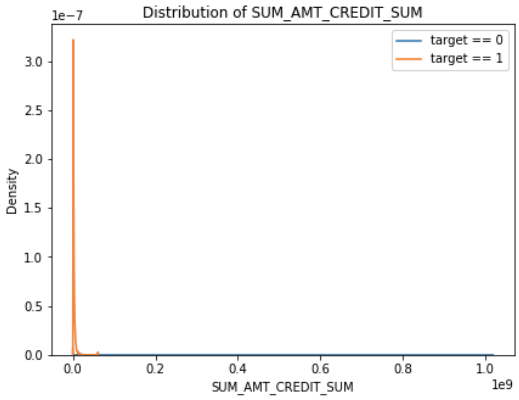

Distribution of FLOORSMAX_AVG
target == 0
target == 1
Density
FLOORSMAX_AVG

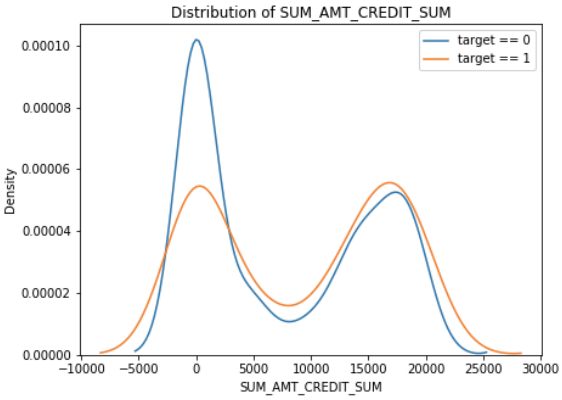

Distribution of SUM_AMT_CREDIT_MAX_OVERDUE
1e−8
target == 0
target == 1
Density
SUM_AMT_CREDIT_MAX_OVERDUE
1e8

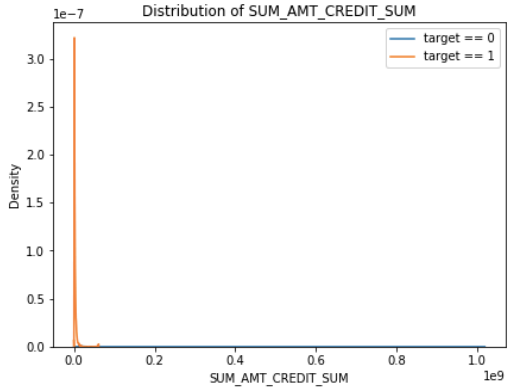

Distribution of SUM_AMT_CREDIT_MAX_OVERDUE
target == 0
target == 1
Density
SUM_AMT_CREDIT_MAX_OVERDUE

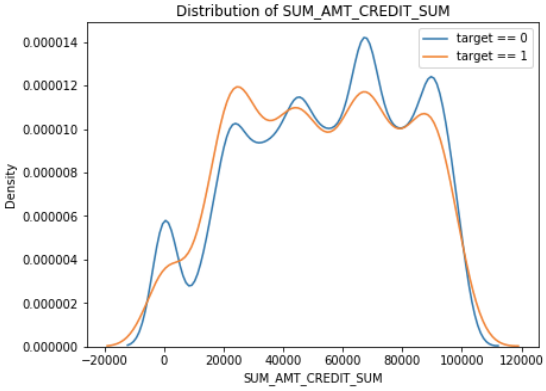

Distribution of MAX_CREDIT_DAY_OVERDUE
target == 0
target == 1
Density
MAX_CREDIT_DAY_OVERDUE

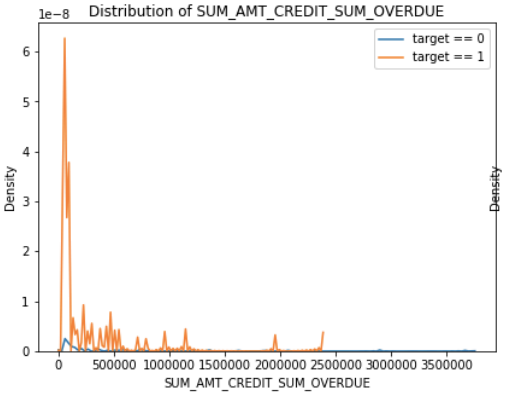

Distribution of MAX_CREDIT_DAY_OVERDUE
target == 0
target == 1
Density
MAX_CREDIT_DAY_OVERDUE

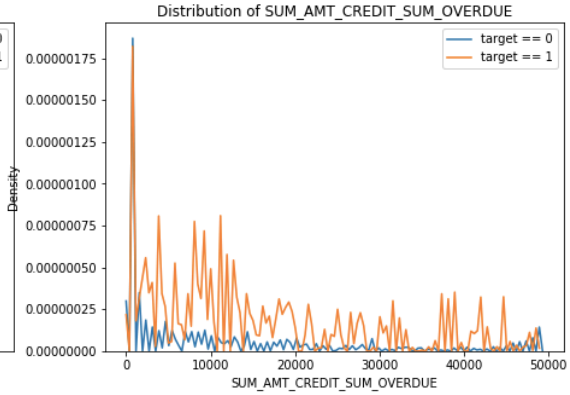

Distribution of SUM_AMT_CREDIT_SUM
1e−7
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM
1e9

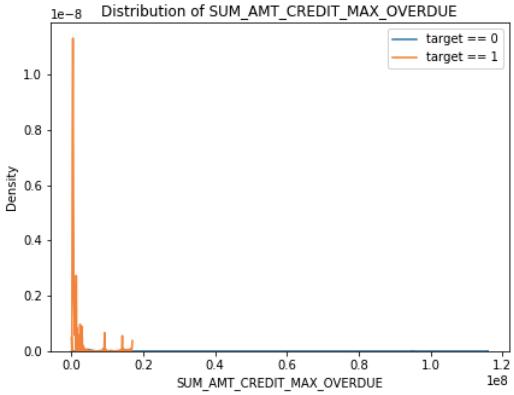

Distribution of SUM_AMT_CREDIT_SUM
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM

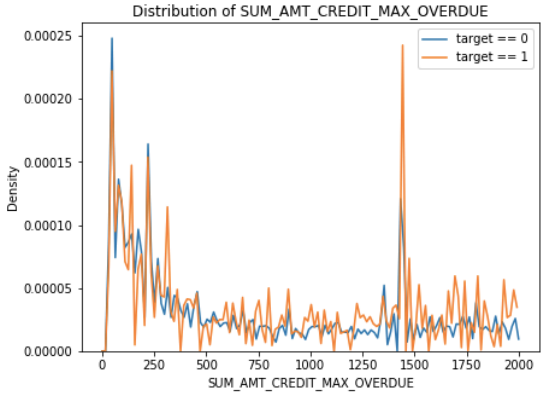

Distribution of SUM_AMT_CREDIT_SUM
1e−7
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM
1e9

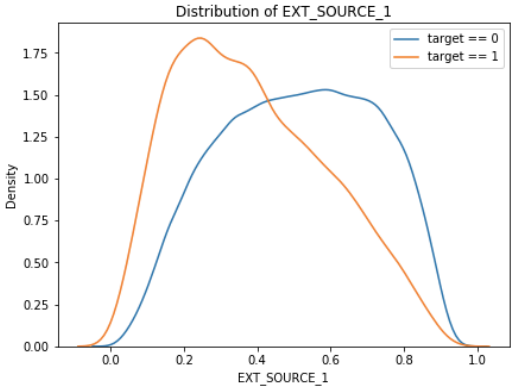

Distribution of SUM_AMT_CREDIT_SUM
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM

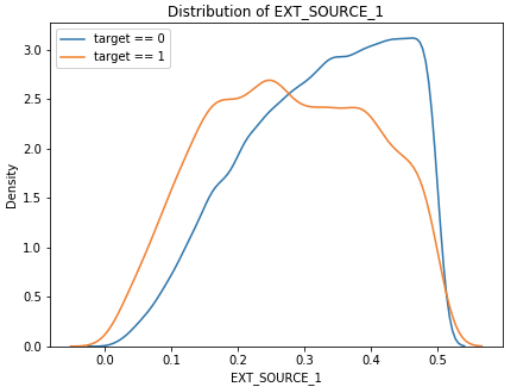

Distribution of SUM_AMT_CREDIT_SUM_OVERDUE
1e−8
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM_OVERDUE

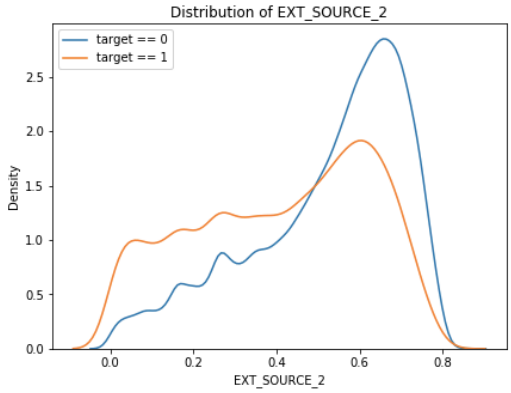

Distribution of SUM_AMT_CREDIT_SUM_OVERDUE
target == 0
target == 1
Density
SUM_AMT_CREDIT_SUM_OVERDUE

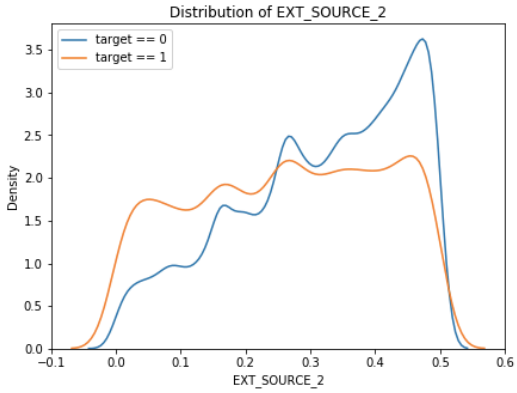

Distribution of SUM_AMT_CREDIT_MAX_OVERDUE
1e−8
target == 0
target == 1
Density
SUM_AMT_CREDIT_MAX_OVERDUE
1e8

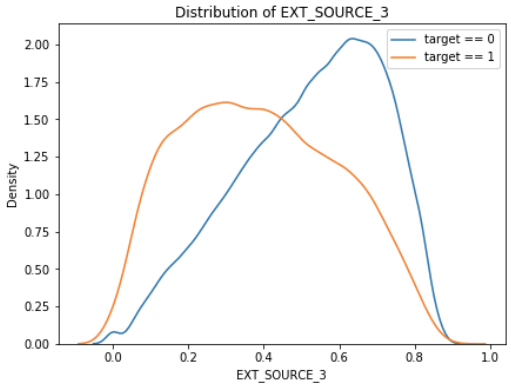

Distribution of SUM_AMT_CREDIT_MAX_OVERDUE
target == 0
target == 1
Density
SUM_AMT_CREDIT_MAX_OVERDUE

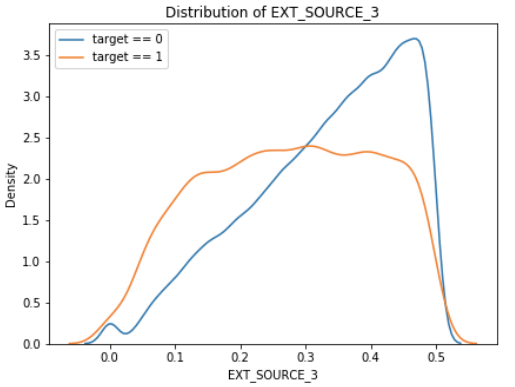

Distribution of EXT_SOURCE_1
target == 0
target == 1
Density
EXT_SOURCE_1

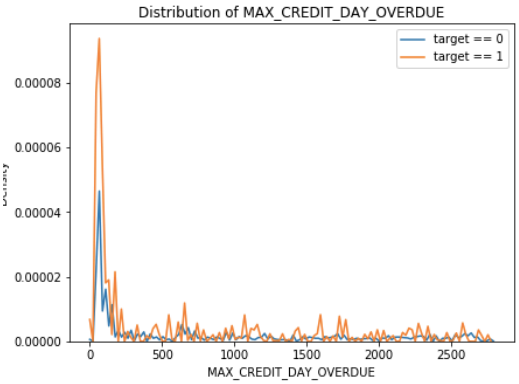

Distribution of EXT_SOURCE_1
target == 0
target == 1
Density
EXT_SOURCE_1

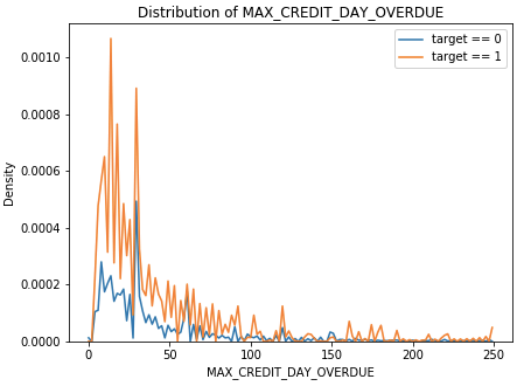

Distribution of EXT_SOURCE_2
target == 0
target == 1
Density
EXT_SOURCE_2

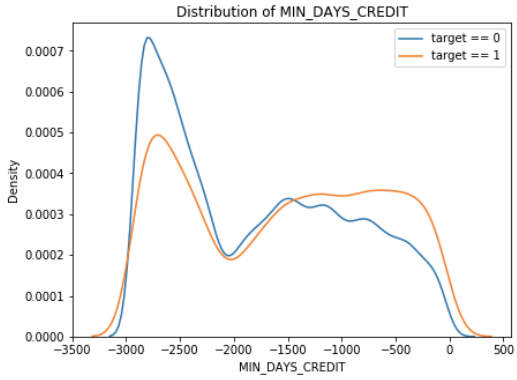

Distribution of EXT_SOURCE_2
target == 0
target == 1
Density
EXT_SOURCE_2

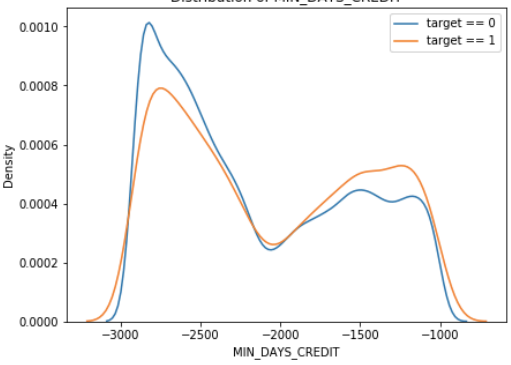

Distribution of EXT_SOURCE_3
target == 0
target == 1
Density
EXT_SOURCE_3


Distribution of EXT_SOURCE_3
target == 0
target == 1
Density
EXT_SOURCE_3

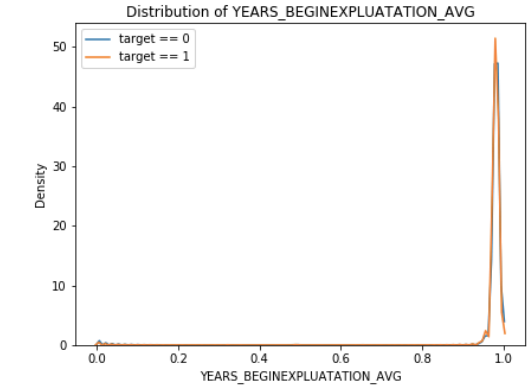

Distribution of MAX_CREDIT_DAY_OVERDUE
target == 0
target == 1
MAX_CREDIT_DAY_OVERDUE

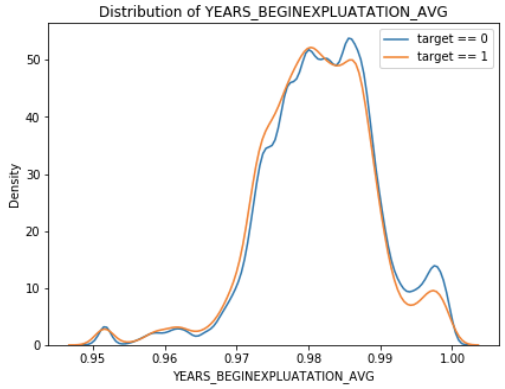

Distribution of MAX_CREDIT_DAY_OVERDUE
target == 0
target == 1
Density
MAX_CREDIT_DAY_OVERDUE

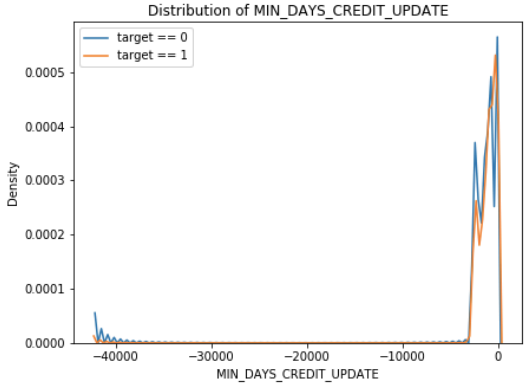

Distribution of MIN_DAYS_CREDIT
target == 0
target == 1
Density
MIN_DAYS_CREDIT

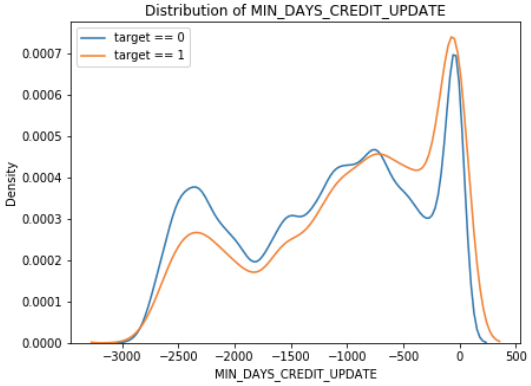

Distribution of MIN_DAYS_CREDIT
target == 0
target == 1
Density
MIN_DAYS_CREDIT

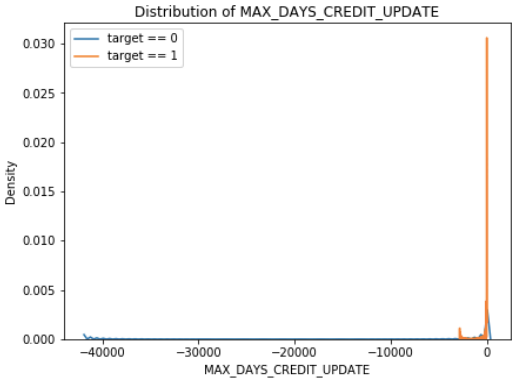
3) Below are KDE plots for some left skewed numeric variables. Left panel shows the original range of the variable, and the right panel shows the zoomed in view of the head in the distribution plot.

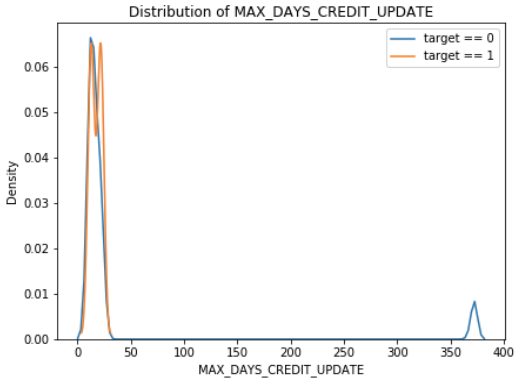


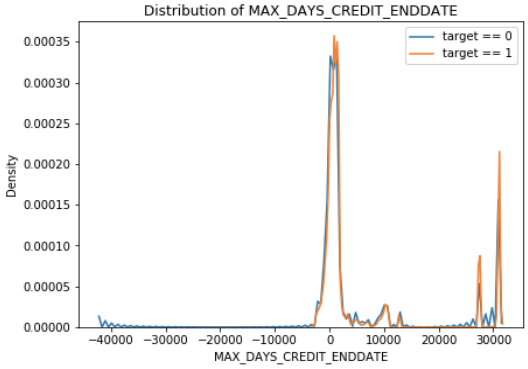


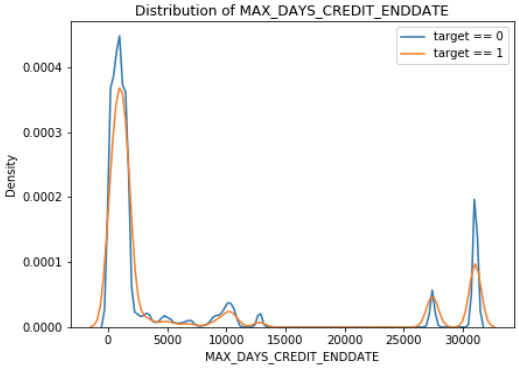




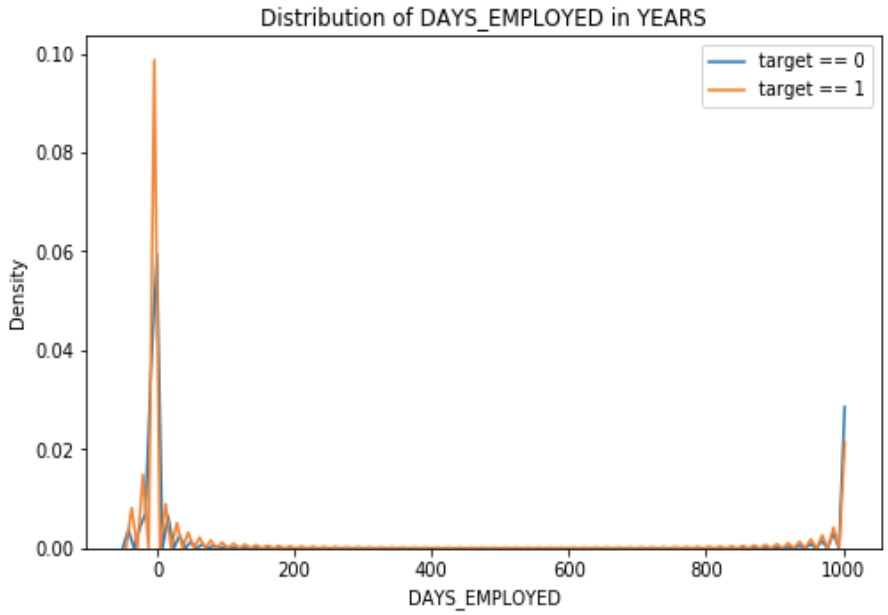


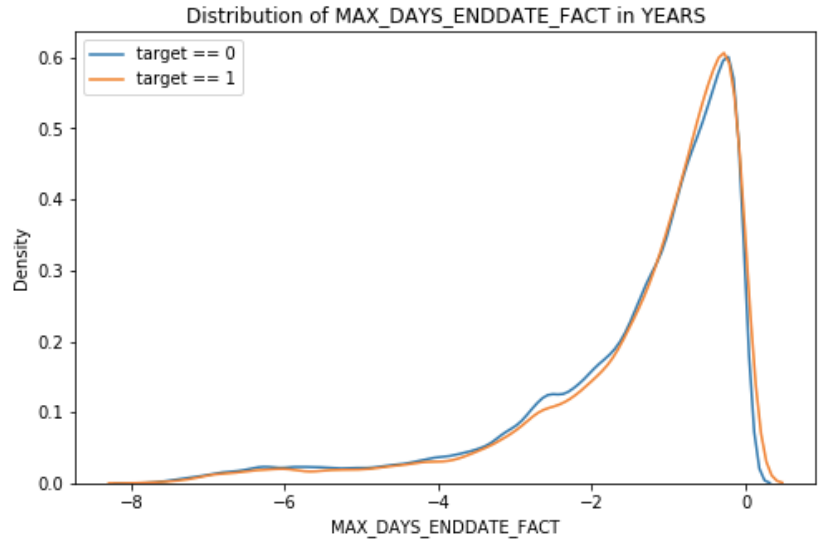


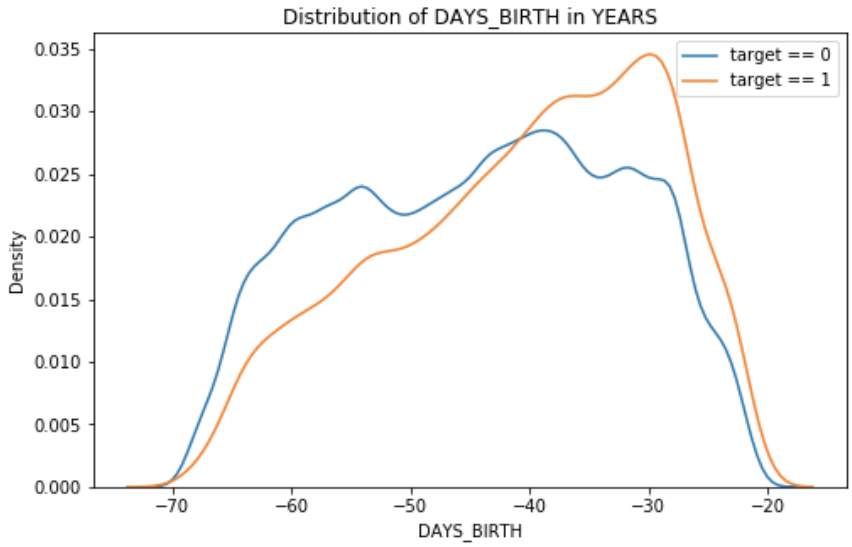


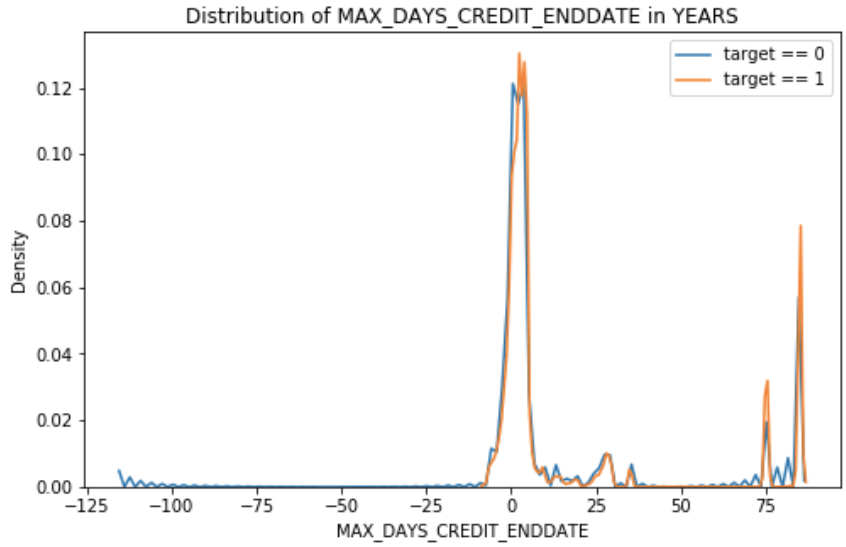


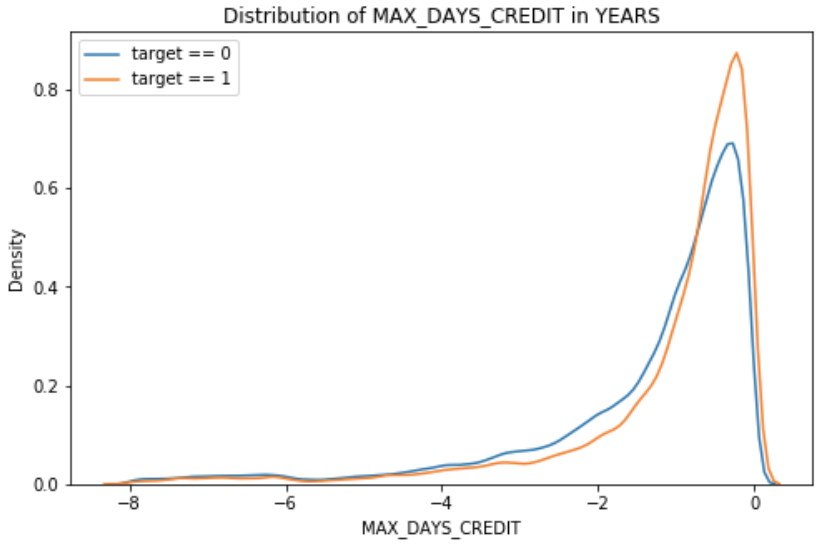
4) Variables below are originally represented in days in the dataset, convert them into years to better visualize the ranges. Here are some observations:
    a. "DAYS_EMPLOYED" has values over 1000 years, which are apparently errors. Based on other age type of variables, they are all counting backwards, meaning the maximum should be 0 (current application date), so cap the value at 0.
    b. '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.
    c. 'MAX_DAYS_CREDIT_UPDATE' is an aggregated field similar to (ii). It means over all the previous applications within the same current application ID, the maximum days before current application when the last information about the Credit Bureau credit come. This should be a negative number as it's counting backwards from the current application date. For all the positive numbers, define them as 0.
    d. 'MIN_DAYS_CREDIT_UPDATE' is treated the same way as MAX_DAYS_CREDIT_UPDATE

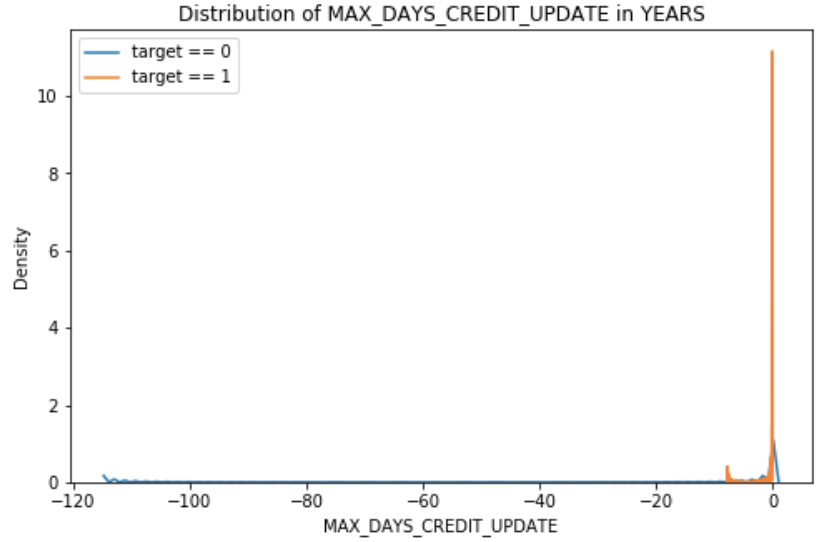


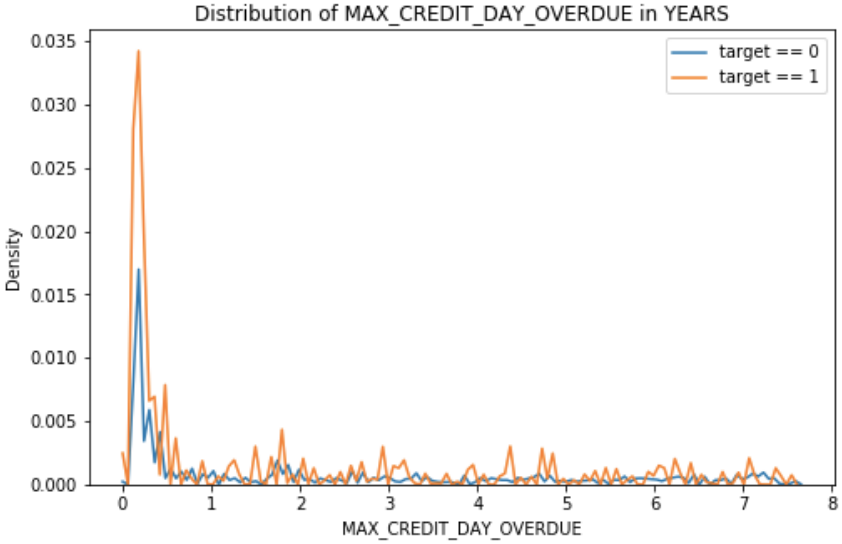

Distribution of MAX_DAYS_ENDDATE_FACT in YEARS
target == 0
target == 1
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
−8
−6
−4
−2
0
MAX_DAYS_ENDDATE_FACT

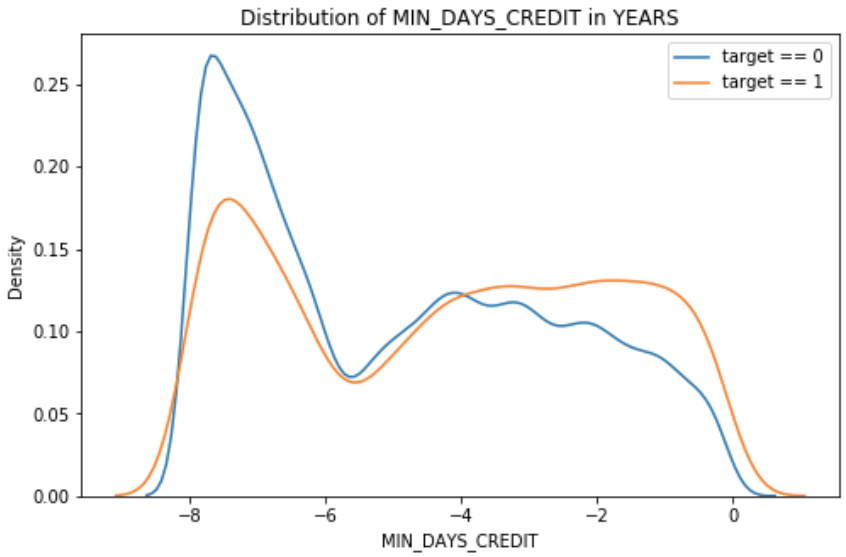

Distribution of DAYS_BIRTH in YEARS
target == 0
target == 1
Density
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
−70
−60
−50
−40
−30
−20
DAYS_BIRTH

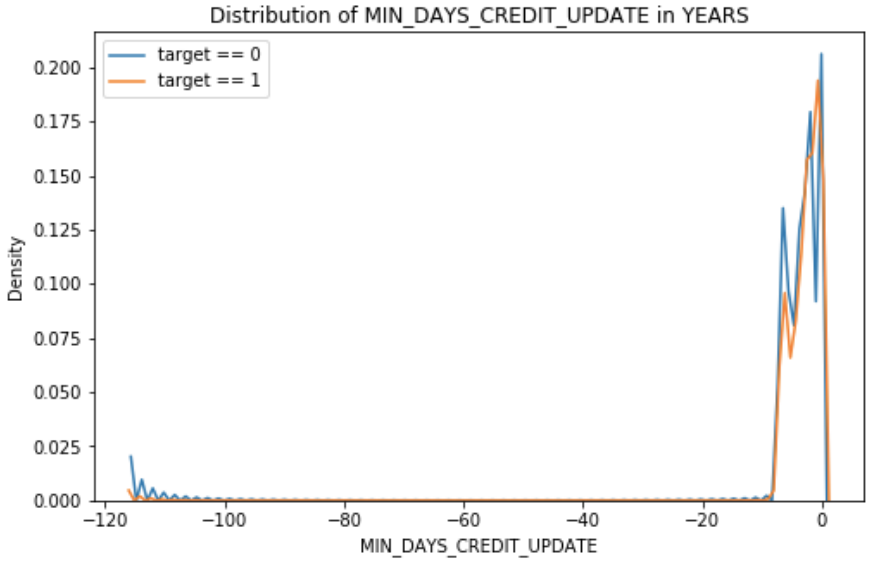

Distribution of MAX_DAYS_CREDIT_ENDDATE in YEARS
target == 0
target == 1
Density
0.00
0.02
0.04
0.06
0.08
0.10
0.12
−125
−100
−75
−50
−25
0
25
50
75
MAX_DAYS_CREDIT_ENDDATE

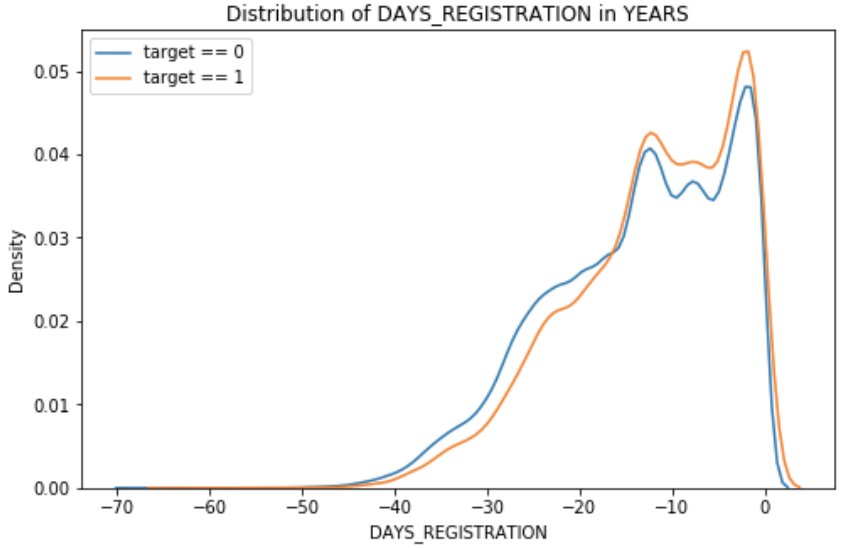

Distribution of MAX_DAYS_CREDIT in YEARS
target == 0
target == 1
Density
MAX_DAYS_CREDIT

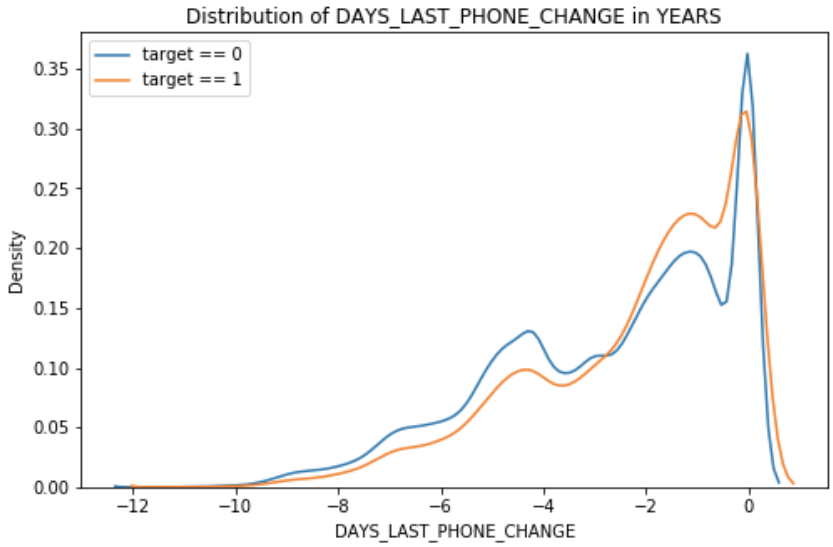

Distribution of MAX_DAYS_CREDIT_UPDATE in YEARS
target == 0
target == 1
Density
MAX_DAYS_CREDIT_UPDATE

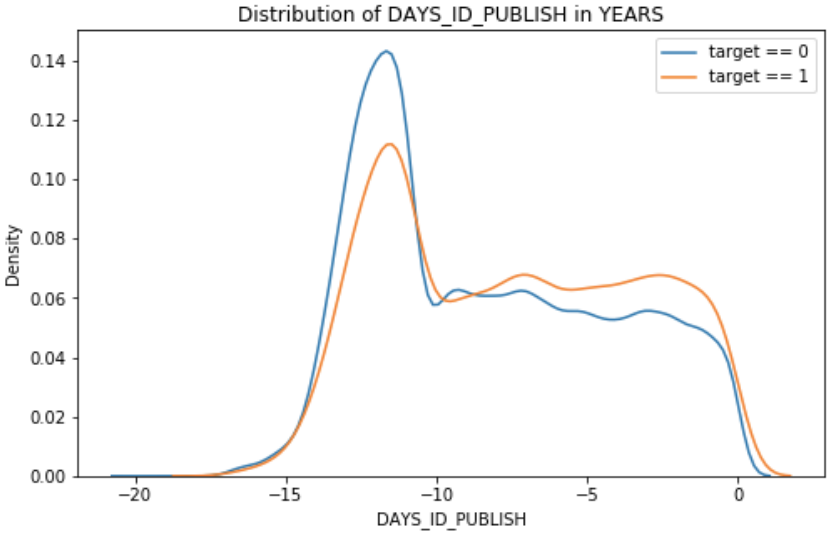

Distribution of MAX_CREDIT_DAY_OVERDUE in YEARS
target == 0
target == 1
Density
MAX_CREDIT_DAY_OVERDUE


Distribution of MIN_DAYS_CREDIT in YEARS
target == 0
target == 1
Density
0.25
0.20
0.15
0.10
0.05
0.00
−8
−6
−4
−2
0
MIN_DAYS_CREDIT


Distribution of MIN_DAYS_CREDIT_UPDATE in YEARS
target == 0
target == 1
Density
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−120
−100
−80
−60
−40
−20
0
MIN_DAYS_CREDIT_UPDATE

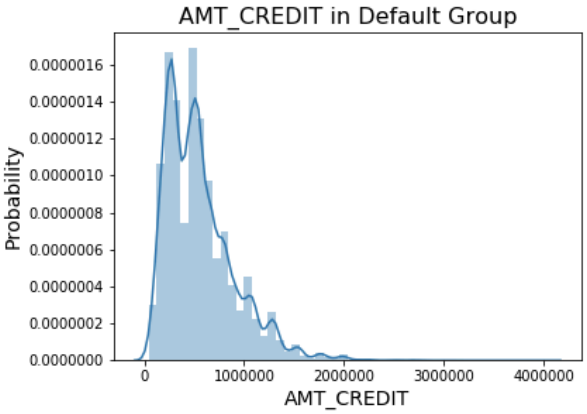

Distribution of DAYS_REGISTRATION in YEARS
target == 0
target == 1
Density
0.05
0.04
0.03
0.02
0.01
0.00
−70
−60
−50
−40
−30
−20
−10
0
DAYS_REGISTRATION

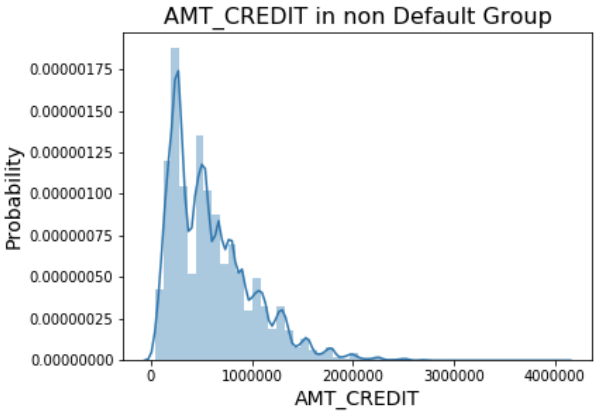


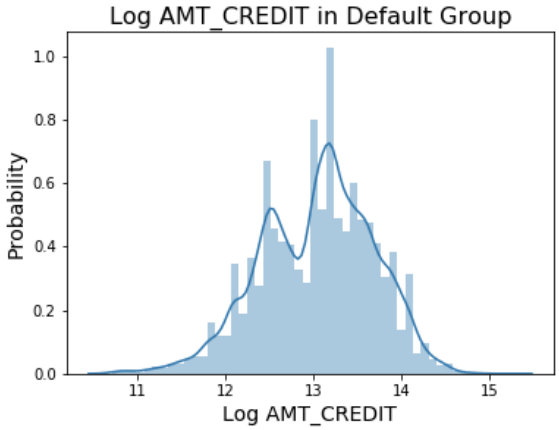


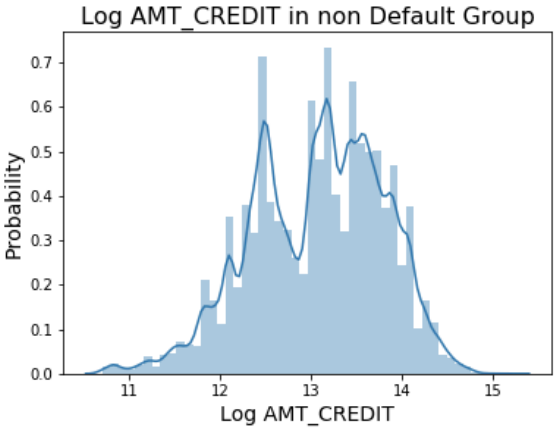
5) Distributions of 'AMT_ANNUITY', 'AMT_CREDIT' and 'AMT_GOODS_PRICE' are skewed to the right, we recommend to take the log transformation to normalize the data.

Amount / Log AMT_CREDIT

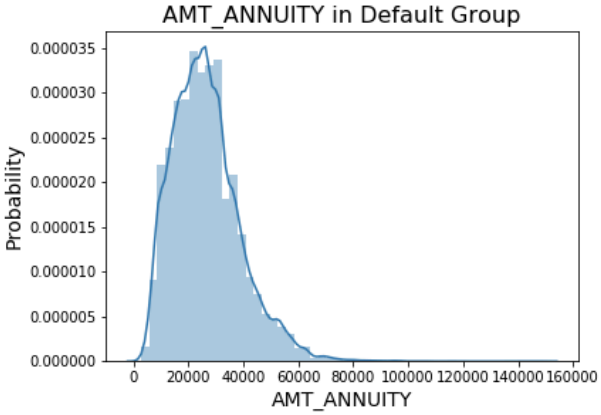


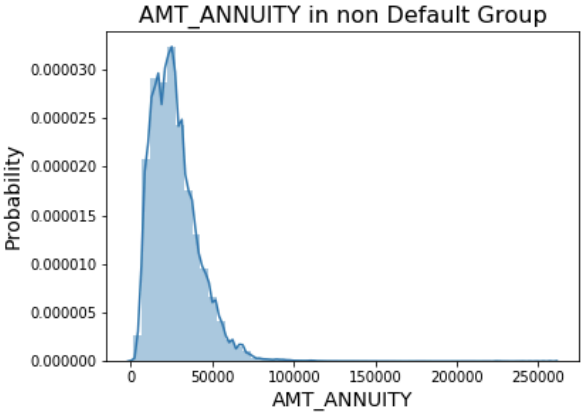


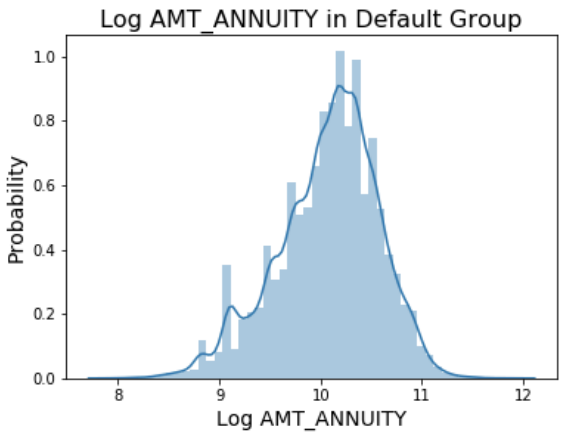


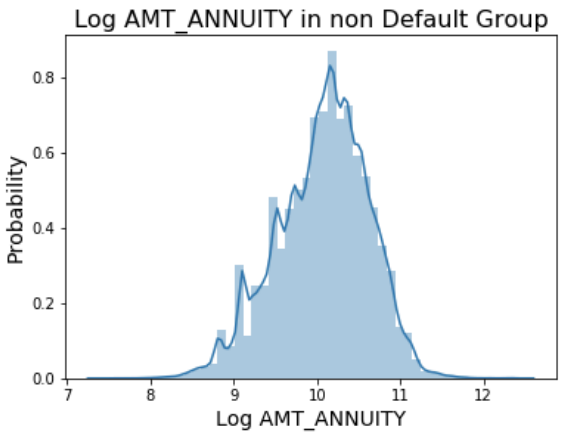


Amount / Log AMT_ANNUITY

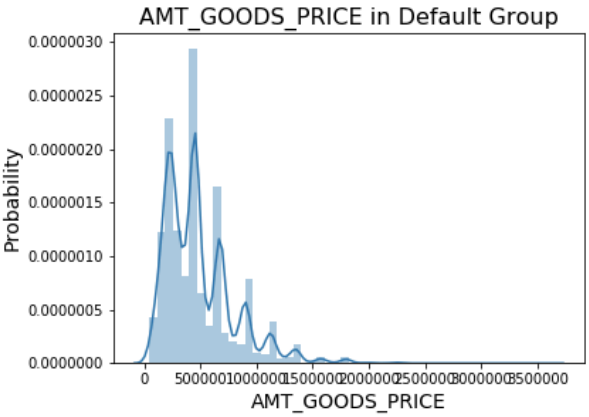


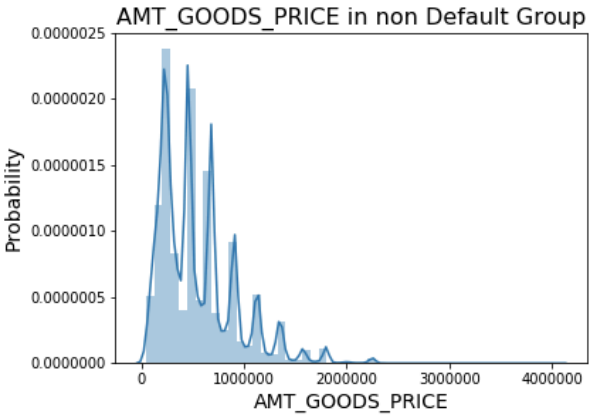


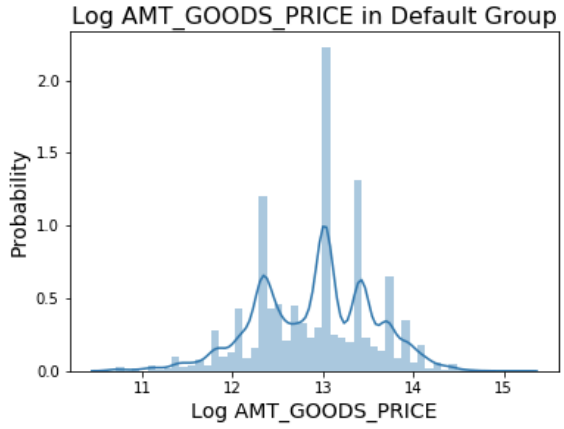


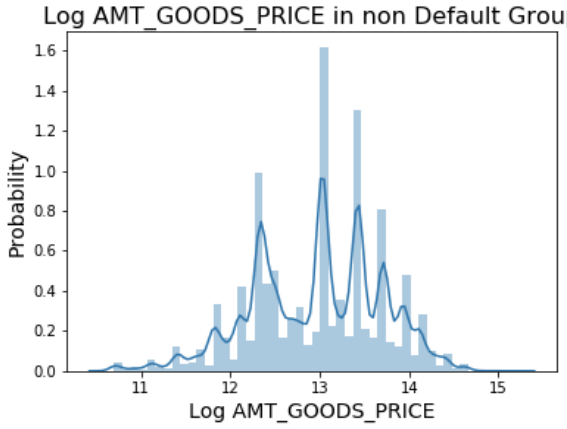


Amount / Log AMT_GOODS_PRICE

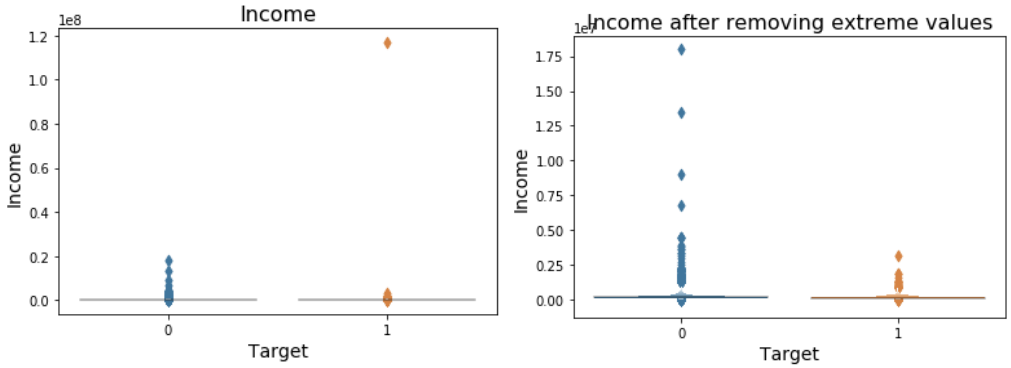


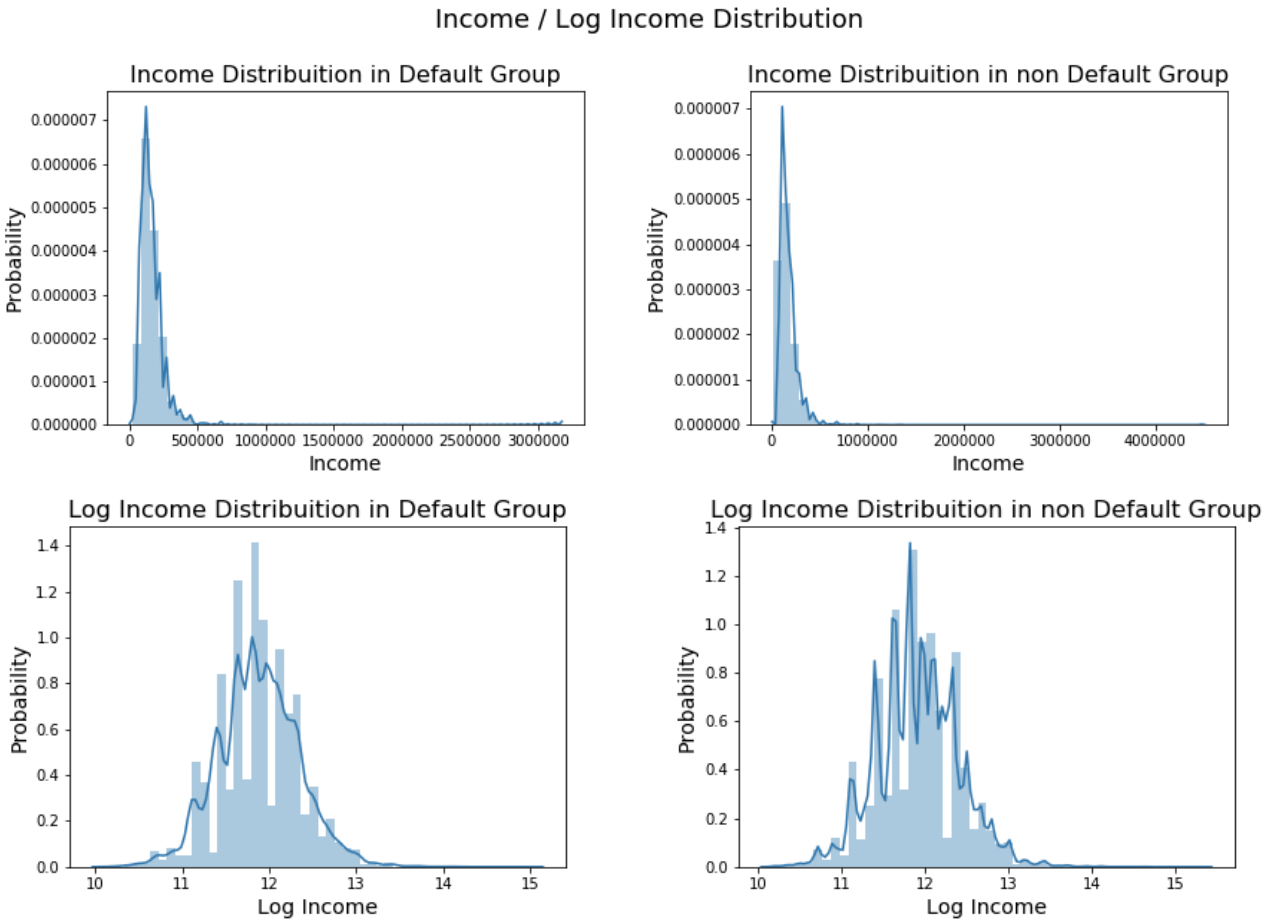




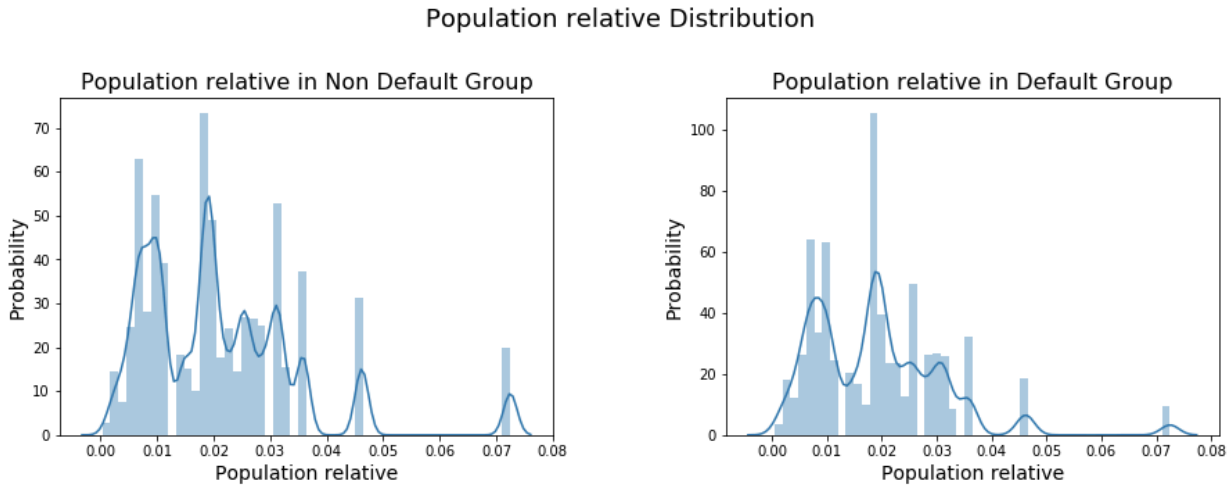


6) There is 1 extremely large income value in the default group ($120 million), and 4 very large values (> $7.5 million) in the non-default group. Recommend to remove the outliers and then do log transformation to normalize the income data.

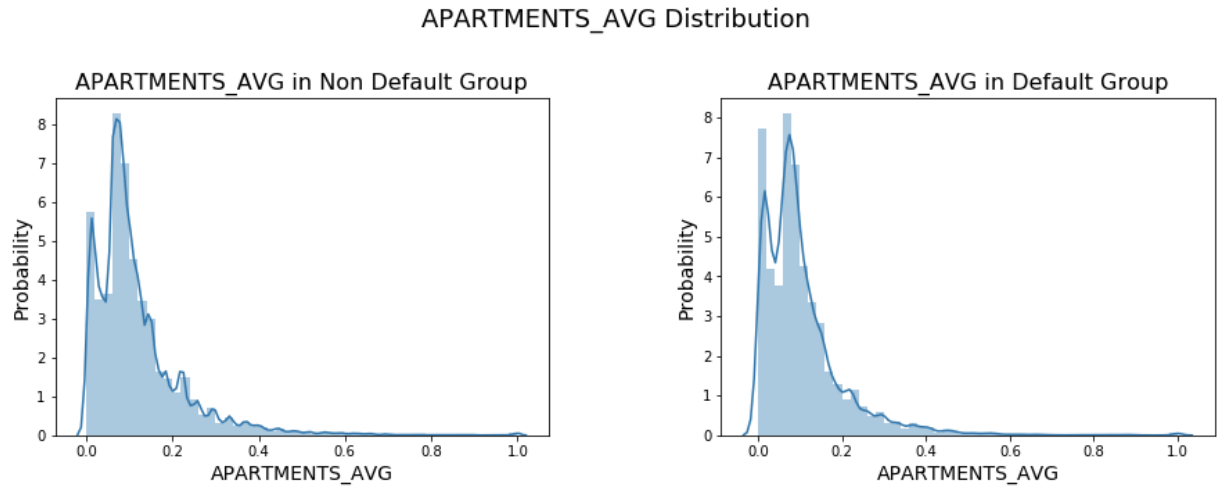




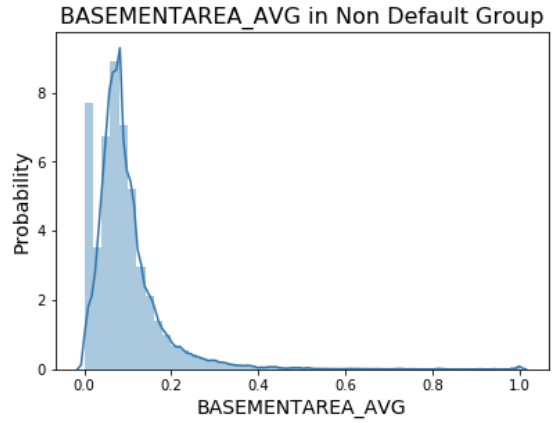
7) 'REGION_POPULATION_RELATIVE' variable has multiple modes, most of the clients live in lower population density places. Visually the distribution for the default and non-default groups are similar.

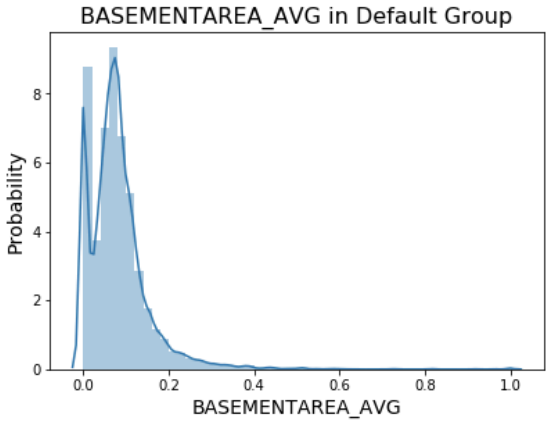


8) The distributions for _AVG, _MODE and _MEDI numeric variables are all kind of skewed, but since they are already normalized to range 0 to 1, no additional transformation is done to these variables. For missing values in these variables, can impute using the median. Here we are only showing the KDE plot for a few variables.

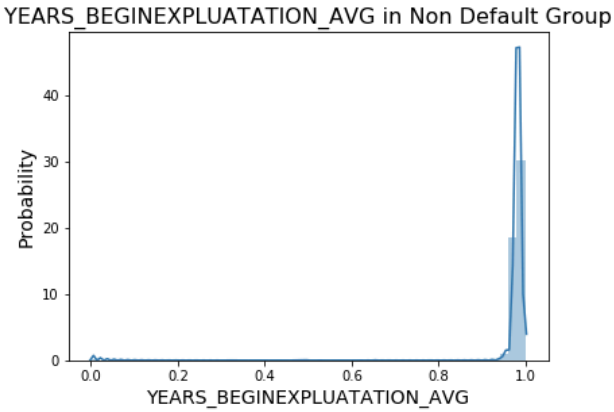


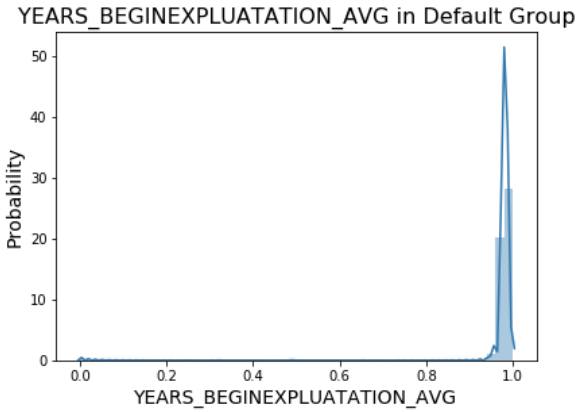
BASEMENTAREA_AVG Distribution



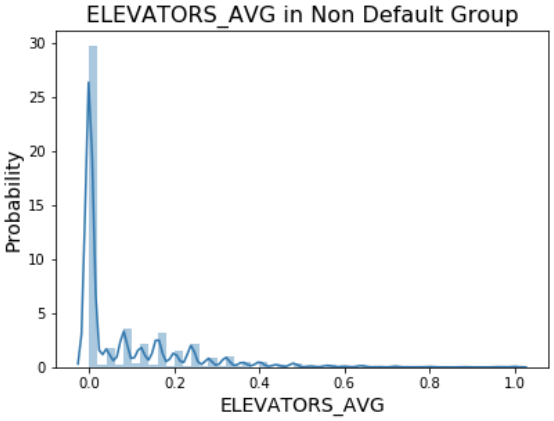


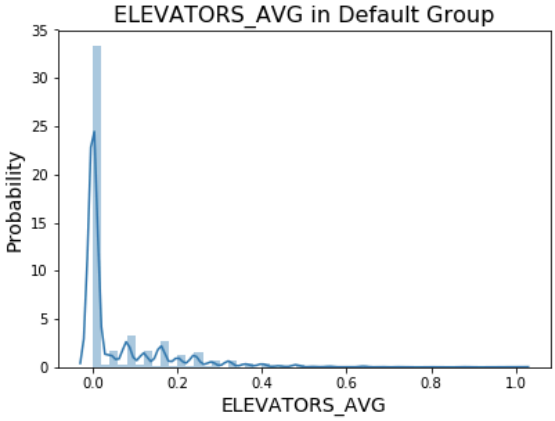
YEARS_BEGINEXPLUATATION_AVG Distribution



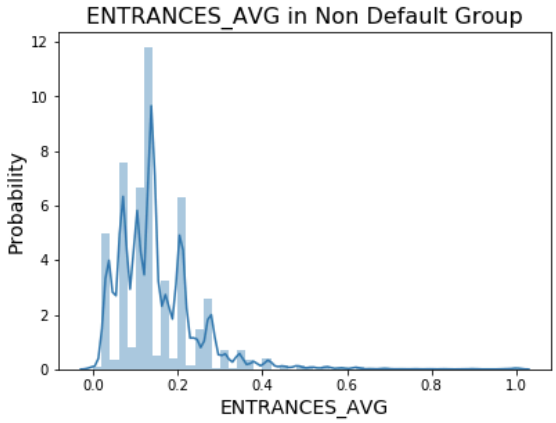


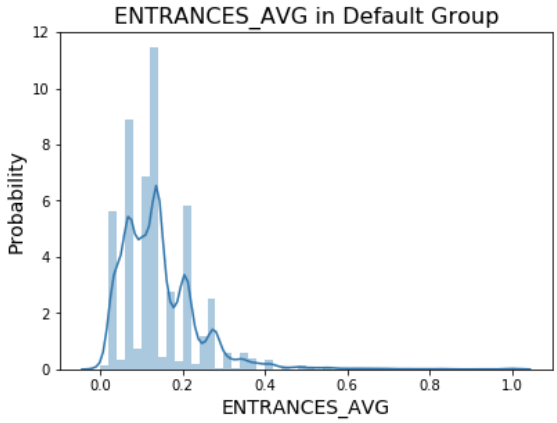
ELEVATORS_AVG Distribution



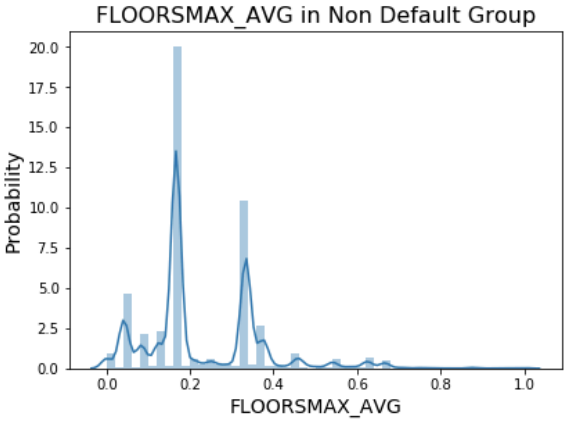


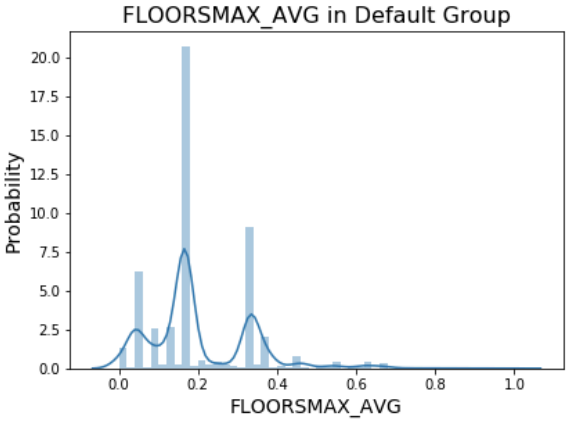
ENTRANCES_AVG Distribution



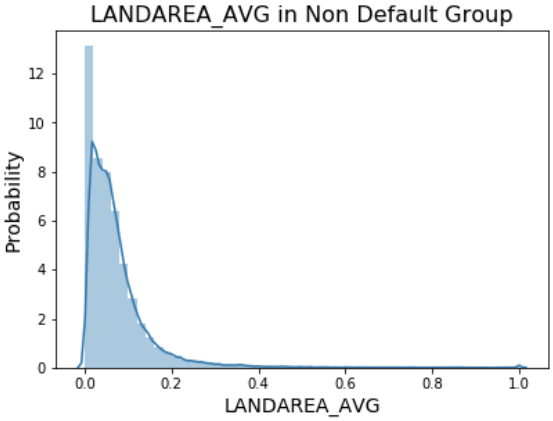


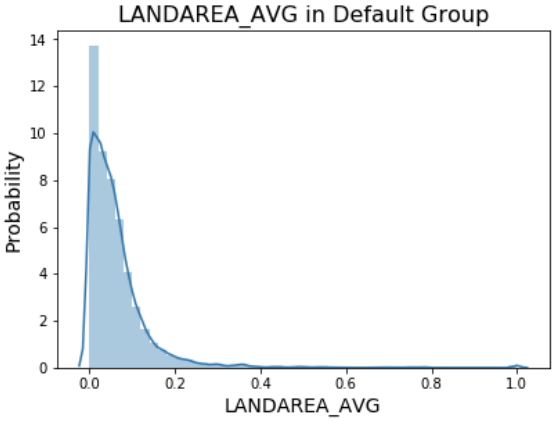
FLOORSMAX_AVG Distribution



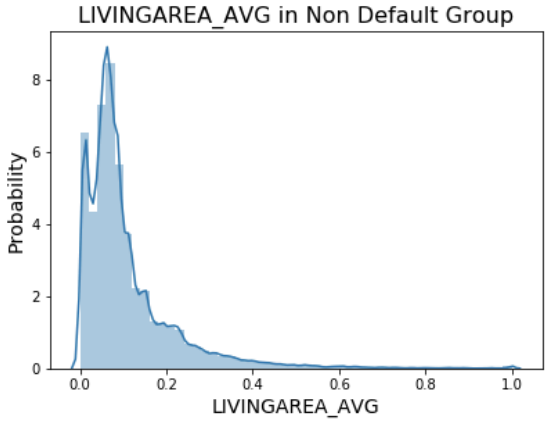


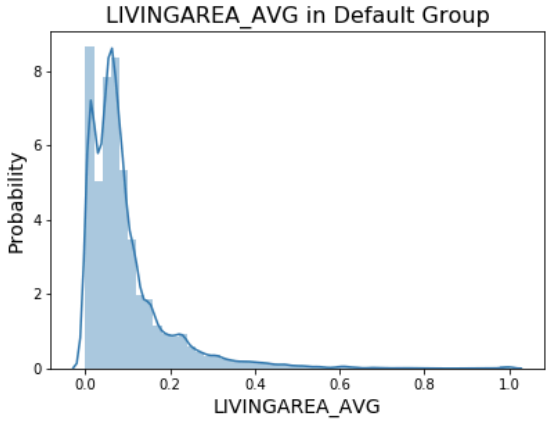
LANDAREA_AVG Distribution



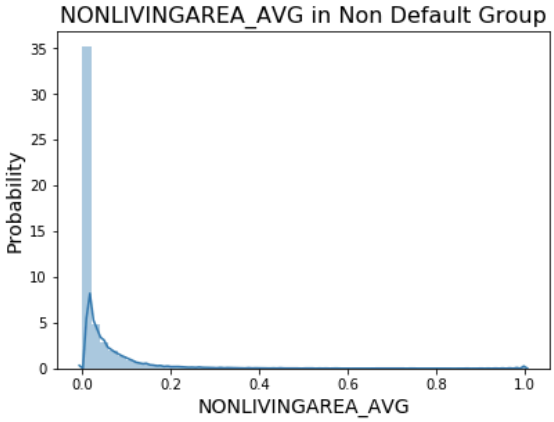


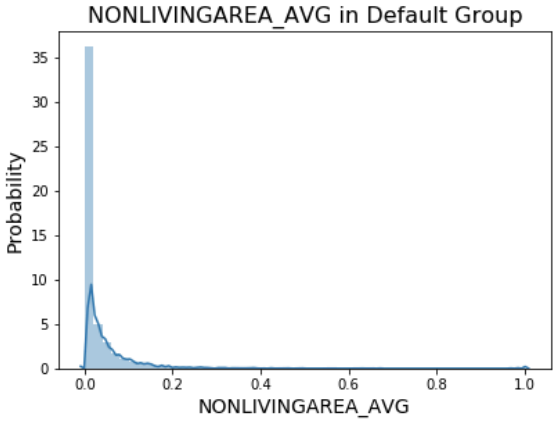
LIVINGAREA_AVG Distribution



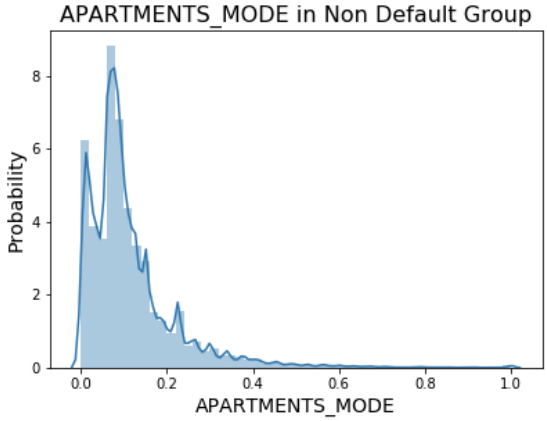


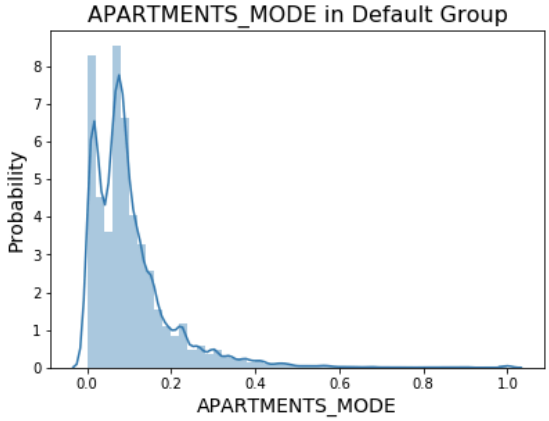
NONLIVINGAREA_AVG Distribution



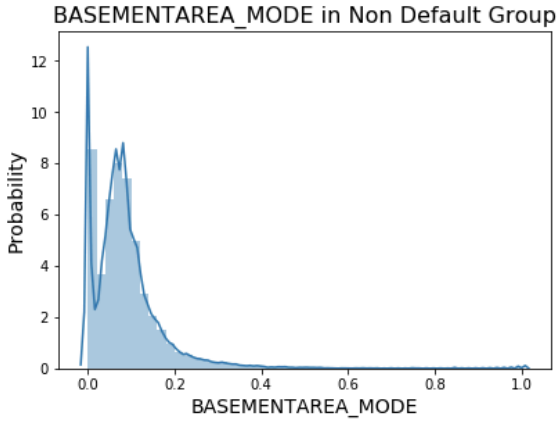


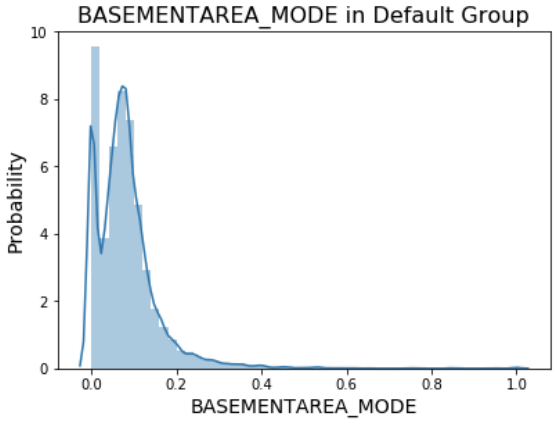
APARTMENTS_MODE Distribution



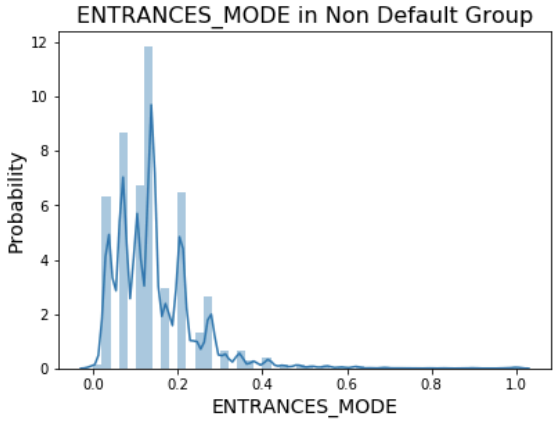


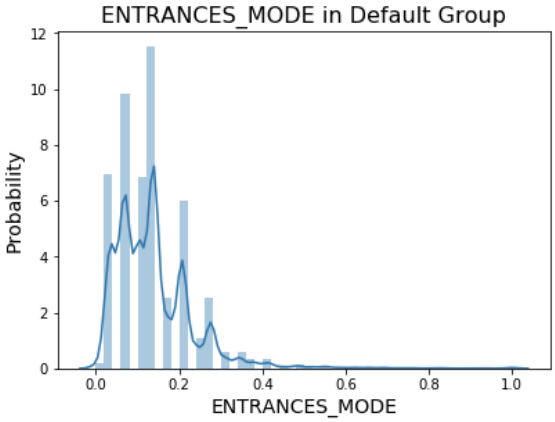
BASEMENTAREA_MODE Distribution



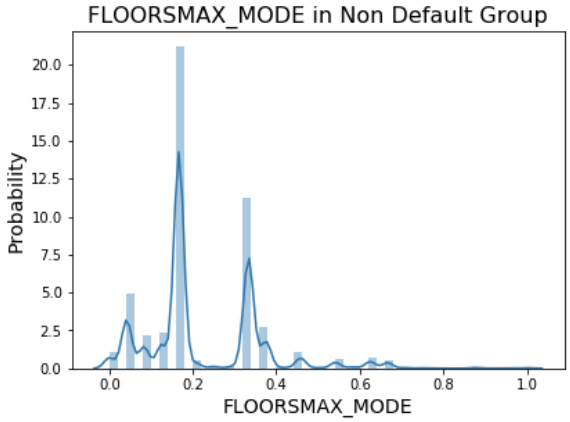


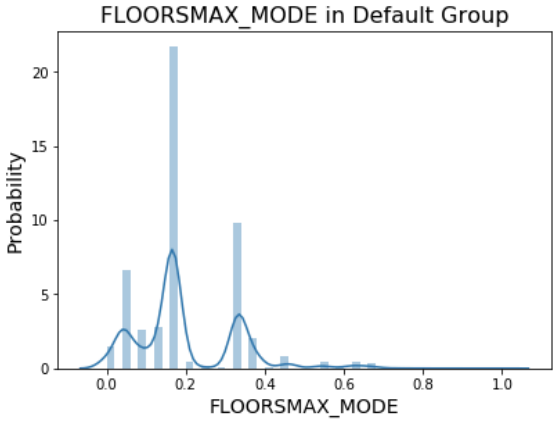
ENTRANCES_MODE Distribution



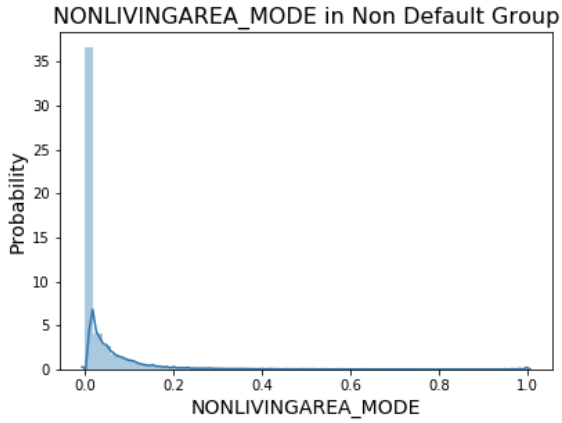


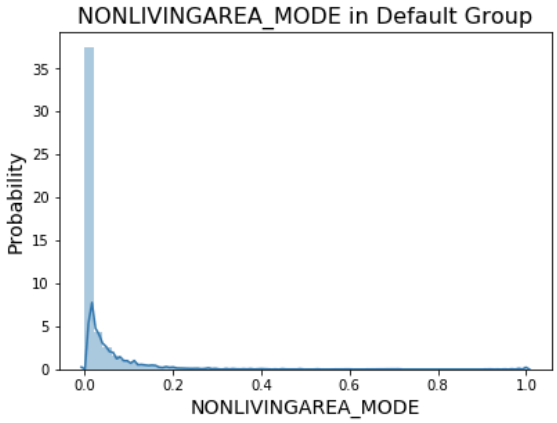
FLOORSMAX_MODE Distribution



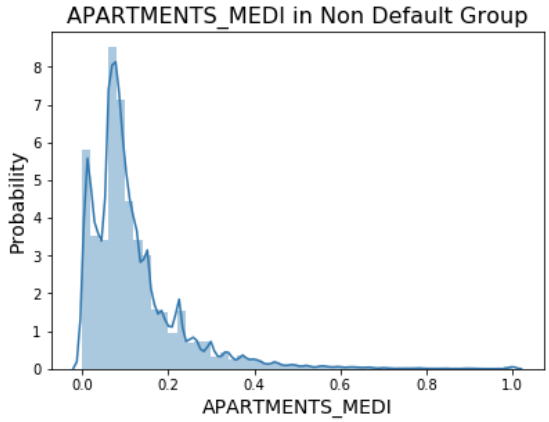


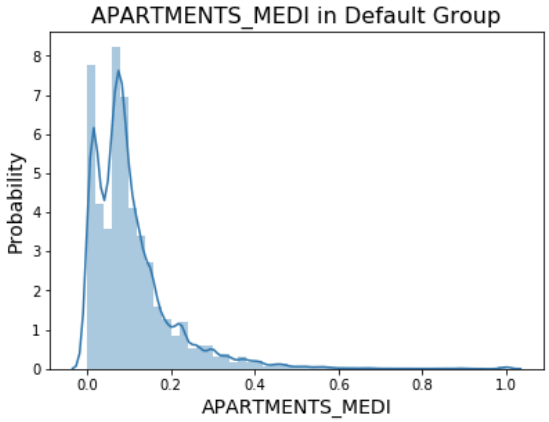
NONLIVINGAREA_MODE Distribution



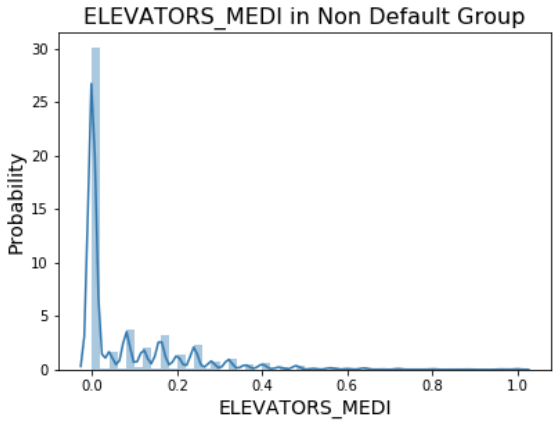


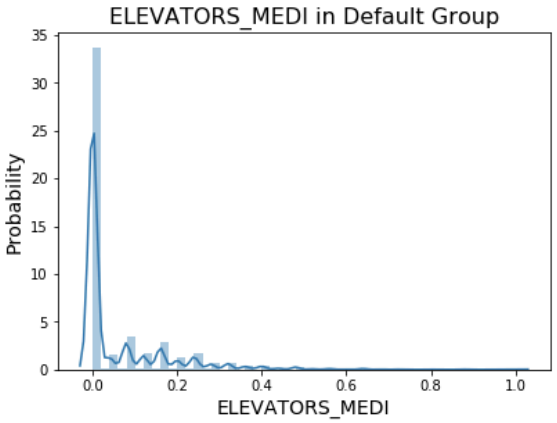
APARTMENTS_MEDI Distribution

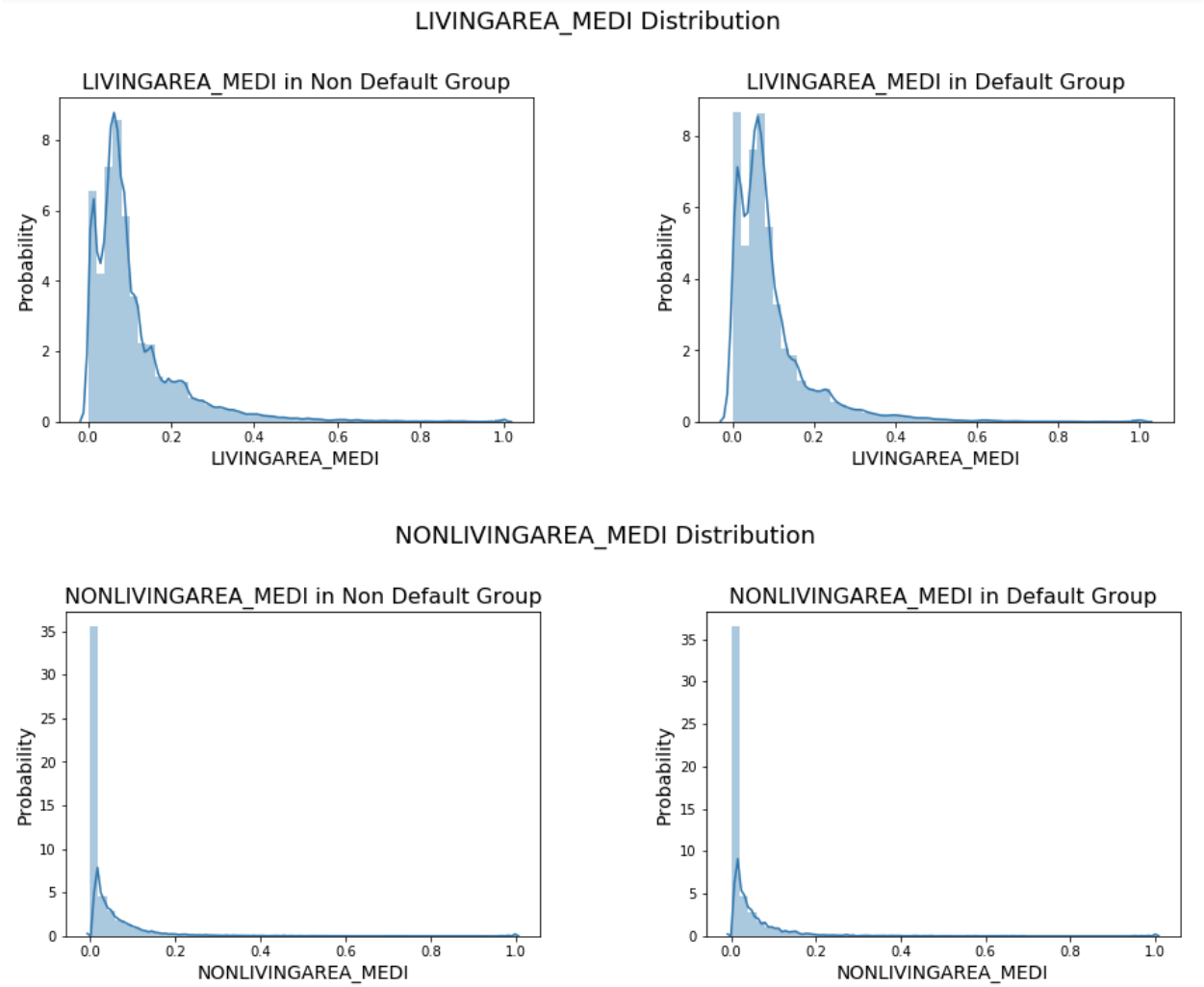




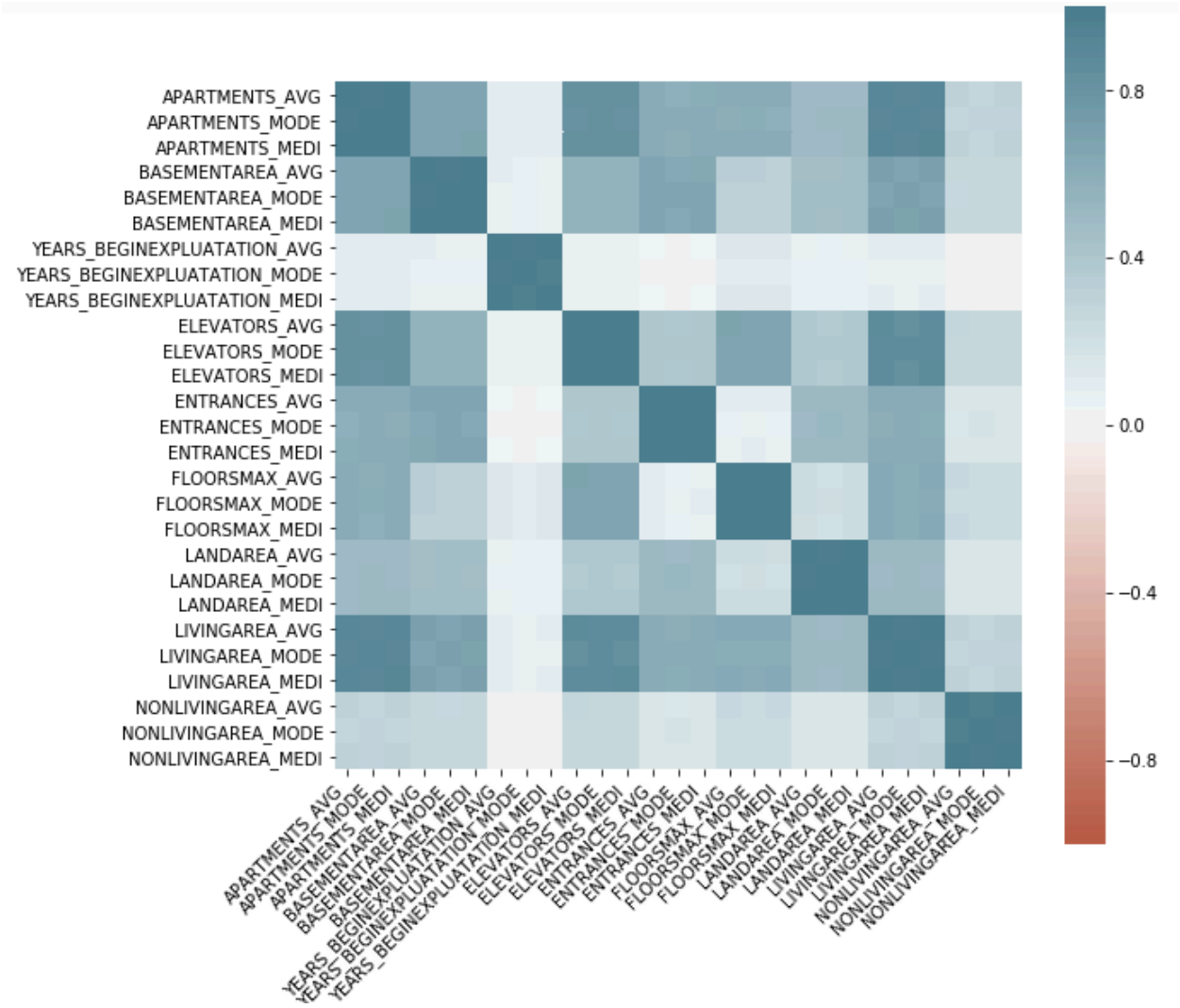
ELEVATORS_MEDI Distribution







9) Correlations between the _AVG, _MODE and _MEDI variables. Some groups of variables are highly correlated with some groups, for example, 'APARTMENTS' variables are highly correlated with 'LIVINGAREA' variables. At this point we don't remove any one of them**.**



10) Data cleaning and transformation, high correlation column removal and outlier removal, missing value imputations.

   a. Remove outliers with extremely large income accounts. Remove gender = XNA rows.
   b. For categorical features: (i) When cardinality is high, combine some low frequency categories. (ii) When missing values (‘nan’) are present, code it as a new category. (iii) Use one hot encoding to convert all categorical variables into numeric dummy variables and drop the original categorical variables.
   c. For numeric features: (i) Drop variables with high or perfect linear correlations with others. (ii) Do log transformation to normalize some “AMT” variables. (iii) Floor or cap some “DAYS” variables based on exploratory column analysis. (iv) Impute all missing values using the median of the non-missing part.

Appendix C. Metadata for Variables

	Table	Row	Description	Special
1	application_{train|test}.csv	SK_ID_CURR	ID of loan in our sample	
2	application_{train|test}.csv	TARGET	Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases)	
5	application_{train|test}.csv	NAME_CONTRACT_TYPE	Identification if loan is cash or revolving	
6	application_{train|test}.csv	CODE_GENDER	Gender of the client	
7	application_{train|test}.csv	FLAG_OWN_CAR	Flag if the client owns a car	
8	application_{train|test}.csv	FLAG_OWN_REALTY	Flag if client owns a house or flat	
9	application_{train|test}.csv	CNT_CHILDREN	Number of children the client has	
10	application_{train|test}.csv	AMT_INCOME_TOTAL	Income of the client	
11	application_{train|test}.csv	AMT_CREDIT	Credit amount of the loan	
12	application_{train|test}.csv	AMT_ANNUITY	Loan annuity	
13	application_{train|test}.csv	AMT_GOODS_PRICE	For consumer loans it is the price of the goods for which the loan is given	
14	application_{train|test}.csv	NAME_TYPE_SUITE	Who was accompanying client when he was applying for the loan	
15	application_{train|test}.csv	NAME_INCOME_TYPE	Clients income type (businessman, working, maternity leave,…)	
16	application_{train|test}.csv	NAME_EDUCATION_TYPE	Level of highest education the client achieved	
17	application_{train|test}.csv	NAME_FAMILY_STATUS	Family status of the client	
18	application_{train|test}.csv	NAME_HOUSING_TYPE	What is the housing situation of the client (renting, living with parents, ...)	
19	application_{train|test}.csv	REGION_POPULATION_RELATIVE	Normalized population of region where client lives (higher number means the client lives in more populated region)	normalized
20	application_{train|test}.csv	DAYS_BIRTH	Client's age in days at the time of application	time only relative to the application
21	application_{train|test}.csv	DAYS_EMPLOYED	How many days before the application the person started current employment	time only relative to the application
22	application_{train|test}.csv	DAYS_REGISTRATION	How many days before the application did client change his registration	time only relative to the application
23	application_{train|test}.csv	DAYS_ID_PUBLISH	How many days before the application did client change the identity document with which he applied for the loan	time only relative to the application
24	application_{train|test}.csv	OWN_CAR_AGE	Age of client's car	
25	application_{train|test}.csv	FLAG_MOBIL	Did client provide mobile phone (1=YES, 0=NO)	
26	application_{train|test}.csv	FLAG_EMP_PHONE	Did client provide work phone (1=YES, 0=NO)	
27	application_{train|test}.csv	FLAG_WORK_PHONE	Did client provide home phone (1=YES, 0=NO)	
28	application_{train|test}.csv	FLAG_CONT_MOBILE	Was mobile phone reachable (1=YES, 0=NO)	

29	application_{train|test}.csv	FLAG_PHONE	Did client provide home phone (1=YES, 0=NO)	
30	application_{train|test}.csv	FLAG_MOBIL	Did client provide email (1=YES, 0=NO)	
31	application_{train|test}.csv	OCCUPATION_TYPE	What kind of occupation does the client have	
32	application_{train|test}.csv	CNT_FAM_MEMBERS	How many family members does client have	
33	application_{train|test}.csv	REGION_RATING_CLIENT	Our rating of the region where client lives (1,2,3)	
34	application_{train|test}.csv	REGION_RATING_CLIENT_W_CITY	Our rating of the region where client lives with taking city into account (1,2,3)	
35	application_{train|test}.csv	WEEKDAY_APPR_PROCESS_START	On which day of the week did the client apply for the loan	
36	application_{train|test}.csv	HOUR_APPR_PROCESS_START	Approximately at what hour did the client apply for the loan	rounded
37	application_{train|test}.csv	REG_REGION_NOT_LIVE_REGION	Flag if client's permanent address does not match contact address (1=different, 0=same, at region level)	
38	application_{train|test}.csv	REG_REGION_NOT_WORK_REGION	Flag if client's permanent address does not match work address (1=different, 0=same, at region level)	
39	application_{train|test}.csv	LIVE_REGION_NOT_WORK_REGION	Flag if client's contact address does not match work address (1=different, 0=same, at region level)	
40	application_{train|test}.csv	REG_CITY_NOT_LIVE_CITY	Flag if client's permanent address does not match contact address (1=different, 0=same, at city level)	
41	application_{train|test}.csv	REG_CITY_NOT_WORK_CITY	Flag if client's permanent address does not match work address (1=different, 0=same, at city level)	
42	application_{train|test}.csv	LIVE_CITY_NOT_WORK_CITY	Flag if client's contact address does not match work address (1=different, 0=same, at city level)	
43	application_{train|test}.csv	ORGANIZATION_TYPE	Type of organization where client works	
44	application_{train|test}.csv	EXT_SOURCE_1	Normalized score from external data source	normalized
45	application_{train|test}.csv	EXT_SOURCE_2	Normalized score from external data source	normalized
46	application_{train|test}.csv	EXT_SOURCE_3	Normalized score from external data source	normalized
47	application_{train|test}.csv	APARTMENTS_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
48	application_{train|test}.csv	BASEMENTAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
49	application_{train|test}.csv	YEARS_BEGINEXPLUATATION_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized

50	application_{train|test}.csv	YEARS_BUILD_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
51	application_{train|test}.csv	COMMONAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
52	application_{train|test}.csv	ELEVATORS_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
53	application_{train|test}.csv	ENTRANCES_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
54	application_{train|test}.csv	FLOORSMAX_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
55	application_{train|test}.csv	FLOORSMIN_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
56	application_{train|test}.csv	LANDAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
57	application_{train|test}.csv	LIVINGAPARTMENTS_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
58	application_{train|test}.csv	LIVINGAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized

59	application_{train|test}.csv	NONLIVINGAPARTMENTS_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
60	application_{train|test}.csv	NONLIVINGAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
61	application_{train|test}.csv	APARTMENTS_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
62	application_{train|test}.csv	BASEMENTAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
63	application_{train|test}.csv	YEARS_BEGINEXPLUATATION_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
64	application_{train|test}.csv	YEARS_BUILD_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
65	application_{train|test}.csv	COMMONAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
66	application_{train|test}.csv	ELEVATORS_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
67	application_{train|test}.csv	ENTRANCES_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized

68	application_{train|test}.csv	FLOORSMAX_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
69	application_{train|test}.csv	FLOORSMIN_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
70	application_{train|test}.csv	LANDAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
71	application_{train|test}.csv	LIVINGAPARTMENTS_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
72	application_{train|test}.csv	LIVINGAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
73	application_{train|test}.csv	NONLIVINGAPARTMENTS_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
74	application_{train|test}.csv	NONLIVINGAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
75	application_{train|test}.csv	APARTMENTS_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
76	application_{train|test}.csv	BASEMENTAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized

77	application_{train|test}.csv	YEARS_BEGINEXPLUATATION_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
78	application_{train|test}.csv	YEARS_BUILD_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
79	application_{train|test}.csv	COMMONAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
80	application_{train|test}.csv	ELEVATORS_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
81	application_{train|test}.csv	ENTRANCES_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
82	application_{train|test}.csv	FLOORSMAX_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
83	application_{train|test}.csv	FLOORSMIN_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
84	application_{train|test}.csv	LANDAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
85	application_{train|test}.csv	LIVINGAPARTMENTS_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized

86	application_{train|test}.csv	LIVINGAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
87	application_{train|test}.csv	NONLIVINGAPARTMENTS_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
88	application_{train|test}.csv	NONLIVINGAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
89	application_{train|test}.csv	FONDKAPREMONT_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
90	application_{train|test}.csv	HOUSETYPE_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
91	application_{train|test}.csv	TOTALAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
92	application_{train|test}.csv	WALLSMATERIAL_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
93	application_{train|test}.csv	EMERGENCYSTATE_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor	normalized
94	application_{train|test}.csv	OBS_30_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings with observable 30 DPD (days past due) default	
95	application_{train|test}.csv	DEF_30_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings defaulted on 30 DPD (days past due)	

96	application_{train|test}.csv	OBS_60_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings with observable 60 DPD (days past due) default	
97	application_{train|test}.csv	DEF_60_CNT_SOCIAL_CIRCLE	How many observation of client's social surroundings defaulted on 60 (days past due) DPD	
98	application_{train|test}.csv	DAYS_LAST_PHONE_CHANGE	How many days before application did client change phone	
99	application_{train|test}.csv	FLAG_DOCUMENT_2	Did client provide document 2	
100	application_{train|test}.csv	FLAG_DOCUMENT_3	Did client provide document 3	
101	application_{train|test}.csv	FLAG_DOCUMENT_4	Did client provide document 4	
102	application_{train|test}.csv	FLAG_DOCUMENT_5	Did client provide document 5	
103	application_{train|test}.csv	FLAG_DOCUMENT_6	Did client provide document 6	
104	application_{train|test}.csv	FLAG_DOCUMENT_7	Did client provide document 7	
105	application_{train|test}.csv	FLAG_DOCUMENT_8	Did client provide document 8	
106	application_{train|test}.csv	FLAG_DOCUMENT_9	Did client provide document 9	
107	application_{train|test}.csv	FLAG_DOCUMENT_10	Did client provide document 10	
108	application_{train|test}.csv	FLAG_DOCUMENT_11	Did client provide document 11	
109	application_{train|test}.csv	FLAG_DOCUMENT_12	Did client provide document 12	
110	application_{train|test}.csv	FLAG_DOCUMENT_13	Did client provide document 13	
111	application_{train|test}.csv	FLAG_DOCUMENT_14	Did client provide document 14	
112	application_{train|test}.csv	FLAG_DOCUMENT_15	Did client provide document 15	
113	application_{train|test}.csv	FLAG_DOCUMENT_16	Did client provide document 16	
114	application_{train|test}.csv	FLAG_DOCUMENT_17	Did client provide document 17	
115	application_{train|test}.csv	FLAG_DOCUMENT_18	Did client provide document 18	
116	application_{train|test}.csv	FLAG_DOCUMENT_19	Did client provide document 19	
117	application_{train|test}.csv	FLAG_DOCUMENT_20	Did client provide document 20	
118	application_{train|test}.csv	FLAG_DOCUMENT_21	Did client provide document 21	
119	application_{train|test}.csv	AMT_REQ_CREDIT_BUREAU_HOUR	Number of enquiries to Credit Bureau about the client one hour before application	
120	application_{train|test}.csv	AMT_REQ_CREDIT_BUREAU_DAY	Number of enquiries to Credit Bureau about the client one day before application (excluding one hour before application)	
121	application_{train|test}.csv	AMT_REQ_CREDIT_BUREAU_WEEK	Number of enquiries to Credit Bureau about the client one week before application (excluding one day before application)	
122	application_{train|test}.csv	AMT_REQ_CREDIT_BUREAU_MON	Number of enquiries to Credit Bureau about the client one month before application (excluding one week before application)	
123	application_{train|test}.csv	AMT_REQ_CREDIT_BUREAU_QRT	Number of enquiries to Credit Bureau about the client 3 month before application (excluding one month before application)	
124	application_{train|test}.csv	AMT_REQ_CREDIT_BUREAU_YEAR	Number of enquiries to Credit Bureau about the client one day year (excluding last 3 months before application)	
125	bureau.csv	SK_ID_CURR	ID of loan in our sample - one loan in our sample can have 0,1,2 or more related previous credits in credit bureau	hashed

126	bureau.csv	SK_BUREAU_ID	Recoded ID of previous Credit Bureau credit related to our loan (unique coding for each loan application)	hashed
127	bureau.csv	CREDIT_ACTIVE	Status of the Credit Bureau (CB) reported credits	
128	bureau.csv	CREDIT_CURRENCY	Recoded currency of the Credit Bureau credit	recoded
129	bureau.csv	DAYS_CREDIT	How many days before current application did client apply for Credit Bureau credit	time only relative to the application
130	bureau.csv	CREDIT_DAY_OVERDUE	Number of days past due on CB credit at the time of application for related loan in our sample	
131	bureau.csv	DAYS_CREDIT_ENDDATE	Remaining duration of CB credit (in days) at the time of application in Home Credit	time only relative to the application
132	bureau.csv	DAYS_ENDDATE_FACT	Days since CB credit ended at the time of application in Home Credit (only for closed credit)	time only relative to the application
133	bureau.csv	AMT_CREDIT_MAX_OVERDUE	Maximal amount overdue on the Credit Bureau credit so far (at application date of loan in our sample)	
134	bureau.csv	CNT_CREDIT_PROLONG	How many times was the Credit Bureau credit prolonged	
135	bureau.csv	AMT_CREDIT_SUM	Current credit amount for the Credit Bureau credit	
136	bureau.csv	AMT_CREDIT_SUM_DEBT	Current debt on Credit Bureau credit	
137	bureau.csv	AMT_CREDIT_SUM_LIMIT	Current credit limit of credit card reported in Credit Bureau	
138	bureau.csv	AMT_CREDIT_SUM_OVERDUE	Current amount overdue on Credit Bureau credit	
139	bureau.csv	CREDIT_TYPE	Type of Credit Bureau credit (Car, cash,...)	
140	bureau.csv	DAYS_CREDIT_UPDATE	How many days before loan application did last information about the Credit Bureau credit come	time only relative to the application
141	bureau.csv	AMT_ANNUITY	Annuity of the Credit Bureau credit	
142	bureau_balance.csv	SK_BUREAU_ID	Recoded ID of Credit Bureau credit (unique coding for each application) - use this to join to CREDIT_BUREAU table	hashed
143	bureau_balance.csv	MONTHS_BALANCE	Month of balance relative to application date (-1 means the freshest balance date)	time only relative to the application
144	bureau_balance.csv	STATUS	Status of Credit Bureau loan during the month (active, closed, DPD0-30,… [C means closed, X means status unknown, 0 means no DPD, 1 means maximal did during month between 1-30, 2 means DPD 31-60,… 5 means DPD 120+ or sold or written off])	
145	POS_CASH_balance.csv	SK_ID_PREV	ID of previous credit in Home Credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit)	
146	POS_CASH_balance.csv	SK_ID_CURR	ID of loan in our sample	
147	POS_CASH_balance.csv	MONTHS_BALANCE	Month of balance relative to application date (-1 means the information to the freshest monthly snapshot, 0 means the information at	time only relative to the application

			application - often it will be the same as -1 as many banks are not updating the information to Credit Bureau regularly)	
148	POS_CASH_balance.csv	CNT_INSTALMENT	Term of previous credit (can change over time)	
149	POS_CASH_balance.csv	CNT_INSTALMENT_FUTURE	Installments left to pay on the previous credit	
150	POS_CASH_balance.csv	NAME_CONTRACT_STATUS	Contract status during the month	
151	POS_CASH_balance.csv	SK_DPD	DPD (days past due) during the month of previous credit	
152	POS_CASH_balance.csv	SK_DPD_DEF	DPD during the month with tolerance (debts with low loan amounts are ignored) of the previous credit	
153	credit_card_balance.csv	SK_ID_PREV	ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit)	hashed
154	credit_card_balance.csv	SK_ID_CURR	ID of loan in our sample	hashed
155	credit_card_balance.csv	MONTHS_BALANCE	Month of balance relative to application date (-1 means the freshest balance date)	time only relative to the application
156	credit_card_balance.csv	AMT_BALANCE	Balance during the month of previous credit	
157	credit_card_balance.csv	AMT_CREDIT_LIMIT_ACTUAL	Credit card limit during the month of the previous credit	
158	credit_card_balance.csv	AMT_DRAWINGS_ATM_CURRENT	Amount drawing at ATM during the month of the previous credit	
159	credit_card_balance.csv	AMT_DRAWINGS_CURRENT	Amount drawing during the month of the previous credit	
160	credit_card_balance.csv	AMT_DRAWINGS_OTHER_CURRENT	Amount of other drawings during the month of the previous credit	
161	credit_card_balance.csv	AMT_DRAWINGS_POS_CURRENT	Amount drawing or buying goods during the month of the previous credit	
162	credit_card_balance.csv	AMT_INST_MIN_REGULARITY	Minimal installment for this month of the previous credit	
163	credit_card_balance.csv	AMT_PAYMENT_CURRENT	How much did the client pay during the month on the previous credit	
164	credit_card_balance.csv	AMT_PAYMENT_TOTAL_CURRENT	How much did the client pay during the month in total on the previous credit	
165	credit_card_balance.csv	AMT_RECEIVABLE_PRINCIPAL	Amount receivable for principal on the previous credit	
166	credit_card_balance.csv	AMT_RECIVABLE	Amount receivable on the previous credit	
167	credit_card_balance.csv	AMT_TOTAL_RECEIVABLE	Total amount receivable on the previous credit	
168	credit_card_balance.csv	CNT_DRAWINGS_ATM_CURRENT	Number of drawings at ATM during this month on the previous credit	
169	credit_card_balance.csv	CNT_DRAWINGS_CURRENT	Number of drawings during this month on the previous credit	
170	credit_card_balance.csv	CNT_DRAWINGS_OTHER_CURRENT	Number of other drawings during this month on the previous credit	
171	credit_card_balance.csv	CNT_DRAWINGS_POS_CURRENT	Number of drawings for goods during this month on the previous credit	
172	credit_card_balance.csv	CNT_INSTALMENT_MATURE_CUM	Number of paid installments on the previous credit	

173	credit_card_balance.csv	NAME_CONTRACT_STATUS	Contract status (active signed,...) on the previous credit	
174	credit_card_balance.csv	SK_DPD	DPD (Days past due) during the month on the previous credit	
175	credit_card_balance.csv	SK_DPD_DEF	DPD (Days past due) during the month with tolerance (debts with low loan amounts are ignored) of the previous credit	
176	previous_application.csv	SK_ID_PREV	ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loan applications in Home Credit, previous application could, but not necessarily have to lead to credit)	hashed
177	previous_application.csv	SK_ID_CURR	ID of loan in our sample	hashed
178	previous_application.csv	NAME_CONTRACT_TYPE	Contract product type (Cash loan, consumer loan [POS] ,...) of the previous application	
179	previous_application.csv	AMT_ANNUITY	Annuity of previous application	
180	previous_application.csv	AMT_APPLICATION	For how much credit did client ask on the previous application	
181	previous_application.csv	AMT_CREDIT	Final credit amount on the previous application. This differs from AMT_APPLICATION in a way that the AMT_APPLICATION is the amount for which the client initially applied for, but during our approval process he could have received different amount - AMT_CREDIT	
182	previous_application.csv	AMT_DOWN_PAYMENT	Down payment on the previous application	
183	previous_application.csv	AMT_GOODS_PRICE	Goods price of good that client asked for (if applicable) on the previous application	
184	previous_application.csv	WEEKDAY_APPR_PROCESS_START	On which day of the week did the client apply for previous application	
185	previous_application.csv	HOUR_APPR_PROCESS_START	Approximately at what day hour did the client apply for the previous application	rounded
186	previous_application.csv	FLAG_LAST_APPL_PER_CONTRACT	Flag if it was last application for the previous contract. Sometimes by mistake of client or our clerk there could be more applications for one single contract	
187	previous_application.csv	NFLAG_LAST_APPL_IN_DAY	Flag if the application was the last application per day of the client. Sometimes clients apply for more applications a day. Rarely it could also be error in our system that one application is in the database twice	
188	previous_application.csv	NFLAG_MICRO_CASH	Flag Micro finance loan	
189	previous_application.csv	RATE_DOWN_PAYMENT	Down payment rate normalized on previous credit	normalized
190	previous_application.csv	RATE_INTEREST_PRIMARY	Interest rate normalized on previous credit	normalized
191	previous_application.csv	RATE_INTEREST_PRIVILEGED	Interest rate normalized on previous credit	normalized
192	previous_application.csv	NAME_CASH_LOAN_PURPOSE	Purpose of the cash loan	
193	previous_application.csv	NAME_CONTRACT_STATUS	Contract status (approved, cancelled, ...) of previous application	
194	previous_application.csv	DAYS_DECISION	Relative to current application when was the decision about previous application made	time only relative to the application

195	previous_application.csv	NAME_PAYMENT_TYPE	Payment method that client chose to pay for the previous application	
196	previous_application.csv	CODE_REJECT_REASON	Why was the previous application rejected	
197	previous_application.csv	NAME_TYPE_SUITE	Who accompanied client when applying for the previous application	
198	previous_application.csv	NAME_CLIENT_TYPE	Was the client old or new client when applying for the previous application	
199	previous_application.csv	NAME_GOODS_CATEGORY	What kind of goods did the client apply for in the previous application	
200	previous_application.csv	NAME_PORTFOLIO	Was the previous application for CASH, POS, CAR, …	
201	previous_application.csv	NAME_PRODUCT_TYPE	Was the previous application x-sell o walk-in	
202	previous_application.csv	CHANNEL_TYPE	Through which channel we acquired the client on the previous application	
203	previous_application.csv	SELLERPLACE_AREA	Selling area of seller place of the previous application	
204	previous_application.csv	NAME_SELLER_INDUSTRY	The industry of the seller	
205	previous_application.csv	CNT_PAYMENT	Term of previous credit at application of the previous application	
206	previous_application.csv	NAME_YIELD_GROUP	Grouped interest rate into small medium and high of the previous application	grouped
207	previous_application.csv	PRODUCT_COMBINATION	Detailed product combination of the previous application	
208	previous_application.csv	DAYS_FIRST_DRAWING	Relative to application date of current application when was the first disbursement of the previous application	time only relative to the application
209	previous_application.csv	DAYS_FIRST_DUE	Relative to application date of current application when was the first due supposed to be of the previous application	time only relative to the application
210	previous_application.csv	DAYS_LAST_DUE_1ST_VERSION	Relative to application date of current application when was the first due of the previous application	time only relative to the application
211	previous_application.csv	DAYS_LAST_DUE	Relative to application date of current application when was the last due date of the previous application	time only relative to the application
212	previous_application.csv	DAYS_TERMINATION	Relative to application date of current application when was the expected termination of the previous application	time only relative to the application
213	previous_application.csv	NFLAG_INSURED_ON_APPROVAL	Did the client requested insurance during the previous application	
214	installments_payments.csv	SK_ID_PREV	ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit)	hashed
215	installments_payments.csv	SK_ID_CURR	ID of loan in our sample	hashed
216	installments_payments.csv	NUM_INSTALMENT_VERSION	Version of installment calendar (0 is for credit card) of previous credit. Change of installment version from month to month signifies that some parameter of payment calendar has changed	
217	installments_payments.csv	NUM_INSTALMENT_NUMBER	On which installment we observe payment	
218	installments_payments.csv	DAYS_INSTALMENT	When the installment of previous credit was supposed to be paid (relative to application date of current loan)	time only relative to the application

219	installments_payments.csv	DAYS_ENTRY_PAYMENT	When was the installments of previous credit paid actually (relative to application date of current loan)	time only relative to the application
220	installments_payments.csv	AMT_INSTALMENT	What was the prescribed installment amount of previous credit on this installment	
221	installments_payments.csv	AMT_PAYMENT	What the client actually paid on previous credit on this installment	