**1. Introduction**

Small businesses and giant corporations are having difficulty surviving in the current economic recession. A sophisticated risk control can help reduce unnecessary cost and avoid unwanted situations. Credit score check is one of the risk control methods in the financial industry and banks. They examine personal information and data from credit card applicants to predict the probability of debts occurring. Credit score checking is crucial for the bank to decide whether to issue a credit card to the applicant.

The goal of this project is building a prediction model that can determine the credit approval rate from each customer. XGboost, logistic regression, and decision tree will be constructed to answer our research question. Our dataset has a lot of personal information and attributes about applicants, which will be used to build our models. In addition, we will also investigate the most influential factor in determining “good” clients and “bad” clients.

**2. Exploratory Data Analysis**

The dataset used in this project is from Kaggle website called “Credit Card Approval Prediction ''. The dataset contains information of credit card users including genders, number of children, number of properties, their repayment dates, etc. Machine learning models could be established based on this dataset to help banks and financial institutions in judging whether a customer has good credit or not, so that the bank could make decisions on whether to lend a credit card to a certain customer or not. (Seanny, 2019)

***Table 1. Variables explanations - application dataset***

|  |  |  |
| --- | --- | --- |
| **Feature name** | **Explanation** | **Remarks** |
| ID | Client number |  |
| CODE\_GENDER | Gender |  |
| FLAG\_OWN\_CAR | Is there a car |  |
| FLAG\_OWN\_REALTY | Is there a property |  |
| CNT\_CHILDREN | Number of children |  |
| AMT\_INCOME\_TOTAL | Annual income |  |
| NAME\_INCOME\_TYPE | Income category |  |
| NAME\_EDUCATION\_TYPE | Education level |  |
| NAME\_FAMILY\_STATUS | Marital status |  |
| NAME\_HOUSING\_TYPE | Way of living |  |
| DAYS\_BIRTH | Birthday | Count backwards from current day (0), -1 means yesterday |
| DAYS\_EMPLOYED | Start date of employment | Count backwards from current day(0). If positive, it means the person is currently unemployed. |
| FLAG\_MOBIL | Is there a mobile phone |  |
| FLAG\_WORK\_PHONE | Is there a work phone |  |
| FLAG\_PHONE | Is there a phone |  |
| FLAG\_EMAIL | Is there an email |  |
| OCCUPATION\_TYPE | Occupation |  |
| CNT\_FAM\_MEMBERS | Family size |  |

***Table 2. Variables explanations - credit\_record dataset***

|  |  |  |
| --- | --- | --- |
| **Feature name** | **Explanation** | **Remarks** |
| ID | Client number |  |
| MONTHS\_BALANCE | Record month | The month of the extracted data is the starting point, backwards, 0 is the current month, -1 is the previous month, and so on |
| STATUS | Status | 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 |

**2.1 Analysis Description**

In this part, several analyses have been implemented through the goal of creating model-ready dataset. During EDA, the following tasks would be carried out:

* Analysis of the structure of the dataset
* Outliers’ detection and solution
* NA and duplicate value analysis and solution
* Feature analysis, synthesis, and transformation
* Encode categorical variables
* Correlation analysis

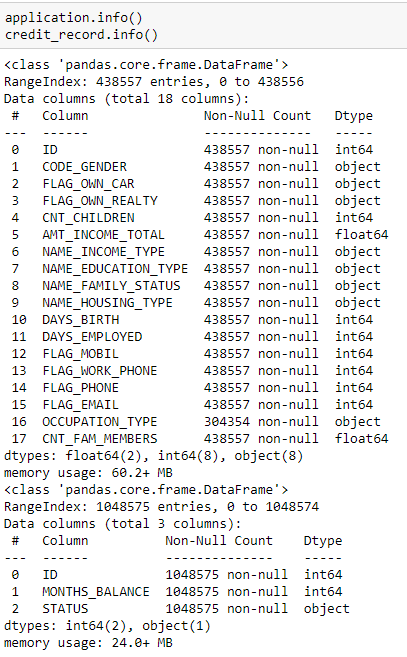
**2.2 Data Extraction**

For extracting the dataset, we download two required dataset and input them into the Jupyter Notebook from the Kaggle website using “open datasets” packages in Python. (Gupta, 2022) This package allows direct downloading of datasets from a given website (Kaggle in this case) by only inputting values of the username and the corresponding key, which is extracted from the Kaggle account and noted in the notebook.

**2.3 Data Cleanup**

**2.3.1 Structure of the Datasets**

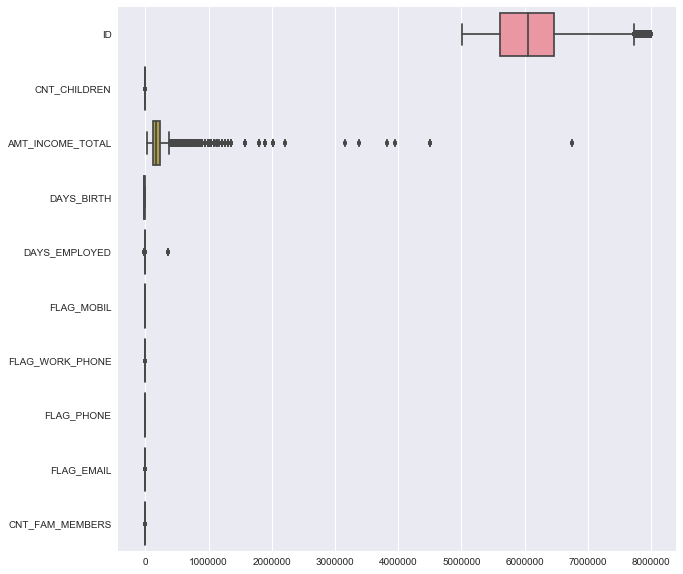
There are two datasets: one is called “application” which stores the information of customers who applied for a credit card, the second one is called “credit\_record” which stores the information regarding credit histories for each customer. We checked the structure of two datasets and the result is shown in Figure 1. There are 18 variables and 438,557 records in the “application” dataset and there are 3 variables and 1,048,575 records in the “credit\_record” dataset. The data types are also shown in Figure 1. As we could see, there are both numerical and categorical variables in the datasets, but there is no response variable indicating whether a customer is "good" or "bad". Therefore, we need to create a dependent variable for supervised learning models after.

****

***Figure 1. Structure of datasets***

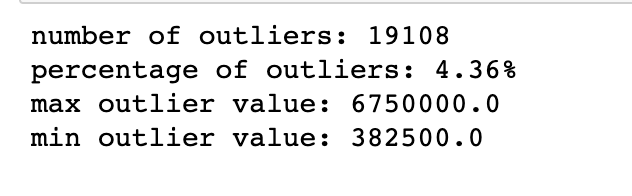
**2.3.2 Check outliers**

Outliers are checked by looking at the boxplots of all numerical variables in the dataset, as shown in Figure 2.



***Figure 2. Boxplot of numerical variables in the dataset***

As we could see from the boxplot, except the “ID” column, most of the outliers are from the “AMT\_INCOME\_TOTAL” variable. The number of outliers and percentage are also investigated in Figure 3, the figure shows that the outliers only count 4.36% of the total amount of data. Therefore, we decide to drop outliers.



***Figure 3. Statistics of outliers in “AMT\_INCOME\_TOTAL” variable***

**2.3.2 Check Duplicated and Null values**

There are no duplicate records in either dataset. There are no null values in the “credit\_record” dataset, but there are null values in the ‘OCCUPATION\_TYPE’ variable in the “application” dataset, both the number and percentage of null values in the dataset are shown in Table 3. Since Null values account for 30% of the total number of records in the “OCCUPATION\_TYPE” variable, for better analysis, we replace these null values with "Unknown".

***Table 3. The number and percentage of NA values in the dataset***

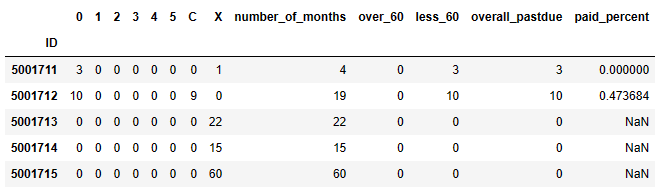
|  |  |  |
| --- | --- | --- |
| Variable Name | Number of Null Values | Percentage of Null Values (%) |
| ID | 0 | 0 |
| CODE\_GENDER | 0 | 0 |
| FLAG\_OWN\_CAR | 0 | 0 |
| FLAG\_OWN\_REALTY | 0 | 0 |
| CNT\_CHILDREN | 0 | 0 |
| AMT\_INCOME\_TOTAL | 0 | 0 |
| NAME\_INCOME\_TYPE | 0 | 0 |
| NAME\_EDUCATION\_TYPE | 0 | 0 |
| NAME\_FAMILY\_STATUS | 0 | 0 |
| NAME\_HOUSING\_TYPE | 0 | 0 |
| DAYS\_BIRTH | 0 | 0 |
| DAYS\_EMPYOYED | 0 | 0 |
| FLAG\_MOBIL | 0 | 0 |
| FLAG\_WORK\_PHONE | 0 | 0 |
| FLAG\_PHONE | 0 | 0 |
| FLAG\_EMAIL | 0 | 0 |
| OCCUPATION\_TYPE | 129662 | 30.6 |
| CNT\_FAM\_MEMBERS | 0 | 0 |

**2.3.3 Convert 'DAYS\_BIRTH' and 'DAYS\_EMPLOYED'**

The numbers in 'DAYS\_BIRTH' and 'DAYS\_EMPLOYED' count backwards from current day (0), which is hard to understand or read as they are all negative numbers. For better understanding, we converted "DAYS\_BIRTH" to "age", which is the age of the customer. Additionally, we converted "DAYS\_EMPLOYED" to the number of days the customer was employed, which is a positive number.

**2.3.4 Create Dependent Variable**

We created the dependent variable to identify the clients based on their credit status. Users at risk should typically be in the 3% range, so we selected customers who were past due for more than 60 days as our target risk customers. These bad customers are designated as "0". Customers who paid off all debts or had no debts during the month or were less than 60 days past due were identified as good customers. These good customers are designated as "1". There are 667 risky clients and 45,318 good clients in the dataset (shown in Figure 4).

****

****

***Figure 4. Create Dependent Variable***

**2.3.5 Merge Datasets and Delete Duplicate IDs**

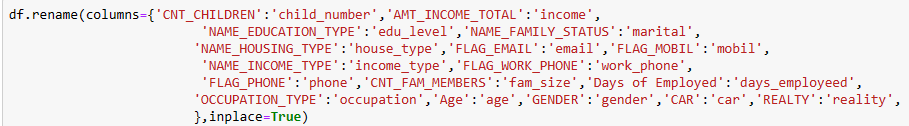
We merged the target variable "good\_or\_bad" with the ‘application’ dataset. There are 34,928 unique IDs in the new dataset, which are the records we are going to analyze.



***Figure 5. Number of unique IDs in the new dataset***

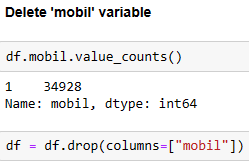
**2.3.6 Rename Variables and Delete Useless Variables**

We renamed all variables for readability. The new variables’ names are shown in Figure 6.

****

***Figure 6. Rename variables***

Then we deleted the ‘mobil’ variable as it is useless for model building. All clients have a mobile phone, which will not help to identify the clients.

****

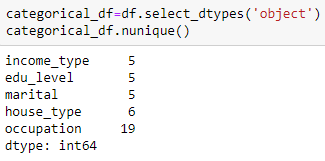
***Figure 7. Delete useless variables***

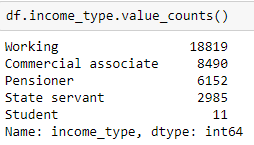
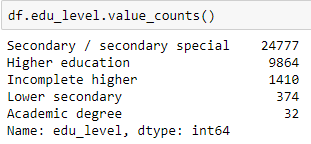
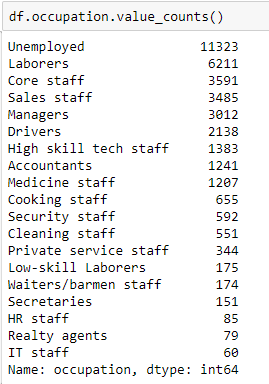
**2.3.7 Encode Categorical Variables**

First, we replaced ‘F’ by 0 and ‘M’ by 1 in the ‘gender’ variable. Then we replaced ‘N’ by 0 and ‘Y’ by 1 in the ‘car’ and ‘reality’ variables. After that, there are 6 categorical variables in the dataset.

The number of unique values for each variable is shown in Figure 7. We defined a covert\_dummy function to encode all the other categorical variables. There are 19 unique values in the ‘occupation’ variable, which will generate too many dummy variables, so we separated them into 3 different groups: 'labor work’, ‘office work’, and ‘hightec work’ based on the types of work. Also, as there are too few ‘academic degrees’ in the ‘edu\_level’ variable, we added it into the ‘higher education’ category. For the same reason, we added ‘student’ into the ‘state servant’ category in the ‘income\_type’ variable.

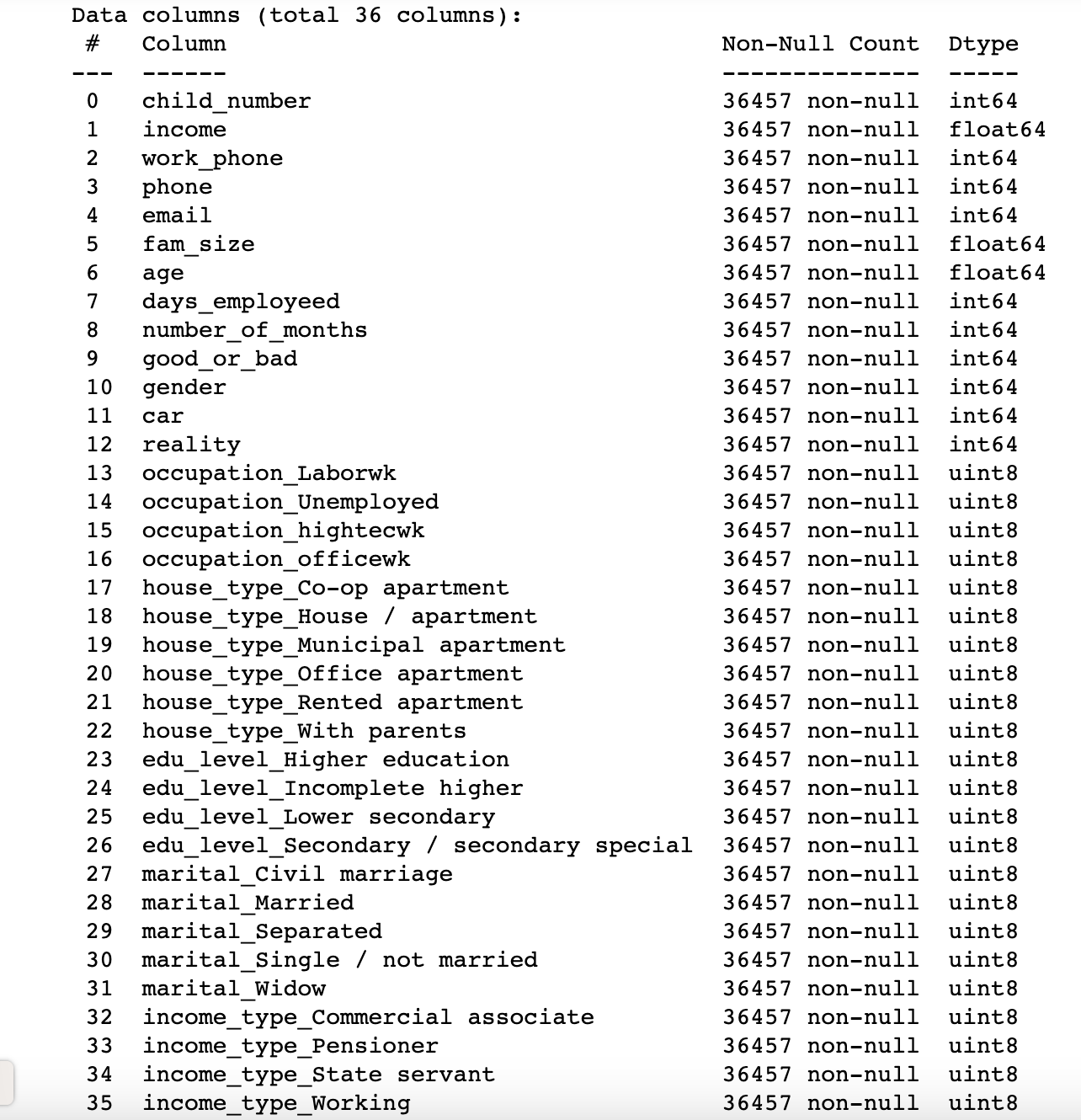
****

****

****

***Figure 8. Encode Categorical Variables***

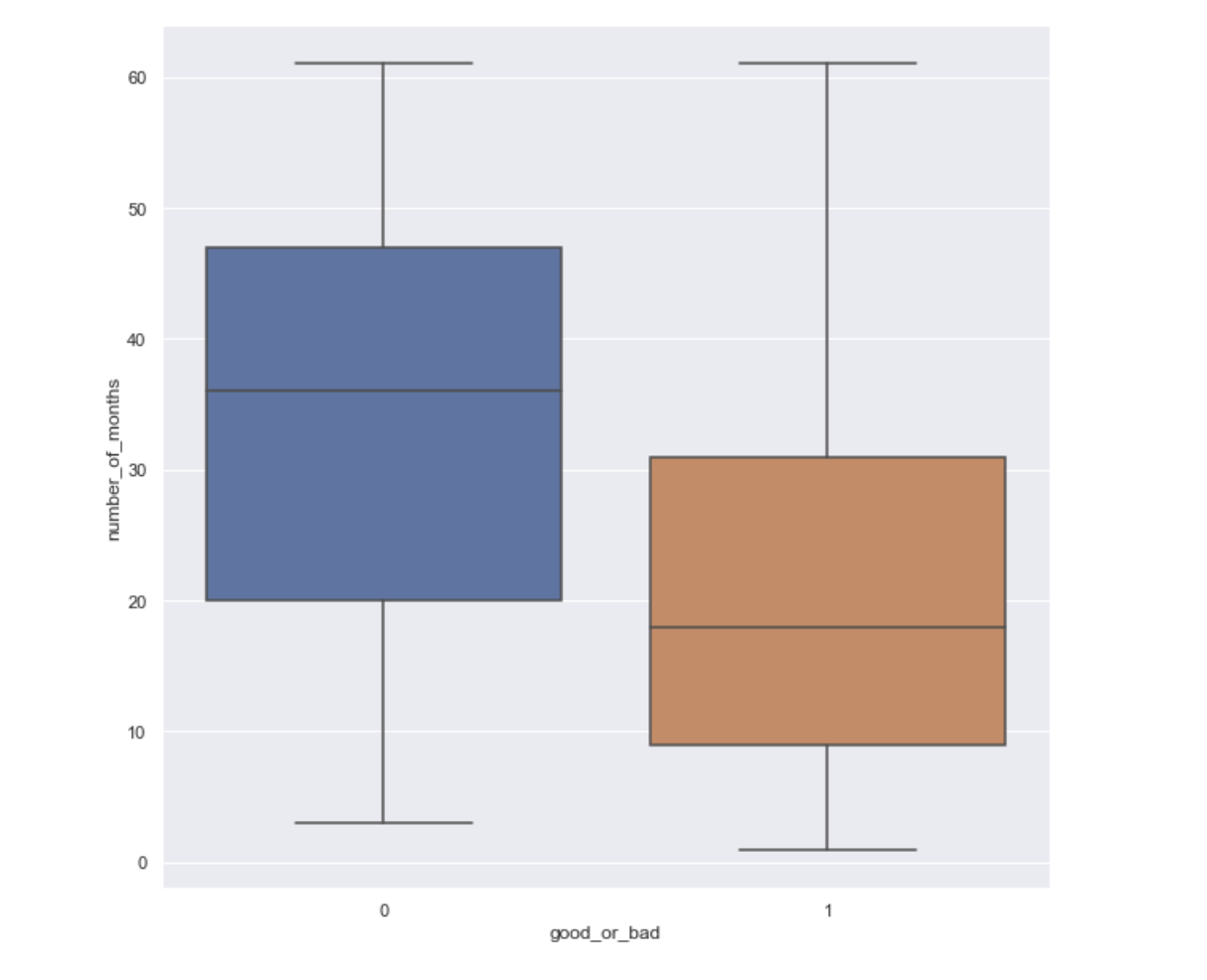
Finally, we got a cleaned dataset with 36 variables and 34,928 records (shown in Figure 9).

****

***Figure 9. Structure of the cleaned dataset***

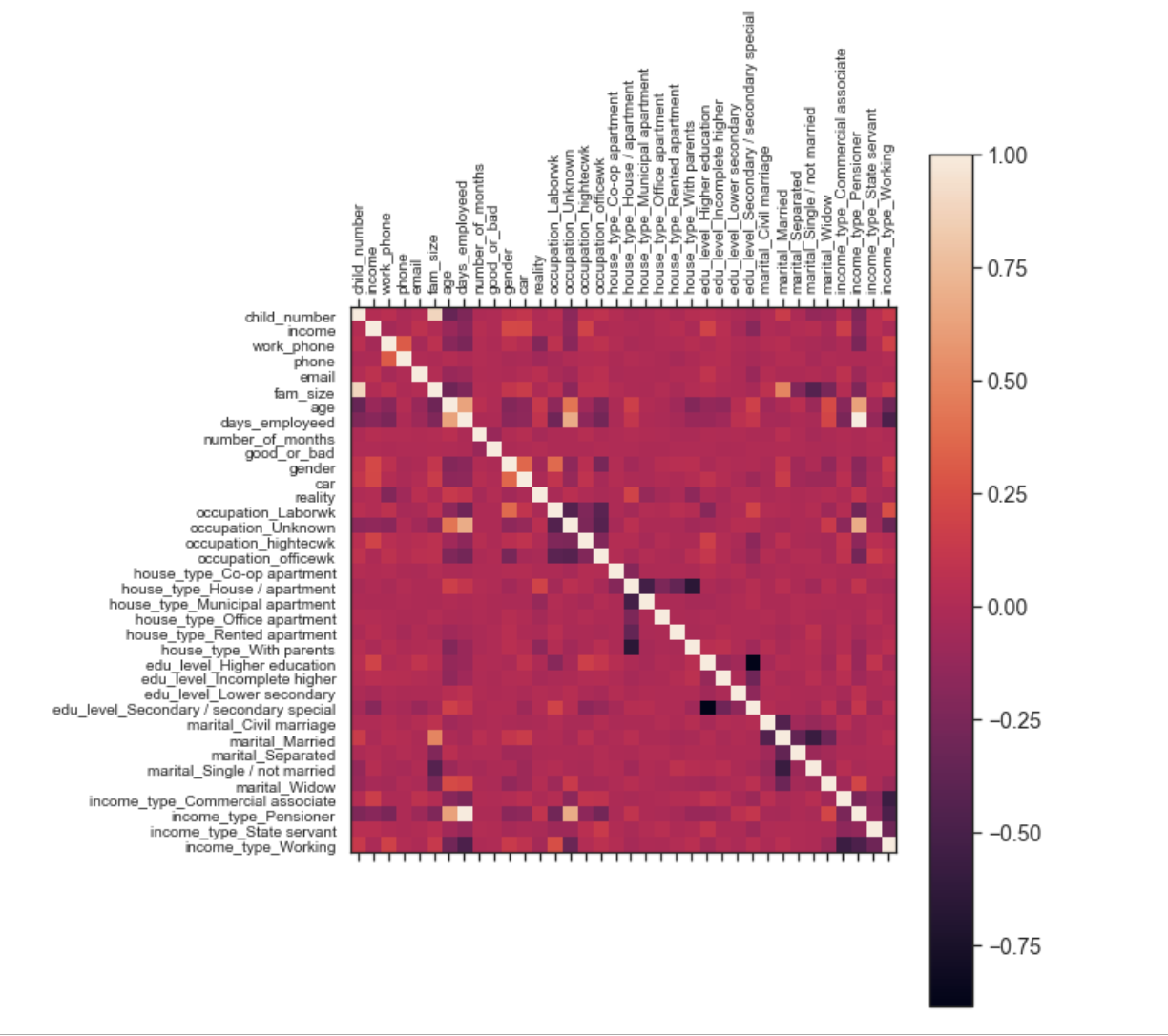
**2.4 Data Visualizations***Table1.*

The first thing we would like to compare is the distribution of the number of months in both good credit and bad credit record groups of customers. By looking at both boxplots we could find out that both the median and IQR of month balance for good credit record customers are lower than customers with bad credit record.

****

***Figure 10. Distribution of Number of Months for Good/ Bad Credit Customers***

Table 5 shows the second thing we were looking at: the correlation matrix. A correlation plot is shown in Figure 11, we could see that there are not significant correlations among numerical variables in the dataset, as there are few light grids in the figure. In Table 5 we collect the most correlated 5 variables with the target label “good\_or\_bad” customers. As we could see, even those top 5 most correlated variables don’t have much correlation with the target variable, but those variables would first be considered in building models after.



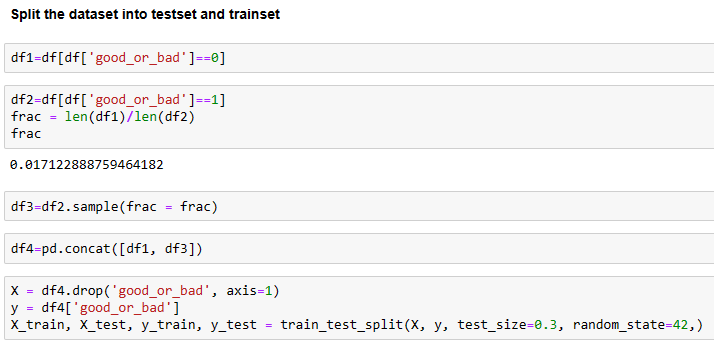
***Figure 11. Correlation Plot***

***Table 5. Top 5 correlation matrix to the target variable***

|  |  |
| --- | --- |
| good\_or\_bad | 1.000 |
| reality | 0.020 |
| martial\_Married | 0.013 |
| income\_type\_State servant | 0.009 |
| occupation\_officewk | 0.009 |

**3. Predictive Models**

Due to the uneven distribution of good customers and bad customers, the proportion of good customers is about 99%. For better modeling, we selected some good customers to build a model, so that the ratio of good customers to bad customers is 50:50. We then randomly split the dataset into training and testing sets with a ratio of 0.3. 70% of the data is in the training set, and 30% of the data is in the test set.

****

***Figure 12. Test set and Train set***

**3.1 XGBoost**

XGBoost is a tree-based ensemble machine learning technique that improves on the Gradient Boosting framework by introducing certain precise approximation algorithms. It has improved prediction power and performance.

In general, it's this algorithm's effectiveness, accuracy, and viability.

It features both tree learning methods and linear model solvers. Therefore, its ability to perform parallel processing on a single machine is what makes it quick.

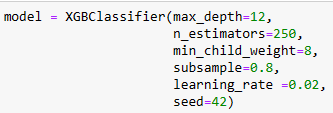
Additionally, it offers tools for detecting significant variables and doing cross-validation.

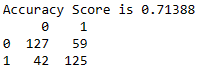
XGBoost may be used for any supervised machine learning task that meets the following requirements:

* When the training data contains a large number of observations.
* In the training set, there are more observations than features.
* When the data is entirely numerical or contains both numerical and categorical features, it performs well.
* When model performance metrics should be taken into account.

We will use XGBoost in this study since all requirements have been fulfilled.

First, we used the train set to build the XGBoost model (shown in Figure 13).





***Figure 13. Build XGBoost model***

The model has an accuracy of 0.71. The accuracy, precision, sensitivity, and specificity values are around 0.73, indicating that the model is accurate with very low Type 1 and Type 2 errors.

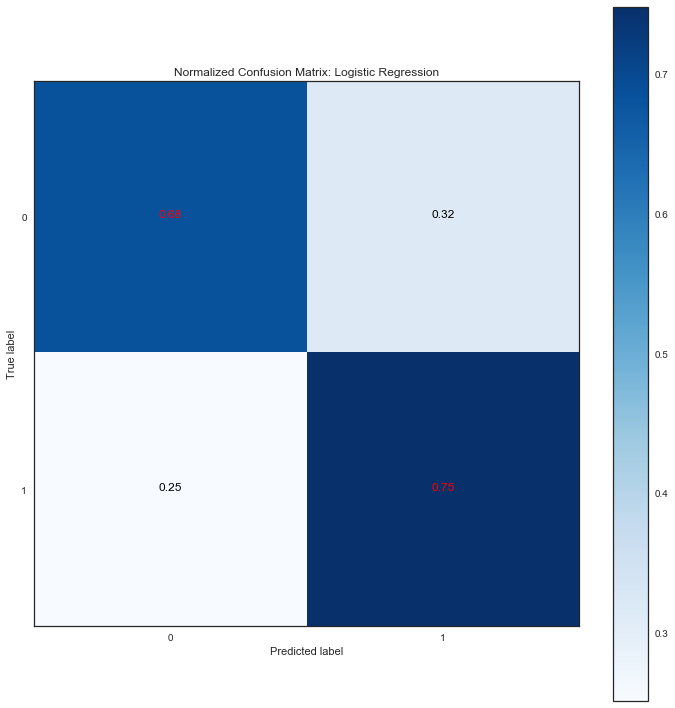
The confusion matrix is shown in Table 7, and the Type 1 error is 0.25, which is not high. A type 1 error means that a customer is predicted to be "bad" when the customer is actually "good". Type 1 errors should be minimized because penalizing good customers is very harmful.

***Table 6. XGBoost Model performance***

|  |  |
| --- | --- |
| **Accuracy** | 0.71 |
| **Mis-Classification** | 0.29 |
| **Sensitivity** | 0.68 |
| **Specificity** | 0.73 |
| **Precision** | 0.73 |
| **f1\_score** | 0.7 |
| **AUC** | 0.71 |

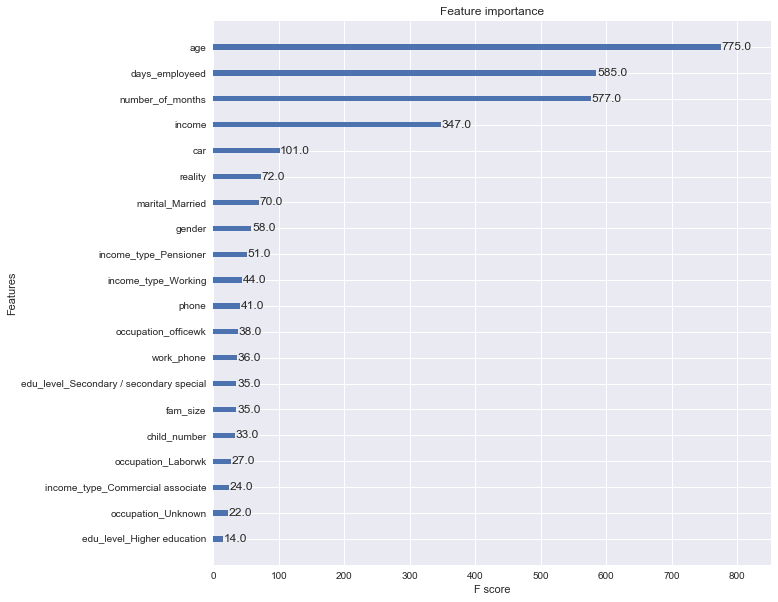
***Table 7. XGBoost Confusion matrix***

|  |  |  |
| --- | --- | --- |
|  | **Positive (1)** | **Negative(0)** |
| **Positive (1)** | 130 | 56 (FP)  Type I Error |
| **Negative(0)** | 48 (FN)  Type II Error | 119 |



***Figure 14. XGBoost Confusion matrix***

The importance of features is ranked in Figure 15. The top 5 important features are ‘age’, ‘days\_employeed’, ‘month\_balance’, ‘income’, and ‘car’. These are the important factors that affect the clients’ credit records and are the key factors to consider when deciding if the credit card application should be approved or not.



***Figure 15. Features importance***

**3.2 Decision Tree**

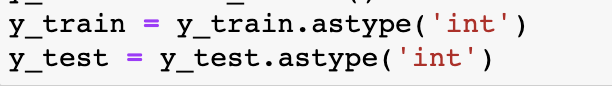
Decision Tree is a supervised learning model used in the decision prediction process, a decision tree consists of nodes and branches, each internal node in the tree represents a certain feature and each branch represents a decision rule. Since the algorithm mimics a human decision-making process, and unlike other machine learning algorithms like neural networks whose internal process cannot be seen, we could investigate the whole process of decision making in a certain decision tree. Therefore, the decision tree algorithm is easy to understand and interpret when facing classification problems.

The basic idea behind the decision tree algorithm is:

* In each node, select the best attribute using Attribute Selection Measures to split the dataset into two subsets.
* Repeating this process recursively until one of the condition matches:
  + All tuples belong to the same attribute
  + There are no more attributes remain
  + There are no more data remain

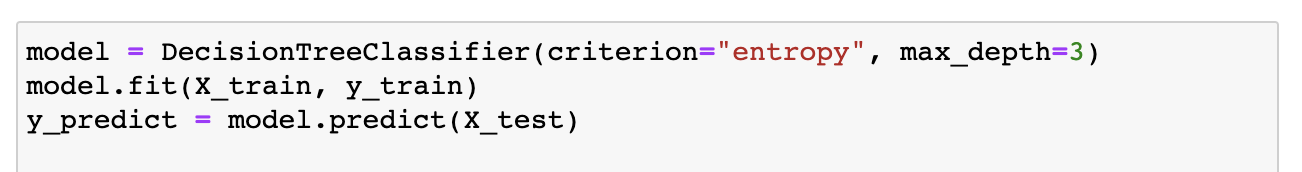
(Navlani, 2018)

Before putting the dataset into the decision tree algorithm, we first transform the data type of target variable into integer so that it can be transferred into binary values during building the model.



***Figure 16. Transfer the target data type***

To make a better interpretable decision tree, we train the model by selecting the Attribute Selection Measures to Information Gain (“entropy” in the model parameter) and set the max tree depth into 3.

******

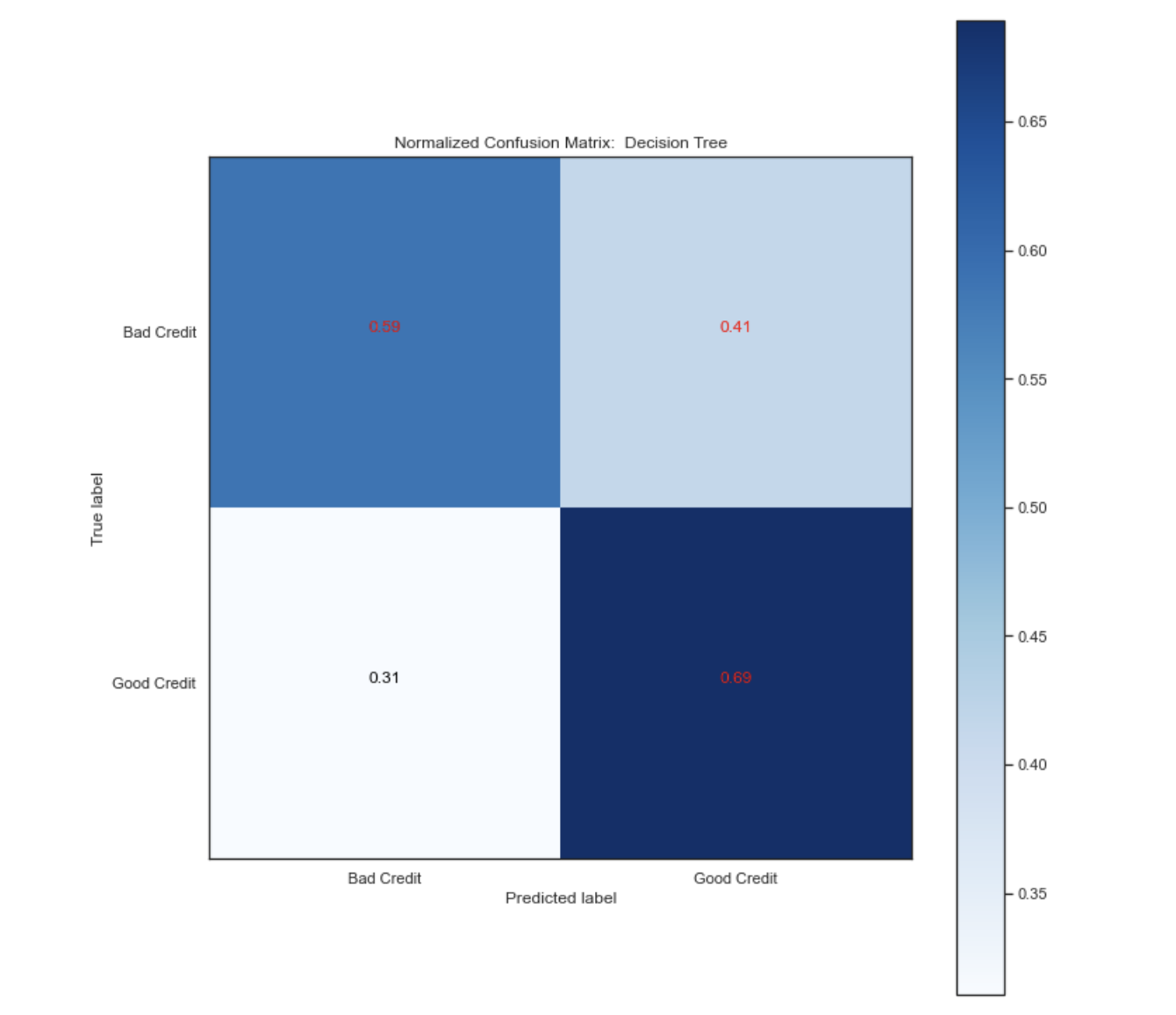
***Figure 17. The code in Building the Decision Tree***

The overall accuracy of the decision tree model is 0.65, the model summary is shown below. As we could see, the model has not enough data of customers who are the bad credit customers to predict, the next step we might consider is to put more data on customers who are bad credit and put them into training the model.

***Table 8. The performance summary of the decision tree model***

|  |  |
| --- | --- |
| **Accuracy** | 0.65 |
| **Mis-Classification** | 0.35 |
| **Sensitivity** | 0.69 |
| **Specificity** | 0.63 |
| **Precision** | 0.63 |
| **f1\_score** | 0.66 |
| **AUC** | 0.64 |

The confusion matrix is shown in Figure 18, we could see that the decision tree model performs better in predicting good credit customers than bad ones. In other words, the Type II error (0.31) is lower than the Type I error (0.41), it means that the model is less possible to falsely predict a good credit customer to a bad one than vice versa, which is reasonable to see from a financial institution’s perspective that they wish to punish bad credit customers as few as possible.



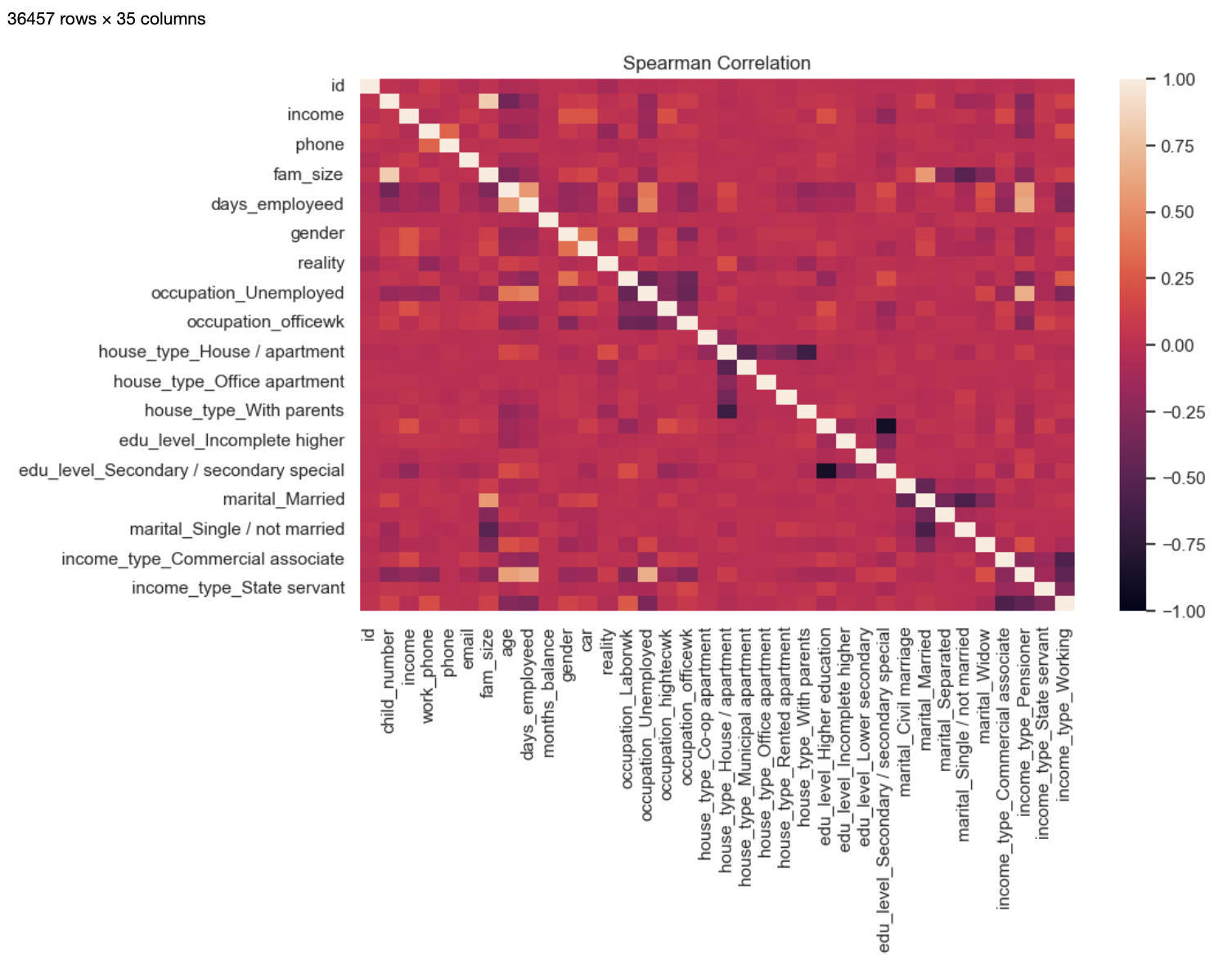
***Figure 18. The confusion matrix of the decision tree model***

**3.3 Logistic Regression**

Logistic Regression is a Machine learning classification algorithm that can determine the probability of a categorical dependent variable. Logistic regression can be used to describe the independent variables and explain the relationship between one dependent binary variable and multiple independent variables. The binary variables are encoded data as 1 (yes, positive) or 0 (no, negative). In our case, 1 means risky and bad clients who will have low credit card approval rates, and 0 means good clients who have good credit scores and high credit card approval rates. Before constructing a logistic regression model, our dataset, including dependent variable and independent variables should meet the following assumptions:

* For a binary regression model, the factor level “1” of the dependent variable should represent the desired outcome. In our case, banks need to pay extra attention to those risky clients who have low credit approval and avoid issuing credit cards to them to cause bad debt. The purpose for this logistic regression model is to detect any bad client. Thus, we define 1 as bad client.
* The independent variables should be independent of each other. An ideal logistic regression model should have little or no multicollinearity. Checking multicollinearity is essential before modeling.
* Logistic regression models should only contain meaningful variables.

Since there are some categorical variables in our dataset, Spearman rank correlation coefficient will be used to check multicollinearity. It measures the monotonic relation between two independent variables. Its values range from -1 to +1. -1 means a perfectly monotonically decreasing relationship. +1 means a perfectly monotonically increasing relationship. 0 means there is non-monotonic relation.



***Figure 19: Spearman rank correlation***

This correlation plot shows that there is no significant relationship between each variable. There are some negative monotonic relationships. However, it can be negligible in a dataset that contains 35 total attributes. Our dataset is eligible to build a logistic regression model.

In order to find the best logistic regression with highest accuracy, we define our model first using our train dataset containing every predictor variable. The parameter for our defined model is shown below:

***Table 9. Tuning parameter for best logistic regression with all variables***

|  |  |
| --- | --- |
| Parameters for best logistic regression model outcome | |
| C | 1 |
| Penalty | l1 |
| Solver | libnear |

Next, we use statsmodels to see what is the evidence that each dependent variable is related to our target variable (good or bad customers). And for the efficiency and simplicity of our logistic model, we will select only 4 variables that have the most significance on the outcome. The four variables are “days\_employed”, “number of months'', “car”, and “reality”.

We re-split our train and test data using these 4 variables and redefine our logistic model. The parameters for redefined logistic regression model are shown below:

***Table 10. Tuning parameter for best logistic regression with selected variables***

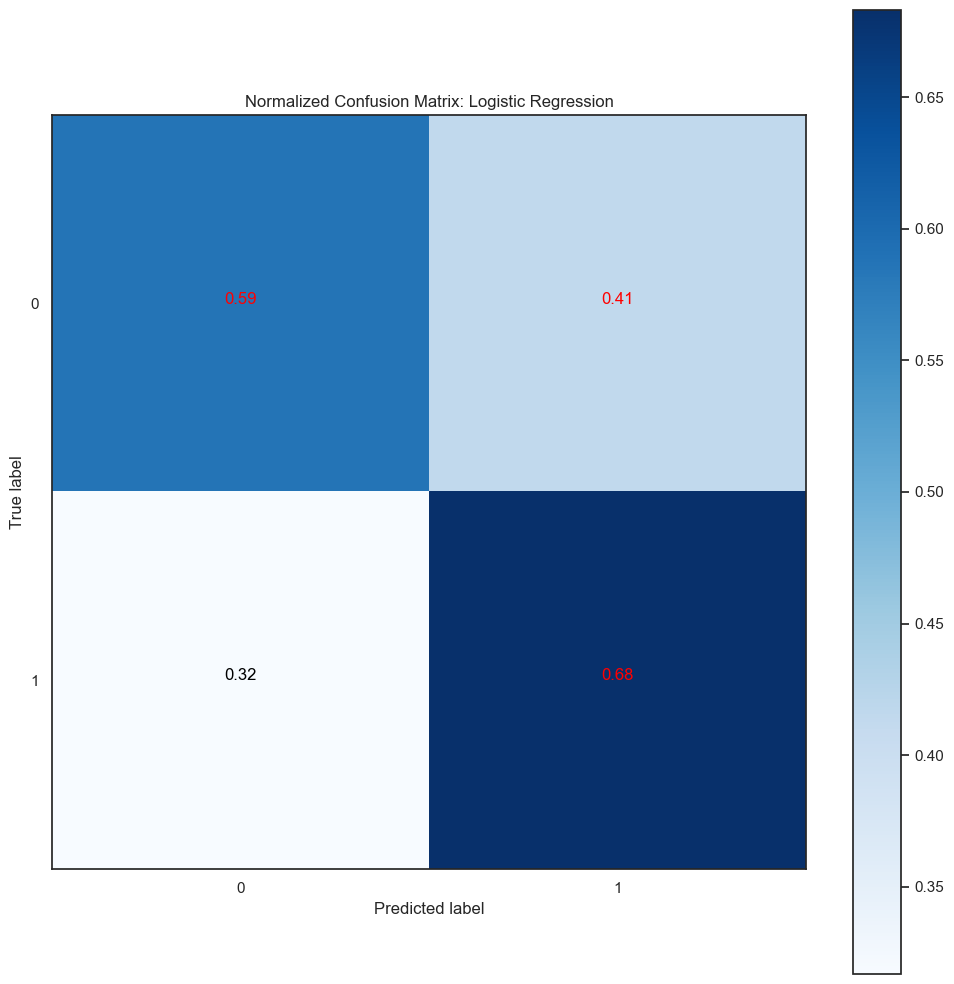
|  |  |
| --- | --- |
| Parameters for best logistic regression model outcome (4 variables) | |
| C | 0.1 |
| Penalty | l1 |
| Solver | liblinear |

The coefficient for our final logistic model is shown below:

***Table 11. Coefficients for our final logistic regression model***

|  |  |
| --- | --- |
|  | Coefficients |
| Days\_Employed | -2.61e-07 |
| Number of Months | -4.12e-02 |
| Car | 2.55-01 |
| Reality | 9.42e-01 |
| Intercept | 1.07 |

The confusion matrix plot is shown below:



***Figure 21: Confusion matrix of logistic regression 4 variables***

Tested performance is displayed below:

***Table 12. The performance summary for Logistic Regression Model***

|  |  |
| --- | --- |
| **Accuracy** | 0.64 |
| **Mis-Classification** | 0.36 |
| **Sensitivity** | 0.61 |
| **Specificity** | 0.68 |
| **Precision** | 0.68 |
| **f1\_score** | 0.64 |
| **AUC** | 0.64 |

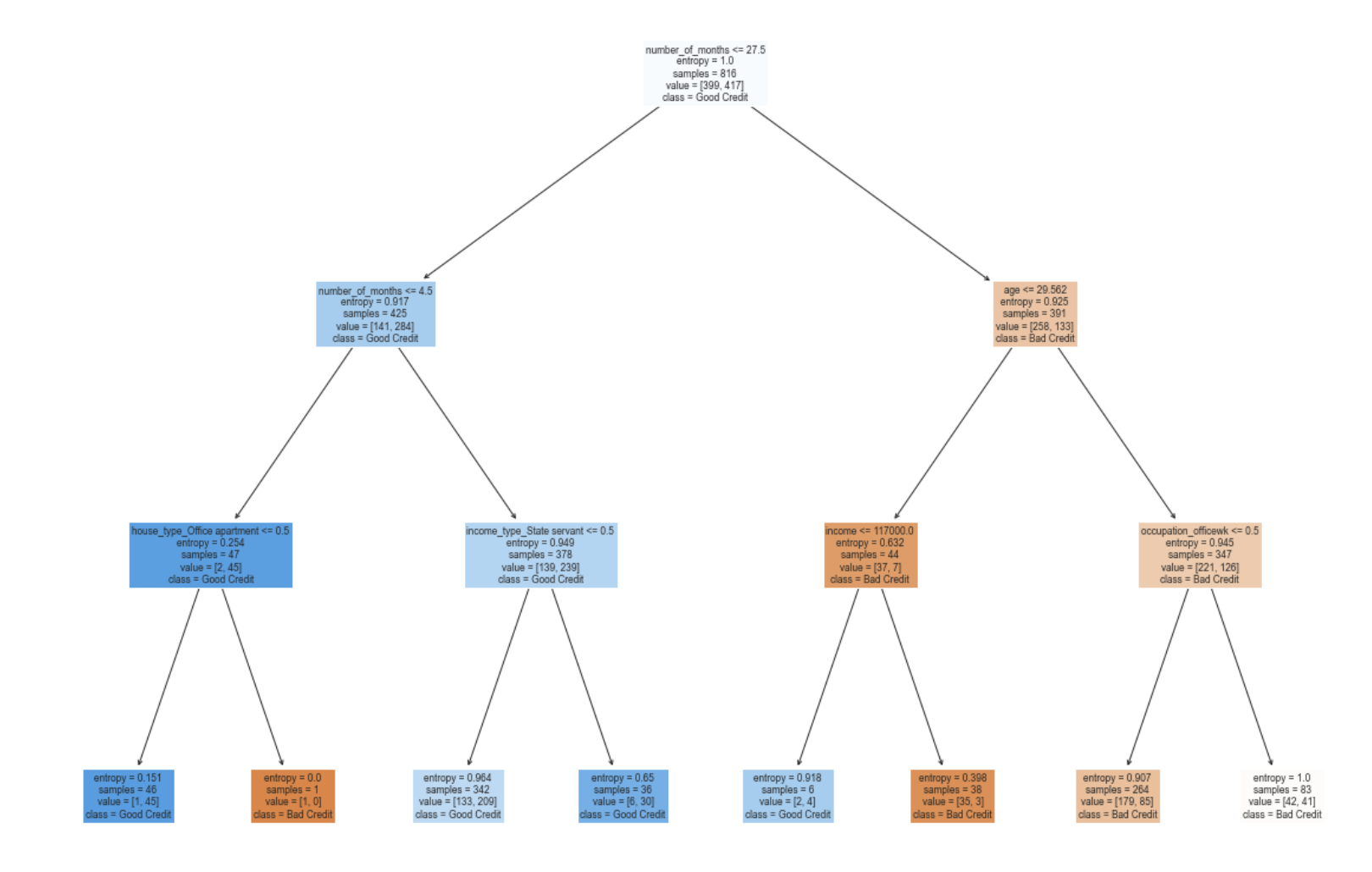
**4. Interpretive & Conclusions**

The model accuracies of XGBoost, Decision Tree and Logistic Regression are all around 0.65 to 0.7, their Type I error are all higher than Type II error, which means that it is easier to predict customers as "good" when they are actually "bad", to the detriment of the bank.

In the XGBoost model, the top 5 important features affecting the ‘target’ variable are ‘age’, ‘days\_employeed’, ‘month\_balance’, ‘income’, and ‘car’. These are the important factors that affect the clients’ credit records and are the key factors to consider when deciding if the credit card application should be approved or not. XGBoost has the highest accuracy and lowest Type 1 and Type 2 errors, indicating that it is a good model to predict if the customer is good or bad.

By looking at the visualization of the decision tree below, “number of months'', “age”, and “Income Type” are mainly considered in making predictions, and the model did good performance in predicting a good credit record customer as the final lead node in the bottom left corner, its entropy is low. When we look at the model’s performance summary in Table 9, even though the model’s accuracy is not as high as the one for the XGBoost, it still performs well in preventing falsely predicting a good customer to a bad one as its Type II error is similar with the XGBoost model.

In conclusion, after comparing the performance of three models implemented upon the dataset, **XGBoost** model would be the **best model** so far to predict whether a credit card user has good credit or not.



***Figure 22: Visualization of Decision Tree***

***Table 9. The summary of performance of three models***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Models** | **Accuracy** | **Precision** | **Sensitivity** | **Specificity** | **AUC** | **Type I Error**  **(FP)** | **Type II Error (FN)** |
| **XGBoost** | 0.71 | 0.75 | 0.68 | 0.75 | 0.72 | 56 | 48 |
| **Decision Tree** | 0.65 | 0.63 | 0.69 | 0.63 | 0.64 | 78 | 50 |
| **Logistic Regression** | 0.64 | 0.68 | 0.61 | 0.68 | 0.64 | 51 | 78 |

**5. Reference**

Ali, R. (n.d.). *Predictive Modeling: Types, Benefits, and Algorithms*. Oracle NetSuite. Retrieved

November 15, 2022, from https://www.netsuite.com/portal/resource/articles/financial-management/predictive-modeling.shtml

Gupta, S. (2022, June 9). *How to download kaggle datasets into Jupyter Notebook*. GeeksforGeeks. Retrieved November 27, 2022, from https://www.geeksforgeeks.org/how-to-download-kaggle-datasets-into-jupyter-notebook/

Navlani, A. (2018, December 28). *Python decision tree classification tutorial: Scikit-Learn Decisiontreeclassifier*. DataCamp. Retrieved November 28, 2022, from https://www.datacamp.com/tutorial/decision-tree-classification-python

Seanny. (2019). *Credit card approval prediction*. Kaggle. Retrieved November 27, 2022, from https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction?resource=download

XGBoost: What it is, and when to use it. KDnuggets. (n.d.). Retrieved November 27, 2022, from <https://www.kdnuggets.com/2020/12/xgboost-what-when.html>

Li, Susan. “Building A Logistic Regression in Python, Step by Step.” Medium, February 27,

2019.<https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8>.

Stack Abuse. “Calculating Spearman’s Rank Correlation Coefficient in Python with Pandas,” August 23, 2021.<https://stackabuse.com/calculating-spearmans-rank-correlation-coefficient-in-python-with-pandas/>.