HOME CREDIT DEFUALT RISK

Made by Group 2

Group member ID:

450019689

460255402

450151439

2. November. 2018

Executive summary

This report provides an analysis of the home credit default risk by making the correct prediction of whether home loan applicants will default. Our clients are home loan providers whose business objective is to maximum profit and reduce the proportion of default loans. With the correct default prediction, lenders can make a beneficial decision for them.

The datasets are provided by home credit group, and the exploratory data analysis is conducted for having a better understanding on this. We also implement four models to solve this problem and the result of logistic regression is arbitrarily chosen as the benchmark. The results show that the LightGBM perform the best prediction accuracy and AUC, and also give the lowest average error loss. Besides, this report also finds the features importance of the explanatory variables by comparing their SHAP value, and the result shows that some features such as three external sources and gender could have significant impact on the default prediction. Finally, we investigate the fact that this analysis conducted has some limitations and most of the limitations are caused by the computational constraint. For instance, only 3/4 of training set was used, and some good prediction performance models such as XGBoost and Neural Network are not applied due to the computational inefficiency.

Problem

Nowadays, more and more people rely on bank loans to pay for their home purchases, while many of them struggle to get loans due to insufficient or non-existent credit histories. And, unfortunately, there are untrustworthy lenders taking advantages from this population who can potentially repay the loan but get rejected by formal loan lenders. In addition to this, loan groups are also suffering from those non-performing loans, which could cause bankrupt and even instability in national financial system.

Our clients are home loan providers who are trying to maximum their profit and reduce the proportion of default loans. Therefore, for increasing the profit and minimizing non-performing loans, it is essential to measure each home loan applicant’s probability of default as target variable. Based on datasets provided by home credit group, we will develop several binary classification models including tree-based models, linear models and ensemble learning methods to classify whether a customer will default. By doing so, home loan providers could estimate each applicant’s probability of default, thereby adjust the home loan interest and loan limit according to the default risk. The underserved population will also be benefited if they could provide the related evaluation information. To test the reliability of our predictive models, the AUC and cross-validation score will be used as the reference standard for our model selection. The home credit group also provides a test dataset so that we could know the predictive accuracy for each model.

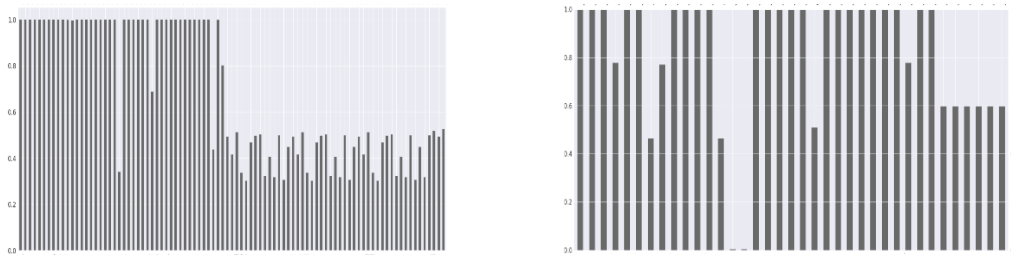
From our models, we also expect to find out which evaluation variables have high correlation with the target variable, and how does these variables affect the applicant's probability of default? In general, we expect that one's ability to repay a loan is most strongly correlated with factors such as income and credit history. However, since the data set consists of a various types of consumer information, such as loans from other financial institutions and credit card balance/repayments, a lot of other significant variables may emerge.

Data

The data is provided by Home Credit, a service dedicated to provide lines of credit (loans) to the unbanked population. There are 7 different sources of data: *Application_train/test* which is the main training and testing data with information about each loan application at Home Credit. Every loan has its own identification number, and the data comes with Target variable indicating 0 if the loan was repaid or 1 the loan was not repaid. *Bureau* and *Bureau_balance* contains the client's previous credits from other financial institutions and monthly credit balance. *Previous_application* shows previous applications for each borrower at Home Credit. POS_CASH_BALANCE, credit_card_balance and instalments_payment provide the details of previous monthly cash balance, credit balance and instalment payment of each client. The variable descriptions and relationships among these dataset is explained in Appendix 9.1.

Starting with the dataset, we identify and sum up all the column types over all data frames which contain money amount variables, ordinal categorical variables and time-length variables. The percentage of non-missing values for each variable has been calculated, and shown in figure1.1.

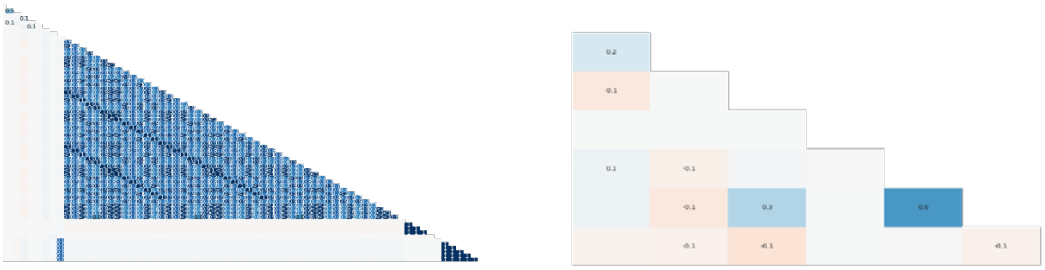
Figure 1.1: Percentage of missing values (more tables shown in appendix 9.2)



We found that the information about building where the client lives, the external source data and date-related information in previous applications tend to be absent, which may be due to borrowers with no house or incomplete data collecting.

The second step we plot the missing correlations (Figure 1.2) between columns with absent values, the blue colour indicates positive correlation and the red colour indicates negative correlation. The darker the colour, the higher the correlation.

Figure 1.2: Missing value correlations (More correlation tables in appendix 9.3)

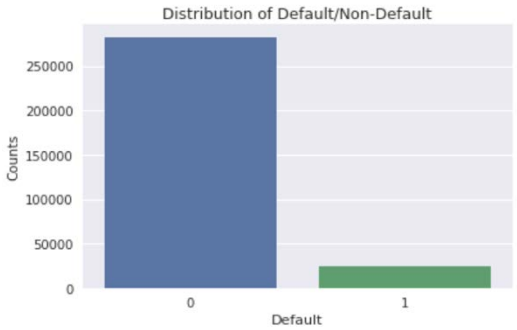


Application_train *Bureau_balance*

From figure1.2, it is apparently that there is correlation between absent value in the train dataset. For example, clients provide no information about living building if there is no other building related information. However, missing values such as money balance and credit balance in the *Bureau_balance* dataset shows less correlations between each other, which could be caused by different sources of data collection. More correlation tables are included in appendix 9.3.

For the purpose of filling missing values, we use values extremely different with these in each feature, which could help tree-based model to recognize the information underlying the missing values. For example, we fill positive value 1 for the time-length variables since the original value only occurs negative in the dataset and fill the negative 1 for the external source features since its original value is between 0 and 1. Alternatively, we use mean values to fill missing data for generalized linear model.

Figure 1.3: Target variable



As shown by the Figure1.3, the target variable has serious imbalanced class issue with only 8% as default applicants, which could affect our model's predictive ability. Therefore, methods like sub-sampling, over-sampling should be used to increase the number of default thereby train the model more useful.

Figure1.4: Correlations to the target variable

	TARGET
TARGET	1.000000
EXT_SOURCE_3	0.178919
EXT_SOURCE_2	0.160472
EXT_SOURCE_1	0.155317
DAYS_BIRTH	0.078239
REGION_RATING_CLIENT_W_CITY	0.060893
REGION_RATING_CLIENT	0.058899
DAYS_LAST_PHONE_CHANGE	0.055218
DAYS_ID_PUBLISH	0.051457
REG_CITY_NOT_WORK_CITY	0.050994

The figure 1.4 demonstrates the correlations between target and other features, we can see that the external sources information are the most correlated variables to the target variable. These are the normalized score generated by external credit assessment institutions, while the exact formula of them

are unknown. Since they are normalized to be between 0 and 1. The missing values could be replaced by -1 as mentioned above.

Figure 1.5: Paired Plots for External Sources and Target Variable

The green column indicates default and blue column indicates non-default in figure 1.5. In *EXT_SOURCE_3*, most of the default events occur at -1 (missing values), which give us some underlying information about missing values. Borrowers may tend to default their repayments if they do not have *EXT_SOURCE_3*.

Figure 1.6: Correlation of Each External Source with Other Variables

	EXT_SOURCE_1		EXT_SOURCE_2		EXT_SOURCE_3
DAYS_BIRTH	0.60061	REGION_RATING_CLIENT	0.292895	DAYS_BIRTH	0.205478
FLAG_EMP_PHONE	0.294147	REGION_RATING_CLIENT_W_CITY	0.288299	EXT_SOURCE_1	0.186846
DAYS_EMPLOYED	0.289848	EXT_SOURCE_1	0.213982	TARGET	0.178919
EXT_SOURCE_2	0.213982	REGION_POPULATION_RELATIVE	0.198924	DAYS_ID_PUBLISH	0.131597
FLAG_DOCUMENT_6	0.190874	DAYS_LAST_PHONE_CHANGE	0.195764	FLAG_EMP_PHONE	0.115293

From figure 1.6, we know that these external scores are correlated with other attributes of an applicant, which tell us what information the external sources used to make prediction. For instance, both external source 1 and external source 3 are highly correlated with date of birth, so that we know both sources use this feature as a explanatory variable.

Figure1.7: Correlation Matrix (More matrix in appendix 9.4)

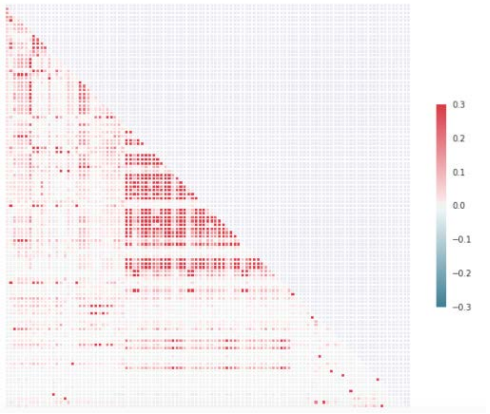
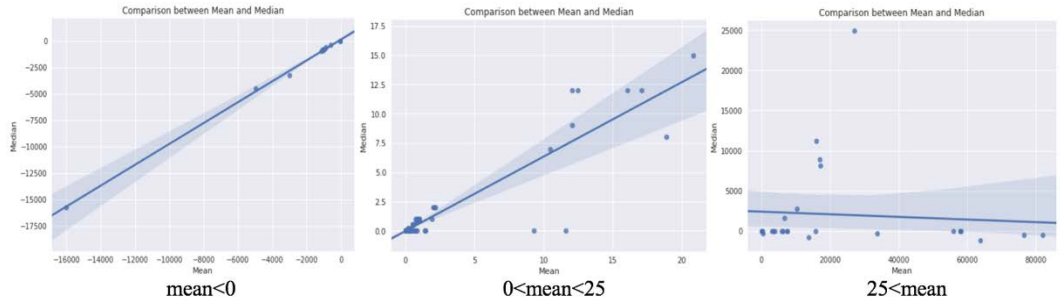


Figure1.7 shows the correlations between each feature in the train dataset. It donates some high correlations between each other. For example, there is 0.35 correlation between credit payment and credit balance, which may be reasonable since higher payment require higher credit balance to pay. Besides, the external sources are also highly correlated with some features as we mentioned previously.

Figure1.8: The distribution of mean and median of variables in the main application data frame



Basically variables with all value less than 0 are the number of days such as days_birth, days_credit. Since the dataset record these values as the length of time back to the date. We use 1 to fill the missing values. In the category of mean between 0 and 25, most of the variables are ordinal variables such as the number of children, the number of cars. The replacement for missing values is using a negative value. The variables with mean value greater than 25 are mainly the monetary features like credit balance and annuity payment. While it shows a significant positive skew since the mean value is much larger than the median value. It indicates some positive outliers in these variables.

Model

Feature Engineering

The first step is to aggregate all information into one data frame. For example, each bureau account has multiple records of past bureau balance, each current applicant has multiple previous application records. The aggregation methods used include mean, count, last figure, and number of unique of each column of the child data frame and append the aggregated figures to the parent data frame. At the same time, the time index is transformed to week, day, month, and year. We also use one hot encoding which is a process to convert categorical variables into a form that could be provided to ML algorithms to do a better job in prediction.

Since some child data frames have a time index, it is likely that the most recent figures of a column have stronger impact on the default probability. Therefore, training windows of 6 months, and 3 years are calculated as well in order to capture the potential trend effect.

Domain Knowledge: there could be more meaningful variables by combining existing variables. 20 features are hand crafted. Some examples include:

$$\text{payment rate} = \frac{\text{Amount of Annuity}}{\text{Amount of Credit}}$$

which is a measure of the lifetime of a loan. The longer the period, the more risk the loan carries.

$$\text{car\_to\_employ\_ratio} = \frac{\text{age of client's car}}{\text{the numbe of days the person stay in current employment}}$$

which is a measure of the financial status of a client.

$$\text{credit per non child} = \frac{\text{amount of credit}}{\text{number of nonchild family members}}$$

which is a measure of the pressure of repayment on an adult member of a family.

Feature Selection: the above aggregation method generated over **1600** features, among which some are redundant or noisy. This calls for feature selection tool to reduce the feature size. Stochastic Gradient Descent Classifier using support vector machine with l1 regularization term can do feature selection by forcing the coefficients of meaningless variables to zero. It managed to reduce the feature size to **781**, which is half of the original figure.

Feature Scaling: Since one of candidate model is linear regression model. The feature set must be scaled to 0 mean and 1 variance to make sure that no feature can dominate others. However, this brings up an issue that the number of decimal places becomes too large that the data frame size increases exponentially, which results in slow file reading and modelling. Therefore, all floats are round to three decimals places for faster computation.

Model Building

Four models were implemented for this problem, including logistic regression with l1 regularization term, random forest classifier, LightGBM and a neural network. Logistic regression serves as a benchmark model but its predictive power is not supposed to the best among three given the complex patterns hidden in the data. Since random forest uses bagging to ensemble trees while LightGBM uses boosting, both of them are implemented for comparison purpose. LightGBM is preferred over XGBoost due to the former's computational efficiency.

Sampling: At this stage, unfortunately it was found out that the training size is too large to the tree-based models to run. Therefore, a random sample with size as 3 / 4 of training set is drawn without replacement as the input of the models.

Under-sampling: Under sampling was used to mitigate the imbalanced class problem of there being much more non-defaults than defaults. The exact method is as simple as randomly drawing a portion of data from the majority class (non-defaults) to make sure the ratio between classes is more balanced:

$$\text{ratio} = \frac{N_{\text{re,Majority}}}{N_{Minority}}$$

where, $N_{re,Majority} =$ the number of samples in the majority class after resampling,
$N_{Minority} =$ the number of samples in the minority class
After running a simple logistic regression solely on the training set, the best ratio was determined to be 10/3 (based on ROC-AUC score). The resampled data set then has an equivalent of 23.1% of rows being defaults (up from 8%), hence allowing the model to learn the attributes of both $Y = 0, 1$.

Logistic Regression: Logistic regression with stochastic gradient descent was chosen as our baseline model. Logistic regression is a generalized linear model which assumes that the response values given the predictors follow a Bernoulli distribution. It also assumes that the log of the odds ratio of the mean follows a linear distribution. While this gives the model nice properties, its predictive power is not expected to outperform the other three models given the complex patterns hidden in the data. The hyper parameters were 1) regularization term: this determines the size of the penalty for added coefficients 2) learning rate 3) l1 ratio: this controls the proportion allocated to ridge and lasso penalty (1 denotes more to lasso) 4) balanced weights or unbalanced weights for different classes.

Random Forest: Random Forests are tree based models that implement bagging. Random subsets of the training data are used to train many trees and predictions for particular points are aggregated over the 'forest' of trees. In addition, random forests add an extra degree of randomness by randomly selecting a subset of features to use for each split. This produces predictions which have low variance and low bias, provided that the number of trees is large enough. Hyper parameters tuned include 1) the number of estimators, 2) the minimum sample split, 3) the minimum samples for a leaf, 4) the criterion for split, 5) the maximum depth of a tree, 6) balanced weights for different classes or not. This ensures that the best estimator from the Search algorithm has a good balance between complexity and accuracy.

LightGBM: Light gradient boosting is a tree based boosting method. The key concept behind boosting is to train many individual weak learners, usually trees and output a single strong leaner. Weak learners improve upon previous learners by fitting the previous iterations' pseudo residuals (in the form of the gradient of the loss function with respect to predicted value), thereby producing better fits with lower variance. Since this model was previously known to fit the dataset accurately and efficiently, a wide range of hyper parameters were tuned including: 1) number of leaves: complexity regularize which controls maximum amount of tree leaves for base learners 2) maximum depth of a tree: this limits the maximum depth for tree models and is used to control over fitting 3) bagging fraction and feature fraction: reduces variance of estimators by training learners on random percentage of dataset and using a random subset of features to split trees 4) learning rate: controls how much each individual tree contributes towards the final prediction 5) the number of boosting iterations. These parameters are used to control the complexity of the model, which affects the bias and variance of predictions.

Neural Network: A simple multilayer perceptron neural network was performed to learn the complex relationships between loan default and the other explanatory variables. Neural networks produce complex combinations of the original predictors as derived features, thereby potentially making more accurate predictions. Hyper parameters tuned include 1) number of layers: this controls number of hidden layers (more layers usually results in better fits) 2) neurons: this controls the number of neurons for each layer. Together with number of layers, it controls the number of weights in the network 3) batch size: the size of the subsets used to iteratively train the network 4) epochs: the number of times the full training set is passed through the model 5) dropout: regularization term that controls the number of inputs to each subsequent layer 6) learning rate and 7) optimisation algorithm.

Hyper parameter Tuning

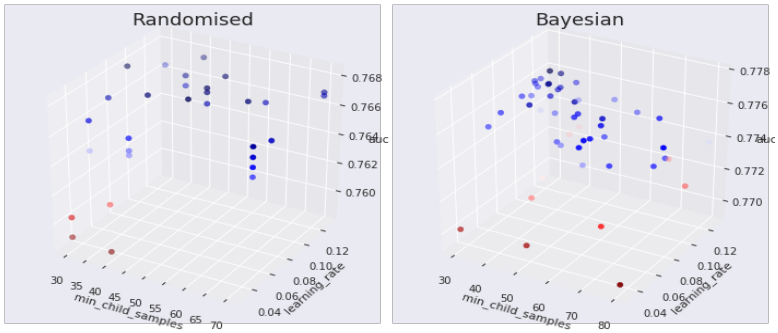
The parameters of the above models were tuned using randomized search CV and Bayesian optimisation using the library hyper opt. Grid search CV was excluded given its computational power and the size of the dataset.

Randomized Search CV: This method searches for the optimal set of parameters for a model by randomly selecting potential parameters from a dictionary of parameters. It is preferred over Grid Search CV given the size of the dataset and is likely to return the optimal result provided the optimal combination of parameters is included within the dictionary of possible values.

Bayesian Optimisation: Bayesian optimisation is an automated method for finding the optimal hyper parameters. It uses informed search algorithms that attempt to find the optimal set of parameters which minimize a function (e.g. RMSE) whilst limiting calls to the optimisation function (i.e. less fitting). The key difference with Bayesian optimisation is that is uses an algorithm which evaluates the probability of loss given the current set of parameters, thereby using past evaluation results to choose the next values to evaluate.

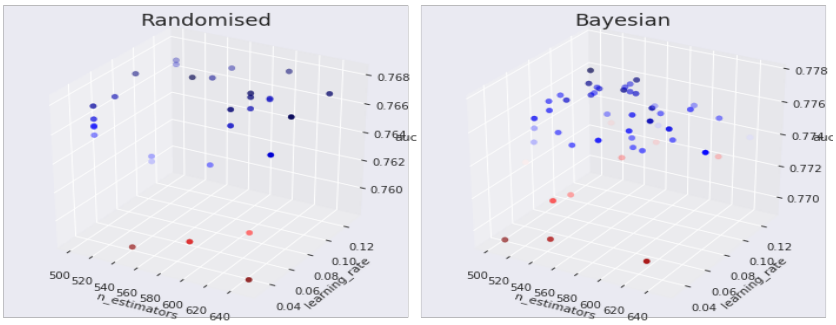
Below are various plots of the effect of different parameters on the average cross validation AUC. Note, that the plots are only for LGB and neural network as they were clearly the best performing models:

Figure2.1: LGB (1)



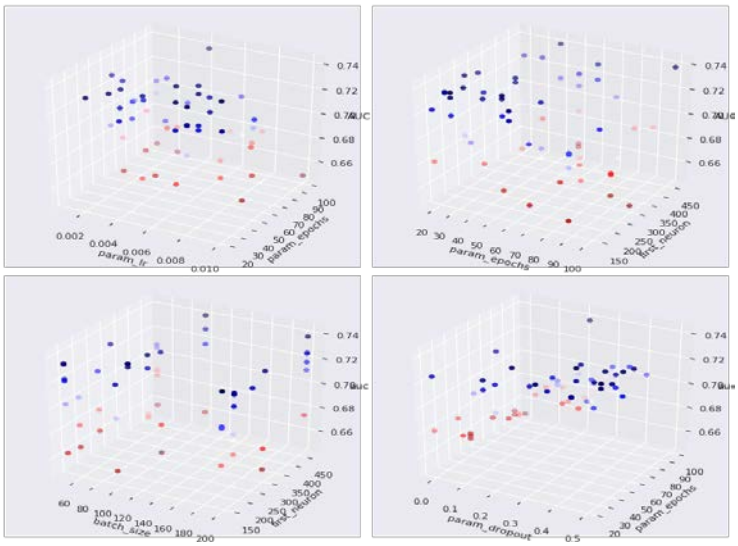
As can be seen from the above graphs, a higher learning rate coupled with lower number of data in each leaf results in better scores. This is quite intuitive as lower min_child_samples usually result in lower bias (better fits) whilst higher learning rates increase the contribution each tree makes to the final predictions

Figure2.2: LGB (2)



Lower number of learners with a higher learning rate seems to result in higher AUC scores. This can be seen quite clearly from Bayesian graph where data points are clumped towards the far corner

Figure2.3: Neural Network



In general, from the figure2.3, we notice several trends. First, a lower learning rate combined with a higher number of epochs leads to better AUC. This is intuitive as we would expect the network to improve with a greater number of iterations. Second, a higher dropout rate also seems to result in higher AUC. This could mean that lower number of neurons for each layer would result in better fits.

Model Evaluation

The optimal models for both randomise search CV and hyper opt were evaluated using three fold stratified cross validation to obtain the following results:

	TNR	TPR	Accuracy	AUC	Loss
lgb_hyperopt_clf	0.709	0.694	0.703	0.767	-1046.996
lgb_random_clf	0.810	0.562	0.708	0.766	-949.994
rf_hyperopt_clf	0.947	0.141	0.615	0.690	-1093.273
logistic_random_clf	0.745	0.506	0.646	0.683	-1173.131
Random_Forest_random_clf	0.707	0.555	0.644	0.685	-1208.153
logistic_hyperopt_clf	0.752	0.506	0.651	0.685	-1154.393
NN_Random_clf	0.802	0.530	0.690	0.741	-1007.934

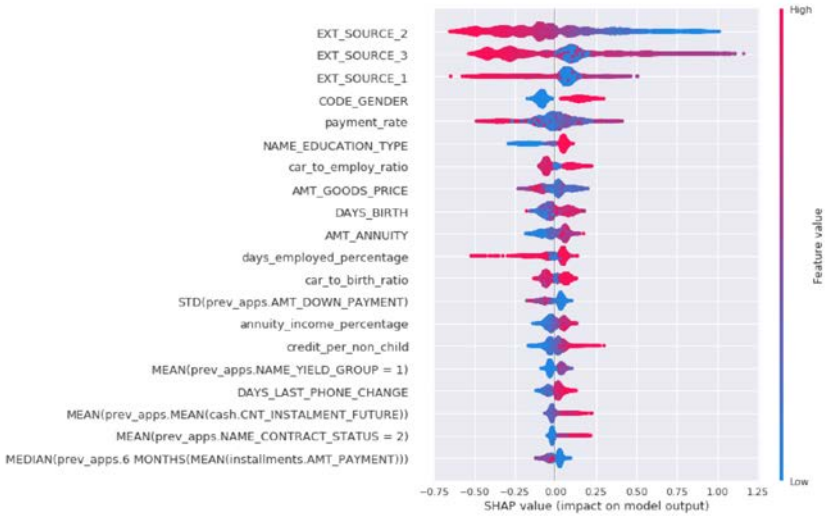
LightGBM outperforms the other models in terms of AUC and accuracy. This is expected given the model is known to fit the dataset well. However, other models give higher TNR and TPR, which are arguably more important than the AUC. In this context, TNR refers to the model's ability to predict actual negatives (i.e. non default) while TPR refers to the model's ability to correctly predict defaults. Most models demonstrate a lower TPR given the lower proportion of default rates. However, LGB parameters found determined using hyper opt outperforms the other models substantially in terms of TPR.

In terms of a business context however, lenders would focus much more on type I and type II error rather than the above metrics. Hence, a loss metric was also calculated to determine the loss Home Credit Group would incur from making these errors. The probability of type I and type II errors are determined by subtracting the TNR and TPR from 1 respectively. In this case, type I error means incorrect rejection of the while type II error means that we fail to reject the loan application when the borrower actually defaults. Therefore, the loss metric was calculated by multiplying the type I error with average loss of net interest income and adding this to the probability of type II error multiplied with average loss of principle. These figures were drawn from Home Credit Group's financial statements. Again, LightGBM is the best model with regards to this metric.

Analysis

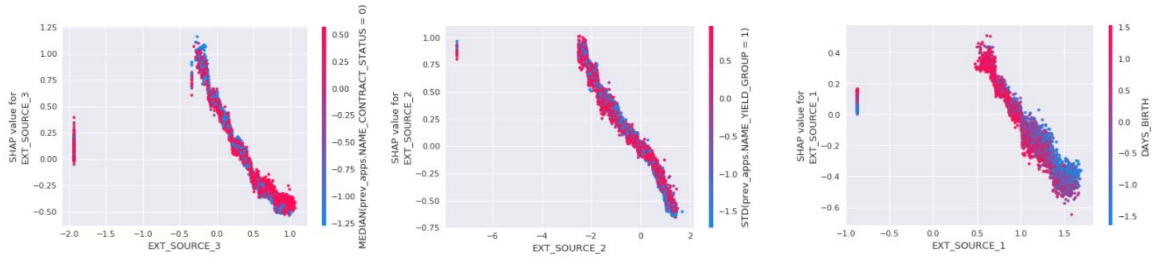
This part answers the question that what predicts the default of an applicant. A model explanation tool called SHapley Additive exPlanations (SHAP) is used. It assigns each feature an importance value for a particular prediction. SHAP connects game theory with local explanations, uniting several previous methods [1-7] and representing the only possible consistent and locally accurate additive feature attribution method based on expectations. There are three kinds of plots below, namely the feature importance graph, the feature dependence plots, and the prediction decomposition.

Figure3.1: Feature Importance Graph (Feature descriptions in appendix 9.5)



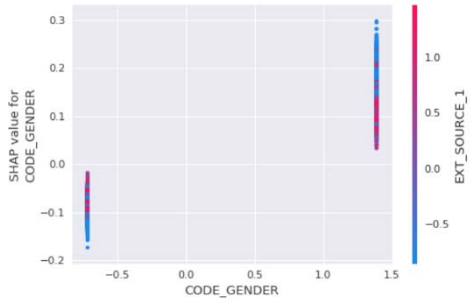
From the graph above, one can tell that the external sources of information are the strongest predictors of default rates by absolute SHAP value. In general, a higher external source score (represented by red) is correlated with more negative SHAP values, meaning lower chances of default. This is followed by variables relating to the applicants' characteristics, such as gender, education background and age, as well as monetary variables such as, payment rate, amount of annuity and payment rate. At the same time, these features may also exhibit interaction effects as they are unlikely to independently influence probabilities of default. These interactions are examined below:

Figure3.2: The Feature Dependency Plot of External Sources 1-3



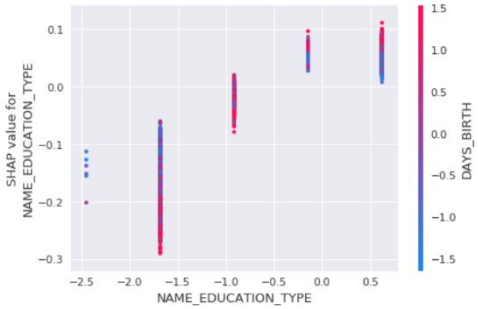
The above feature dependency plots show the interaction of each of the external sources of information with another feature automatically chosen by SHAP. For example, EXT_SOURCE_1 has a lower effect on the model output when DAYS_BIRTH are lower, as represented by lower SHAP values for the blue dots. It is also interesting to see that applicants with missing values (outside 0 and 1) in EXT_SOURCE (especially source 2) indicate that they are more likely to default.

Figure3.3: The Dependency Plot of Other Importance Features

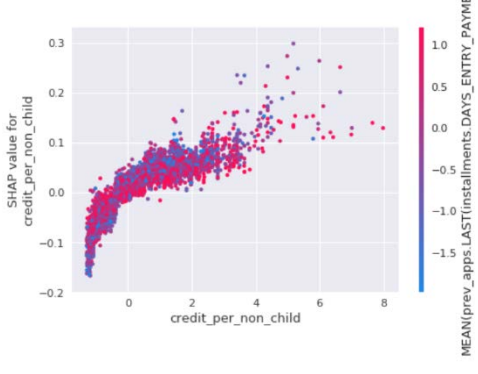


On average, males (code 1) have a higher probability of default than females, as evidenced by more positive SHAP values. Interestingly enough, for males, higher EXT_1 scores represent a lower chance of default whilst for females, it has the opposite effect.

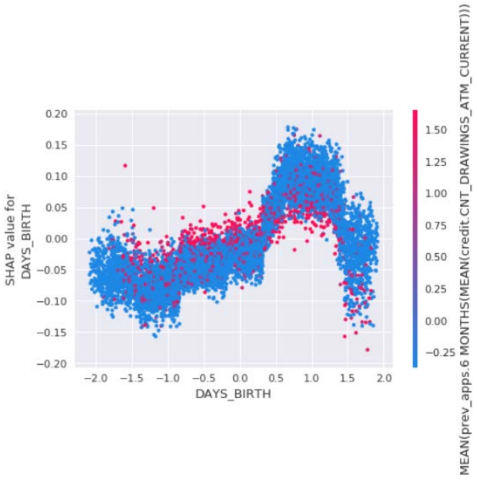
Payment rate is the ratio of credit amount and annuity. The payment rate has a threshold between 1 and 2. For values before the threshold, higher value shows higher chance of default. For values above the threshold, the chance of defaults drops considerably and then increases, which is a quite non-intuitive and complex pattern.



From the graph, it is quite clear that different education types Different education backgrounds groups have different average chances of default, and different variations as well. The interaction with age is interesting. Since days_birth represents the number of days since birth, the higher this figure, the elder the person. In this graph, elder ages often reverse the default probability to the base value of 0.5.



This variable represents the credit amount an adult family member share. It demonstrates a nonlinear pattern where the default chance jumps up at a low value, and linearly increases as the value increases. The variance of default probability also increases. It seems that log transformation of this variable can reveal more information. Then the interpretation would be intuitive as that the heavier the loan every adult family member has on its shoulder, the higher chance the family would default.

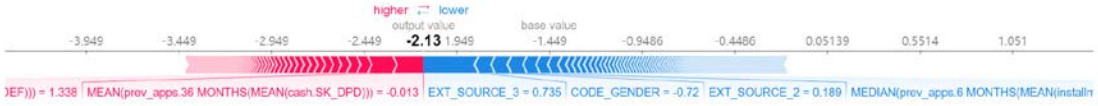


This graph reveals the effect of ages. The DAYS_BIRTH has negative values. So larger figure means younger age. Youth (18 -25) group has the highest chance of default. As the age grows, the default chance would be decreasing. But at a much elder age, the chance reverses up a little bit. This is an intuitive result. Since youth has less work experience and thus less income. Their family are gradually retiring. Therefore, they have less money to repay the loan as other age groups.

The amount of annuity is the amount an applicant must pay every year until the full credit amount is repaid, which is a positive figure. The graph demonstrates a log effect, where the small increase in annuity increases the default chances significantly. For intermediate level of annuity, the default chance varies considerably among applicants. Surprisingly, the default chance drops back to the base value as the annuity increases further but with some outliers.

There are many important features that could be interesting to demonstrate, but they are not included given the page limit. As we have discussed, if variables that have non-linear relationship with the default chance could be engineered, such as log or power transformation, for example, the log (credit_per_non_child), the model could potentially be more accurate and robust.

SHAP values could also decompose the prediction in terms of the contribution from every variable. For example, for this applicant, he/she is expected to be able to repay the loan. The major factors are the external sources 2, 3 and the gender.



This makes sure that whenever this model makes a prediction, it also lists out the reasons for its decision to assist the human decision maker to make more informed decision.

Criticism

Assumptions

1) In feature creation, when combining multiple records for a single entity, only a limited amount of quantities has been calculated. Some meaningful quantities such as standard deviation, skewness, kurtosis, and inter-quantile range would have been added if the computational costs were affordable and the computational speed was fast enough.

2) In feature scaling, there might be better scalers such as “robustscaler”, “minmaxscaler” from scikit-learn. The former adjusts any distribution towards normal distribution. The latter handles outliers well. Unfortunately, due to the resource constraint, only the standard normalization method was used.

Reliability

1) In the sampling part, only 3/4 of training set were used given the computational constraint. If conditional allows, the whole training set should be definitely used. More advanced under-sampling methods such as K-nearest neighbors or even combination of over- and under-samplings could be experimented if they do not take such long time.

2) In the model building part, a wider range of models, such as XGBoost, Neural Network are good candidates as well. The only reason they did not show up in our report is their computational inefficiency. Even more, the model stacking would increase our model’s stability and accuracy as well. But since there is no formula for that, it is essentially a trial and errors process that takes lots of computational resources and time.

3) In the hyper parameter tuning part, the Randomized Search CV does a good job in finding good hyper parameter sets but this is not guaranteed to be the best. Grid Search CV ensures the best feature set is picked, at the cost of ten times more computational resources.

Reproducibility

1) The major randomness comes from Randomized Search CV. Since the potential combinations of different hyper-parameters values is large while the number of iterations is limited given the computational cost, the resulting best estimator would vary for every new search, thus the model accuracy varies.

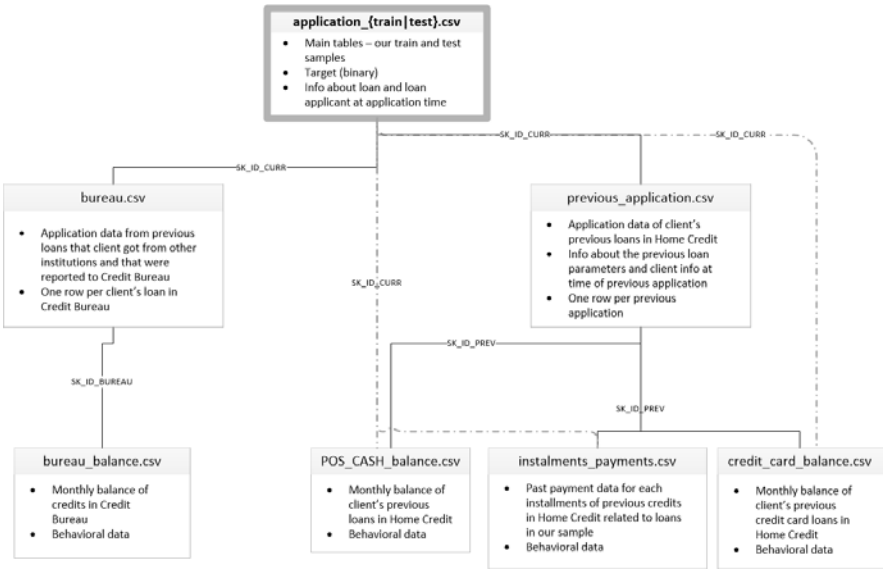
2) The second major randomness comes from the random feature selection and sampling inside LightGBM model building. This parameter serves the purpose of reducing over-fitting, and increases model randomness at the same time.

3) The number of folds for cross validation is 5, not a very large number. This figure is picked for computation cost reason. Therefore, the cross-validation score and the resulting best estimator could vary for each run.

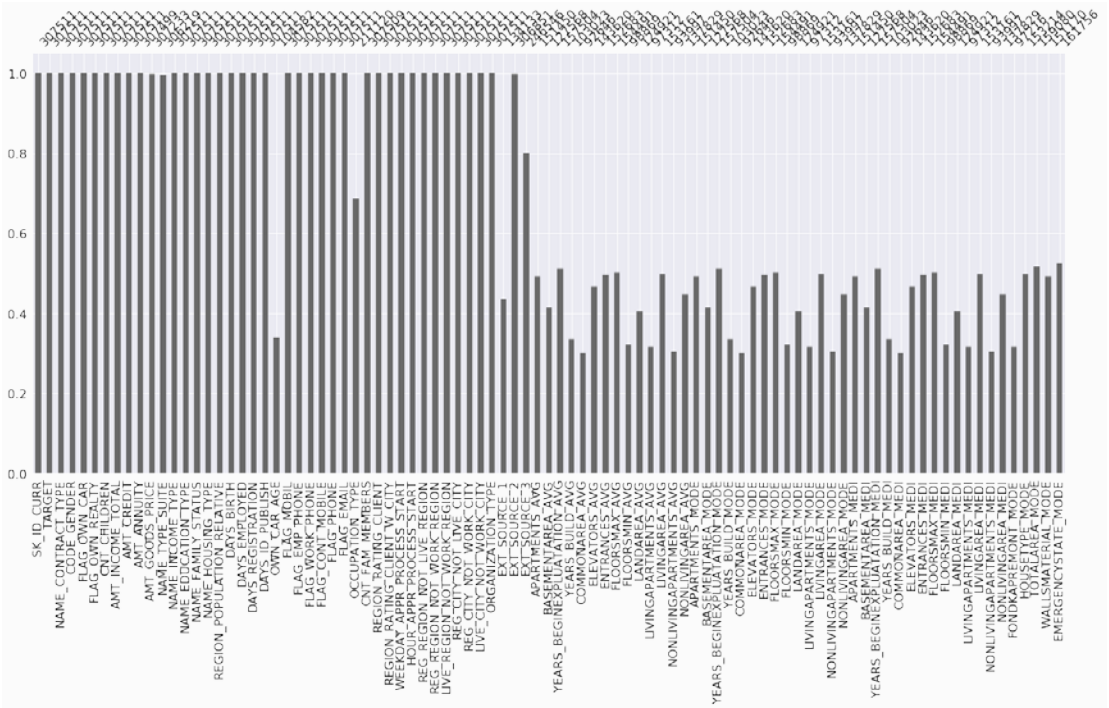
4) Another randomness comes from the random sampling and under-sampling. However, this is modified by specifying the random state to certain values.

Appendix

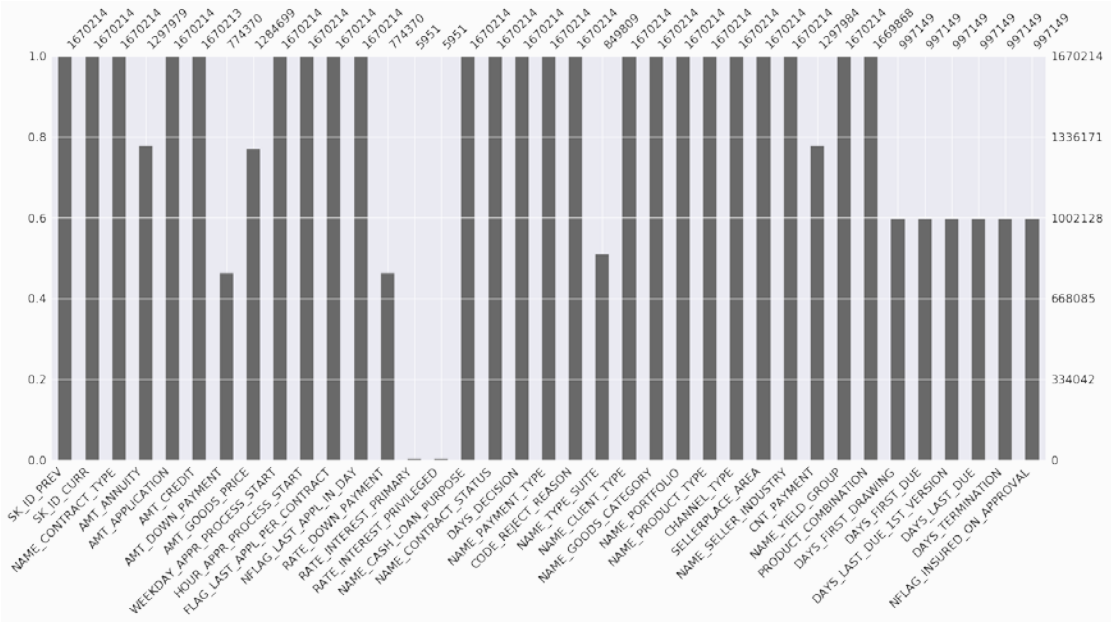
9.1: The Relationship between Data frames

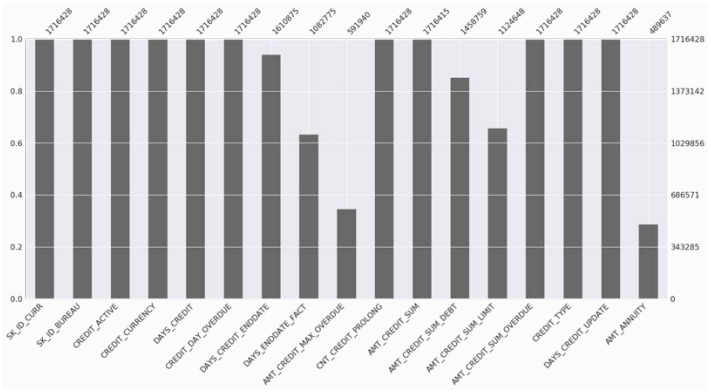


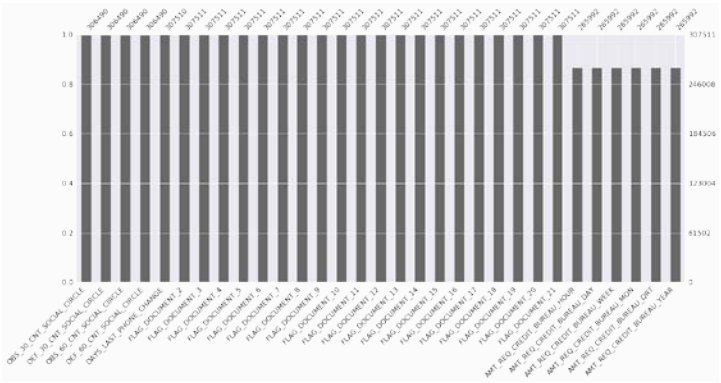
9.2: Percentage of missing values in all data frames

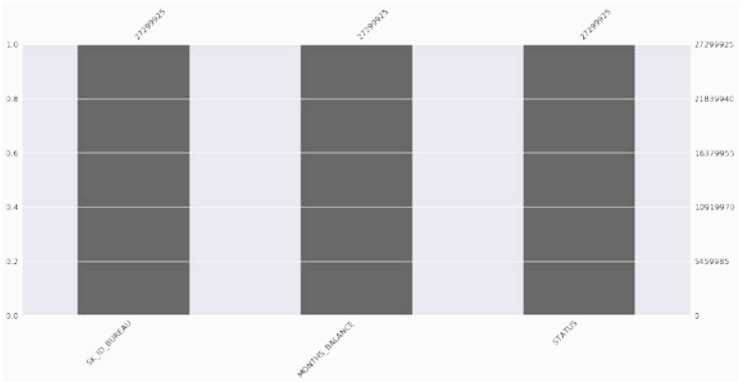


Group 2: 460255402, 450151439, 450019689

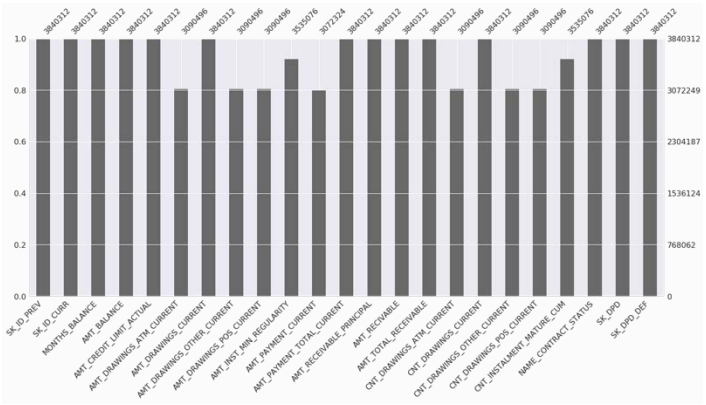


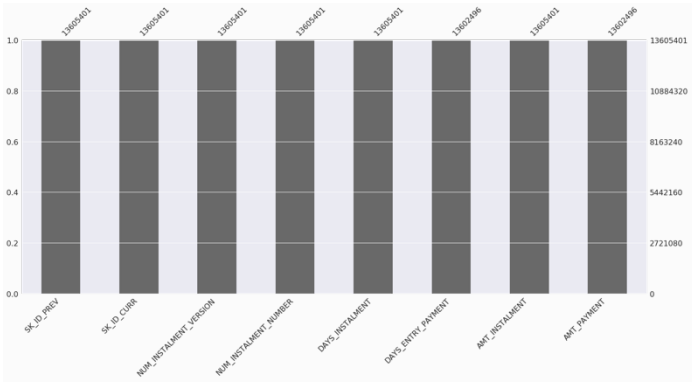


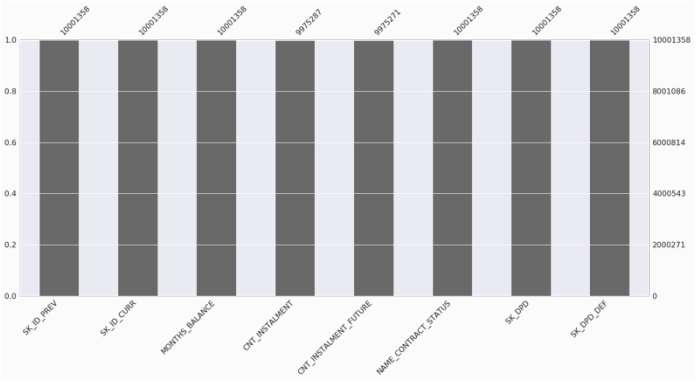




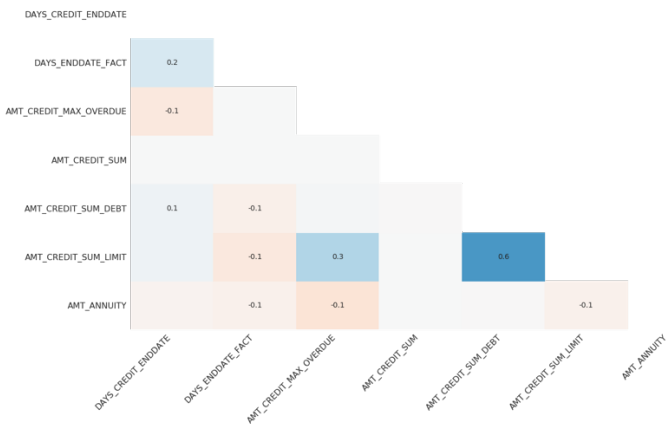
Group 2: 460255402, 450151439, 450019689



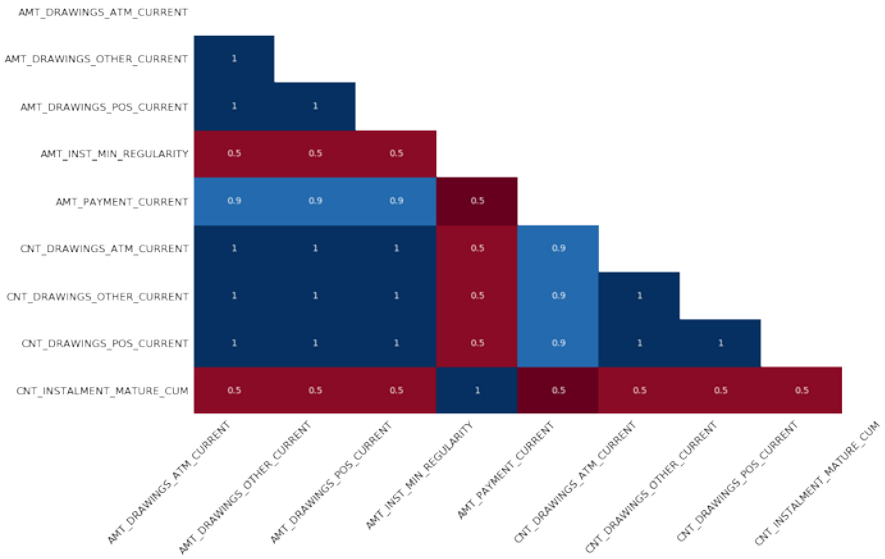


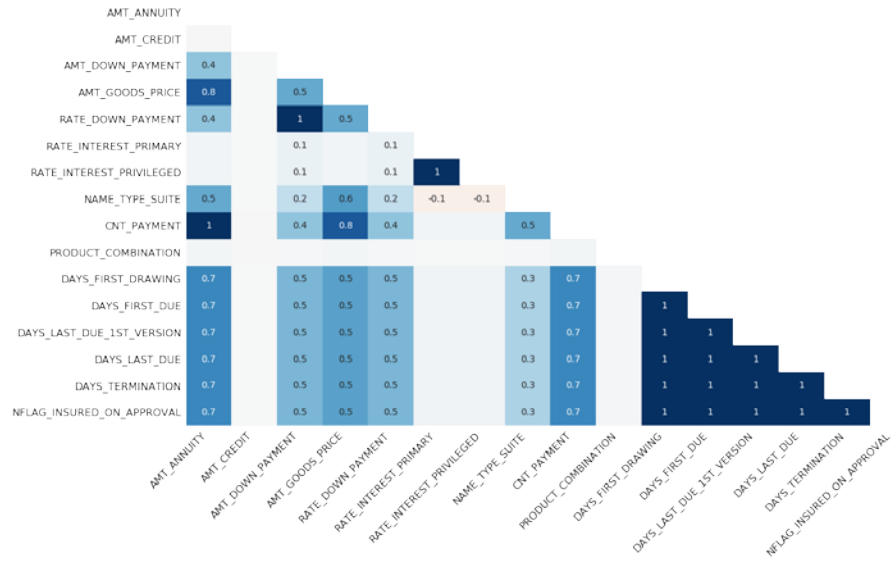


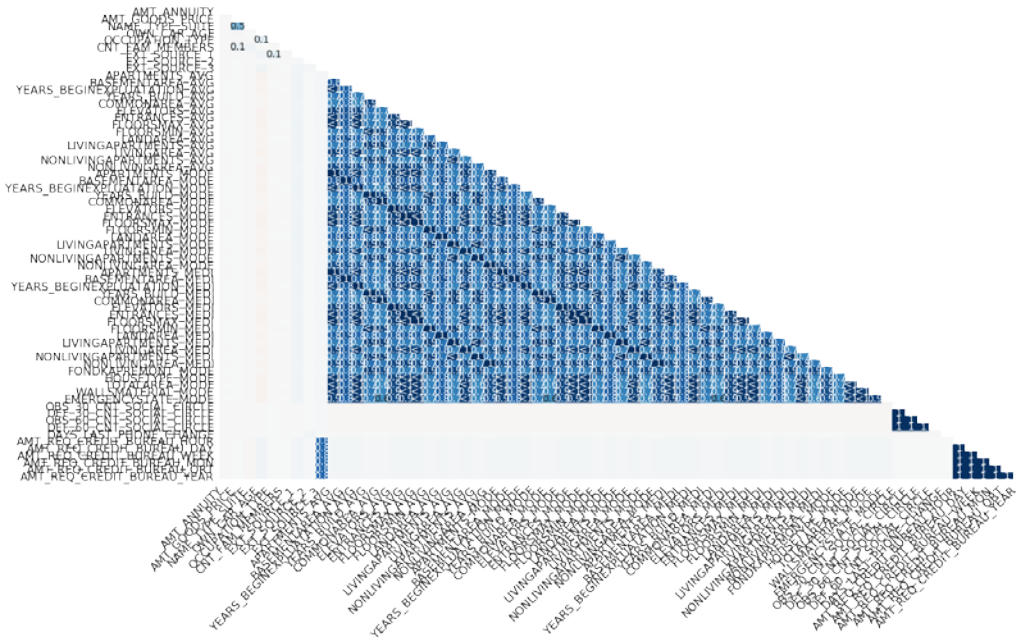
9.3: Missing value correlations in all data frames



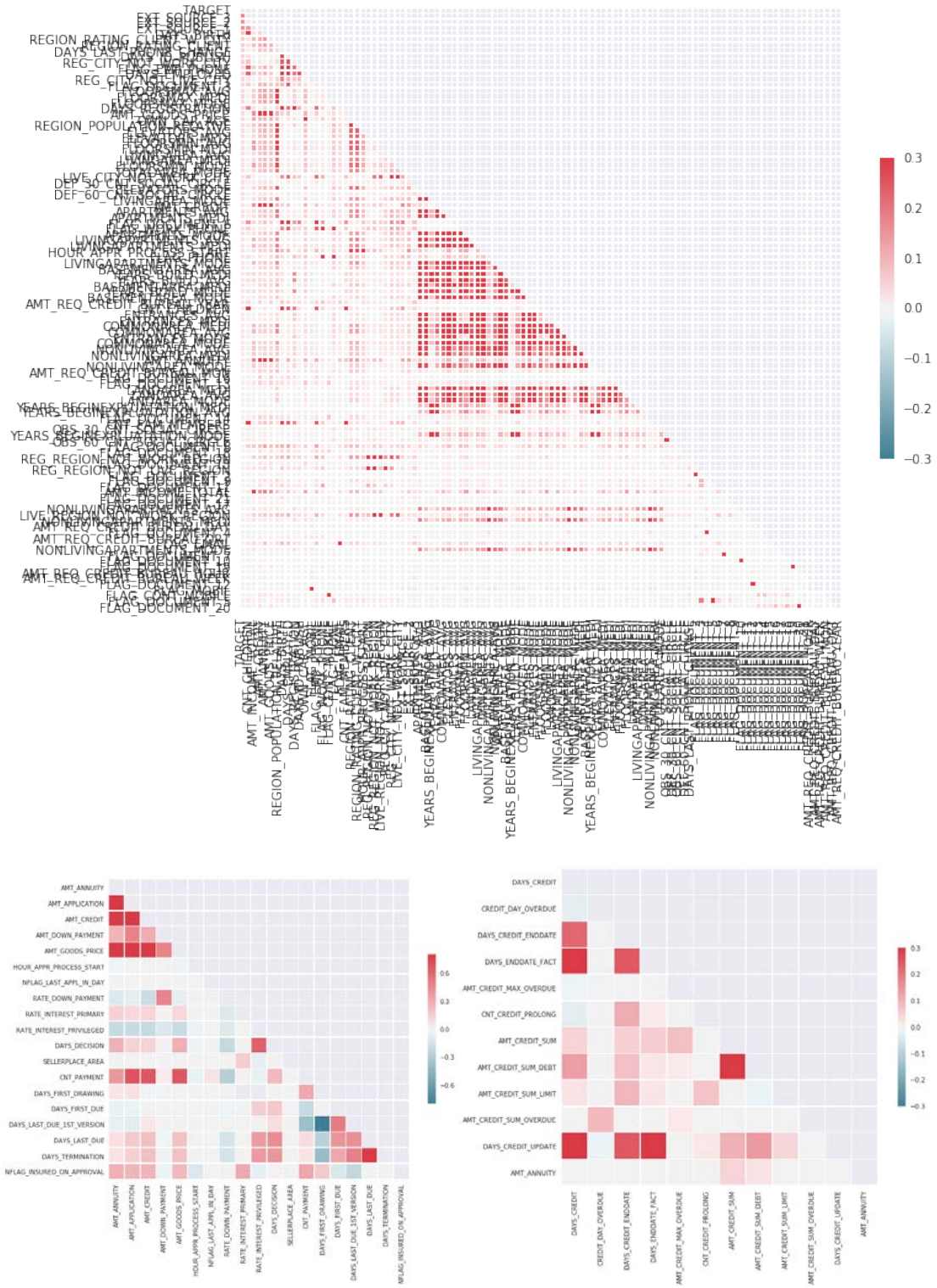
Group 2: 460255402, 450151439, 450019689

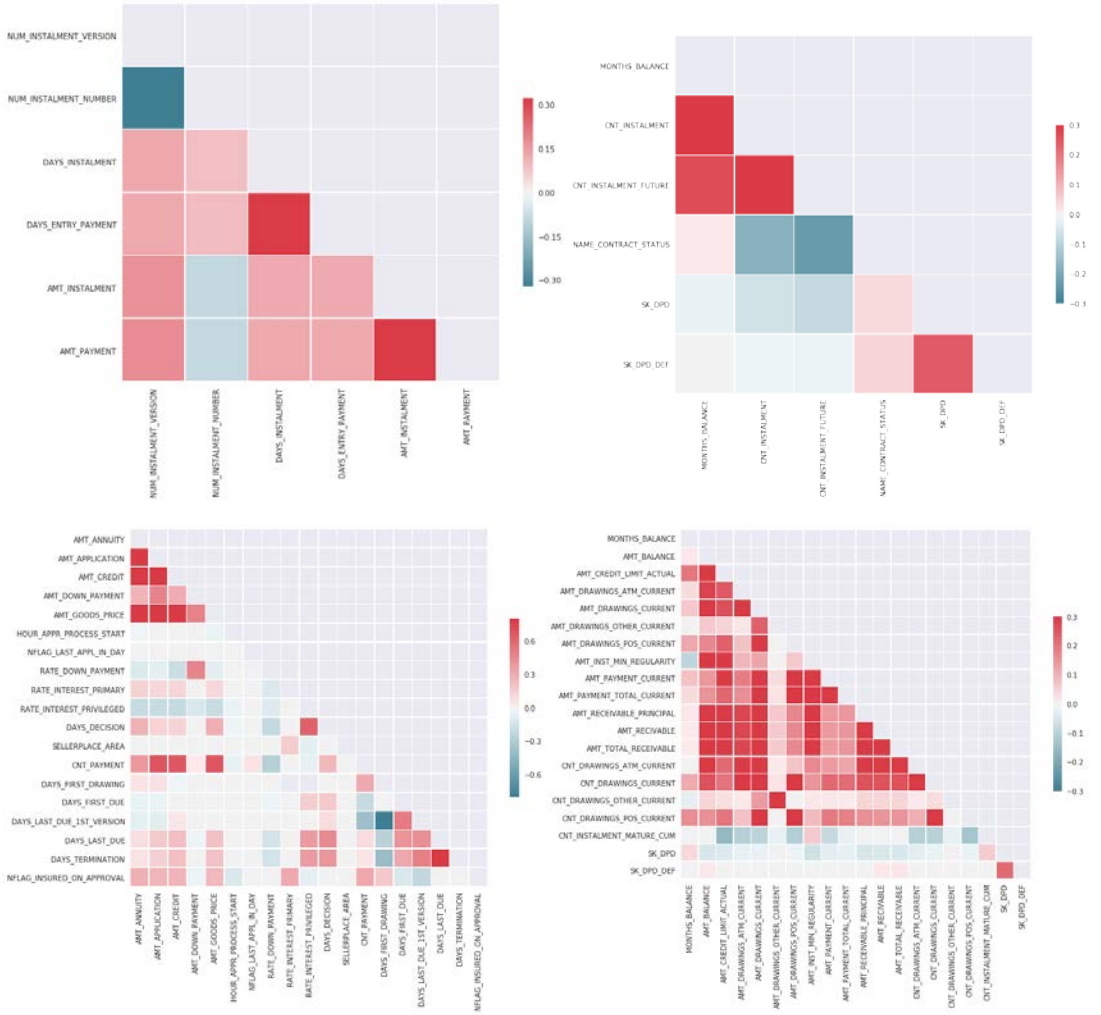






9.4： Correlation Matrix in all data frames





9.5: Descriptions of important features

Name	Description	Range
EXT_SOURCE_1	Normalized score from external data source 1	(0,1)
EXT_SOURCE_2	Normalized score from external data source 2	(0,1)
EXT_SOURCE_3	Normalized score from external data source 3	(0,1)
CODE_GENDER	Gender of the client	(0,1)
Payment rate	Payment rate	Continuous number
EDUCATION_TYPE	Level of highest education the client achieved	Continuous number
Credit_per_non_child	People’s credit balance without child	Continuous number

Group 2: 460255402, 450151439, 450019689

DAYS_BIRTH	Client's age in days at the time of application	Continuous number
AMT_ANNUITY	Annuity of the Credit Bureau credit	Continuous number