Springboard Data Science Career Track Program

Capstone Project I

Credit Card Risk Analysis

A case study of classification algorithms

Ly Nguyen

April 2020

Contents

1 Introduction

1.1 Overview

The increasing number of loan applicants with the effects of global financial crisis lead to the increase of credit risk analysis. That aims to prevent the loss due to a borrower's failure to make payments, to improve the overall performance and secure a loan risk and increase the possibility of getting potentially valued customers. Credit risk analysis require the ability to store, categorize financial data based on a variety of criteria, assess risk levels associated with borrowers' profiles, their capability to repay the loan, and their borrowing history. It is also an approach for financial firms to gather and use data in order to keep up with new demands, get more impact in the market.

The most critical questions for the lenders are 1) How risky is the borrower? and 2) Given the borrower's risk, should we lend to him/her? With the answer to these questions, we can then determine if the borrower is eligible for the loan. Lenders provide loans to borrowers in exchange for the profit of repayment interest. That means they only make a profit if the borrowers can pay off the loans. If they do not pay the loan, then the lender loses money. We study the development of machine learning models to support banks and lenders in their decision-making by processing each applicant data and give prediction on paid or unpaid loan likelihood. An ideal system would process the data and automate decision-making, but in this study, we leave that for future work.

The models implemented in this project address a binary classification problem where the output is the prediction of the probability for loan application to be be paid or not. Date are labeled as 0 or 1, which means that a loan has been repaid or not. We use the dataset provided by Home Credit Default Risk from Kaggle.com for this project.[1]

Since the two classes associated with this dataset are imbalanced, the baseline models implemented in this project perform poorly on Recall and mostly cannot predict the minority class which is unpaid loans. In this project we explore methods to improve the baseline models' performance on those loans.

The project was implemented in various phases: data wrangling, hypothesis testing, visualizations of data distributions and applying classification algorithms like Logistic Regression, kNN, and Random Forest. We illustrate how to deal with missing values and techniques used with imbalanced classification problems. We also analyze and compare the performance of each algorithms with the baseline models. The models are implemented in Python with data analysis libraries like Pandas, Matplotlib, s=Seaborn and Scikit-learn. The model are effective in processing datasets, which can help business to evaluate, quantify model risk and performance for every individual credit decision, and determine the safety constraints and key decision factors for lending risks, therefore it helps them to improve their business and serve customers better.

The Jupyter notebooks that were developed for this project are available from a GitHub repository.[2]

---

[1] https://www.kaggle.com/c/home-credit-default-risk/data

[2] https://github.com/khanhly4682/Springboard_MachineLearning/tree/master/Home_Credit_Default_Risk

2 Approach

2.1 Data Acquisition and Wrangling

2.1.1 Dataset

There are 7 different sources of data that were processed:

- *application_train/application_test*: the main training and testing data with information about each loan application at Home Credit. Every loan has its own row and is identified by the feature SK_ID_CURR. The training application data comes with the TARGET indicating 0: the loan was repaid or 1: the loan was not repaid.
- *bureau*: data concerning client's previous credits from other financial institutions. Each previous credit has its own row in bureau, but one loan in the application data can have multiple previous credits.
- *bureau_balance*: monthly data about the previous credits in bureau. Each row is one month of a previous credit, and a single previous credit can have multiple rows, one for each month of the credit length.
- *previous_application*: previous applications for loans at Home Credit of clients who have loans in the application data. Each current loan in the application data can have multiple previous loans. Each previous application has one row and is identified by the feature SK_ID_PREV.
- *POS_CASH_balance*: monthly data about previous point of sale or cash loans clients have had with Home Credit. Each row is one month of a previous point of sale or cash loan, and a single previous loan can have many rows. _ *credit_card_balance*: monthly data about previous credit cards clients have had with Home Credit. Each row is one month of a credit card balance, and a single credit card can have many rows.
- *installments_payment*: payment history for previous loans at Home Credit. There is one row for every made payment and one row for every missed payment.

The below diagram shows how these datasets are related:

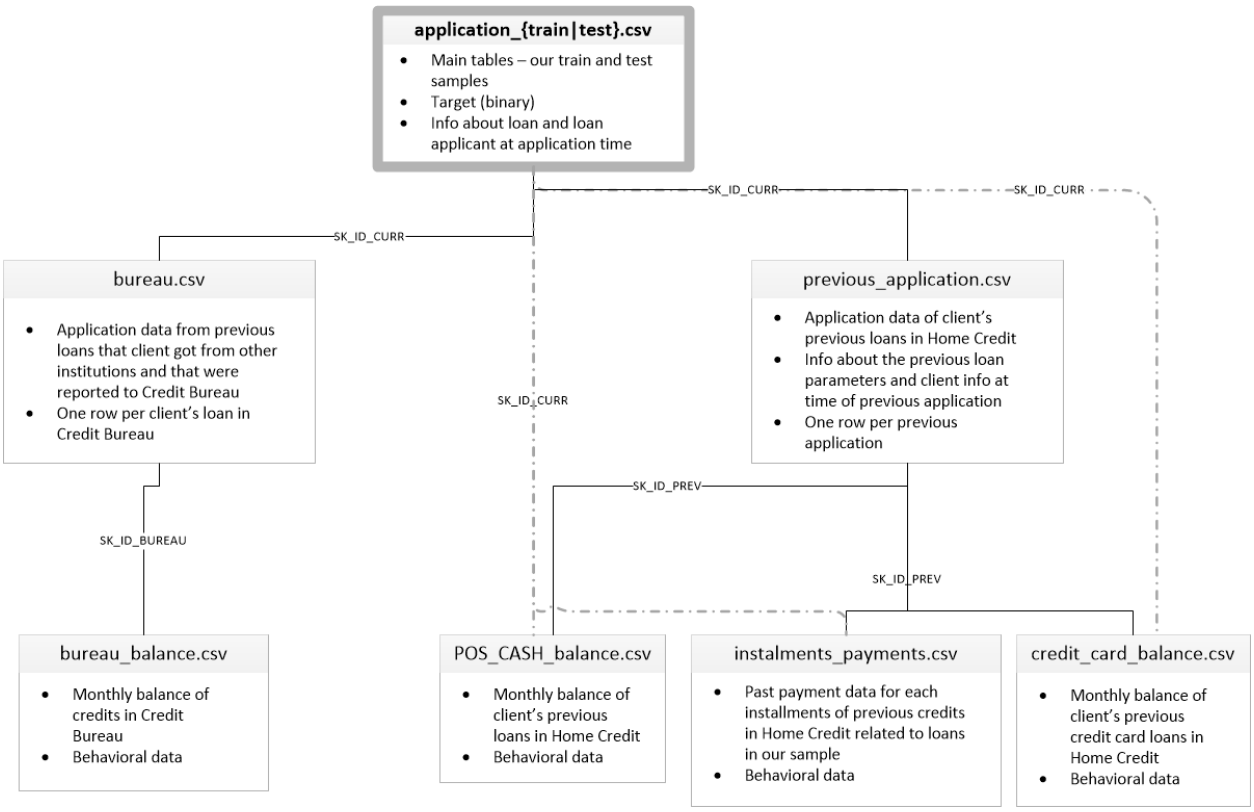


Figure 1. Data Overview

We combine all of the data files into one file based on each individual borrower's unique ID SK_ID_BUREAU. In this way, we concatenated all the features together to construct the training and test sets with the maximal usage of the given data. It also helps to easily process and manage the data. The final dataset contain 250,000 applications with nearly 200 features.

2.1.2 Data Preprocessing

The raw dataset is incomplete and likely to contain many errors. Before feeding data into the machine learning models, there are some preprocessing steps that must be performed for further processing. In our project, we preformed the follow steps:

Dealing with Missing values:

The dataset has 67 features that have missing values include both categorical and numerical data. The missing values can be due for the transformations while collecting data. There are some of the most common strategies for dealing with missing values:

Delete all values that have any missing values. This is usually done if the number of missing values is very small compared with the size of the data and the missing values are deemed to be random.

Impute missing values using mean or median of each feature separately, although other summary statistics can also be considered.

In our project, for categorical features that have missing values, we replace them with 'NaN'; while for numerical features, we replace missing values with the common method of mean value. Imputing missing values of categorical variables with Python's 'NaN' has the advantage of treating this value as a "Not Available" value, which is in itself a category that indicates absence of a value.

Checking data quality:

In our data, the "number of employed days" variable has values that exceed the normal range. We divided the days of employment to 365 days a year to calculate the number of employment years for each applicant. The maximum value we have is 1000 years, that is not normal and these values can skew the summary distribution and mislead the data representations. Therefore, we remove these erroneous values to make sure they do not affect the accuracy of the model.

Label encoding:

As machine learning algorithms most often accept only numerical inputs, it is important to encode the categorical variables into some specific numerical values. In the dataset, there are 16 categorical features including contract type, code gender, occupation type, organization type, etc. We encoded categorical values using dummy variable. A dummy variable is a variable that takes only 0 or 1 to indicate the absence or presence of a unique value of the feature (including 'NaN', which in this context means absence of value.)

2.1.3 Data Set Characteristics: Class Imbalance

A problem of many existing classification problems is that they are defined datasets that are inherently imbalanced. So when making predictions, the models tends to achieve high accuracy just for the majority class (the class with more data points). This high accuracy does not tell much since the accuracy is equivalent to the proportion of majority class in the test set. Therefore, the models show a poor performance on data points that belong to the minority class.

In the dataset for this project, 92% of the loan is paid and 8% of the loan is not paid. The ratio of unpaid loans and paid loans causes the baseline model perform badly. The model misclassifies almost all the loan as no risk and the recall of the baseline model for the minority class is very poor. We will address this issue further to handle the problem with resampling techniques in the next section.

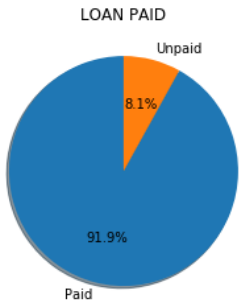


Figure 2. Loan Paid

2.1.4 Loan Applicant Exploratory Data Analysis (Data Storytelling)

When banks or lenders make the decision to give loans to applicants, it is necessary to analyze information of clients who apply for the loan to identify possible financial risks. This problem motivates to do an exploratory data analysis on the given dataset.

In this section we covered exploratory questions that were posed, and our analysis of what the dataset shows.

The following figures show that more female applicants tend to apply for loans, in comparison to male applicants. Most of income sources are from people who are working. Laborers tend to apply for a loan with more frequency, compared to other occupations. And the top organizations who applied for a loan mostly come from group of business entities of “type 3”.

- Business

- Self employed

- Medicine

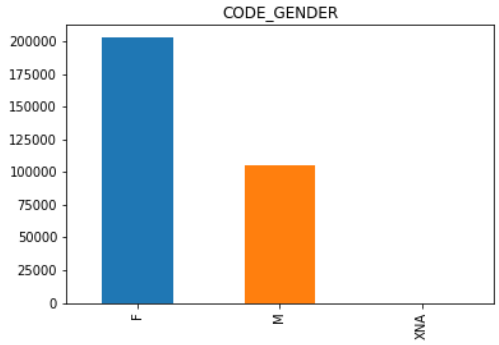


Figure 3. Code gender

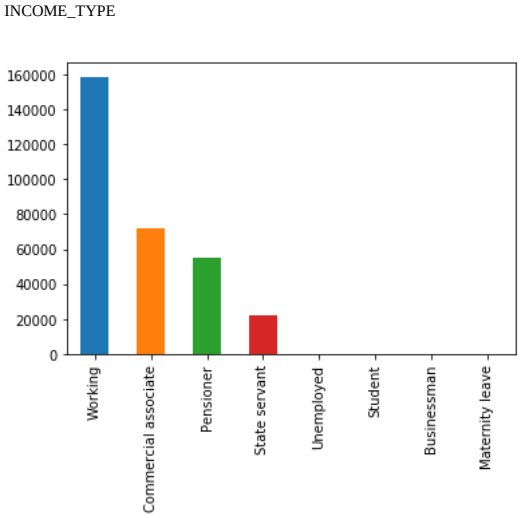
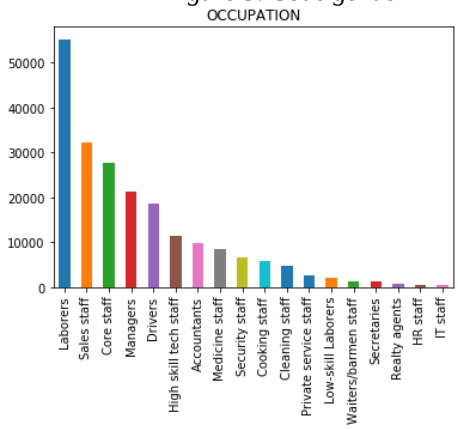


Figure 4. Income type



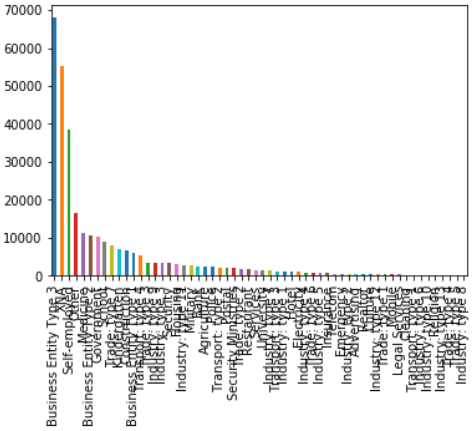


Figure 5. Occupation

Figure 6. Organization Type

Also, 69% of people who own a property apply for a loan more than people who not own a property. 66% of people who do not have car applied for a loan. And most of those cars have duration of 12 years usage (Figure 13).

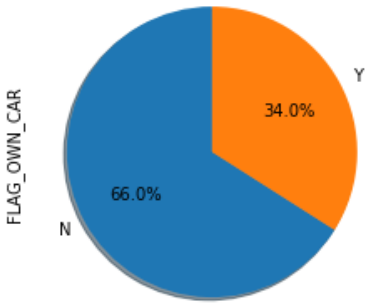


Figure 7. Own Car

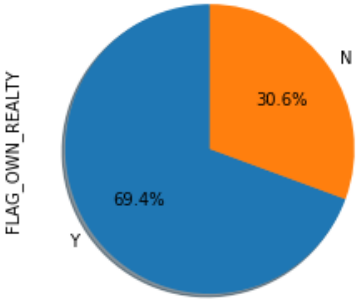


Figure 8. Own real estate

Most of applicants who applied a loan are married, and 71% of them have and education level of secondary school.

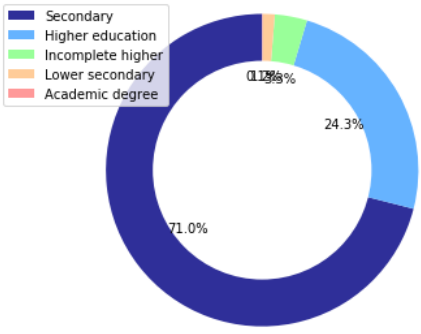


Figure 9. Education Level

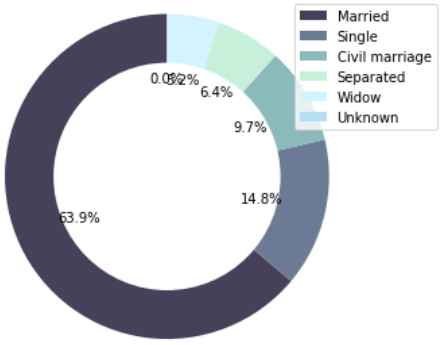


Figure 10.Marital Status

Most of type loans are cash loans and only 9.5% of them are revolving loans. When people apply for loans, they usually go alone (81%).

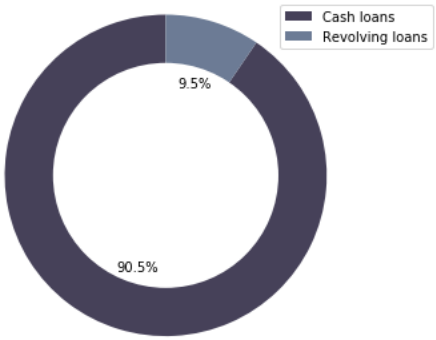


Figure 11. Type of Loans

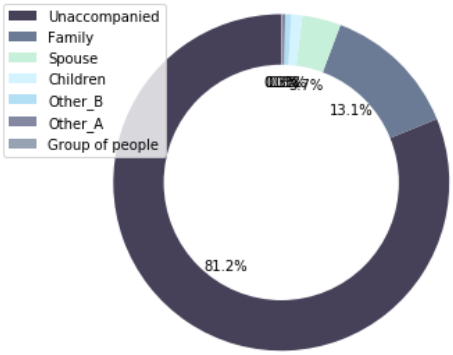


Figure 12. Accompanied Status

Clients mostly applied for a loan in morning from 10 a.m. -1 p.m. (Figure 14)

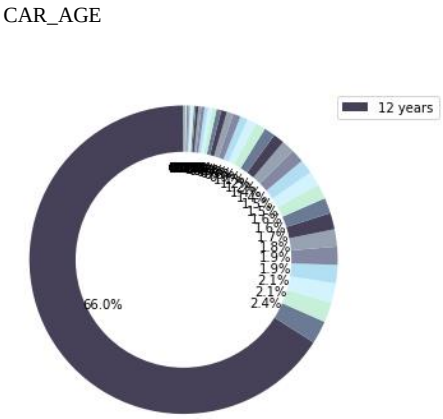


Figure 13. Car age

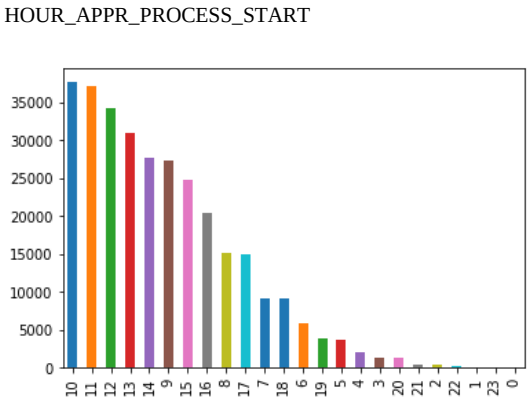


Figure 14. Hours application process tart

The above observations with information of loan applicants give us an idea of the profile of applicants.

Applications of Inferential Statistics

We perform inferential statistics to see if the difference of income between paid and un-paid loan, and the difference of income between male and female groups are statistically significant.

In the first case, we set the null hypothesis H_o there is no difference of income between loan paid and non-paid groups. And the alternative hypothesis is there is a difference of income between paid and non-paid groups. We used the means of income to determine if there is any difference between those groups and calculate 95% confidence interval over 10,000 replicates. We modeled the second case similarly.

According to our results, we cannot conclude whether the difference between income of paid and un-paid loans is statistically significant, and that there is a statistically significant difference of income between males and females.

2.2 Baseline Modeling

We choose Logistic Regression as the baseline model to estimate the probability that a loan will be in default, or not. The model is fitted on the training data set and then predict data samples from the test set. The prediction results for paid loans is very good, however it is under performing for the unpaid loan a Recall of 0%.

	precision	recall	f1-score	support
0	0.91	1.00	0.95	46041
1	0.20	0.00	0.00	4387
accuracy			0.91	50428
macro avg	0.56	0.50	0.48	50428
weighted avg	0.85	0.91	0.87	50428

Table 1: Classification Report for Logistic Regression

The problem is because the dataset is imbalanced, so it causes a skewness in the data distribution by creating a minority class and the majority class. The bias in the data cause the machine learning model to ignore the minority class. To address the problem of class imbalance, we will randomly resample the dataset using under sampling and over sampling techniques to adjust the distribution of the data set (i.e. the ratio between the different classes or categories represented). We will discuss more about resampling in the next section.

2.3 Extended Modeling

The baseline model, Logistic Regression has a bad result for predicting unpaid loans. We want to improve the results of the classification model by applying resampling techniques. And then, we also compare the performance of Logistic Regression, kNN, and Random Forest with the baseline before applying resampling techniques.

The performance evaluations are conducted over the same train-test split (80%-20%) to ensure consistency in the evaluations. Using accuracy to evaluate the model can mislead the results because of the imbalance issue, since the data set is dominated by paid loans (for which the models perform very well). For the Logistic Regression, the Recall is 0%.

Our next goal is to explore techniques to improve Recall on unpaid loans.

2.3.1 Resampling

Our approach to improve the model's Recall on unpaid loans involves the resampling of the imbalanced dataset. Resampling includes over sampling and under sampling. All resampling is applied to the training data after performing a train test split and we will evaluate the models using precisions, recalls on a the same test set.

2.3.1.1 Under sampling:

It aims at balancing class distribution by randomly selecting data points from the majority class the skewness from 1:10 to 1:1. When instances of two different classes are very close to each other, we remove the instances of the majority class to increase the spaces between the two classes (a.k.a. as "near-miss").

Near Miss Under Sampling

In this project, we apply Near Miss under sampling. Near Miss refers to a collection of under sampling methods that select examples based on the distance of majority class examples to minority class examples. The approaches were proposed by Jianping Zhang and Inderjeet Mani in their 2003 paper titled "KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction."[3] The technique will under sample the majority class to have the same number of examples as the minority class

To prevent the bias of using only one sample, we run resampling for a number of times and get the average results for Precision and Recall.

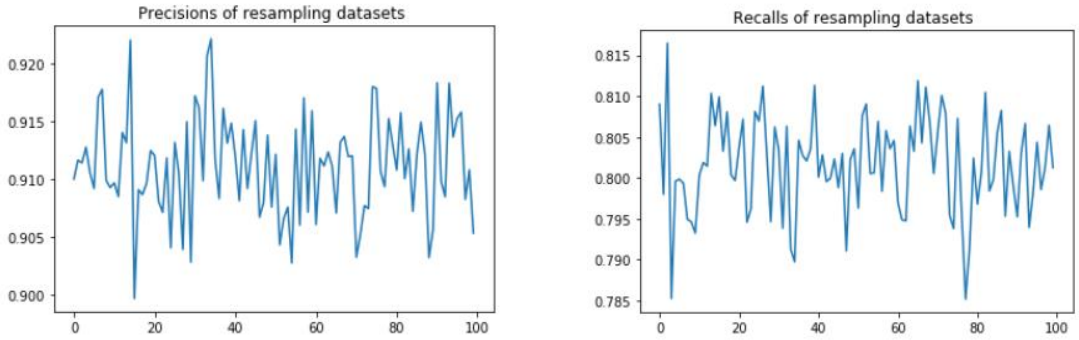


Figure 15. Logistic Regression's Precision and Recall for all samples

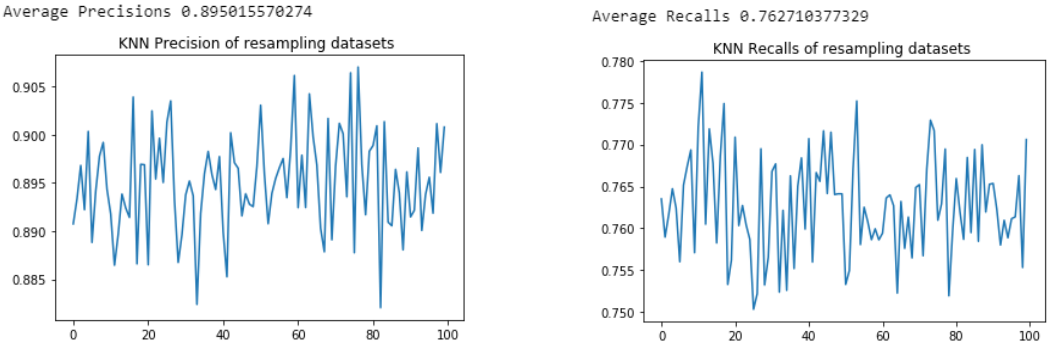
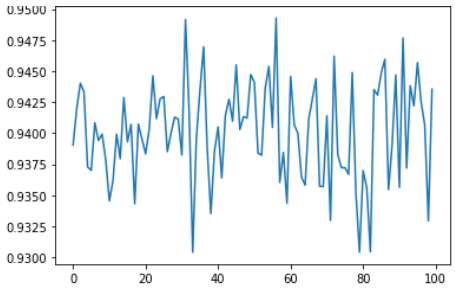
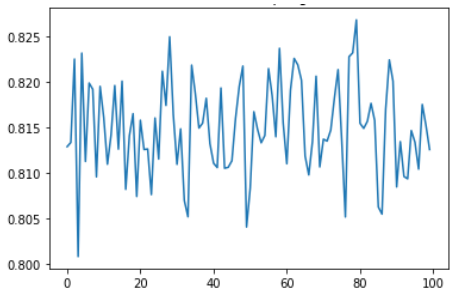


Figure 16. kNN's Precision and Recall for all samples

[3] https://www.site.uottawa.ca/~nat/Workshop2003/jzhang.pdf



RF's Precision



RF's Recalls

Figure 17. Random Forest Precision and Recall for all samples

We compare the results for all models, based on user sampled test data. The results shown that Logistic Regression are much better than kNN and Random Forest with Precision of 0.91 and Recall 0.80. The average Precision and Recall of the 3 models are as follows:

	Precision	**Recall**
Logistic Regression - Under sampling	0.91	0.80
Logistic Regression - Over sampling	0.87	0.6
kNN – Under sampling	0.895	0.763
Random Forest – Under sampling	0.895	0.815

Table 2. Results of classification models

2.3.1.2 Over sampling:

The goal of over sampling is to balance class distribution by randomly increasing minority class examples by creating synthetic data points derived from the original minority class. SMOTE (Synthetic Minority Oversampling Technique) synthesizes new minority instances between existing minority instances. It randomly picks up the minority class and calculates the K-nearest neighbor for that particular point. Finally, the synthetic points are added between the neighbors and the chosen spot.

	precision	recall	f1-score	support
0	0.94	0.60	0.73	46098
1	0.12	0.61	0.21	4329
accuracy			0.60	50427
macro avg	0.53	0.60	0.47	50427
weighted avg	0.87	0.60	0.69	50427

Table 3. Logistic Regression with SMOTE.

The performance of oversampling classifier is better in identifying target classes with Precision of 0.87 and Recall of 0.6. However, the result is not as good as under sampling classifiers.

3 Limitations

We evaluated our models based on Precision and Recall and looking to maximize the results with given class. The problem of Precision is a result of overlapped classes in the feature space, and Recall also has problem as result of imbalanced class. Our analysis shows that there is a tradeoff to improve the model

performance with respect to a metric. To improve the Recall, we apply resampling techniques however it makes Precision a little bit worse.

Therefore, we would like to explore more relevant features that describe each loan.

4 Conclusions and Future Work

Identifying the risk score for a potential applicant is crucial for the lending providers and datasets with class imbalance problems should be taken care of to ensure the classifier performance. To predict the risk of giving a loan to an applicant, we used the dataset published by Home Credit.

In this project, we explored three different supervised classification approaches to the credit loan dataset to predict unpaid and paid loan likelihood including Logistic Regression, kNN and Random Forest.

The models are fitted using 80% of the training data set, and 20% of the data is used to evaluate the model performance. In addition, we also introduced resampling methods to address the problem of imbalanced classes. We found that the Logistic Regression classifier with under-sampling gives the best performance compared with kNN and Random Forest. Thus, it may be an efficient classifier to compute risk scores for loan applicants.

Loan lenders can take advantage of prediction modeling discussed in this study to make decisions when evaluating loan applications to prevent financial loss.

However, the study is not complete, there are different methods to improve the models that we have not considered, like applying cross validation, selecting best features, dimensional reduction, etc.

5 Recommendations for the Clients

With the results of the classification models we have implemented, it shows that these models are not suitable for fully automated decision-making of loan applications. The models are good for predicting the loan risk analysis with 60% of improvement. However, for the remaining 40% of the data we suggest that it should be further analyzed by financial specialists to determine the risk likelihood of applicants. For example, a loan with on-time installments or good credit history can have higher probability of paying the loans. The models are effectively in support decision-making by interpreting a large collection of data and transform them to a smaller output that can be easily interpreted by human experts.