Default Analysis for Home Credit Group

Cohort A Team 5: Weifu Shi, Xiaorui Shen, Sizhe Fan, Hang Zhang, Yinghui Wei

Repository link: *https://github.com/leo950811/ba888-team5.git*

Problem statement, Setting and Significance

The main purpose of our project is to help HomeCredit Group to better evaluate the paying capability of customers who have little or no credit history. There are two main goals for our project: the first one is to build a supervised machine learning model to predict whether customers have difficulties repaying the loans. In addition, we will try to calculate the possibility of loan defaulting of each applicant..The second goal is to use unsupervised machine learning to cluster these customers into different groups. By studying the characteristics of each group, HomeCredit group can provide different levels of management to each group such as supervision level, strategy adjustment and alarm calls.

Setting

After the financial crisis occurred in 2008, there are correspondent regulatory agreements and requirements specifically designed for the banking industry in order to reduce the ratios of non-performing loans such as Basel III, Dodd-Frank Wall Street Reform and Consumer Protection Act in 2010 . Since 2008, with the quantitative easing policy and strict banking regulatory policy, the whole capital market seems to favor the middle class customers and upper class customers who normally have relatively well credit history or equivalent collaterals. In contrast, people without credit histories or no collaterals are being less appreciated than the former group. Or in most cases, this group of people are often taken advantage of by untrustworthy lenders who may charge ridiculous high interest rates or implement misleading policies of ripping off the wealth of the customers. However, the mission statement of HomeCredit group is to build a responsible lending model to empower underserved customers with little or no credit history to access financing. For our group, we will use various statistical and machine models to design a better prediction model for the company.

Significance

It is extremely important to make sure the efficiency and effectiveness of the lending model. Our group has decided to use scientific machine learning models to predict the paying capability of these customers. We will also ensure clients who are capable of repayment are not getting rejected. For the potential customers, they can use these models to evaluate their own repaying ability before applying for the loans and to understand some of the importance of certain attributes. In addition, the design of our machine learning model can be a reference or case study for other financial institutions and academic institutions to have a glimpse of the credit default algorithms and banking industry.

A brief summary of the data

The dataset we are using is the Home Credit Default Risk retrieved from the Kaggle website (https://www.kaggle.com/c/home-credit-default-risk/data). The dataset contains demographic information of customers such as birthday, gender, number of children, education, income sources, family status, and housing info.

The dataset was updated two years ago and had high-quality accuracy, timeliness, and completeness. The datasets include seven different tables. But right now, we only use one file: application_{train}.csv. This is the main table we are going to use. The training dataset consists of 122 columns with 307512 observations. The dataset includes customer's basic application info such as family status, income sources, education level, and loan type.

The target variable measures the ability of customers to repay the loan. Number 1 means that customers have difficulties in paying, and 0 means customers not. In the training dataset, most customers have no difficulties.

The dataset has two different loan types: revolving loans and cash loans. Revolving loans means that "Arrangement which allows for the loan amount to be withdrawn, repaid, and redrawn again in any manner and any number of times until the arrangement expires." Most loans are cash loans. The purpose of the loan is to own a car or own reality.

The main income sources of applicants are from working salary, and 63.9 % of applicants are married. The top occupation of applicants is laborers, and 71% of the applicant's education level is secondary, which means that customers may be low-income and less educated.

Exploratory data analysis - EDA

Knowing that there are 122 columns and 307511 rows, we decide to figure out which data points should be removed first. We had a glance at the data using the `skim` function to see how our variables look like. There are 16 character variables and 106 numeric variables.

Basically, there are three kinds of variables in the dataset, that are continuous, binary, and categorical. For numeric data, when we take a look at the distribution of our variables, most of them are not normalized instead they are right or left-skewed. Besides, because there are many variables, their units are different. Even though variables have the same unit, numbers may have a big difference. Then, it is necessary to make data in normalization before we put them in any model to avoid that models learn bias from variables. We would apply box-cox to transform non-normalized data to change the value of these numeric columns set to a common scale without changing the difference in the value range.

Here we move to data cleaning. Starting with the numerical ones, we first remove the variables that have insignificant variance, because it won't make sense if the column doesn't have enough difference values. Our standard in this dataset is if the variance of the column is less than 0.01, we remove it. After this first step, we have 65 variables left.

Second, we found out that there are variables that have too many missing values. Among the 65 variables, we noticed that 20 variables have more than half values are missing. There are limited ways to deal with missing values: 1. Impute with means or regression predictions; 2. Replace with 0, -999 or -9999 ; 3. Delete the columns that have too many missing values. After researching and discussing, we think that impute with means could be a biased decision because there are huge tails in our data, for example, there are more than 9000 people whose lowest annual income is still more than 100 billion dollars. Imputation with predictions of regression would cause a vital problem of creating multicollinearity among the variables. Replacing missing value with -999 and -9999 is not a familiar approach, so we did some research on that. We found several articles stating that replacing with -999 or -9999 was a good approach in the old days when software could not recognize and deal with "missing" values. It was not correct to treat missing values as 0, so they encoded missing values with "obviously invalid" numbers like -9999 to let the computer recognize the abnormality of the data. But nowadays, it doesn't make sense to replace missing values with "obviously invalid" numbers because those don't exist, and computer software knows what are missing values. Therefore treating the missing value as 0 would also create a huge bias.

We also found reliable resources saying that it is important to identify if the data is structurally missing or logically missing. So we dig into the description of those variables that have too many missing values, most of them don't have a persuasive description that tells us to keep them. After removing those variables, we have 49 numeric variables left in total. However, some of them could be recorded or combined in further analysis, for example, we merged `FLAG_DOCUMENTS` 2-21, because each flag documents represent whether the applicant submit the specific document (0 or 1), but we don't know the specific name for each document, so we summed them up to recode them into "how many documents were submitted". Yet, this approach is not well supported either, so we finally decide to use a random forest to decide what are the important documents and we are going to only keep those.

Third, we also decide to remove the highly correlated variables, not removing pairwise but only one of the highly correlated pairs. Here our criterion is if the correlation is bigger than 0.8 we remove them. The reason is that the linear models have the assumption that the predictors should be independent. Also, we want to reduce the dimension first because, 1. The decrease in computational time and complexity; 2. Removing highly correlated variables with the same underlying information; 3. Removing predictors with degenerate distributions.

Above are the discussions about numeric columns. There are 13 character variables left and their missing values are less than 50%. Then, we have further actions about two categorical variables in these left ones. We decided to delete EMERGENCYSTATE_MODE since it doesn't have a detailed explanation about this variable, it would be hard to give impute NAs with reasonable data and its missing value is almost 50%. We also deleted NAME_TYPE_SUITE that means who accompanies a customer when applying for a loan, but the data includes two confusing categories - OTHER_A and OTHER_B that doesn't provide more details.

Our plan to tackle the problem

For this project, our project is to predict the loan repaying capability of each applicant and use 0/1 to show the result. So our plan is to build a model to predict 0/1, use our existing data to train our model, apply the trained model to cases that may be encountered in the future, and help people make decisions.

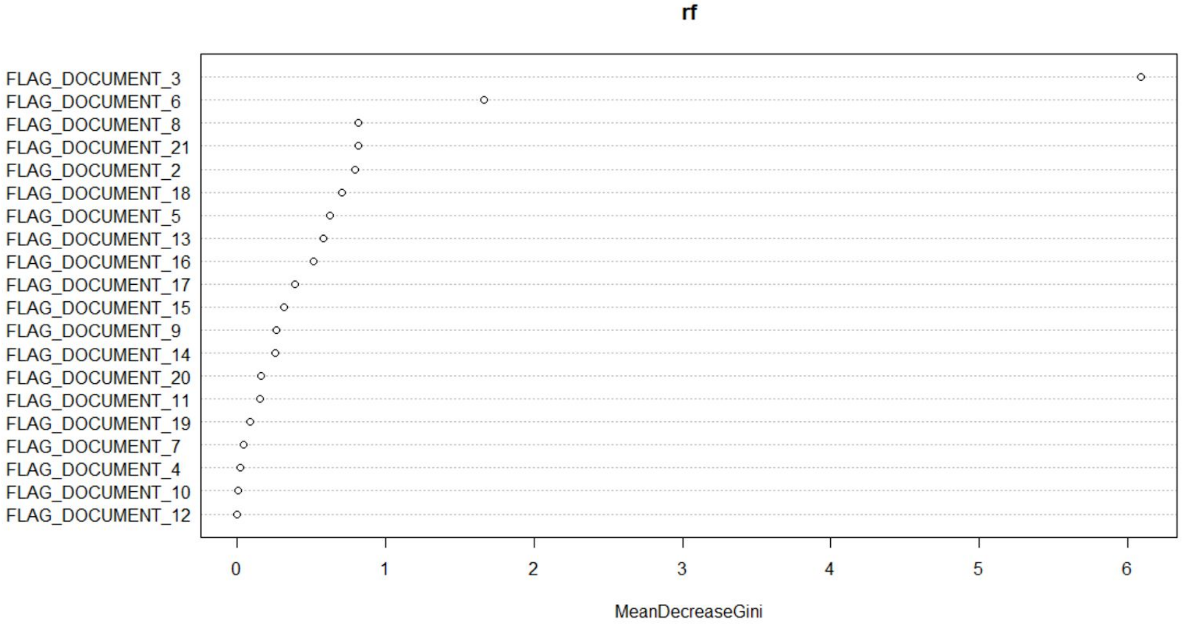
In addition to using models to predict results and help people make decisions, we also hope that our model will enable people to understand why their loan application was not approved. This requires our model to be interpretable. So in the process of choosing a model, we will also consider this appropriately.

Considering that many variables are covered in our data -- even after data cleaning, we still have more than 50 variables. We worried including all the variables in the model would bring too much noise to our prediction. So before we use the whole data to build our final model, we will use some way to pick up some of those variables to improve our model's accuracy and efficiency.

For example, when our clients apply for the loan, our bank will require them to provide 22 documents. So that in our dataset, we have 22 variables that describe those documents. If the customer provided the document, we recorded it as 1, otherwise, the record will be 0. Because there is a lot of data missing in some document records, and we don't want to use 22 variables in the model to discuss the impact of submitting documents on the results. So we plan to sort out some of this data. Unfortunately, we only know the number of the document but don't know the content of each document behind the number. We were very distressed by this problem.

At the very beginning, we plan to simply add them all. In other words, we will see how many documents they can submit to us. But we quickly realized it was bad. Some of these 22 documents must be more important than others. Simple addition and counting totals will reduce the impact of these documents. Therefore, we try to use models to analyze the importance of documents.

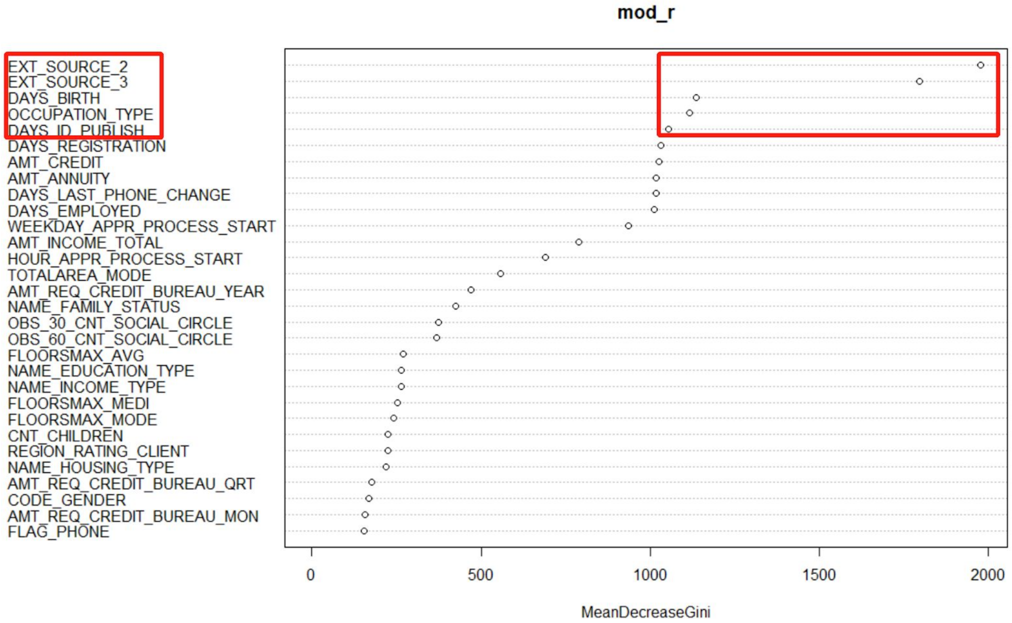
Because Random Forest can handle nonlinear parameters efficiently and provide a reliable feature importance estimate. We finally decided to use it to help us select the document and find that document 3 has the greatest impact on the results. So we will keep D3 for our further steps.



Supervised machine learning algorithm (Credit default Analysis)

Random Forest

Using random forest to do the classification of applicants, which is an ensemble tree-based learning algorithm aggregates the votes from different decision trees to decide the final class of the data object. We splitted the applicant data into a train dataset with 260000 observations and a test dataset with 47511 observations. Random forest in R can't allocate too large vectors at once and our data has the extreme imbalance problem of independent variables - TARGET. Because of these two points, we decided to try to balance the class manually by extracting all applicants who have payment difficulty from the train and randomly sampling the same rows of applicants who can make payments on time. Then our new train had a balanced class distribution. After training the model, we got the error of class 0 (applicants who don't have payment difficulty) is 34% and the other class error is 31%. Then we applied test data into the model to see the performance of the new dataset. To understand the generalization of capability of the model, we need to measure it by some metric. In general, we can naturally think of accuracy, which is defined as the percentage of total samples that predicted the correct result. The test accuracy is about 66% from the model, which is not very good. Although the accuracy can judge the total accuracy, it can't be used as a good indicator to measure the results in the case of unbalanced samples. Because of this, we would pay more attention to precision and recall rate. Precision refers to the prediction result. It means the probability of the sample that is actually positive among all the samples that are predicted to be positive. Our precision rate is about 16%, which means there is only a small part of predictions where clients have real payment difficulty. Recall refers to the original sample, and its meaning is the probability of being predicted to be a positive sample in the sample that is actually positive. And our recall rate is 70%. We care more about bad or difficult applicants. Because if we mistake bad applicants for good ones too much, the amount of default may occur in the future will far exceed the amount of loan interest repaid by good applicants, resulting in serious losses. The higher the recall rate, the higher the probability that the actual bad applicants will be predicted.



From this model, we can see the rank of features importance. The first two variables are normalized scores from external data sources and there is no further detailed explanation, but in terms of how important they are, it may be an overall score that combines different attributes. Some other important features are applicants' age and occupation type. DAYS_IN_PUBLISH means how many days before the application did the client change the identity document with which he applied for the loan (time only relative to the application). So we may have the insight that the top features of clients or loans have more weight on identifying if the client would have payment difficulty in the future.

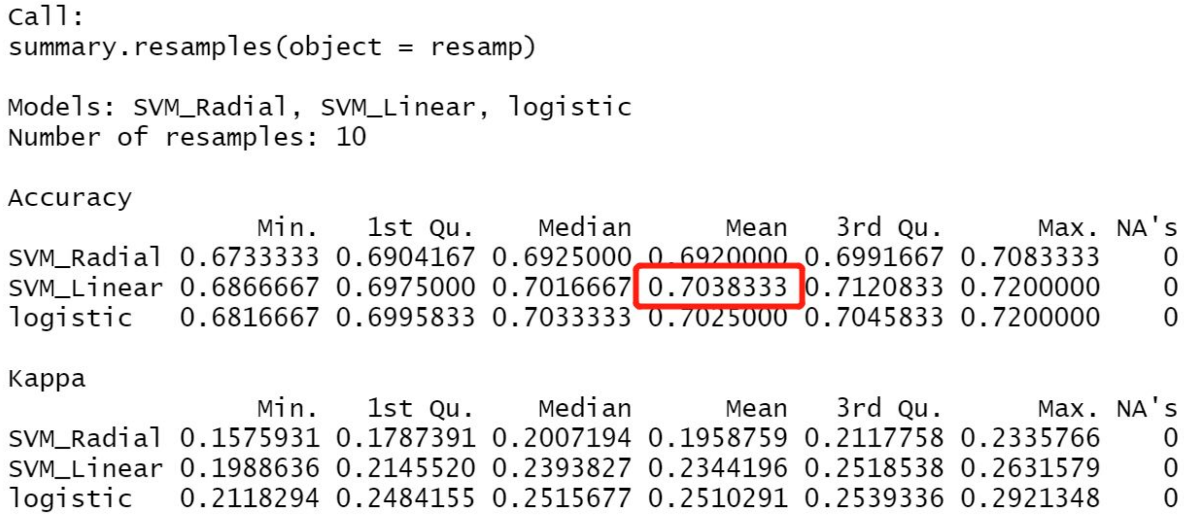
However such a low accuracy would indicate some problems, the main reason that caused the accuracy is there is a fair amount of observations in class 0 that are predicted as class 1. So, test specificity, ability of the test to correctly identify those without the payment difficulty, is pretty low about 66%.

Logistic regression vs Support Vector Machine

Typical classification models are logistic regression and SVM, so we are going to compare both these methods to get better prediction.

Logistic regression basically uses sigmoid function to find relationships between variables. When variables are fairly independent, logistic regression could be a nice choice to solve classification problems. We removed the highly correlated data in the data pre-processing, so using logistic regression with all the variables we have would be fine.

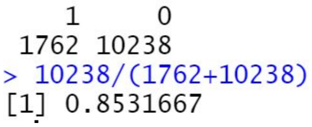
Support Vector Machine (SVM) is a supervised machine learning method that builds a non-probabilistic model that assigns new examples to one category or the other. It can efficiently perform a non-linear classification implicitly mapping their inputs into high-dimensional feature spaces. There are different kernels built-in the SVM in R, here we considered the radial basis and the linear kernel, which are known to be effective enough to solve predictive problems. In the SVM models, we used 5-fold cross validation and repeated the process twice for both linear and radial kernels. Below are the comparisons:



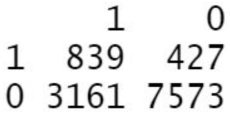
```
Call:
summary.resamples(object = resamp)

Models: SVM_Radial, SVM_Linear, logistic
Number of resamples: 10

Accuracy
                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
SVM_Radial 0.6733333 0.6904167 0.6925000 0.6920000 0.6991667 0.7083333    0
SVM_Linear 0.6866667 0.6975000 0.7016667 0.7038333 0.7120833 0.7200000    0
logistic   0.6816667 0.6995833 0.7033333 0.7025000 0.7045833 0.7200000    0

Kappa
                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
SVM_Radial 0.1575931 0.1787391 0.2007194 0.1958759 0.2117758 0.2335766    0
SVM_Linear 0.1988636 0.2145520 0.2393827 0.2344196 0.2518538 0.2631579    0
logistic   0.2118294 0.2484155 0.2515677 0.2510291 0.2539336 0.2921348    0
```

From the comparison, we can see that among the three classification and regression models, the accuracy of the linear kernel SVM is the highest (70.4%). Although the kappa statistic is not the highest, three of them are in the same range level (0.2), so we decided to stick with the highest accuracy.

The result of the linear kernel SVM follows below:



```
    1     0
 1762 10238
> 10238/(1762+10238)
[1] 0.8531667
```

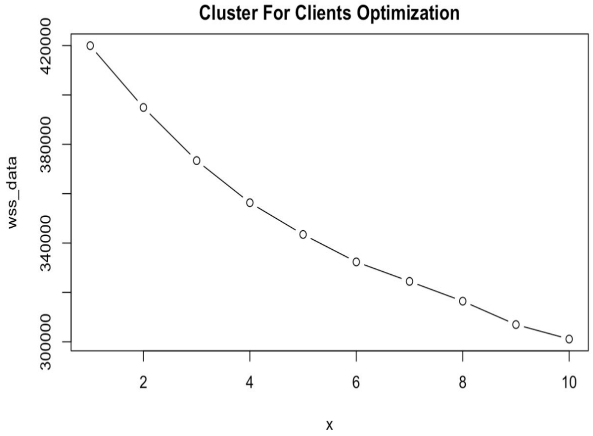


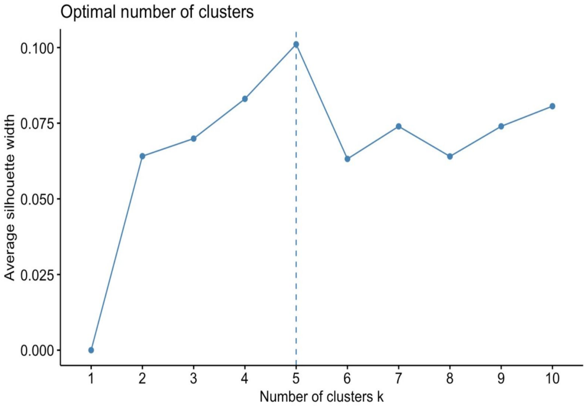
```
     1    0
1  839  427
0 3161 7573
```

This means that it predicts 85% of the test data are 0 and 15% of the rest data are 1, with 70% accuracy. The sensitivity of this model is 0.21, and the specificity is 0.95.

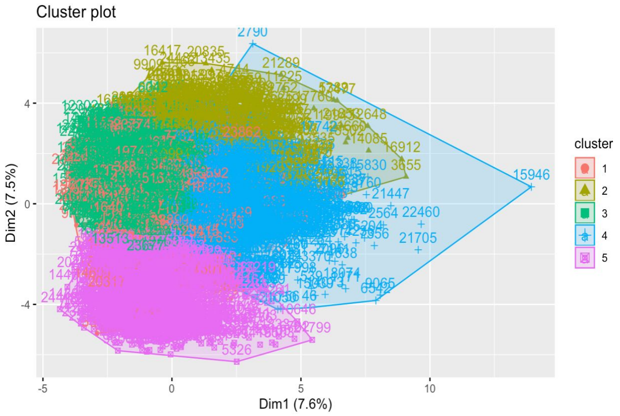
Unsupervised Machine Learning Machine Algorithm

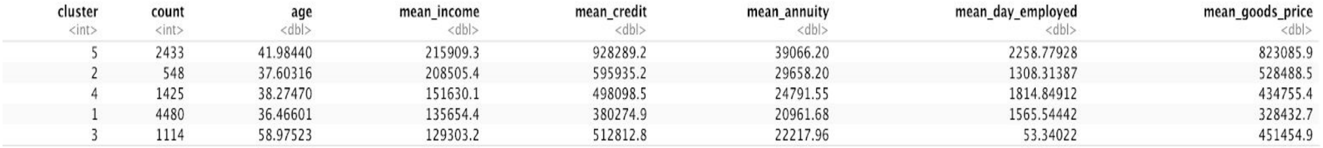
Default Customer Segmentation





We have randomly sampled 10,000 customers and implemented K-Means unsupervised ML algorithms. In terms of Silhouette and WSS methods, the optimal number of clusters would be five. We have tried 1-10 numbers of clusters for these 10,000 clients and looked at the summary result. Eventually we have reached a consensus that five clusters would be the most reasonable cluster for these ten thousand clients

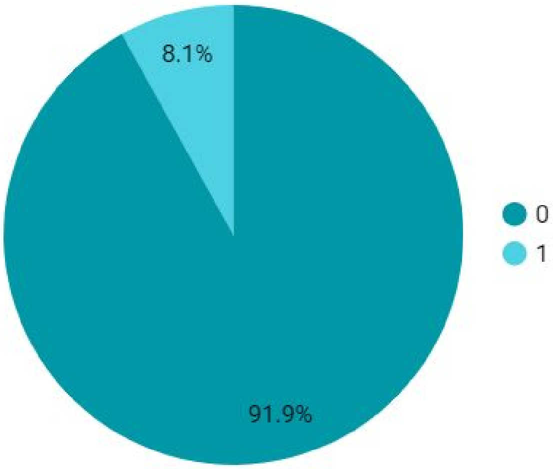




cluster	count	age	mean_income	mean_credit	mean_annuity	mean_day_employed	mean_goods_price
<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
5	2433	41.98440	215909.3	928289.2	39066.20	2258.77928	823085.9
2	548	37.60316	208505.4	595935.2	29658.20	1308.31387	528488.5
4	1425	38.27470	151630.1	498098.5	24791.55	1814.84912	434755.4
1	4480	36.46601	135654.4	380274.9	20961.68	1565.54442	328432.7
3	1114	58.97523	129303.2	512812.8	22217.96	53.34022	451454.9

As shown above, HomeCredit Group should set different levels of alarm call.For example, We would suggest the management team of Home Credit group to be extremely cautious about the clients in cluster 3. This is because Cluster 3 is considered as a high risk group and should be alerted and immediately: relatively elder age, low income and low employment day. Their risk of defaulting is higher than the other clusters. Other groups might have to postpone/delay their payments at a later date considering that they are still able to repay the loan. However, for cluster 3, they have a relatively serious problem of repaying the loans that they are not able to pay off the loan at a later date. Therefore, Home Credit Group should consider implementing other corporate measures such as preparing enough Allowance for Doubtful Accounts in the accounting practice, reselling these non-performing loans to other entities, hiring professionals to deal with this specific group etc. One important risk indicator of the financial institution is the ratio of non-performing loans. Companies would have to keep it as low as possible in order to mitigate the liquidity risks and default risk of the company.

Challenge of the results



Our data set is imbalanced since our target variable is not a 50/50 distribution. Number 1 means that customers who have difficulties to pay their loan and number 0 means not. From the pie chart above, number 0 has 91.9% percentage so our data set is very imbalanced. Therefore, we decided to use under sampling which means that we randomly select the target variable 0 and 1 with the ratio 2:1 (or 1:1) and then combine all results to balance it. We thought if there are any imbalanced variables over their values will also affect the predicted results.

Because our data is unbalanced, we decided to use sensitivity and specificity to evaluate our model. Sensitivity and specificity are important measures of the performance of a binary classification test. Accuracy is not enough to measure the performance of a binary classification. For example, our data has 91.9% percent of customers who can pay their loan. If we predict a customer can pay his loan, then we get 91.9% percentage correct, which does not make any sense.

Conclusion and future work

In order to better accomplish our goal, we use many models including supervised and unsupervised to analyze our data and make predictions. Before using the model for data analysis, we preprocessed our data to improve the accuracy of the model as much as possible and try to reduce the noise in the data that may interfere with the model. After repeated weighing and careful comparison, we left some of them as our final result.

For those supervised types, we finally kept random forest and the linear kernel SVM. Their accuracy is 66% and 70%. It is worth noting that they have their own advantages and disadvantages, which is why we finally decided to keep the two models. We believe that combining them together will achieve better results than using them alone. Random forest model's sensitivity is 0.7. In other words, it can help us identify users who may be at risk. Using this model can effectively help us monitor and reduce risks, which is undoubtedly important to us.

And SVM will help us from another cover. The specificity for our SVM is 0.95. It can help us find the customers who have the repaying ability to pay their loan on time. If we want to find out which customer is worth our loan and able to repay in time, this model will provide great help to us.

We also try to use the unsupervised model to classify our customers. The K-Means roughly divides our customers into five categories. After analyzing the results of the model, we believe that we should pay special attention to the following three customers: low income, low employment day and older age. In some cases, these three characteristics can help our employees quickly and easily measure whether they can lend money to these customers. In actual operation, this point is very helpful.

For the further step, although our current models have their own strengths, in the future we still hope to build a unified integrated model to help us solve problems better. No matter how to let the operator choose the model according to the situation, it is still easy to cause deviation. If we can build a model that can solve this problem independently, then I can eliminate this bias very well. For the future model, we will not only focus on accuracy, but also pay attention to both the sensitivity and the specificity to make our model more trustworthy.