Credit Risk Analysis and Modeling

Author: Florence Sun

1. Introduction

Credit scoring is using statistical modeling to analyze relevant customers' data and transform the messy data information into numeric measures to guide credit decisions. The ultimate goal of credit scoring is to minimize loss that banks may suffer.

In the report, a data set with 50,000 credit card application records with 53 features and 1 binary response from a bank in Brazil in 2007 is used. This report covers topics listed as below:

- data preparation (Section 2)
- scorecard development using weight of evidence, information value and logistic regression (Section 3)
- random forest (Section 4)
- XGBoosting (Section 5)
- model comparison (Section 6)
- two-cut-off point strategy (Section 7)

2. Data Preparation

The data preparation mainly includes dealing with missing values, treating outliers and other operations.

2.1 Identity feature

The feature SEX is an identity feature and should not be included in the scorecard model. Thus, SEX feature is dropped.

2.2 Missing values

Missing values occur when no data value is stored for the variable in an observation. The missing values should be dealt before feeding data into models and a significant point is that the data is split into training and test data set with ratio of 0.8 before working on missing values, which is to avoid information leakage from test data set into training data set.

Table 1 shows features where missing values exist. For example, the feature STATE_OF_BIRTH at the first row has 1676 missing values, which takes 4.19% of the whole data.

According to the ratio of missing values in each column shown in Table 1, different strategies are used when dealing with missing values.

Table 1. The number and ratio of missing values in the data set

	Feature	Number_of_null_values	Ratio_of_null_values(%)
0	STATE_OF_BIRTH	1676	4.1900
1	CITY_OF_BIRTH	1676	4.1900
2	RESIDENCIAL_BOROUGH	5	0.0125
3	RESIDENCIAL_PHONE_AREA_CODE	6597	16.4925
4	RESIDENCE_TYPE	1080	2.7000
5	MONTHS_IN_RESIDENCE	3048	7.6200
6	PROFESSIONAL_STATE	27455	68.6375
7	PROFESSIONAL_CITY	27309	68.2725
8	PROFESSIONAL_BOROUGH	27799	69.4975
9	PROFESSIONAL_PHONE_AREA_CODE	29279	73.1975
10	PROFESSION_CODE	6186	15.4650
11	OCCUPATION_TYPE	5854	14.6350
12	MATE_PROFESSION_CODE	23033	57.5825
13	MATE_EDUCATION_LEVEL	25812	64.5300

- Fill the missing values with median for numerical variable or mode for categorical variable when the ratio of missing values is small. For example, RESIDENCIAL_BOROUGH only has 5 missing records, so the mode of the column is filled. RESIDENCE_TYPE, MONTHS_IN_RESIDENCE, STATE_OF_BIRGH, CITY_OF_BIRTH use this method as well.

- Drop the columns where the missing value ratio is larger than 60% and impossible to fill using other information. For example, PROFESSIONAL_BOROUGH and PROFESSIONAL_PHONE_AREA_CODE.

- Fill missing values using information from other columns. For example, values from RESIDENTIAL_STATE are used to fill missing values in PROFESSIONAL_STATE. The

reason is that for the rows where both columns have not null values, the ratio of rows where RESIDENTIAL_STATE and PROFESSIONAL_STATE have same values is up to 98.22%. Therefore, it is confident to say that at least 98.15% of data of these two columns have the same values. Meanwhile, it makes sense since most of individuals work and live in the same state. Similarly, missing values in the PROFESSIONAL_CITY are filled with values from RESIDENTIAL_CITY since 65.31% of the data in these two columns have the same value.

- Group feature which has missing values into several groups and use mode of each group to fill null in each group. PROFESSION_CODE has 15.47% of missing values. When looking further into the missing values, the customer records have PROFESSIONAL_CITY and PROFESSIONAL_STATE data. So, it is not appropriate to assume that missing values are caused by unemployment and just use a placeholder to represent it. Instead, a more appropriate way is grouping PROFESSION_CODE by PREOFESSIONAL_CITY and finding mode of PROFESSION_CODE for each PROFESSIONAL_CITY group and at last filling the mode into the missing values. The same operation also works for OCCUPATION_TYPE.

- Create a new category for missing values. For instance, not all customers have a mate. The missing values in MATE_PROFESSION_CODE and MATE_EDUCATION_LEVEL features are likely caused by the fact that customers do not have a mate. Thus, a new category is created to represent them.

2.3 Outlier

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. Outliers will highly affect model parameters if models are not robust to outliers.

In the data set, the numerical variables have some outliers. Figure 1 shows the distribution of 7 variables. Most of customers have 0~4 dependents. When the quantity of dependents is over 6, the ratio of data records takes only 0.2% and thus it is reasonable to exclude records where quantity of dependents is over 6. The distribution of MONTHS_IN_RESIDENCE is shift to the left and most of them are between 0 and 50. It is a judgmental decision if the scorecard model includes prediction power in customers who live in the same residence for more than 60 months. Here I simply exclude these data. Figure 1 clearly shows that there are some extreme data in PERSONAL_MONTLY_INCOME, OTHER_INCOMES and PERSONAL_ASSETS_VALUE variables. The method is just simply deleting them.

In Brazil, the minimum age to sign a legal contract is 16 years-old and thus a reasonable assumption for the AGE variable is that customers have to be at least 16 years-old to apply for credit cards. Therefore, any records with age lower than 16 and higher than 100 are deleted.

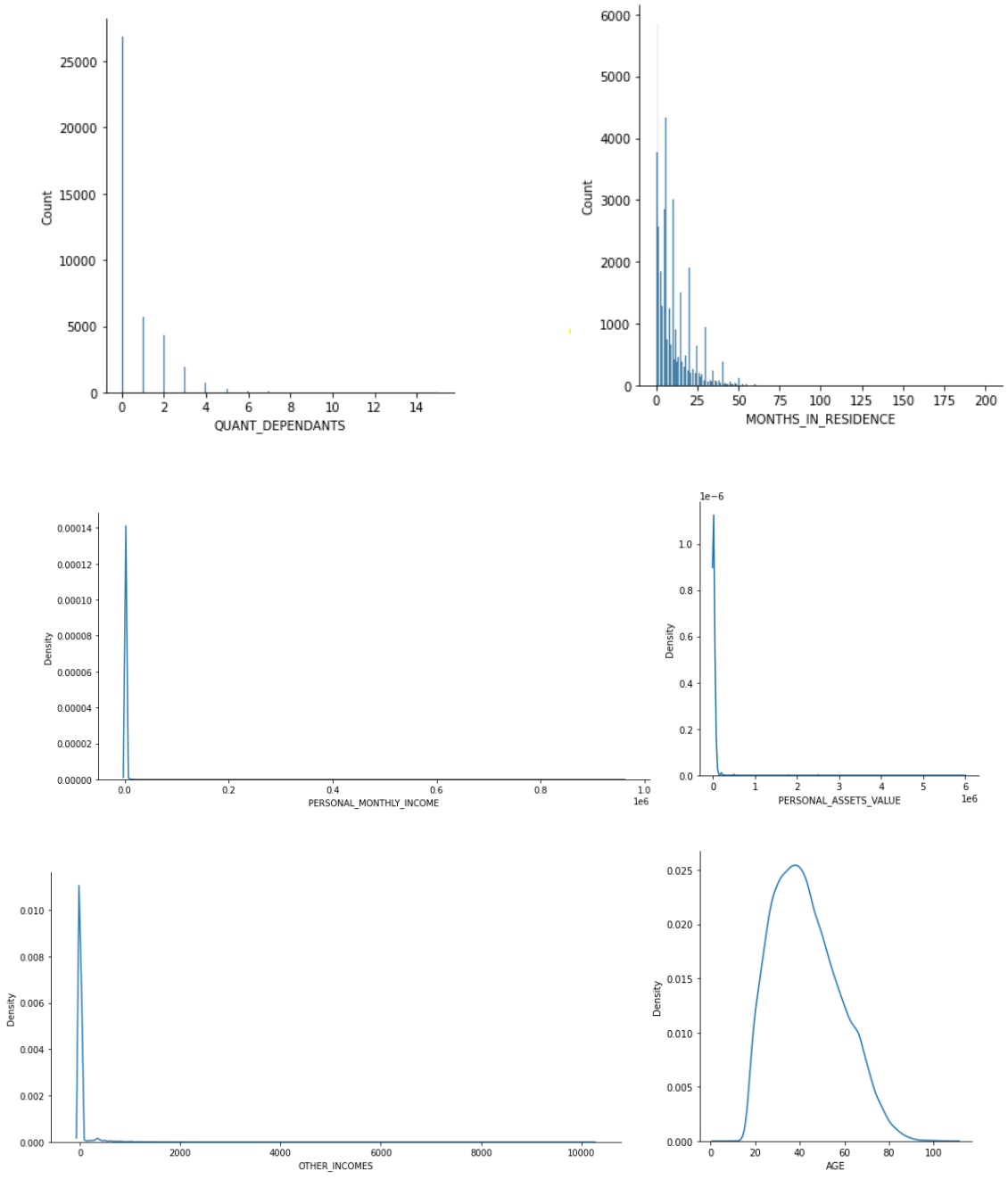


Figure 1. Distributions of seven numerical variables

2.4 Features containing one value

There are many variables with only one value, which are not useful in predicting response. Figure 2 shows some of variables containing only one single value. Thus, these variables are deleted.

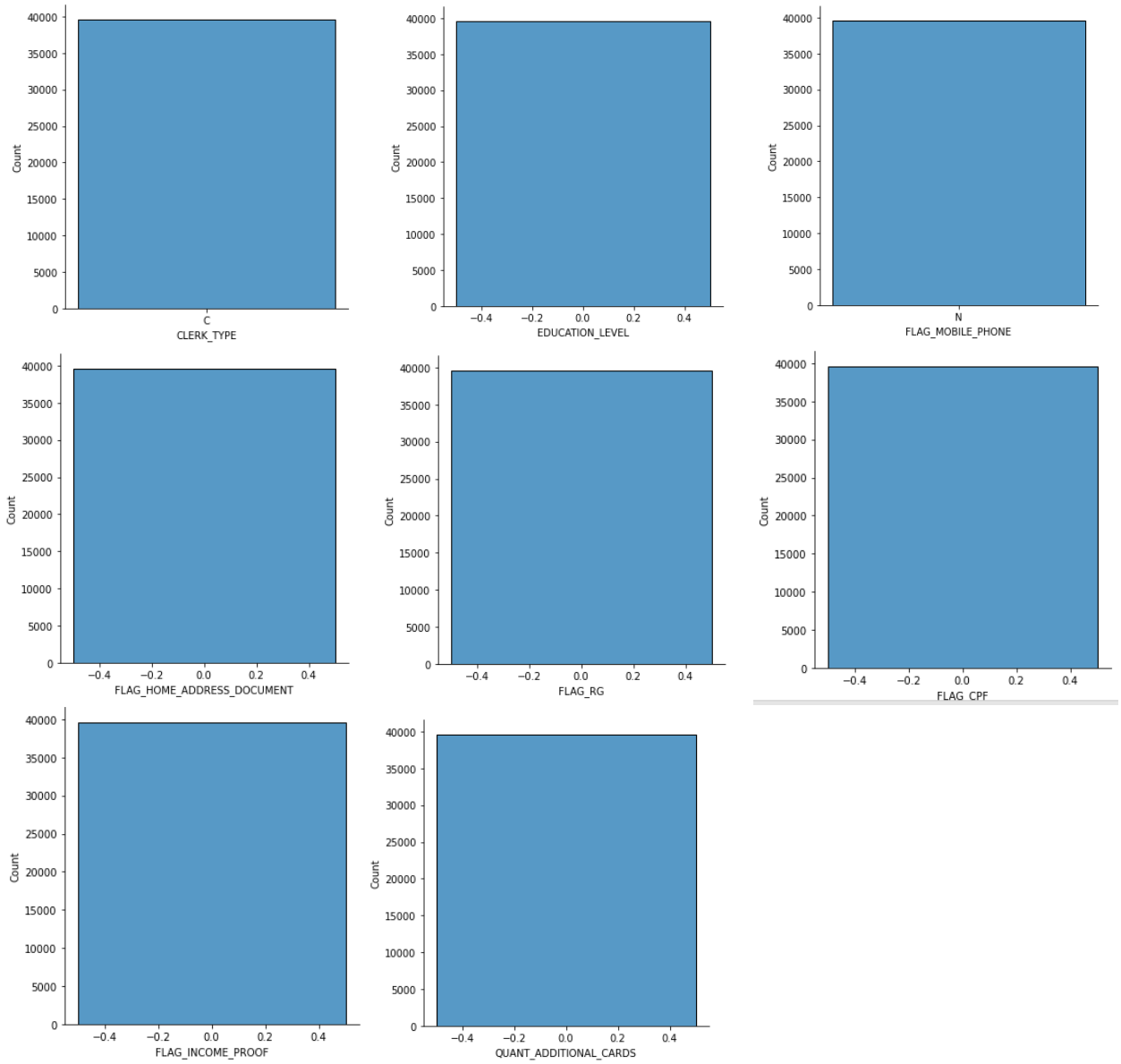


Figure 2. Features containing only one value

2.5 Others

Variable APPLICATION_SUBMISSTION_TYPE is supposed to have only two values: “Web” and “Carga”. However, it contains another value “0” shown in figure 3. Without any further information, I assume it is caused by input error and simply use the mode to replace the “0” value.

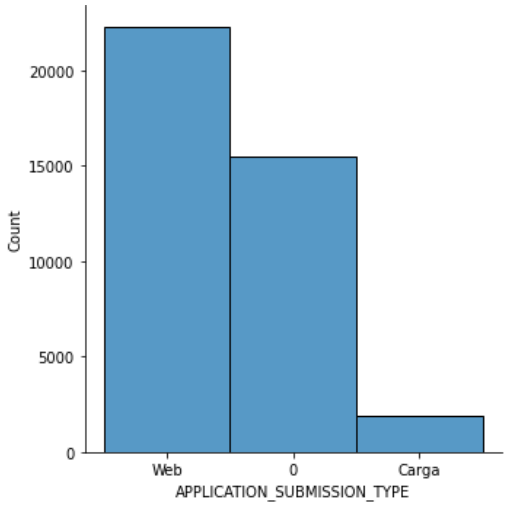


Figure 3. Feature APPLICATION_SUBMISSION_TYPE contains not only “Web” and “Carga”, but also “0”.

2.6 Normalization

Normalization refers to rescale data to bring all the values of numeric columns in the dataset to a common scale. The reason of performing normalization is that variables at different scales do not contribute equally to the analysis. Thus, usually normalization is applied to data before feeding to models. But in scorecard model, weight of evidence (WOE), which will be discussed in the next section does the normalization automatically.

Figure 4 shows the violin plot of normalized numerical variables. Most of variables look pretty good in a reasonable range and distribution. However, the distributions of PERSIONAL_MONTHLY_INCOME, OTHER_INCOME and PERSOANL_ASSETS_VALUE are right skewed with a very long right tail even after cutting off the outliers.

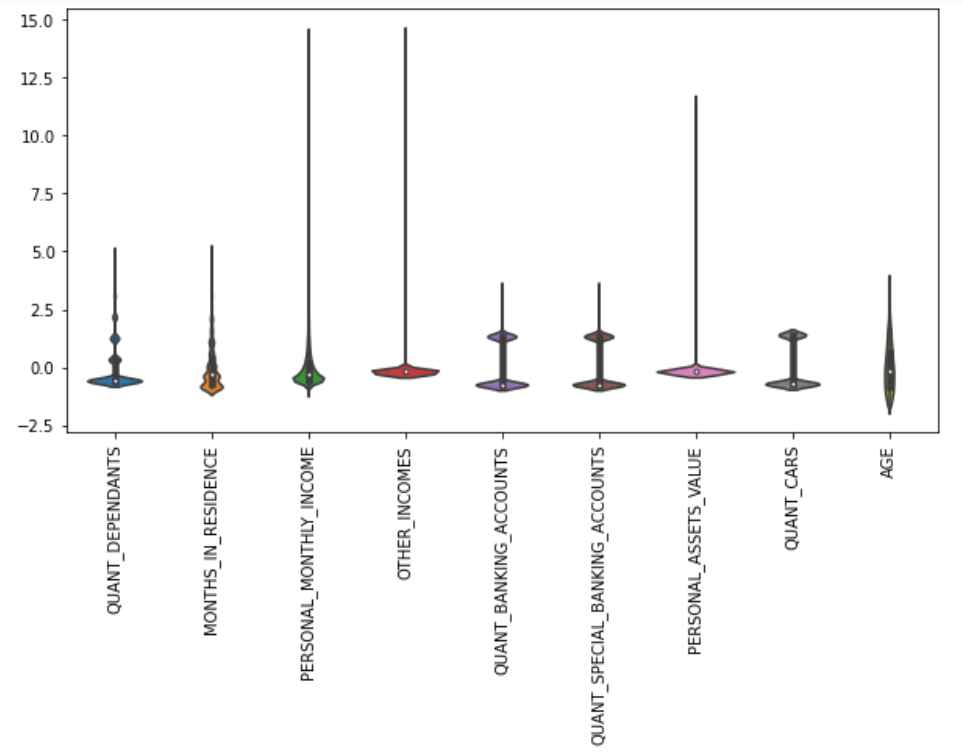


Figure 4. Violin plot of numerical variables

2.7 New Variables

Machine learning algorithm only learns from the data whatever we give, so creating features that are relevant to a task is absolutely crucial. In this report, three types of new variables are created.

- Aggregation variables: it is useful to reduce data redundancy and improve data stability. Table 2 shows three variables created by aggregation method.

Table 2 New variables created by aggregation (addition) method

New variable	Adding Method
TOTAL_INCOMES	PERSONAL_MONTHLY_INCOME+OTHER_INCOMES
FLAG_CARDS	FLAG_VISA+FLAG_MASTERCARD+ FLAG_DINERS+FLAG_AMERICAN_EXPRESS +FLAG_OTHER_CARDS
QUANT_ALL_BANKING_ACCOUNTS	QUANT_BANKING_ACCOUNTS+ QUANT_SPECIAL_BANKING_ACCOUNTS

- Average variables: the average of personal monthly income for each group may be an important indicator by utilizing two variables' information.

Table 3 New variables created by averaging method

New variable	Average method
AVG_INCOME_STATE	Get the average personal monthly income for each residential state
AVG_INCOME_OCCUPATION_TYPE	Get the average personal monthly income for each occupation type
AVG_INCOME_PROFESSION_CODE	Get the average personal monthly income for each profession code

- Ratio variable: Ratio is a good indicator of how two variables vary compared with each other. RATIO_ASSET_INCOME created by dividing PERSONAL_ASSETS_VALUE by PERSONAL_MONTHLY_INCOME.

3 Scorecard Model Development

3.1 Weight of Evidence

Weight of evidence (WOE) measures the predictive power of predictors in relation to the response variable.

$$WOE = \ln\left(\frac{\text{Distribution of Goods}}{\text{Distribution of Bads}}\right)$$

where distribution of goods is percentage of good customers and distribution of bads is percentage of bad customers (defaulters) in a particular group.

The scorecard python package (scorecardpy) automatically calculates WOE values for generated groups for each variable. However, the trending of bad probability (probability of defaulters) should make sure.

Figure 5 shows the bin count distribution and bad probability for some variables. The reasonable assumption is that if customers stay longer in the current residence, have higher personal monthly income and total income, the probability of default would decrease. However, it clearly shows waves in the first three plots. For the AGE (bottom right) plot, the bad probability decreases as the age increases, which makes sense since older customers stabilize their financial status. When the age reaches 70, the bad probability starts to increase. It is reasonable since people with age above 70 years-old have much higher mortality and thus the default numbers increase among these age range.

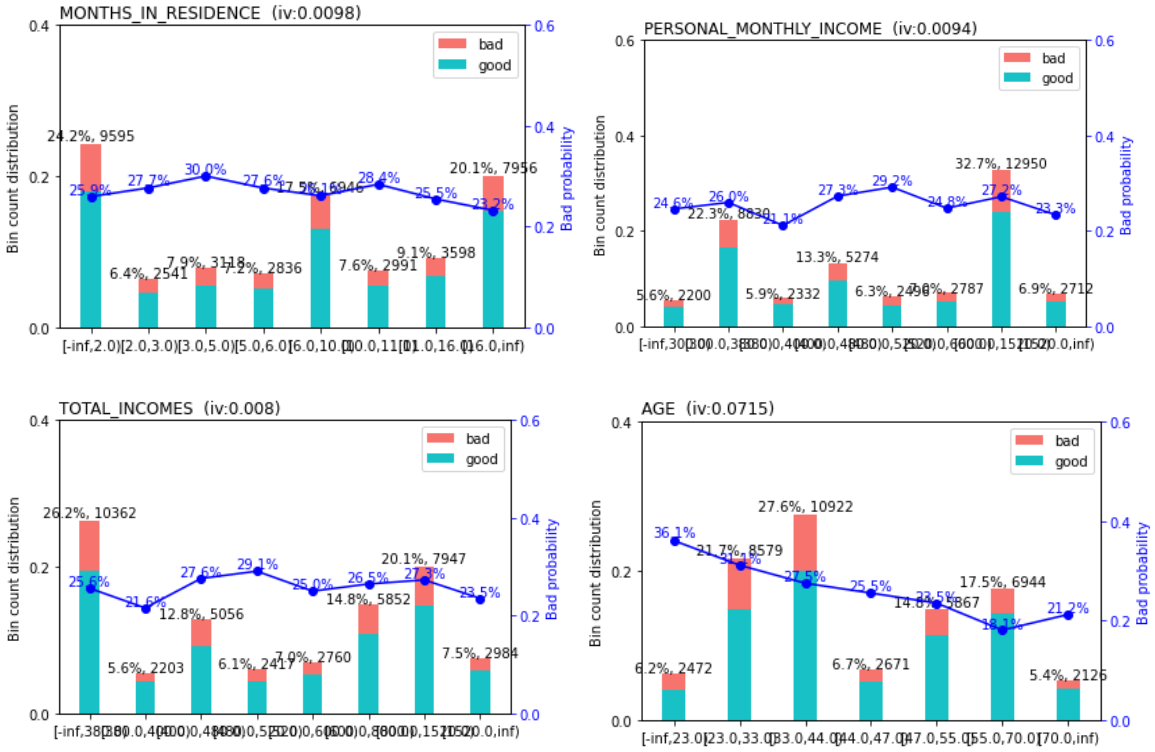
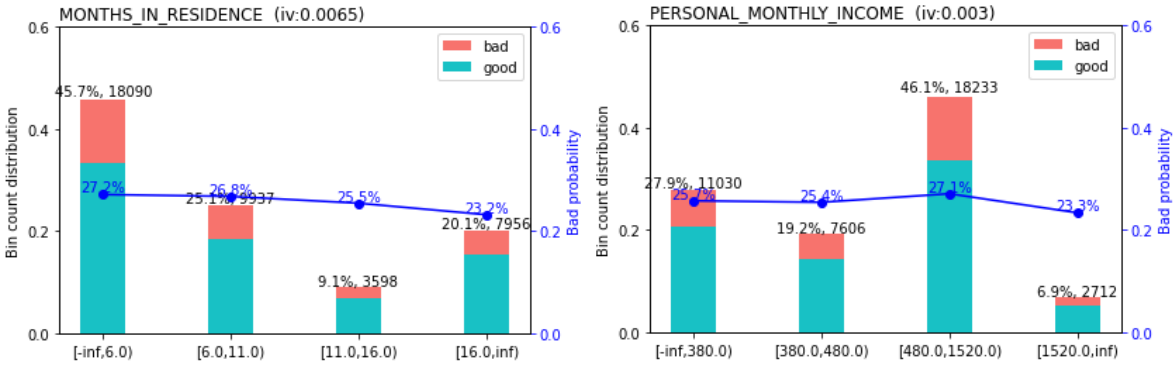


Figure 5. Bin count distribution and bad probability plots for MONTH_IN_RESIDENCE, PERSONAL_MONTHLY_INCOME, TOTAL_INCOMES and AGE variables

3.2 Manual Adjustment

Bins of variables can be adjusted using scorecard library to make the trend of bad probability reasonable. For example, for the variable MONTHS_IN_RESIDENCE, bins are adjusted to [6.0,11.0,16.0]. Bins are adjusted to [380.0, 480.0, 1520.0] for PERSONAL_MONTHLY_INCOME and [520.0, 800.0,1520.0] for TOTAL_INCOMES. For the AGE feature, the bins stay the same since the bad probability is already in a reasonable trend. Figure 6 demonstrates the plots after bins adjustment.



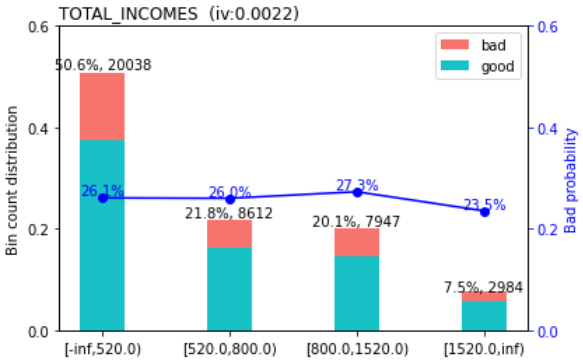


Figure 6. Bin count distribution and bad probability plots for MONTH_IN_RESIDENCE, PERSONAL_MONTHLY_INCOME, TOTAL_INCOMES and AGE variables after adjustment.

3.3 Information Value and Variable Selection

Information value (IV) helps to rank variables on the basis of their importance. The equation of calculating is as below:

$$IV = \sum\left(\left(p_{good} - p_{bad}\right) * WoE_{category}\right)$$

where p_{good} is the percentage of good customers, p_{bad} is the percentage of bad customers and $WoE_{category}$ is the weight of evidence for that particular category.

Table 4 shows the variables which have information value above 0.003. The most important variable is AGE_woe, followed by PAYMENT_DAY_woe and MARITAL_STATUS_woe. Usually, information value above 0.02 are selected to feed into models. However, considering the quality of this data set, variables with information value above 0.01 are chosen in this report.

Table 4. Variable importance shown by information value above 0.003

variable	info_value
AGE_woe	0.071529
PAYMENT_DAY_woe	0.030830
MARITAL_STATUS_woe	0.025127
OCCUPATION_TYPE_woe	0.021743
AVG_INCOME_OCCUPATION_TYPE_woe	0.021431
FLAG_RESIDENCIAL_PHONE_woe	0.016742
PROFESSIONAL_STATE_woe	0.014476
RESIDENCIAL_STATE_woe	0.013873
STATE_OF_BIRTH_woe	0.011393
AVG_INCOME_STATE_woe	0.007489
MONTHS_IN_RESIDENCE_woe	0.006477
RESIDENCE_TYPE_woe	0.005199
PROFESSION_CODE_woe	0.003106
AVG_INCOME_PROFESSION_CODE_woe	0.003105
MATE_PROFESSION_CODE_woe	0.003064
PERSONAL_MONTHLY_INCOME_woe	0.003001

Before fitting data into the logistic regression model, let's check the correlation. Correlation measures the degree to which two variables are related. High correlation reduces the precision of the estimated coefficients and statistical significance of the variables. Figure 7 illustrates the correlation heat map among selected variables with information value above 0.01.

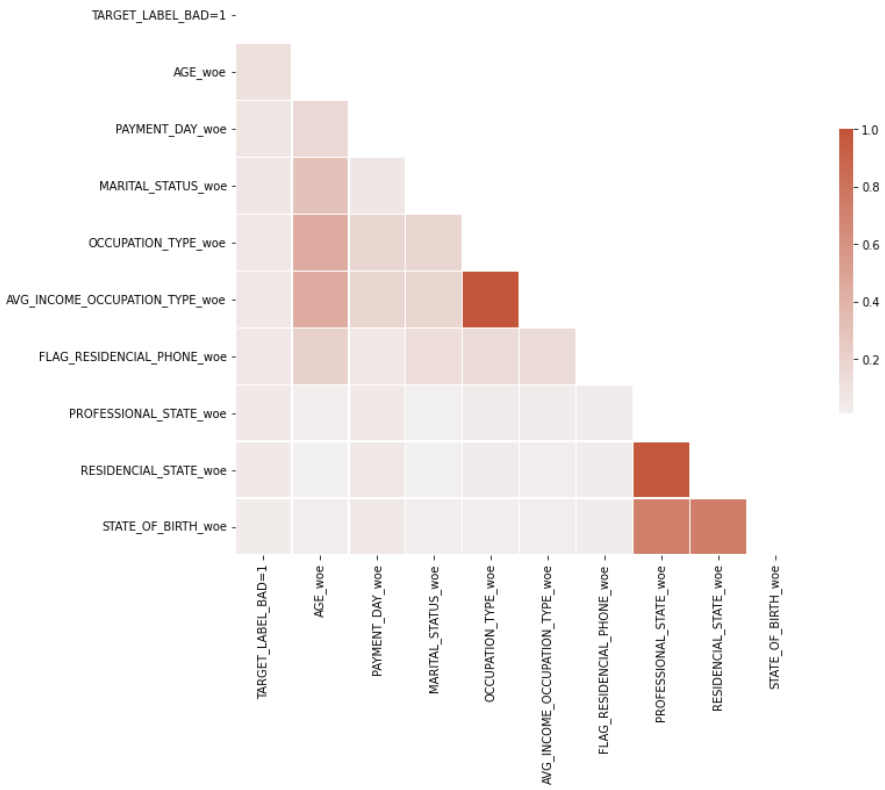


Figure 7 Correlation heat map for the selected variables (IV>0.01)

AVG_INCOME_OCCUPATION_TYPE_woe and OCCUPATION_TYPE_woe are highly correlated since the former feature is derived from the latter one. With the similar information values, AVG_INCOME_OCCUPATION_TYPE_woe is kept as it contains personal monthly income information instead of OCCUPATION_TYPE_woe. RESIDENTIAL_STATE_woe instead of PROFESSIONAL_STATE_woe is kept since the missing value in the latter one was filled by the value from the former feature. STATE_OF_BIRTH_woe is deleted since it has high correlation with PROFESSIONAL_STATE_woe and RESIDENTAIL_STATE_woe. In conclusion, 6 variables are selected.

3.4 Logistic Regression

The objective of logistic regression is to determine the probability of default for a given customer. The equation to determine the probability is as below.

$$p(y_i = 1|\mathbf{x}_i) = \frac{1}{1 + e^{-\left(\beta_0 + \sum_{j=1}^{V} \beta_j x_{ij}\right)}}$$

where β_0 is the intercept, and β_j is the regression coefficient associated to variable x_j. These parameters are unknown and need to be estimated using algorithm like maximum likelihood. In the logistic regression model, several hyperparameters are required to set. For example, tolerance

is set to 0.0001 since most of banks accept this value. Parameter class_weight is set to balance as the data is imbalanced between good customers and defaulters. The parameter cv is set to 3 for cross validation, reasonable for the size of this data set. Then the training data set is fitted into the model and at last the unseen test data is used for model performance evaluation.

Figure 8 illustrates the confusion matrix on the test data set. The performance is balanced since the type I error and type II error are almost equal and the top left and bottom right are the same, which indicates the parameters of the model are set appropriately.

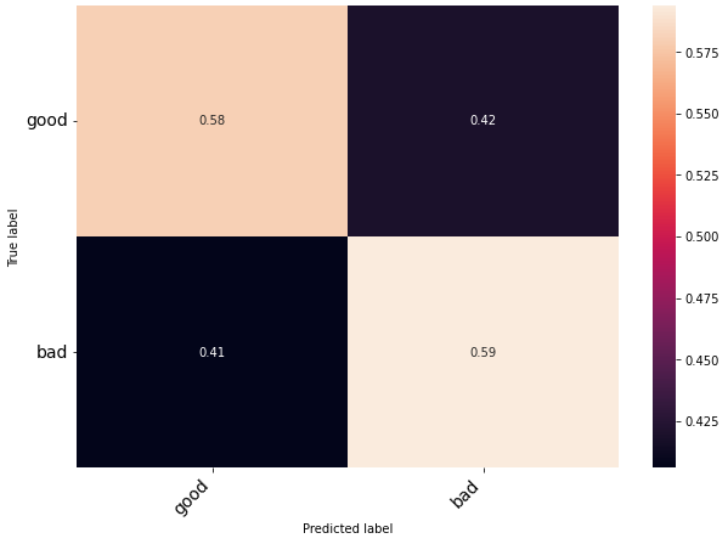


Figure 8 Confusion matrix on test data set

Another metric to measure model performance is receiver operating characteristic (ROC) curve and the AUC value (the area under the ROC curve). Figure 9 shows the ROC curve of the logistic regression and the AUC is 0.62.

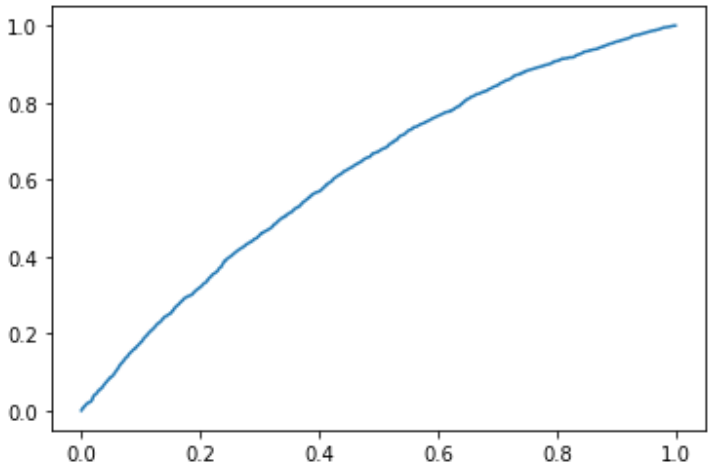


Figure 9 ROC curve of logistic regression

3.5 Scorecard

After the model training, it is ready for building the scorecard. In the scorecard model setting, the base model is logistic regression trained in section 3.4. Base points are 750, odds is 0.02 and points to double the odds (pdo) is 50. Thus, the descriptive values for the scorecard are shown in the left table in Table 5. The minimum score is 370, while the maximum score is 546.

Now a scorecard model is fully developed to predict the test (unseen) data. The result is displayed in the right table in Table 5.

Table 5. Score descriptive values for the scorecard model (left) scores for the train data (right) predicted scores for the test data

	score
count	39581.000000
mean	470.251004
std	28.483069
min	370.000000
25%	452.000000
50%	468.000000
75%	489.000000
max	546.000000

	score
count	9908.000000
mean	471.685103
std	28.772126
min	374.000000
25%	452.000000
50%	469.000000
75%	491.000000
max	546.000000

4. Random Forest

Random forest is an ensemble method to construct many single trees together and improve the prediction power. Figure 10 illustrates how random forest works. It has 9 decision trees and each single tree has their own decision. Since the prediction has label 1 six times and label 0 three times, the final decision is label 1 by taking the most frequent label.

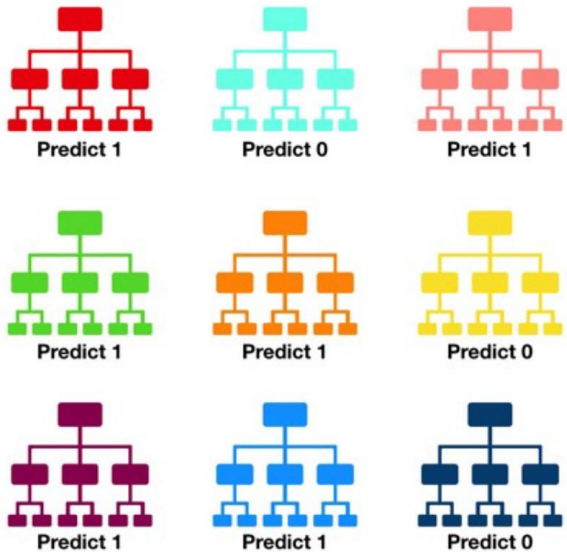


Figure 10 Illustration of random forest

In the random forest model, the optimal hyperparameters can be found by using cross validation and grid search. For example, parameter n_estimators with a range from 500 to 2500, max_features with "auto" and "sqrt" selection, max_depth with a range between 10 and 110, and min_samples_split with 2, 5, 10 were fed into the model. After the cross validation and grid search, the best hyperparamers are n_estimators as 500, max_depth as 70, max_features as "auto" and min_samples_leaf as 1.

Data used for the random forest is the data set after clean and new variable creation, but before the weight of evidence. For some variables which are numerical, for example PAYMENT_DAY, they are converted to categorical variable first and then encoded. Figure 11 shows the confusion matrix of random forest, which clearly indicates imbalanced distribution. Among good customers, 98% of them are identified as good. However, among defaulters, 96% of them are identified as good customers.

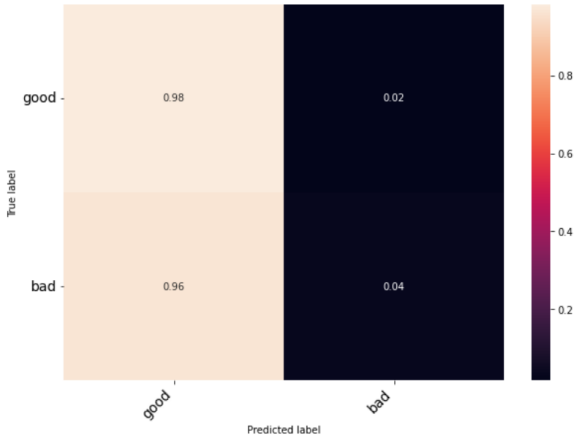


Figure 11 Confusion matrix for random forest

Another metric to evaluation the classification model is ROC curve and AUC value. Model performs better with higher AUC value. For the random forest model, the AUC is 0.62, same as the AUC for logistic regression in Section 3.

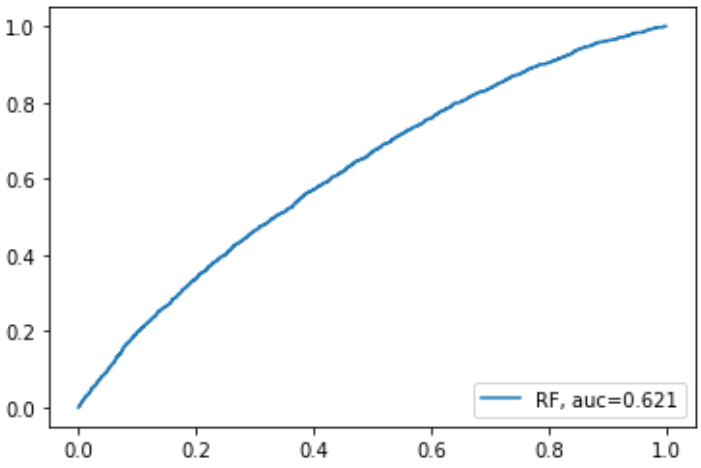


Figure 12 ROC curve for random forest

5. XGBoosting

XGBoosting is also an ensemble method by constructing many decision trees together and highly improve the predictive power. The difference from random forest is that XGBoosting utilizes boosting method instead of bagging method. Errors from the previous single tree are fed into the next small tree until error change is small.

In XGBoosting model, grid search is used to find the optimal hyperparameters as well. The optimal parameters are learning_rate as 0.1, max_depth as 3 and n_estimators as 100.

Figure 13 shows the confusion matrix for XGBoosting, which clearly illustrates an unbalanced matrix. The possible reason is the data quality issue.

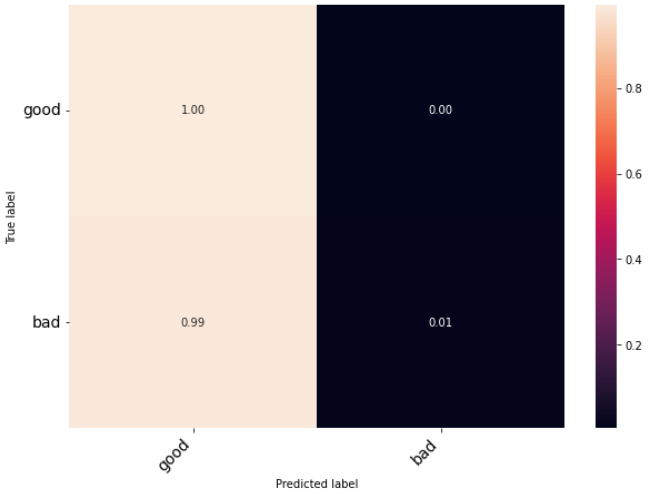


Figure 13 Confusion matrix for XGBoosting

Figure 14 is the ROC curve for XGBoosting. The AUC is 0.625, similar to the AUC for both logistic regression and random forest.

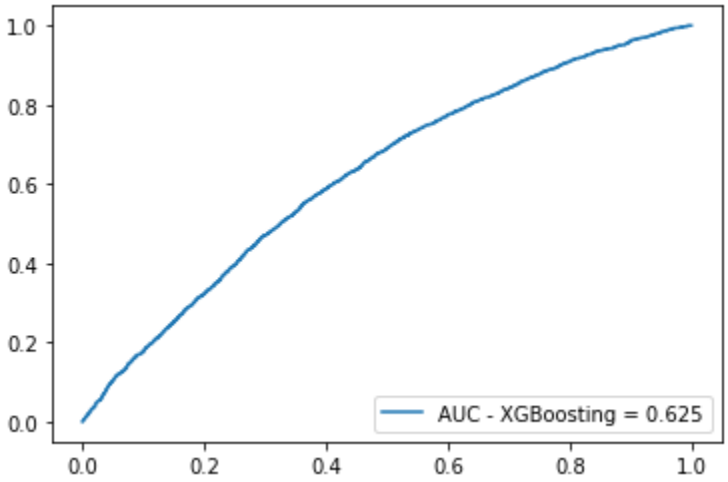


Figure 14 ROC curve for XGBoosting

6. Model Comparison

Section 3, 4 and 5 utilize logistic regression, random forest and XGBoosting respectively to predict the probability of default. The AUC values are almost the same for all three models. However, the confusion matrix is quite different. It is much better for logistic regression than random forest and XGBoosting. Thus, logistic regression is chosen in this report as the best estimator.

Figure 15, 16 and 17 are the variable importance for three models. As we can see, the variable importance for the three models is quite different. In the logistic regression, the most important variable is AGE, followed by PAYMENT_DATY and MARITAL_STATUS. However, in the random forest, the most important variable is AGE, followed by PERSONAL_MONTHLY_INCOME, TOTAL_INCOME and MONTHS_IN_RESIDENCE. PAYMENT_DAY and MARITAL_STATUS variables are not even in the top 10 most important variables in random forest. The three most important variables in logistic regression show up again as the top 8 most important variables in XGBoosing. However, information about residential and professional phone become significant variables for prediction in XGBoosting. Variables such as PERSONAL_MONTHLY_INCOME and AVG_INCOME that are not statistically significant in logistic regression become significant in random forest.

variable	info_value
AGE_woe	0.071529
PAYMENT_DAY_woe	0.030830
MARITAL_STATUS_woe	0.025127
OCCUPATION_TYPE_woe	0.021743
AVG_INCOME_OCCUPATION_TYPE_woe	0.021431
FLAG_RESIDENCIAL_PHONE_woe	0.016742
PROFESSIONAL_STATE_woe	0.014476
RESIDENCIAL_STATE_woe	0.013873
STATE_OF_BIRTH_woe	0.011393
AVG_INCOME_STATE_woe	0.007489
MONTHS_IN_RESIDENCE_woe	0.006477
RESIDENCE_TYPE_woe	0.005199
PROFESSION_CODE_woe	0.003106
AVG_INCOME_PROFESSION_CODE_woe	0.003105
MATE_PROFESSION_CODE_woe	0.003064
PERSONAL_MONTHLY_INCOME_woe	0.003001

Figure 15 Variable importance for logistic regression

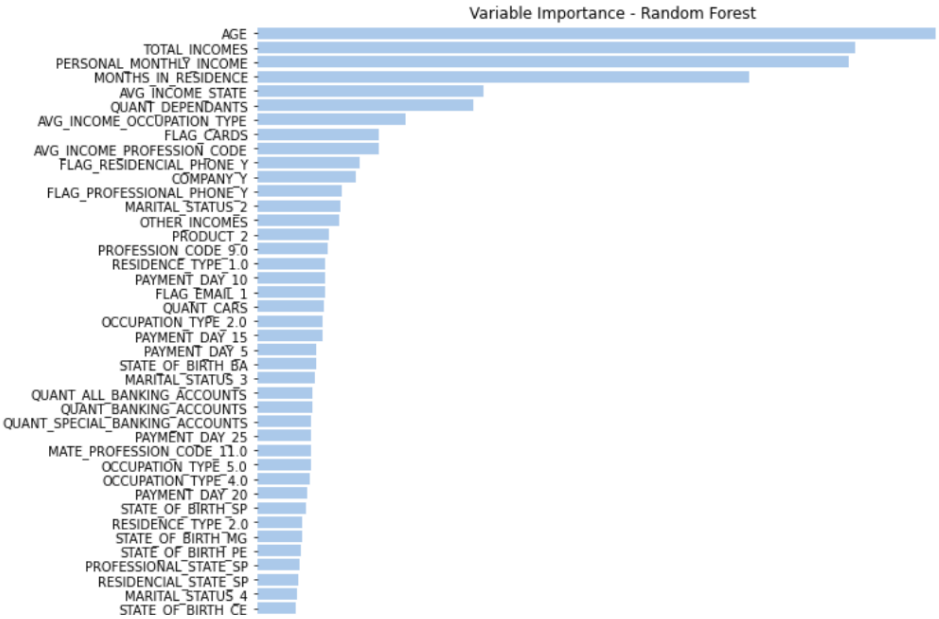


Figure 16 Variable importance for random forest

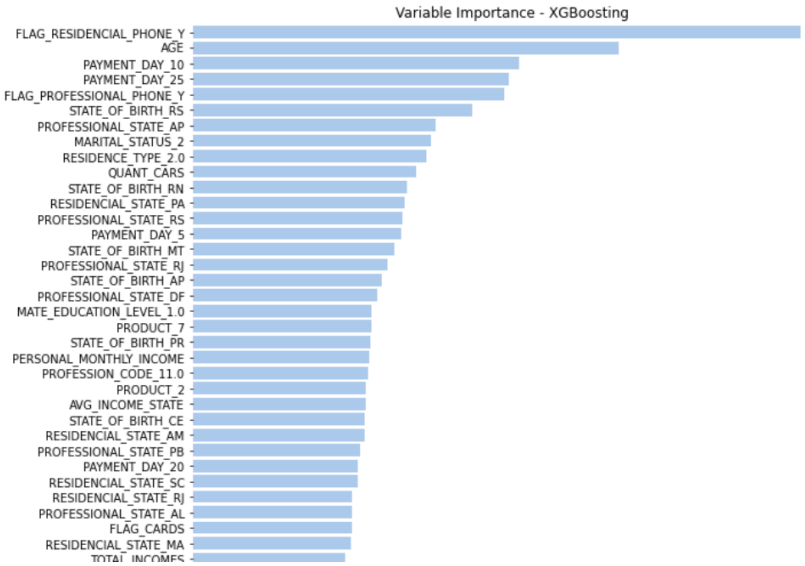


Figure 17 Variable importance for XGBoosting

The main possible reason for the difference in variable importance is that model mechanism is different. The assumption of logistic regress in linear relation between predictors and response is using logit function and cutoff point to determine the probability of default. However, the base model inside the random forest and XGBoosting is decision tree. The splitting decision in the decision tree is entropy, which tries to minimize the entropy after splitting.

7. Two-cut-off Point Strategy

The methodology to determine a cut-off point starts by calculating the cost of accepting a defaulter using the expected loss and the cost of rejecting a good applicant. In this report, cutoff points are selected from 0.4 to 0.9 with a step of 0.05. In this report, the definitions of several calculations are as below:

$$Accpeted_{percentage} = \frac{N(\hat{y} = 0)}{Ntotal}$$

where $N(\hat{y} = 0)$ is the number of predicted good customers, Ntotal is the total number of customers.

$$accuracy\ of\ goods = \frac{N(\hat{y} = 0|y = 0)}{N(y = 0)}$$

where $N(y = 0)$ is the number of true good, $N(\hat{y} = 0|y = 0)$ is the number of predicted good among the true good.

$$accuracy\ of\ bads = \frac{N(\hat{y} = 1|y = 1)}{N(y = 1)}$$

where $N(y = 1)$ is the number of true bad customers, $N(\hat{y} = 1|y = 1)$ is the number of predicted bad among the true bad customers.

$$accuracy\ of\ total = \frac{N(\hat{y} = 0|y = 0) + N(\hat{y} = 1|y = 1)}{Ntotal}$$

where accuracy of total is adding the number of predicted good among the true good and the number of predicted bad among the true bad customers together and then divided by the total number of customers.

$$total\ good\ cost = \sum cost\ for\ each\ rejected\ good\ customer$$

where total good cost is defined by summing the cost for each rejected good customers. The cost of the rejected good customer is calculated by multiplying this customer's monthly income (approved limit) with 0.32 (average utilization of the approved limit) and then 0.2 (interest rate).

$$total\ bad\ cost = \sum cost\ for\ each\ accepted\ bad\ customer$$

where total bad cost is defined by summing the cost for each accepted bad customer. The cost of the accepted bad customer is calculated by multiplying this customer's monthly income with 0.32.

Total cost is simply adding total good cost and total bad cost together. Average good cost is by dividing total good cost with the number of true good customers. Similarly, average bad cost is by dividing total bad cost with the number of true bad customers. Table 6 demonstrates these values for different cutoff points. As we can see, with the increase of cutoff point, accepted percentage and accuracy of good increase, while the accuracy of bad decreases. The total good cost decreases for higher cutoff point, while the total bad cost increases.

Table 6 Cutoff Point Tables

cutoff	accepted_percentage	accuracy_good	accuracy_bad	accuracy_total	avg_good_cost	avg_bad_cost	total_good_cost	total_bad_cost	total_cost
0.40	17.692832	20.567618	90.441176	38.814077	35.089299	23.758867	1.026187e+06	2.455717e+05	1.271758e+06
0.45	31.366059	35.223115	79.547214	46.797706	27.574702	50.718034	8.064222e+05	5.242216e+05	1.330644e+06
0.50	51.403451	55.558215	60.352167	56.810086	17.583334	96.172483	5.142246e+05	9.940388e+05	1.508263e+06
0.55	71.913292	75.113695	37.142028	65.197949	9.112124	148.054920	2.664841e+05	1.530296e+06	1.796780e+06
0.60	87.590005	89.314413	17.289087	70.506051	3.733393	188.789123	1.091831e+05	1.951324e+06	2.060507e+06
0.65	96.048609	96.645580	5.640480	72.880928	1.156036	211.067467	3.380826e+04	2.181593e+06	2.215402e+06
0.70	99.267325	99.408446	1.131966	73.744979	0.181097	219.496239	5.296185e+03	2.268713e+06	2.274009e+06
0.75	99.911574	99.924773	0.125774	73.863722	0.020782	221.226591	6.077722e+02	2.286598e+06	2.287206e+06
0.80	100.000000	100.000000	0.000000	73.886461	0.000000	221.468046	0.000000e+00	2.289094e+06	2.289094e+06
0.85	100.000000	100.000000	0.000000	73.886461	0.000000	221.468046	0.000000e+00	2.289094e+06	2.289094e+06
0.90	100.000000	100.000000	0.000000	73.886461	0.000000	221.468046	0.000000e+00	2.289094e+06	2.289094e+06

Figure 11 shows the defaulter accuracy and total cost as a function of cutoff. It clearly illustrates the defaulter accuracy decreases and the total cost increases with the increment of cutoff point.

Thus, a two-cutoff point strategy is developed to help bank minimize the cost. One cutoff point is 0.5 with high defaulter accuracy and low total cost. Any customer with probability of default below 0.5 is accepted immediately. The other cutoff point is 0.6 because at this point, the total cost drastically increasing while defaulter accuracy sharply decreasing. Any customer with probability of default above 0.6 will be rejected immediately. For the rest of customers with probability of default between 0.5 and 0.6, other further method or expertise from the bank professionals will be used to make a decision.

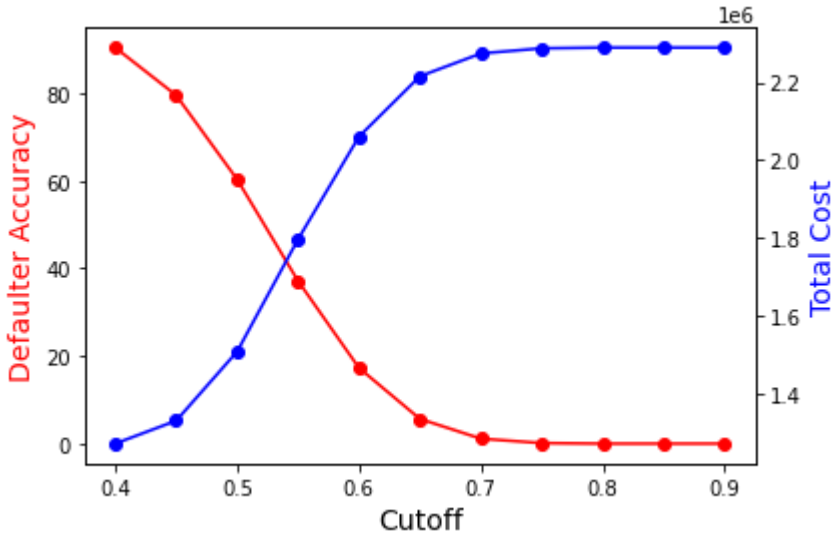


Figure Defaulter accuracy and total cost for each cutoff value.