SCSD 4009 Project Report

Credit Card Approval Prediction

Chan Yuk Yee 56230549

Leung Tsan Lim 56211178

Table of Context

Abstract 3

1. Introduction 4

1.1 Introduction to dataset 4

1.2 Challenge 5

1.3 Objective 5

2. Methods 5

2.1 Data Pre-processing 5

2.1.1 Data Integration 5

2.1.2 Data Cleaning 6

2.1.3 Data Wrangling 7

2.2 Exploratory data analysis 7

2.2.1 Data visualization 7

2.2.2 Analysis importance features 9

2.2.3 PCA 9

3. Results 10

3.1 LogisticRegression 10

3.2 KNN 10

3.3 XGBoost 11

4. Evaluation 11

4.1 5-fold cross-validation 11

4.2 KNN error rate 11

4.3 5-fold cross-validation with SMOTE 12

4.4 Comparing accuracy 12

5. Discussion 12

4.1 Further discussion on balanced data 12

4.2 Further discussion on cross-validation for imbalanced datasets 13

6. Conclusion 13

References 14

**Abstract**

Credit card is a widely accepted method of payment. There is a lot of risk management system for credit cards in commercial banks and financial institutions. However, manually processing applications for credit cards is error-prone and a waste of time. In this paper, we use individual information and data submitted by credit card applicants to predict the possibility of future default. Before that, we conduct exploratory data analysis to explore the discovery of data. Then, we analyzed and compared the data with several supervised machine-learning algorithms. In addition, the most important features that determine whether a customer will become a 'bad' client are extracted from all features. Furthermore, the best-performing algorithm was chosen for predicting credit card approval. The algorithm gave around 90% accuracy in prediction.

**1. Introduction**

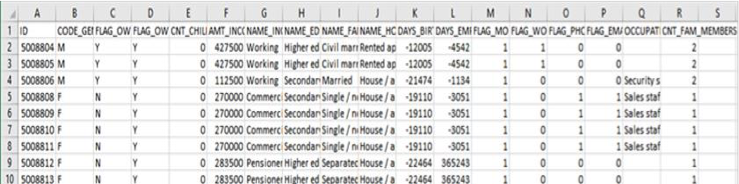
Credit card is a widely accepted method of payment. There is a lot of risk management system for credit cards in commercial banks and financial institutions. However, manually processing applications for credit cards is error-prone and a waste of time. The staff will spend too much time and work inefficiently. In order to increase productivity and efficiency, credit card approval predictions in machine learning are very valuable.

With the advancement of technology and digitalization coming in the 20s, the ability of machines to perform the predictive analysis is very significant in this era of big data. Machine learning is widely used in a lot of applications. For instance, financial institutions and commercial banks sometimes face the challenge of which risk factors to consider when providing credit to customers. To lower the risk of advancing credit to customers, machine learning tools should be used to identify the low-risk customer rather than analysis the person with human feelings. In this paper, we use individual information and data submitted by credit card applicants to predict the possibility of future default. Before that, we conduct exploratory data analysis to explore the discovery of data. Then, we analyzed and compared the data with several supervised machine-learning algorithms. In addition, the most important features that determine whether a customer will become a 'bad' client are extracted from all features. Furthermore, the best-performing algorithm was chosen for predicting credit card approval. The algorithm gave around 90% accuracy in prediction.

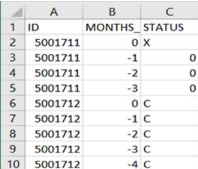
After the completion of the project, a finding of the detection of the credit risk of the customer by machine learning was reported in the following.

***1.1 Introduction to dataset***

**Figure 1. Application record dataset**



**Figure 2. Monthly credit card account status dataset**



The data was collected from the Kaggle Platform[[1]](#footnote-1). Figure 1 showed the credit card application records and it contains around 430,000 rows & 18 features. Figure 2 show monthly credit card account status respectively and it contains around 1,000,000 rows & 3 features.

***1.2 Challenge***

In this project, there are the following challenges:

* The dataset is highly imbalanced.
* The definition of 'good' or 'bad' is not given. The target must be defined.

***1.3 Objective***

Our achievement in this project is to:

* Determine the importance features that affect the credit card approval result.
* Use predictors to predict the credit card approval result with high accuracy

**2. Methods**

***2.1 Data Processing***

***2.1.1 Data Integration – defining target and merging***

In the dataset, due payment days can be observed. Assume that a person is considered a ‘Bad’ client if he has unpaid payments for more than 60 days. Status is defined by the table 1.

**Table 1. The definition of “Status” variable**

|  |  |
| --- | --- |
| Status | Meaning |
| 0 | 1-29 days past due |
| 1 | 30-59 days past due |
| 2 | 60-89 days overdue |
| 3 | 90-119 days overdue |
| 4 | 120-149 days overdue |
| 5 | Overdue or bad debts, write-offs for more than 150 days |
| C | Paid off that month |
| X | No loan for the month |

The target is determined by below logic:

|  |
| --- |
| **Target definition** |

**procedure** TargetDefinition

**for** i in id:

**if** status(i) >= 2:

target(i) 🡨 0 (It means a person is considered to be a bad client)

**else**:

target(i) 🡨 1 (It means a person is considered to be a good client)

|  |
| --- |
|  |

**Figure 3. The flow of target definition**



***2.1.2 Data Cleaning***

***a. Remove duplicate***

The duplicate data will be deleted. After duplicate removal by using pandas (pd.drop\_duplicates(...)), the data left 32713 rows.

***b. Filter unwanted outliers***

99.7 rules applied in CNT\_CHILDREN, AMT\_INCOME\_TOTAL, CNT\_FAM\_MEMBERS since they are continuous features.

***c. Handle missing value***

One variable has 14% missing value, and that is categorical feature. The solution is to ignore the missing value.

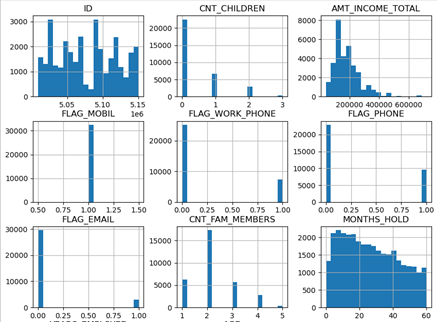
***2.1.3 Data Wrangling***

* One hot encoder is used to divide the Categorical data into different column datasets.
* Train-test Split are applied in 7:3 rate to avoid overfitting.
* SMOTE is used to handle imbalanced data. Note that Applying oversampling after train-test split because the observations from the minority class in the training dataset might end up in the testing dataset. This is in a way allows the algorithm to cheat since it learned from something similar.

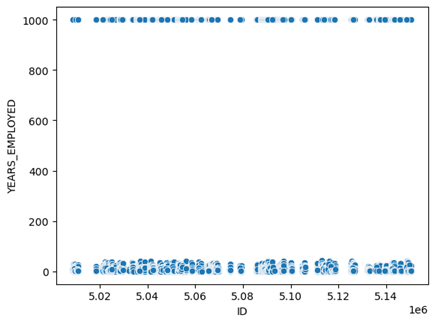
***2.2 Exploratory data analysis***

***2.2.1 Findings of data visualization***

**Figure 4. The distribution of different variables**

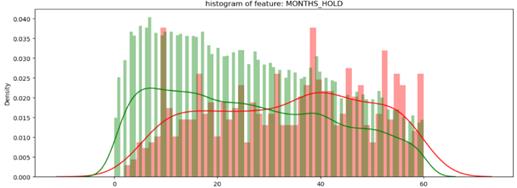


**Figure 5. The distribution of YEARS\_EMPLOYED**

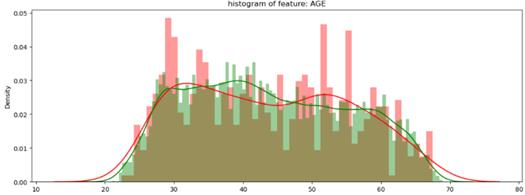


1. Figure 4 showed that FLAG\_MOBIL is useless data since the data record that all people have mobile phone. This column needs to be drop.
2. Figure 5 showed that some people have around 1000 YEARS\_EMPLOYED. After checking, we found that they are both pensioners. Assume that most people around 40-45 YEARS\_EMPLOYED are going to become pensioners. ➔ Solution: transform 1000 to [40,45] uniform randomly.
3. MONTHS\_HOLD should not be a predictor since it refers to the month the user holds the credit card. It may lead to overfitting. This column also needs to be dropped.
4. If a person is at a work, the probability of being a good client will be higher than the person who is unemployed.

**Figure 6: The distribution of MONTHS\_HOLD**



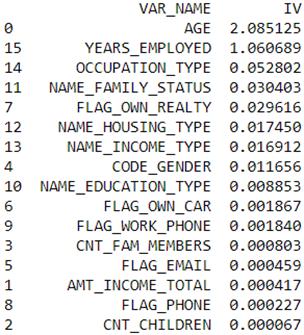
**Figure 7: The distribution of AGE**



***2.2.2 Analysis importance features***

Information value (IV) is measured in the powers among the predictors, which determine the important variables in predictive model. It ranks variables on the basis of the importance[[2]](#footnote-2). Figure 8 shows the result.

**Figure 8: Iv value of features**



**Table 2: Rules related to Information Value**

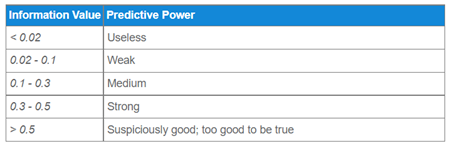
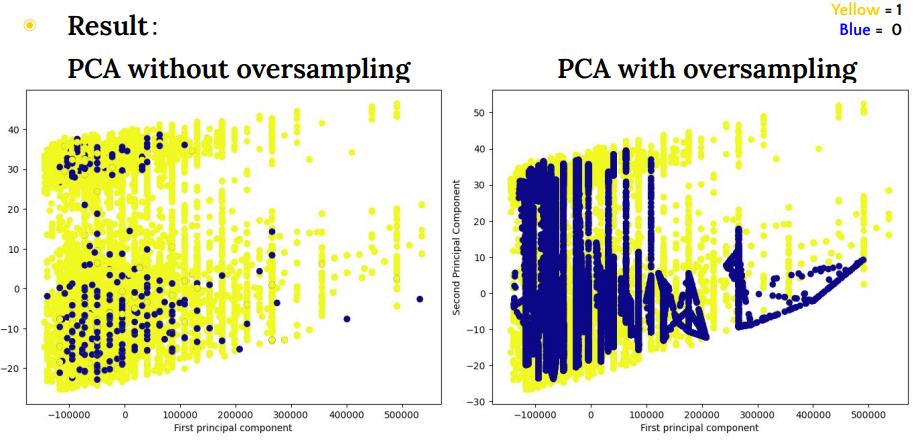


Table 2 showed that AGE and YEARS\_EMPLOYED are top 2 importance features.

***2.2.3 PCA***

Unsupervised learning is applied to test does PCA fit the data well. Figure 9 shows the result.

**Figure 9: PCA result**



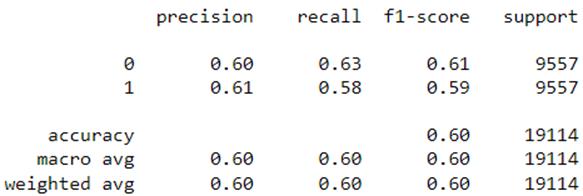
Although unsupervised learning could be applied to the dataset, a clear classification point cannot be found in the PCA graph, it does not result in a clear classification area.

Therefore, as the datasets consist of a certain number of predictors, supervised machine learning should be used.

**3. Results**

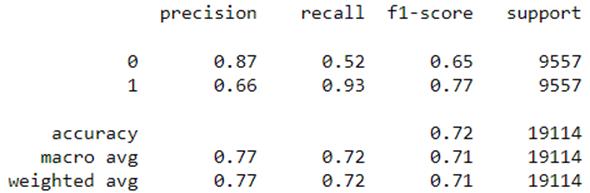
***3.1 Logistic Regression***

Information Value is designed mainly for the binary logistic regression models[[3]](#footnote-3). Features’ IV value < 0.01 is decided to remove. Therefore, the remaining features will be used in the Logistic Regression model. AGE, YEARS\_EMPLOYED, OCCUPATION\_TYPE, NAME\_FAMILY\_STATUS, FLAG\_OWN\_REAITY, NAME\_HOUSING\_TYPE, NAME\_INCOME\_TYPE, CODE\_GENDER are included. The result shows Logistic Regression has an Accuracy of 60%.



***3.1 KNN***

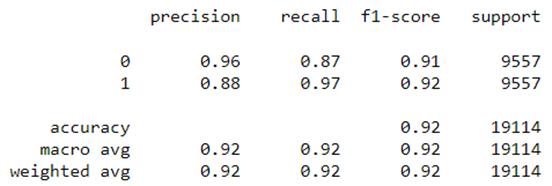
Feature selection via IV value and using them in KNN and XGBoost model might not produce the most accuracy. IV value should not be used in KNN and XGBoost models. The result shows KNN has an Accuracy of 72%.



*\*\* the correlation graph between parameter (n) and the error rate will be tested in validation part.*

***3.1 XGBoost***

The full name of XGboost is eXtreme Gradient Boosting. It is currently the most common algorithm in Kaggle competitions, and it is also the model used by most winners. Result shows XGBoost has Accuracy with 92%.



**4. Results**

***4.1 5-fold cross validation***

After performing 5-fold cross-validation, accuracy rate has changed.

Logistic Regression: 60%→58% (slightly decrease)

KNN: 70%→94% (considerably increase)

XGBoost: 90%→94% (slightly increase)

***4.2 KNN parameters error rate***

**Figure 10: The graph of KNN error rate**

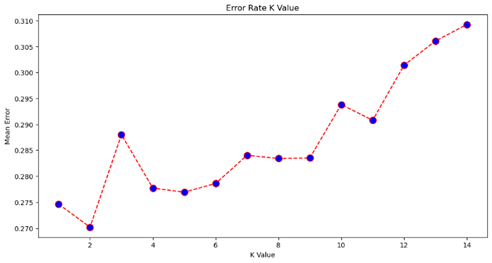


Figure 10 showed that k values bigger, Error Rate bigger.

***4.3 5-fold cross validation with SMOTE***

Using SMOTE to avoid inaccurate evaluation metrics when using 5-fold cross-validation. After performing 5-fold cross-validation with SMOTE, accuracy rate has changed.

**Performance (simple train-test-split v.s. 5-fold with SMOTE):**

Logistic Regression: 60%→58% (slightly decrease)

KNN: 70%→89% (considerably increase)

XGBoost: 90%→93% (slightly increase)

***4.4 Comparing accuracy***

**Table 3. Accuracy comparison**

|  |  |  |  |
| --- | --- | --- | --- |
| **Accuracy** | **Simple train-test** | **5-fold** | **5-fold with SMOTE** |
| Logistics Regression | 60% | 58% | 58% |
| KNN | 70% | 94% | 89% |
| XGBoost | 90% | 94% | 93% |

The performance of the simple train-test spilt is the poorest. The performance of the KNN and XGBoost models via 5-fold cross-validation with SMOTE is lightly decreased compared to the models via 5-fold cross-validation.

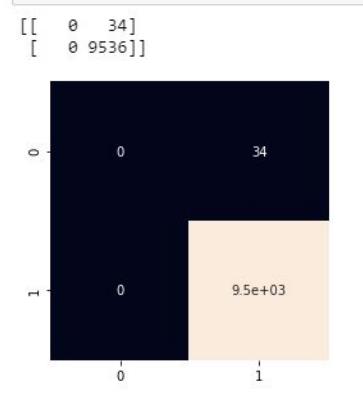
**5. Discussion**

***5.1 Further discussion on balanced data***

Unbalanced data is mentioned at the report's introduction, and this report used SMOTE to simulate target "0" data. The effect of imbalanced data on model training and testing is also examined in this research.

The accuracy of logistic regression is 99% when imbalanced data is fitted into the model after the train-test split, seems that it is a perfect predicting model in this situation. However, Figure 11 is shown, 9536 of the predicted data are accurate and 34 are inaccurate. There are 34 data with "0" labeled in the testing data since the matrix predicts all 1 in the predicted result.

**Figure 11. Confusion Matrix for Imbalanced Data Classification**



This showed that if extremely unbalanced data is fitted to the data set, the training data set and the test data set continue to be unbalanced, causing it to be meaningless modeling with high accuracy. Therefore, data balance should be ensured before the modeling process.

***5.2 Further discussion on cross-validation for imbalanced datasets***

5-fold cross-validation (performing oversampling before executing cross-validation) and 5- fold cross-validation with SMOTE (performing oversampling during cross-validation) are both adopted for model validation in this paper. We found that performing oversampling before executing cross-validation seems "efficient". However, the fact is that we are overestimating the classification performance while adopting this method[[4]](#footnote-4). Therefore, performing oversampling during cross-validation is the correct way of handling imbalanced data.

**6. Conclusion**

In conclusion, AGE and YEAR\_EMPLOYED are the important features in the predictors, which means they are easily interpreted the credit card approval result. On the other hand, XGBoost has the highest performance with 90-95% accuracy. Therefore, XGBoost should be applied to the business situation in order to optimize the prediction result.

**References**

1. Credit card approval prediction. Kaggle. (n.d.). Retrieved November 22, 2021, from <https://www.kaggle.com/rikdifos/credit-card-approval-prediction/metadata>
2. Bhalla, D. (n.d.). Weight of evidence (WOE) and information value (iv) explained. ListenData. Retrieved November 22, 2021, from [https://www.listendata.com/2015/03/weight- of-evidence-woe-and-information.html](https://www.listendata.com/2015/03/weight-%20of-evidence-woe-and-information.html)
3. Santos, M. S., Soares, J. P., Abreu, P. H., Araujo, H., &amp; Santos, J. (2018). Cross- validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches. IEEE Computational Intelligence Magazine, 13(4), 59–76. <https://doi.org/10.1109/mci.2018.2866730>

1. Credit card approval prediction. Kaggle. (n.d.). Retrieved November 22, 2021, from https://www.kaggle.com/rikdifos/credit- card-approval-prediction/metadata. [↑](#footnote-ref-1)
2. Bhalla, D. (n.d.). Weight of evidence (WOE) and information value (iv) explained. ListenData. Retrieved November 22, 2021, from https://www.listendata.com/2015/03/weight-of-evidence-woe-and-information.html. [↑](#footnote-ref-2)
3. Bhalla, D. (n.d.). Weight of evidence (WOE) and information value (iv) explained. ListenData. Retrieved November 22, 2021, from https://www.listendata.com/2015/03/weight-of-evidence-woe-and-information.html. [↑](#footnote-ref-3)
4. Santos, M. S., Soares, J. P., Abreu, P. H., Araujo, H., &amp; Santos, J. (2018). Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [research frontier]. IEEE Computational Intelligence Magazine, 13(4), 59– 76. https://doi.org/10.1109/mci.2018.2866730 [↑](#footnote-ref-4)