

**MGT7179 Advanced Analytics & Machine Learning**

**Semester 2 – 2020/21**

**Assignment 2 – Classification**

Vasileios Gounaris-Bampaletsos

40314803

wordcount

4609

# Contents

[Contents 2](#_Toc70123751)

[Abstract 3](#_Toc70123752)

[Introduction 4](#_Toc70123753)

[Methodology 7](#_Toc70123754)

[Data Understanding 7](#_Toc70123755)

[Data Preparation 16](#_Toc70123756)

[Modeling – Evaluation 19](#_Toc70123757)

[Findings 27](#_Toc70123758)

[Data Understanding 27](#_Toc70123759)

[Data Preparation 27](#_Toc70123760)

[Correlations 28](#_Toc70123761)

[Visualisations 29](#_Toc70123762)

[Models-Predictions 33](#_Toc70123763)

[Conclusions 35](#_Toc70123764)

[References 36](#_Toc70123765)

[Appendix 37](#_Toc70123766)

[Code Photos & Graphs 37](#_Toc70123767)

[Final Visualisations 54](#_Toc70123768)

[Demographics 54](#_Toc70123769)

[Economic 58](#_Toc70123770)

[R Code 67](#_Toc70123771)

# Abstract

Society and business exist and operate in a dynamic and ever-changing environment. Technological developments and the rapid increase of resources have affected many sectors of society, which are looking for ways to be functional and reliable. One such area is financial institutions. Banks are affected and study the financial threats from credit risk, as it seems to be the trend of the time. Banks want to find ways to predict the movements of their customers (late payments, delinquent debts), through their data from the past. For this analysis, traditional statistical media are not functional as the data is often unbalanced, thus affecting the final analysis. For this purpose, this work focuses on the application of machine learning models, so as to achieve better predictions aimed at preparing the bank branches. Data from the city of Taiwan were used, with 30,000 entries and 25 variables, covering data from April 2005 to September 2005. The methods used are Logistic Regression, Random Forest, Decision Tree, Support Vector Machine (linear, gaussian (radial kernel)), XGBoost, Naive Bayes. The best methods were Decision Tree and SVM(kernel), with accuracy 82.04% and 82.08% respectively. This work and the models that have been developed, can be well used in the real world. This is because the data used is from the real world, in real conditions and scenarios. The strongest model, then, could help banks and financial institutions and other relevant agencies to predict the loan defaulter earlier and more accurately.

KeyWords: default payment, banks, machine learning, classification

# Introduction

The financial sector has always had the risk of high default rate on credit loans. Banks and the financial sector need a tool to be able to predict with relative accuracy what will happen in the past and therefore be ready without affecting the economy. According to (Alam, et al., 2020), the default rate index has reached historically high levels and is projected to jump to higher rates in the near future. This trend creates a problem for the banks as money is lost by them creating instability. As they report (Kim, et al., 2018) in their example for the United States of America, in 2013 consumers spent 69% of the USA domestic gross product, out of about 3 trillion, more than 25% was from credit cards, thus a slight increase in accuracy in finding risky loans could prevent financial disasters. Thus, building a wall against the unpredictable default credit cards will improve and reduce the risk of possible collapse of their financial system. This will be done and is being done with the help of technology, engineering learning and artificial intelligence, which are created to add important predictions with an absolute sense of responsibility because the issue is serious.

The problem that banks and financial institutions have to deal with is serious and needs to be addressed immediately. It is important to have a tool with which they can predict the habits and actions of their customers, analyzing what they have done in the past. The problem is the inability of consumers to pay their credit debts to banks. For this reason this work comes to give a fairly correct solution, using statistical models of analysis of data from the city of Taiwan (2005) as well as the creation of machine learning models to predict consumer movements. As mentioned (Yeh & Lien, 2009) banks in Taiwan have a problem with credit cards because non-payment of installments leads to the cessation of issuing new cards. This means that the financial system collapses as the cash flow cycle stops, while those already holding cards continue to owe money to the banks. As mentioned in (Delamaire, et al., 2009) the main concern of financial institutions is to separate the types of debts and focus on the main ones, with a plan and provisions, so that this problem does not exist in the future and at the same time debt creation does not continue.

To understand the problem of non-payment of credit cards, let us give an example. (Delamaire, et al., 2009) mentions the examples of countries that have had, since the beginning of the century, the issuance of credit cards free and unchecked. Among these countries are Greece, Italy and Spain, which experienced the greatest recession in their economies during the economic crisis (2009-2019). So it seems that the widespread use of credit cards and loans without limits and restrictions, brings wrong results in entire countries. Thus, the need to control and anticipate such situations is imperative, because many countries can no longer withstand such economic pressure.

The problem that this work is trying to solve is to find reliable forecasting models so that they can be used by banks around the world. Using these models, banks will be able to organize and plan a highly sustainable future for their internal finances as well as expansion for the national economy of each country. Also the purchasing power of consumers will be shared in an optimal way so that the flow and money cycle is not interrupted by overdue debts and inability to pay credit debts.

Many efforts have been made in the last decade, by science teams and by the financial institutions themselves (banks, lenders), to find reliable models for predicting consumer decision (default or not). These efforts involve tests and trials on a variety of real-world data using different machine learning models. Each team has a different approach to which models it will develop, which is mainly influenced by the data they have at their disposal but also the computing power they can use. Thus, it seems that some surveys use "traditional" algorithms such as Logistic Regression, Naive Bayes, Decision Trees and Random Forest, achieving very good scores, thus significantly helping banks to achieve their goal (Yeh & Lien, 2009; Delamaire, et al., 2009). On the other hand, the use of neural networks (Liu, 2018; Hsu, et al., 2019) to find more accurate predictions is increasingly observed. These algorithms, although powerful and very accurate for classification problems (like the one this work solves), need a lot of computing power and big enough data to perform better and produce accurate predictions. There are also several interesting models, such as JRIP (Repeated Incremental Prunning) which was used in (Sharma & Mehra, 2018) and has a different approach to model development. So it seems that researchers have done a lot of work and research to achieve a good result for this problem. Many different algorithms have been used to help banks with forecasts.

As with most predictive modeling projects, the first task in this work is to understand the data. This is achieved through various descriptive statistics and graphs (histograms). The data were then plotted on graphs for better understanding and then the correlations between the variables were found. It was the turn of finding errors and preparing the data for the final phase of the work. This phase is the development of engineering learning models, their evaluation and the final selection of the best model. Then the final conclusions and the main findings of the analysis that was done for this project are presented and finally some tips and improvements that could be made in a new research in the future are given.

Finally, the preparation of the project from a technical point of view, finding and studying the right tools for data analysis was done using a manual (Kuhn & Johnson, 2013). This book offers important insights into the development of machine learning models while teaching the reader the statistical background needed for such a project.

# Methodology

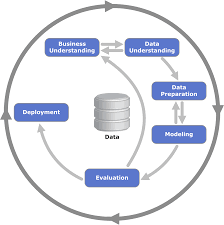
The implementation of this work was done using the CRISP-DM method (Figure 1). This method was followed as the data were real data from the city of Taiwan and so the degree of difficulty was high. For this reason the data had to be displayed first in order to be understood in depth and to understand the business problem that had to be solved.

Figure 1: CRISP-DM process diagram (source:(<https://www.google.com/search?q=crisp+dm&sxsrf=ALeKk01H22jQ7VZGaQejpMv1B0HosOGusA:1616777415908&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiag7mctc7vAhXM4IUKHRGrAgAQ_AUoAXoECAEQAw&biw=1920&bih=962#imgrc=m1Hh09d2zPNgBM>))

As described in the continuation of the work, the problem was first understood through some descriptive statistics and a first representation of the data (histograms), then an Explrotary Data Analysis (EDA) was done along with more specific graphs and a correlation table to give a first answer. in the past behavior of bank customers. Then the variables that would make up the mechanical learning models were selected. Then these models were implemented and evaluated and finally the most appropriate one was selected.

## Data Understanding

Initially, the data were entered in RStudio and their first display was done. The data from the city of Taiwan had 30,000 entries and 25 variables, with the variable target (default.payment.next.month). In the following table are presented for the first time all the variables as well as some of their elements (Table 1) also this (code photo 1) presents the structure of the dataset.

**Table 1: Dataset Dictionary Table**

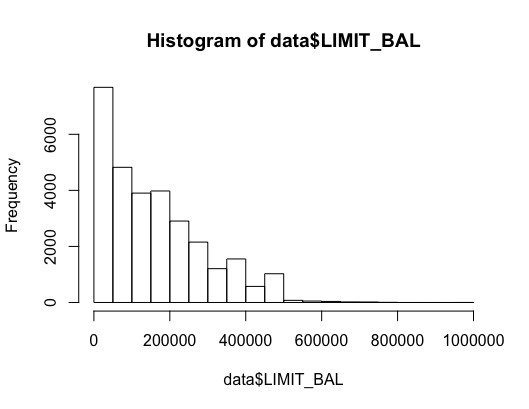
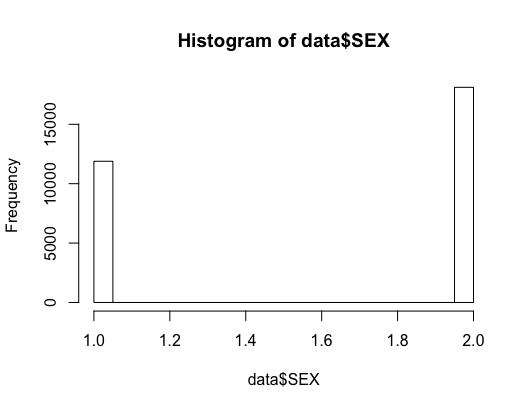
|  |  |  |
| --- | --- | --- |
| **Variables** | **Type** | **Measurement** |
| ID | int | Id of each client |
| LIMIT\_BAL | Num | Amount of given credit in NT dollars |
| SEX | Int | Gender(1=male, 2=female) |
| EDUCATION | Int | Education level (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown) |
| MARRIAGE | Int | Marital status(1=married, 2=single, 3=others) |
| AGE | Int | Age of clients in years |
| PAY\_0 | Int | Repayment status in September 2005(-1=pay duly, 1=payment delay for one month, 2=delay for 2 months,…..9=delay for 9 months and above) |
| PAY\_2 | Int | Repayment status in August 2005(same scale) |
| PAY\_3 | Int | Repayment status in July 2005(same scale) |
| PAY\_4 | Int | Repayment status in June 2005(same scale) |
| PAY\_5 | Int | Repayment status in May 2005(same scale) |
| PAY\_6 | Int | Repayment status in April 2005(same scale) |
| BILL\_AMT1 | Num | Amount of bill statement in September 2005 (NT dollar) |
| BILL\_AMT2 | Num | Amount of bill statement in August 2005 (NT dollar) |
| BILL\_AMT3 | Num | Amount of bill statement in July 2005 (NT dollar) |
| BILL\_AMT4 | Num | Amount of bill statement in June 2005 (NT dollar) |
| BILL\_AMT5 | Num | Amount of bill statement in May 2005 (NT dollar) |
| BILL\_AMT6 | Num | Amount of bill statement in April 2005 (NT dollar) |
| PAY\_AMT1 | Num | Amount of previous payment in September 2005 (NT dollar) |
| PAY\_AMT2 | num | Amount of previous payment in August 2005 (NT dollar) |
| PAY\_AMT3 | Num | Amount of previous payment in July 2005 (NT dollar) |
| PAY\_AMT4 | Num | Amount of previous payment in June 2005 (NT dollar) |
| PAY\_AMT5 | Num | Amount of previous payment in May 2005 (NT dollar) |
| PAY\_AMT6 | Num | Amount of previous payment in April 2005 (NT dollar) |
| Default.payment.next.month | int | Default payment (1=yes, 0=no) |

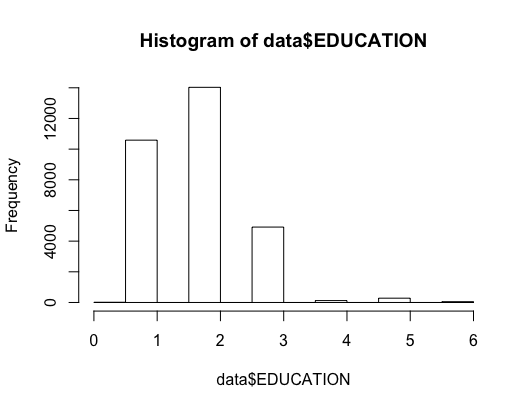
The next move was to check the data for missing values ​​(NAs). As shown in (Code Photos 2, Code Photos 3) there are no NAs anywhere in the entire dataset. Then the descriptive statistics started to see the values ​​and limits of each variable, with the aim of first understanding the data and then dealing with possible outliers. For this reason the summary () command was used first in all the data (Code Photos 4) and then one by one the variables (Code Photos 5, Code Photos 6), for better understanding, and the results are shown in the following table (table 2).

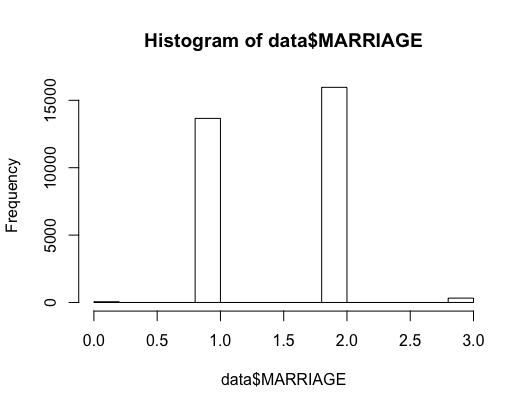
Table 2: Raw data descriptive statistics (source: my R code)

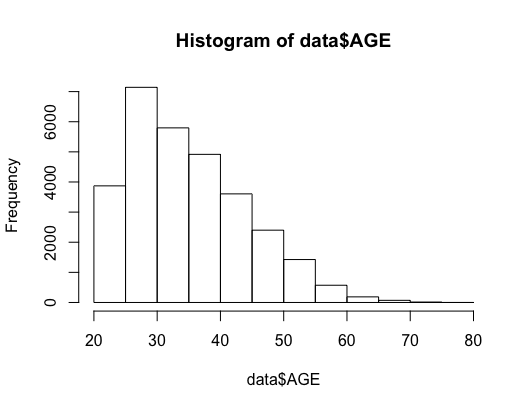
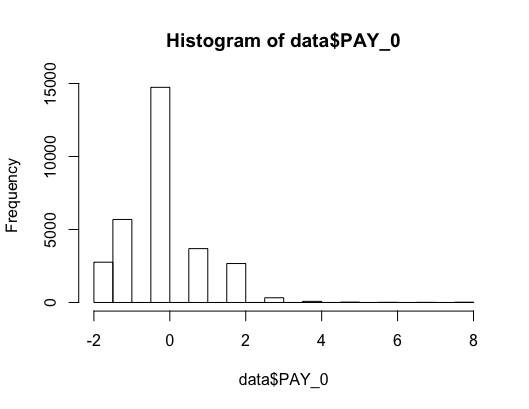
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Variables** | **Missing Values** | **Min.** | **1st Quartile** | **Median** | **Mean** | **3rd Quartile** | **Max.** |
| ID | 0 | 1 | 7501 | 15000 | 15000 | 22500 | 30000 |
| LIMIT\_BAL | 0 | 10000 | 50000 | 140000 | 167484 | 240000 | 1000000 |
| SEX | 0 | 1 | 1 | 2 | 1.604 | 2 | 2 |
| EDUCATION | 0 | 0 | 1 | 2 | 1,853 | 2 | 6 |
| MARRIAGE | 0 | 0 | 1 | 2 | 1.552 | 2 | 3 |
| AGE | 0 | 21 | 28 | 34 | 35.49 | 41 | 79 |
| PAY\_0 | 0 | -2 | -1 | 0 | -0.0167 | 0 | 8 |
| PAY\_2 | 0 | -2 | -1 | 0 | -0.1338 | 0 | 8 |
| PAY\_3 | 0 | -2 | -1 | 0 | -0.1662 | 0 | 8 |
| PAY\_4 | 0 | -2 | -1 | 0 | -0.2207 | 0 | 8 |
| PAY\_5 | 0 | -2 | -1 | 0 | -0.2662 | 0 | 8 |
| PAY\_6 | 0 | -2 | -1 | 0 | -0.2911 | 0 | 8 |
| BILL\_AMT1 | 0 | -165580 | 3559 | 22383 | 51223 | 67091 | 964511 |
| BILL\_AMT2 | 0 | -69777 | 2985 | 21200 | 49179 | 64006 | 983931 |
| BILL\_AMT3 | 0 | -157264 | 2666 | 20088 | 47013 | 60165 | 1664089 |
| BILL\_AMT4 | 0 | -170000 | 2327 | 19052 | 43263 | 54506 | 891586 |
| BILL\_AMT5 | 0 | -81334 | 1763 | 18104 | 40311 | 50190 | 927171 |
| BILL\_AMT6 | 0 | -339603 | 1256 | 17071 | 38872 | 49198 | 961664 |
| PAY\_AMT1 | 0 | 0 | 1000 | 2100 | 5664 | 5006 | 873552 |
| PAY\_AMT2 | 0 | 0 | 833 | 2009 | 5921 | 5000 | 1684259 |
| PAY\_AMT3 | 0 | 0 | 390 | 1800 | 5226 | 4505 | 896040 |
| PAY\_AMT4 | 0 | 0 | 296 | 1500 | 4826 | 4013 | 621000 |
| PAY\_AMT5 | 0 | 0 | 252.5 | 1500 | 4799.4 | 4031.5 | 426529 |
| PAY\_AMT6 | 0 | 0 | 117.8 | 1500 | 5215.5 | 4000 | 528666 |
| Default.payment  Next.month | 0 | 0 | 1 | 0 | 0.2212 | 0 | 1 |

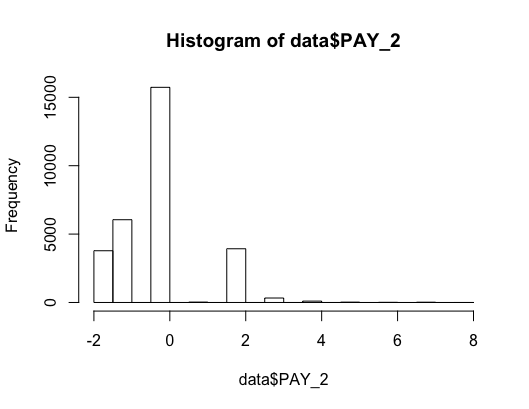
The first problems appear (colored in orange on the board) and will be solved in the next step of the task. First, from the correction and preparation of the data, simple histograms were implemented to better show the distribution of each variable and to better capture the existing errors. For this reason the following histograms were created (their analysis will be done in the next section).

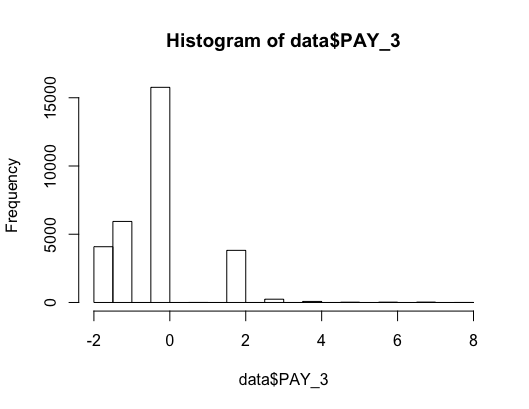


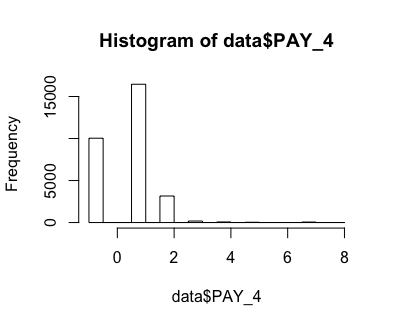


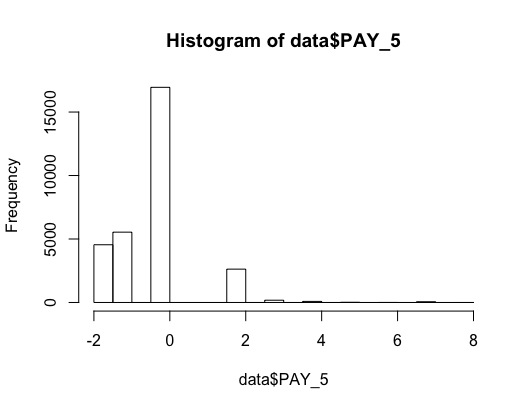


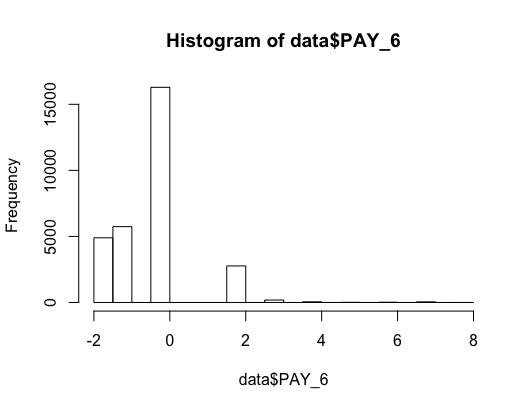
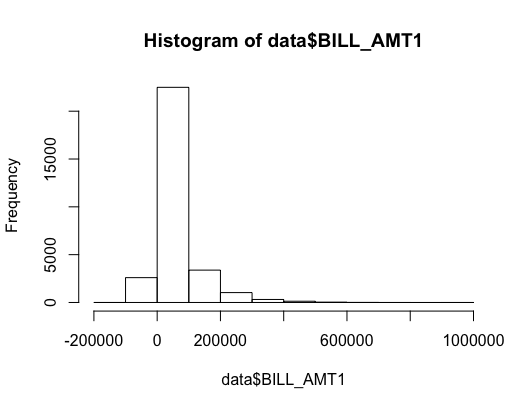


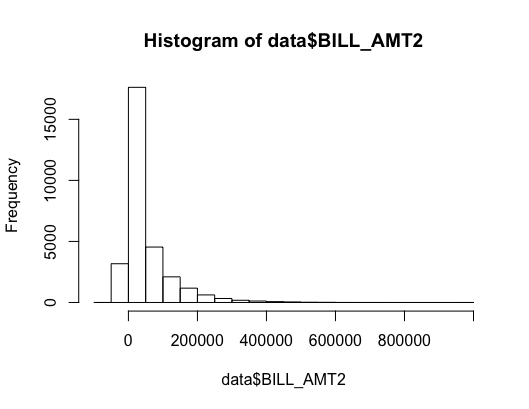


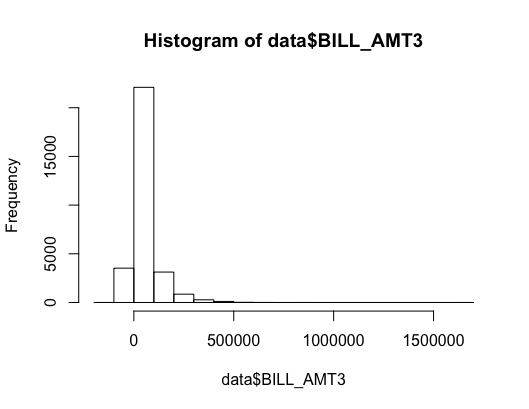


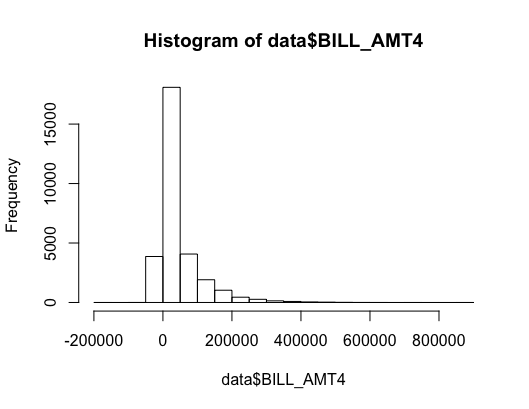


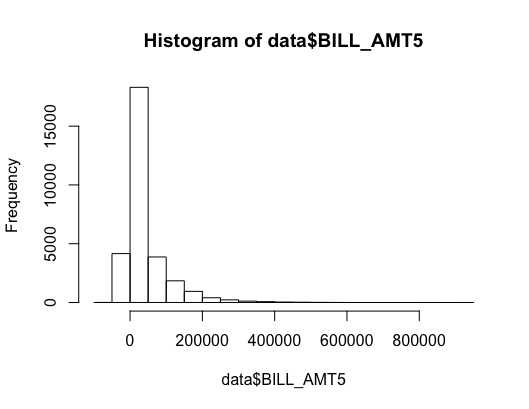


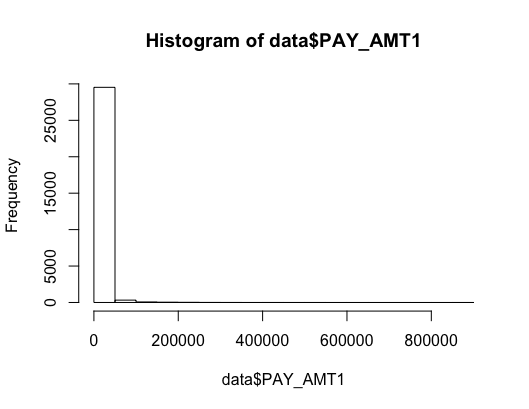
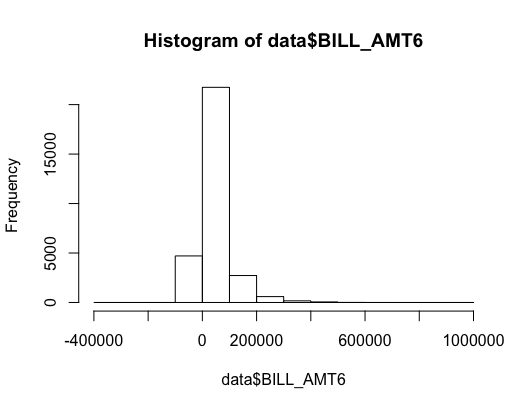


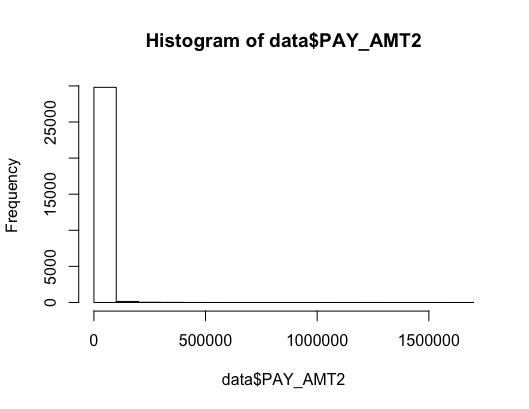


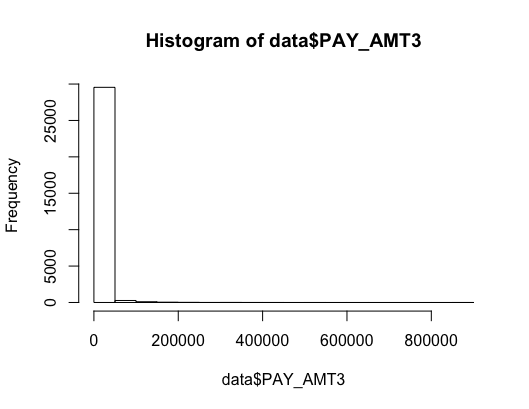


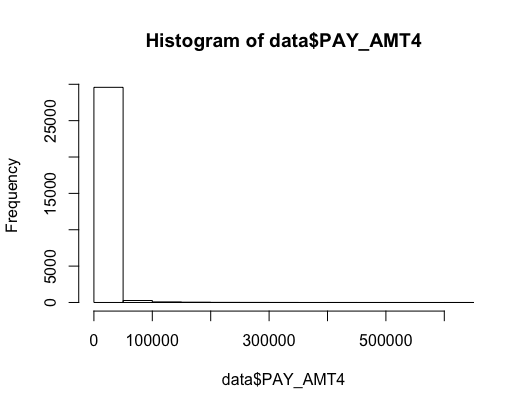


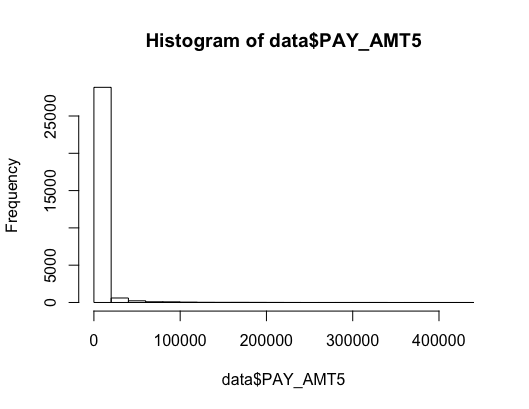


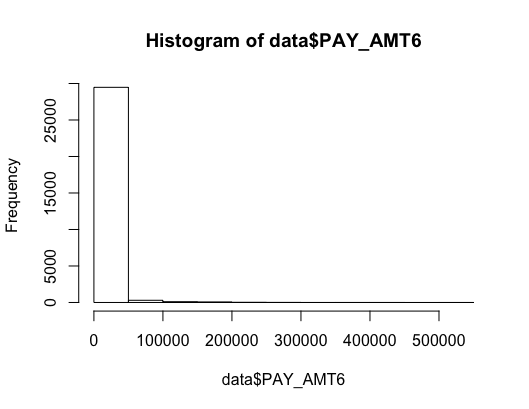
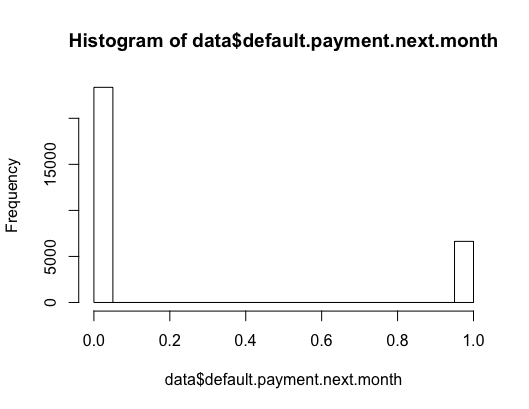






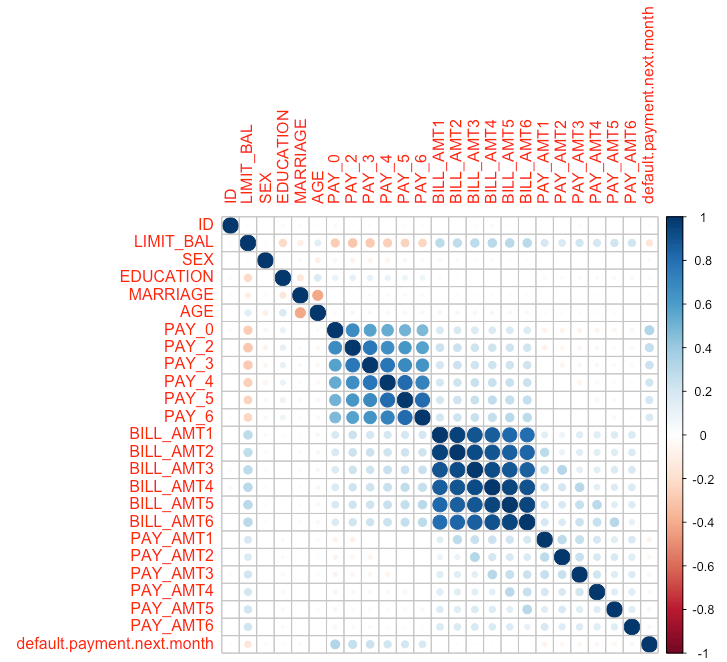






Then the correlation table was created. This shows the correlations between the variables. This is important because knowledge about variables is acquired. The correlation was made based on the pearson method. The analysis of the correlations will be done in the next section. Below are the whole table (Figure 2) and the half table (Figure 3). In this way the correlations are better observed and the variables are understood. The aim is to understand which variables affect or are influenced by each other.

Figure 2: Correlation Matrix(FULL) (source: my R code)



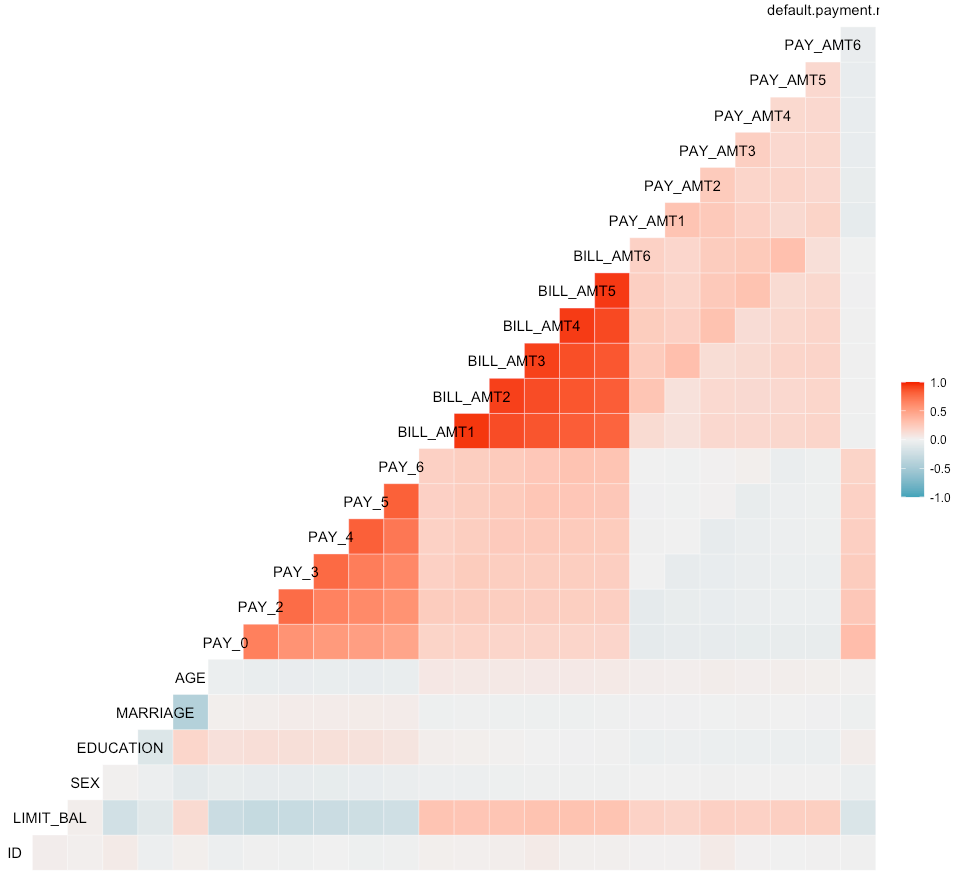


Figure 3: Correlation Matrix(Half)(source: my R code)

## Data Preparation

At this stage, the changes and corrections made to the data will be presented, so that they are ready and correct for the creation of the mechanical learning models. In terms of accuracy, this means that there are no significant errors (outliers), and the values of the variables go hand in hand with the dictionary. The changes and the explanation are presented in the following table (**Table 3**) and follow the flow that exists in the code.

**Table 3: Fix the Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Problem** | **Fix** | **Explanation** |
| Default.payment.next  Month | There is any problem but just rename it to TARGET. It is smaller and good to understand | Rename to TARGET | - |
| 1. PAY\_0 2. PAY\_1 - 6 | 1. It is not good to have pay\_0 and after pay\_2. 2. there are values which there are anywhere in data dictionary(-2 and 0) | 1. Rename to PAY\_1 2. combine the -2 and -1 in -1 value and 0 and 1 in 1 value. | 1. It is more convinient to have pay1 and pay2…. 2. -1 means that the payment is paying on time wihtout delay. Probably it is human error and I believe that it is the best aproach. Also the 0 probably means no delays but in variable 1 there are almost any values. So I combimne (-2, -1 to -1) and (0, 1 to 1). The distribution after this, seems very good. |
| SEX | NO problems | Rename 1-2 to Male-Female | It is more undestandable to have characters |
| EDUCATION | There are 3 “same” values(4others, 5unknown, 6unknown) and 1 “new” value(0). | Combine 0,4,5,6 values and put them in value 4=others  Also rename the 1-2-3-4 values into the characters from dictionary(graduate school, university, high school, others) | In dictionary there are values from 1to6. In the data there is a new value(0). Also there are values with the same meaning(4,5,6 unknown-others). My approach it to combine them in the 4 value = others |
| MARRIAGE | There is a new value in dataset (0) | Combine the new value(0) with value 3(others=divorced,etc) | It was the best approach this combination because the 0 value contains unknown values in it. |

It was not possible to delete the new values ​​-2 and 0, as they are a terribly large percentage of the data. If the value -2.0 was deleted from PAY\_1-6, then 4300 values ​​from 30000 would be left. It is clear that a decision had to be made. An alternative position for these prices was to remain as they are but again in the end it had to be decided what these new prices would symbolize. The strategy followed was probably the most ideal, which will be seen in the next section. The same goes for new values ​​and mergers made on variables EDUCATION and MARRIAGE.

Then the variables PAY\_1-6, SEX, EDUCATION, MARRIAGE, TARGET were converted from integer to factor because they contain values which although numbers (1,2,3,4 ...) correspond to characters therefore this step is important for the development of classification models. Also the variable ID was deleted from the data because it has no value to be used in the models, because it simply contains the number of each customer of the bank. Then, using the table () command, a final attempt was made, after preparing the data, to capture the distribution of the values ​​of the variables which are factors (Code Photos 7, Code Photos 8).

The last and equally important step of understanding the data are the graphs and the representations of the variables. Graphs were used to understand the demographic characteristics of the data concerning the bank's customers (age, gender, education, marriage) and then graphs were made regarding the financial variables and the variables of payment delays (See Appendix-> Final Visualisations). A detailed explanation will follow in the next section.

## Modeling – Evaluation

After the preparation of the data and the creation of the descriptive statistics is completed, it is the turn of the creation of the model predictions. The models will enable banks to predict whether customers are willing to pay their debts next month or not. 7 mechanical learning models were used as well as performance enhancement algorithms of the models such as cross-validation. Initially the data were divided into 2 subsets, train\_set and test\_set, with ratios of 75-25 respectively (**Table 4**).

**Table 4: Datasets after splitting**

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Ratio | Observations | Variables |
| Train\_set | 75% | 22500 | 24 |
| Test\_set | 25% | 7500 | 24 |

This separation was chosen among others (for example 70-30, 80-20) because the data is enough and in this way we give enough data (75% of the total) to the model to train and enough data (25% of the total) to test the model based on the predictions it has made. This helped to achieve better accuracy in almost all the models developed.

The Bayesian Information Criterion (BIC) algorithm was then used to find the most appropriate set of variables from the data, in order to contribute everything to the model, using forward selection. The end result was that all the data variables had to be used, as this algorithm gave as a result (Figure 4, Figure 5).

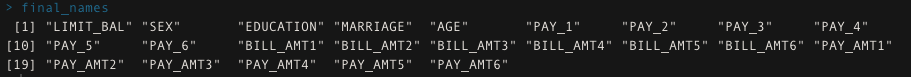


Figure 4: Best Variables for the Models (source: my R code)

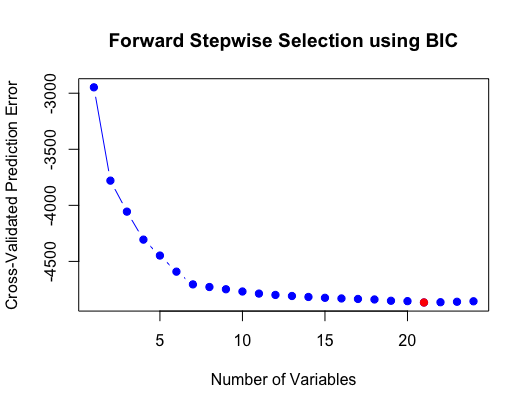


Figure 5: Graph of variables selection using BIC

Also the models’ evaluation been done with the confusion matrix. This matrix contains a lot of significant statistics about the models’ performance but for this project used only 5. (**Table 5**).

**Table 5: Confusion Matrix and Statistics: Explanation**

|  |  |  |
| --- | --- | --- |
| **Statistics** | **Explanation** | **Target Value** |
| Accuracy | It presents how accurate is the model in their predictions. The most valuable value for evaluation and comparison. | Close to 1.  (0 to 1)or(0%to100%) |
| Kappa | It presents how well the classifier preformed as compared to how well it would have performed by chance. | High kappa means that a model has a big difference between the accuracy and the null error rate. (-1 to 1) |
| Test P-Value | Presents the significance of our variables in the model | As lower as better. |
| Sensitivity | It is the number of correct positive predictions divided by the total number of positives. (=recall) | Close to 1  (0 to 1) |
| Specificity | It is the number of correct negative predictions divided by the total number of negatives. | Close to 1.  (0 to 1) |
| Pos Pred Value(Precision) | It is the number of correct positive predictions devided by the total number of positive predictions. | Close to 1. |
| Neg Pred Value | It is the number of incorrect positive predictions divided by the total number of negatives. | Close to 0. |
| Positive Class | It shows the most famous class in the predictions | There is not a target variable. In this project this class was the 0, because this answer had the most appiarences in dataset. |

*Logistic Regression*

It is the most recognizable algorithm for classification problems and the start of each project from what was read in the introduction to understand and find information about the problem. It is used to model the probability of a certain class(yes, no =1,0) Assumptions were also checked to see if the model and data violated them. This was done to simulate as much as possible the process of analyzing and developing machine learning models, in a problem in real conditions (**Table 6**, **Table 7**).

**Table 6: Logistic Regression and Evaluation Tools (sournce: my R code)**

|  |  |  |
| --- | --- | --- |
| Model | Evaluation Tools | Explanation |
| Logistic Regression Model(Code Photos 9) | Confusion Matrix (Code Photos 10) | It contains everything need to evaluate the model’s performance |
| Odds for the model(Code Photos 11) | Greater than 1 = is positive association (higher number for the predictor means that is more possible to predict 1)  Lower than 1 = is negative association (the opposite = more possible to predict 0) |
| Pseudo R^2 (Code Photos 12) | The metrices in this tool must be close to 0. |
| 1. ROC curve(Code Photos 13) 2. Recall-Precision curve(Code Photos 14) | 1. This curve shows the trade-off between sensitivity and specificity. Target curve = closer to the top-left corner. 2. It is the calculation and plotting the precision against the recall(from confusion matrix)   Target curve = begins from top-left corner horizontically to the top-right corner(auc=1). |
| AUC value(Code Photos 15) | It shows the probability that a positive example goes to the right of a negative example. Target value = close to 1 |

**Table 7: Assumptions Checking for Logistic Regression Model**

|  |  |  |
| --- | --- | --- |
| **Assumption** | **Checking Method** | **Result** |
| Linearity | Logarithmic from variables | Assumptions violated because some varibales for example age and education can be in groups |
| Infuential Values | Cook’s Distance(Code Photos 16)  Top 3 values(Code Photos 17)  Potential influence variables(Code Photos 18) | There are no values which are violate the cook’s distance. 3 values are high and for that reason it checked the potential influence values.  ASSUMPTION: OK |
| Multicolinearity | Residuals Fit(Code Photos 19)  Variance Infication Factor(VIF) (Code Photos 20) | 917 values >1.96  1500 values = 0.05 of the total.  917<1500 = OK  The BILL\_AMT1-6 and PAY\_3-5 violate the GVIF>2 |

*Random Forest*

This method is an ensemble learning method for classification and creates a multitude of decision trees in the training time and as result they give the class that is the mode of the classes of the individual trees. This algorithm is more powerful than a decision tree but it depends from the dataset and the variables. In this project are used two random forest algorithms. The first was developed with the simple method(without tuning) and number of trees(depth) equal to 10. After the creation of the model, the cross validation tuning was used to tune the model with k-folds equal to 10. Also was trying to use k=20 but the result(accuracy) was not so better (**Table 8**).

**Table 8: Random Forest Classifier**

|  |  |
| --- | --- |
| **Model** | **Evalutation Tools** |
| Random Forest(Code Photos 21) | Confusion Matrix(Code Photos 22) |
| Cross Validation (k=10) (Code Photos 23) |
| Cross Validation (k=20) (Code Photos 24) |

*Decision Tree*

It is one of the most famous algorithm for classification problems. The algorithm uses a decision tree to go from a value about an item to conclusions about the item’s target value(leaves). The leaves of the tree are the class labeles and the branches are conjuctions of features that lead to those class labels. (**Table 9**)

**Table 9: Decision Tree Model**

|  |  |
| --- | --- |
| **Model** | **Evaluation Tools** |
| Decision Tree Model (Code Photos 25) | Confusion Matrix(Code Photos 26) |
| Decision Tree Graph (Code Photos 27) |

*Support Vector Machine (SVM) – linear*

It is a supervised learning model with associated learning algoritmhs that analyze and explore the data set for regression or classification. The linear SVM take a set from the training set, each marked as belonging to the two categories, and the SVM training example create a model that put new example in one of these 2 categories. With this thing, it creates a non-probanilistic binary linear classifier. Next the algorithm present the training examples to some point in space to find the maximum width of the distance between the 2 categories. The new example then is created and it was in the same space and prediction area with the value which we are looking for.(**Table 10**)

**Table 10: SVM Model (linear)**

|  |  |
| --- | --- |
| **Model** | **Evaluation Tools** |
| Support Vector Machine (linear) (Code Photos 28) | Confusion Matrix(Code Photos 29) |

*Support Vector Machine (SVM) (kernel-gaussian)*

The only difference with the linear SVM is that the kernel SVM can perform a non-linear classification because it implicitly mapping its inputs into a high-dimensional feature space. This helps to maximize the margin of hyperplanes with the help of Gaussian radial basis function. (**Table 11**)

**Table 11: SVM Model (kernel)**

|  |  |
| --- | --- |
| **Model** | **Evaluation Tool** |
| Support Vector Machine (kernel-gaussian bias) (Code Photos 30) | Confusion Matrix (Code Photos 31) |

*Naïve Bayes*

It is a method for classification and it is a simple probabilistic classifier and it is using the bayes’ theorem with strong feeling of independence assumptions between the values. This type of classifier is scalable and requires some parameters linear in the number of variables in a problem. (**Table 12**)

**Table 12: Naive Bayes Model**

|  |  |
| --- | --- |
| **Model** | **Evaluation Tool** |
| Naïve Bayes Model (Code Photos 32) | Confusion Matrix (Code Photos 33) |

*XGBoost*

This method is a decision-tree-based machine learning algorithm and it based in boosted trees algorithms. It is a faster and with better performance edition of gradient boosted machine algorithms. Also in this project, this model was tuned with cross validation with k=5(I want to put higher 10,20 or more, but it took so many time to run and my computer is weak for these models). (**Table 13**)

**Table 13: XGBoost Model (source: my R code)**

|  |  |
| --- | --- |
| **Model** | **Evaluation Tool** |
| XGBoost(Code Photos 34) | Confusion Matrix() |

# Findings

This section will present the results of data analysis (EDA, graphs), evaluate machine learning models and provide answers and solutions based on these results to banks that want to use the logic of this project. The question that had to be answered with this project was what variables affect the decision (possibility) of bank customers not to pay or to delay the payment of their credit debts. So the result, as presented below, had to include those variables (demographic, financial) that affect customers so that they can not pay their debts.

Exploratory Data Analysis (EDA)

## Data Understanding

The first task, as seen in the previous section, was to understand the data through descriptive statistics and histograms. From this it was understood that the data was in excellent condition, in terms of missing values. On the other hand there were issues with some variables and the way in which they were resolved was presented. It became clear from the first recording of the data that the dataset has demographic and financial data for the bank's customers in the city of Taiwan. For this reason, graphs have been developed in this direction. The project approach was to provide answers about which variables influence customer decisions and ultimately how they will affect forecasts.

## Data Preparation

Regarding the new values ​​found in the variables PAY\_1-6 (-2,0) the choice for the merger was made with 2 criteria(Table 2, **Table 3**). First it did not make sense that there was no data at price 1 (payment delay of 1 month) and that there were more entries at price 2 (payment delay of 2 months). Secondly, as in the values ​​of 0.1, so in the values ​​of -2, -1 it was considered a human error to create new values ​​in addition to those provided. So, with this logic, the prices were combined. In the end, the result shows and is probably right, because the contribution of prices has a logical continuity. This means that from the value -1 (there are many entries) to the value 8, the entries decrease in a harmonious way.

Regarding the variables EDUCATION AND MARRIAGE, a similar logic was followed with the new prices found. For example in training, there were 3 variables with the same meaning (uknown) which would be difficult and would prevent good performance on the models.

## Correlations

Regarding the correlation tables that were created (Figure 2, Figure 3), there is a large correlation between the variables related to the amount of bill statement. This means that there is a great deal of influence between these months (from September to April) which indicates how much they are affected by each other. It makes sense because if a customer owes an amount in April and does not pay it, it affects the amount of the following month and so on. Also, as it is logical, the months of late payment between them show a great correlation (PAY\_1-6). Finally, there is a correlation between the variable LIMIT\_BALANCE and all economic variables (PAY\_1-6, BILL\_AMT1-6, PAY\_AMT1-6). This means that the cash balance of each customer is affected by the payments he makes, the debts and the delays in payments to the bank. Finally we see a negative correlation of the available amount of money (LIMIT\_BAL) with EDUCATION. This indicates that the cash flow of each customer is negatively affected (decreases) depending on the level of education it has.

Then the graphs will be analyzed, which helped to the maximum to understand the data and to draw important conclusions.

## Visualisations

The tactic followed to create the charts was as follows. Create 2 types of graphs, one for the demographics and the other for the financial data (**Table 14**). In this way, the data contained in the data from the past were better captured and thus it is easier for banks to understand the habits of their customers.

**Table 14: Visualisations Categories**

|  |  |
| --- | --- |
| **Demographics Variables** | **Economic Variables** |
| SEX  AGE  EDUCATION  MARRIAGE  LIMIT\_BAL  TARGET | PAY\_1-6  BILL\_AMT1-6  PAY\_AMT1-6  TARGET |

Demographics

Graph 1 shows that more women use credit cards than men (according to customer data in Taiwan). It is also observed that in proportion to the size of the 2 genders, the largest percentage has not set the default payment for the following month. Banks should therefore keep in mind that most do not have the default payment option.

Graph 2 shows the option of default payment or not, regarding the age of the customers and the limit balance they have. Thus it seems that older people (50+) tend not to delay payments to banks, unlike young people (21-48). It also seems that the older ones have a marginally higher amount of money available. This figure shows that young people are better and more consistent customers for banks and may need to turn marketing strategies to them.

It follows from Graph 3 that university graduates hold more credit cards than high school graduates. It makes sense, in part, that college graduates will have better jobs and therefore better salaries than others, so banks will trust them more and issue credit cards and loans. A first conclusion is that compared to the size, high school graduates are better in terms of default payment next month. However, all categories have a large number of late payments.

Regarding the ages and the level of education, in Graph 4 and Graph 5it is initially observed, as before, that those who do not have default in the payment next month, have a higher, on average, bank available for expenses. Contradictory, because banks should support good customers and not those who delay debt payments. It also seems that the education per age group and that the older one is the more the balance of one's account increases. A good conclusion that banks should keep in mind is that here too it is observed that high school graduates (and perhaps university graduates) can be considered good customers because they have a low spending limit and are quite good at the default payment. In other words, there is no huge risk of losing capital, as with older people (high limit balance = large loss of capital due to inability to repay).

Graph 6 and Graph 7 reflect the marital status of customers. It is observed that the free clients are slightly more than the married ones, while the rest (divorced, etc.) are minimal. And in this category, customers regardless of marital status do not choose the default payment. It also becomes clear that married people, as is logical, have more money available than single people. This means that, as in the previous figure, a good strategy for banks is to opt for free customers because of the lower risk (in terms of money). It also seems that women have slightly higher (on average) cash available, unless they are married, because men have a slightly higher amount.

Finally, a very interesting, demographic graph, is Graph 8. Describes the concentration of the option (default or not) in relation to the limit balance. As shown in the previous figures, those customers who do not choose the default payment have a greater balance than those who choose it (consistent customers). The interesting thing here is that those who are inconsistent (not default), are consistent in this choice. On the other hand we see that those who are consistent (default), tend to go to the other option (not default). Too bad for the banks, which should take this seriously.

Economics

As for the economic variables, they will be analyzed collectively because something interesting was observed in them (**Table 15**). The charts in this category are designed to enable banks to study past payments and late payments and thus understand their customers and habits, in order to devise better strategies in the future.

**Table 15: Economic Graphs**

|  |  |  |
| --- | --- | --- |
| **Groups – Variables** | **Figures** | **Explanation** |
| BILL\_AMT1-6 | Graph 15, Graph 16, Graph 17, Graph 18, Graph 19, Graph 20 | These graphs show the amount of money that customers have to pay (owe) to the banks. A central explanation was given as to why an interesting statistic was observed. From month to month the amount of money decreases (not significantly). This makes sense as customers pay their debts to the banks every month. |
| PAY\_AMT1-6 | Graph 21, Graph 22, Graph 23, Graph 24, Graph 25, Graph 26 | This group of charts shows the amount paid (actual amount) by customers to the banks for their debts. The important thing is that these graphs, in combination with the above, give an important element. The amount they pay compared to the amount owed by customers is significantly smaller and not enough to repay the debts. We understand that customers extend debt repayment as much as possible. This does not favor the banks at all as a large amount of money remains permanently outstanding. Banks therefore need to find a way to get customers to repay their debts earlier and faster. |
| PAY\_1-6 | Graph 9, Graph 10, Graph 11, Graph 12, Graph 13, Graph 14 | This group reflects the customers' choices regarding payments. Most customers in all groups choose to pay their installment with a delay of one month while the second most popular option is the payment without delay (with a big difference from the first). Banks need to find a way to reduce this difference between a month late and a late payment if they want to avoid future problems. There is also, during the summer months (June to August) an increase of the option "repayable with 2 months delay". This shows that customers use their available money for holidays, leaving their installments for the next month. |

## Models-Predictions

Machine learning models were developed to help banks predict their customers' behavior (default or not) with a view to better strategy. This will bring profits to the banks, thus creating a safety net against the risks of lack of cash. 7 models were developed, which were evaluated in order to enhance their performance as much as possible, based on the available data. The models were evaluated and compared based on accuracy, kappa, sensitivity and specificity. The following table describes all the statistics as well as the selection of the best model developed (**Table 16**).

**Table 16: Machine Learning Models Selection**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Tuning Method** | **Accuracy** | **Kappa**  **(-1 to 1)** | **Sensitivity**  **(0 to 1)** | **Specificity**  **(0 to 1)** |
| Logistic Regession | BIC  Assumption Checking | 78.65% | 0.0706 | 0.99521 | 0.05184 |
| Random Forest | BIC, Cross Validation (CV) (k=10) | 80.4%(no CV)  81.24%(CV,k=10)  81.36%(CV,k=20) | 0.3333 | 0.9324 | 0.3520 |
| Decision Tree | BIC | 82.04% | 0.3567 | 0.9599 | 0.3291 |
| SVM linear | BIC, SVM kernel compare | 77.88% | 0 | 1 | 0 |
| SVM kernel | BIC | 82.08% | 0.3486 | 0.9649 | 0.3134 |
| Naïve Bayes | BIC | 49.77% | 0.1138 | 0.4217 | 0.7655 |
| XGBoost | BIC, Cross Validaton (CV) (k=5) | 80.1% | 0.3134 | 0.9742 | 0.3146 |

All models have a low Cohen's Kappa, which stems from the fact that the data of the city of Taiwan was quite unbalanced (default much much smaller than not default). The logistic regression model performed quite well despite the simplicity of its classical analysis. In the random forest model could achieve greater accuracy by increasing its price k, but we must consider that we lose in time and computing power, therefore it is not selected as the optimal model. The naive bayes model seems to be unable to perform on Taiwan data. Xgboost achieved a good accuracy and with the use of cross validation it might have been significantly enhanced. Finally, the linear SVM does not seem to work for the classification problem that had to be solved and comparing it with the SVM kernel, the huge difference is observed. Closing the 2 models selected are the SVM kernel and the Decision Tree, with similar performance and reliability in predictions. So banks have the ability to use these machine learning models, in similar classification problems so that they are able to predict with great accuracy the habits of their customers, while at the same time they can design new strategies to achieve their goals.

# Conclusions

In summary, this work was done with the aim of answering a question of the real world. The question was how banks could learn from customers 'pasts while at the same time how they would anticipate customers' decisions about whether to pay their installments in the future. Through the analysis of the past, in the data of the customers in the city of Taiwan, it emerged that most of the customers are delaying the repayment of the debts for at least a month. Also, older age groups tend to be bad customers and in combination with the large limit balances they have, make them dangerous for banks. Also, the season, as it seemed, plays a role in the repayment but also in the delays, with the summer being the protagonist here. At the same time, while the amounts of bill statement increase or remain constant, the actual debt payments do not come close to the full repayment of the debts. This is quite worrying for the banks. So there seems to be a tendency for customers to try to delay repayment. It is also important that in the forecast models but also in the correlation tables the economic variables significantly affect the models and the forecasts. Regarding the models and the future uses and improvements, the proposal of this work is to test in balanced data and the models to be tuning in different ways to achieve better accuracy in the predictions. Finally, banks acquire an important tool of analysis and forecasting, which in the future will bring them profits and better organization and business strategy.

# References

Alam, T. M. et al., 2020. An Investigation of Credit Card Default Prediction in the Imbalanced Datasets. *IEEEAcces,* Issue 8, pp. 201173-201198.

Delamaire, L., Abdou, H. & Pointon, J., 2009. Credit card fraud and detection techniques: a review. *Banks and Bank Systems,* 2(4), pp. 57-68.

Hsu, T.-C.et al., 2019. Enhanced Recurrent Neural Network for Combining Static and Dynamic Features for Credit Card Default Prediction. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP,* pp. 1572-1576.

Kim, H., Cho, H. & Ryu, D., 2018. An empirical study on credit card loan delinquency. *Economic Systems,* 3(42), pp. 437-449.

Kuhn, M. & Johnson, K., 2013. *Applied Predictive Modeling.* 1st edition ed. New York: Springer.

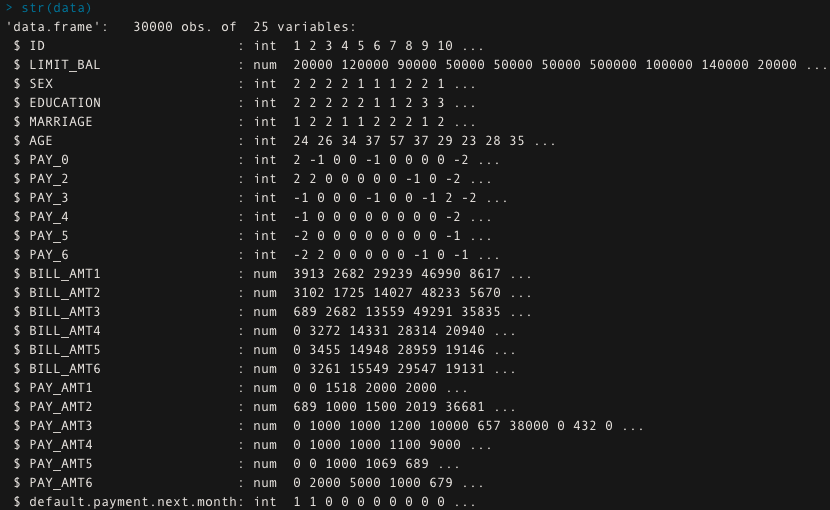
Liu, R., 2018. Machine Learning Approaches to Predict Default of Credit Card Clients.. *Modern Economy,* Issue 9, pp. 1828-1838.

Sharma, S. & Mehra, V., 2018. Default Payment Analysis of Credit Card Clients. *ResearchGate,* Issue 2, pp. 1-7.

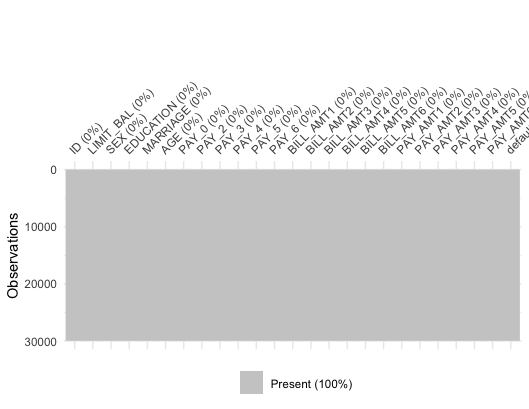
Yeh, I.-C. & Lien, C.-h., 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications,* Issue 36, pp. 2473-2480.

# Appendix

## Code Photos & Graphs



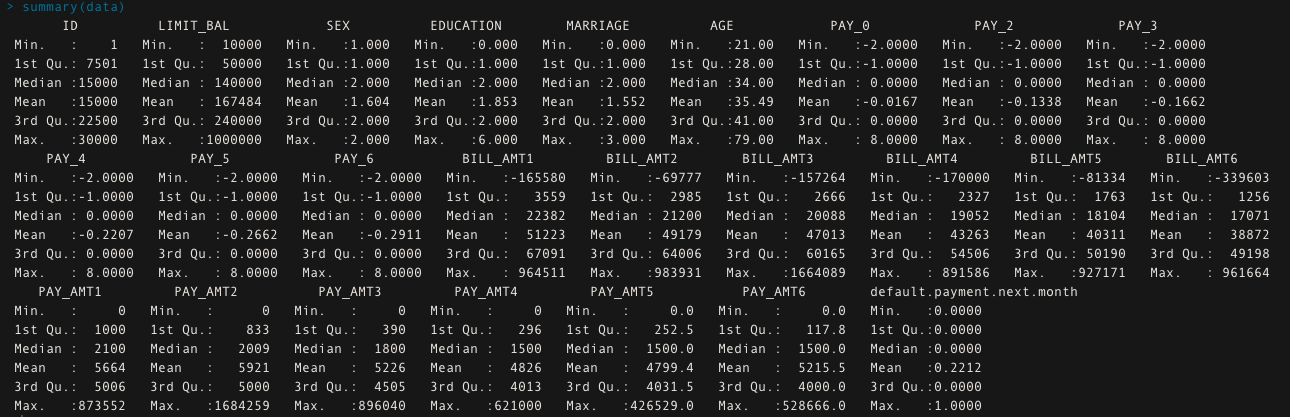
Code Photos 1: Structure of Dataset (source: my R code)



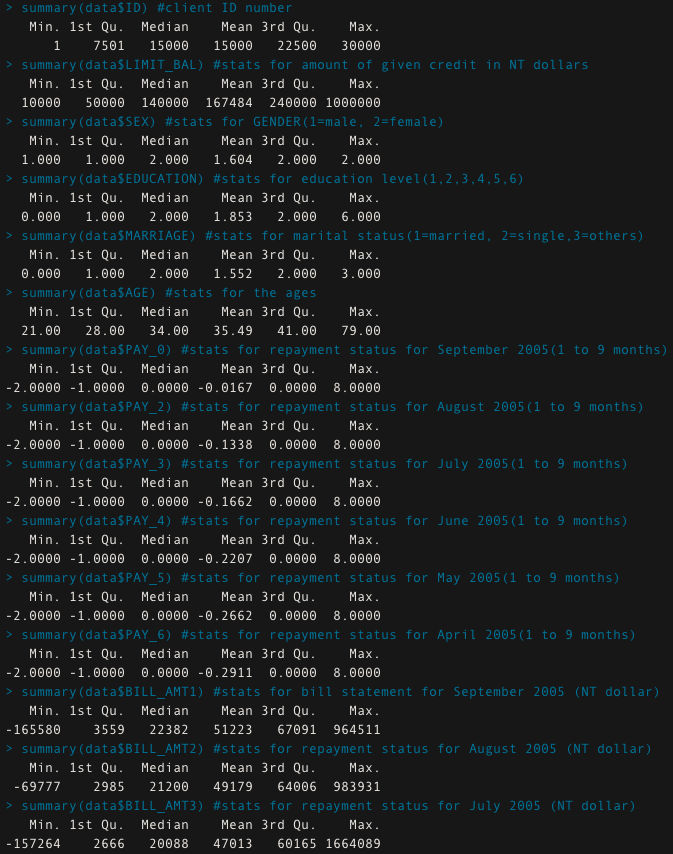
Code Photos 2: Graph to see if we have NAs (source: my R code)



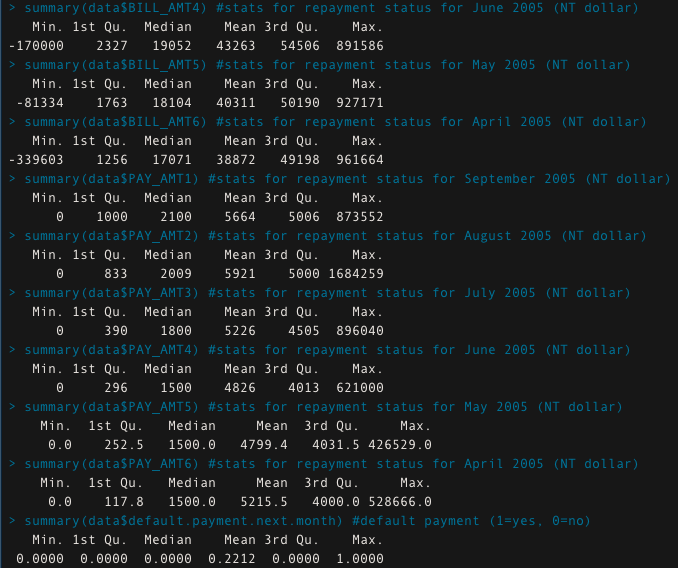
Code Photos 3: Command in R to see the number of NAs (source: my R code)



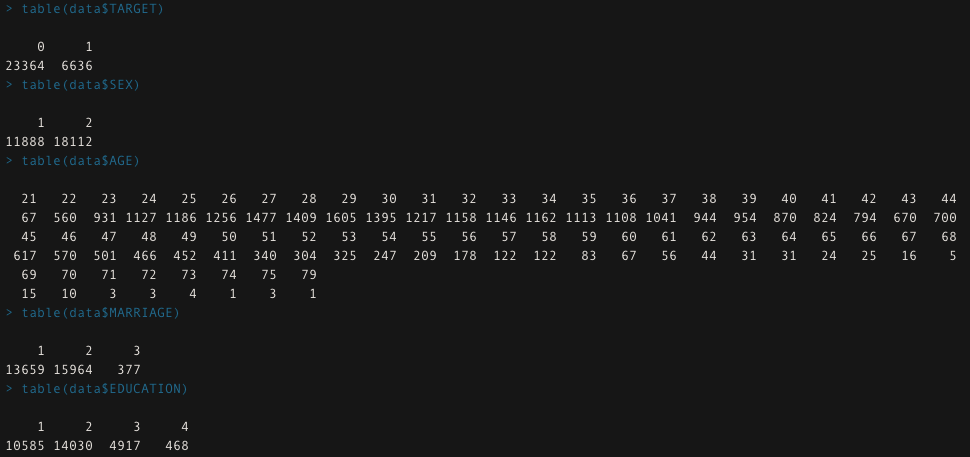
Code Photos 4: Summarise the data for the first time. EDA (Source: my R code)



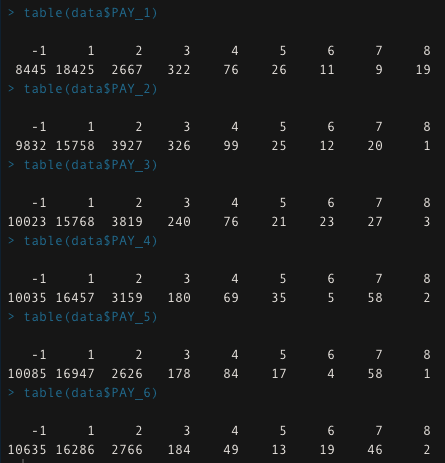
Code Photos 5: Summary per variable 1 (source:my R code)



Code Photos 6: Summary per variable 2 (source: my R code)

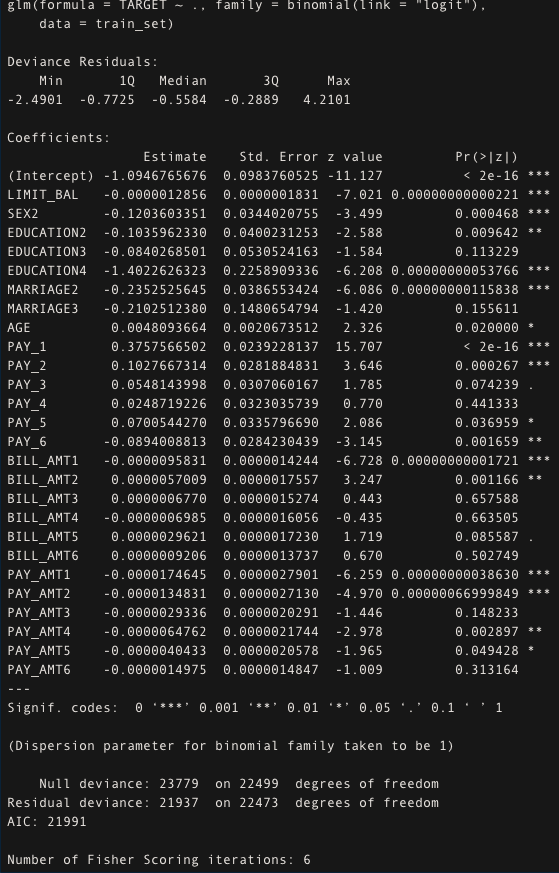


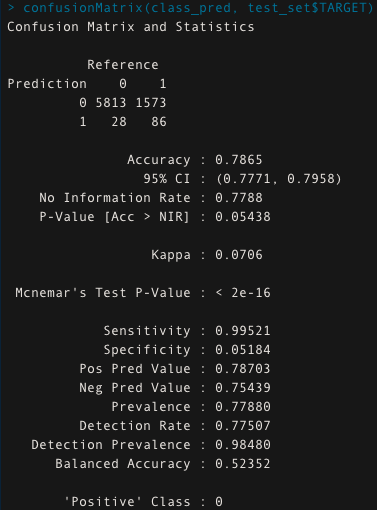
Code Photos 7: Distribution of variables(as.factor) 1 (source:my R code)



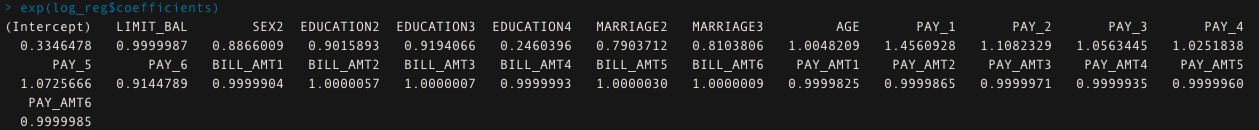
Code Photos 8: Distribution of variables(as.factor) 2 (source:my R code)

Code Photos 9: Logistic Regression Model (source: my R code)

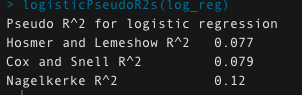




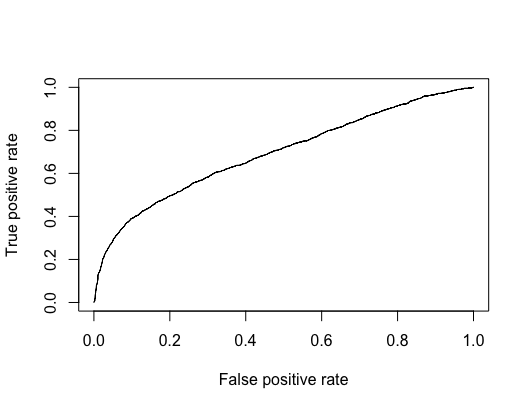
Code Photos 10: Confusion Matrix (Logistic Regression) (source: my R code)



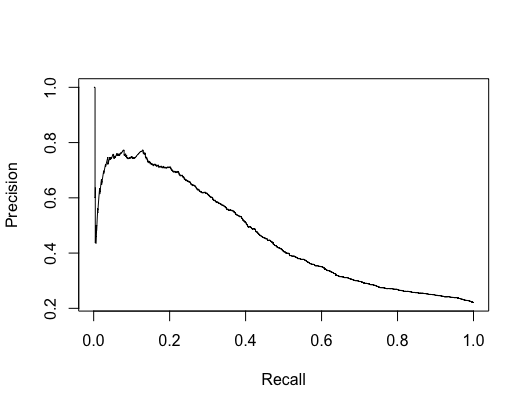
Code Photos 11: odds of the final model's variables (logistic regression) (source: my R code)



Code Photos 12: Pseudo R^2 for logistic regression (source: my R code)



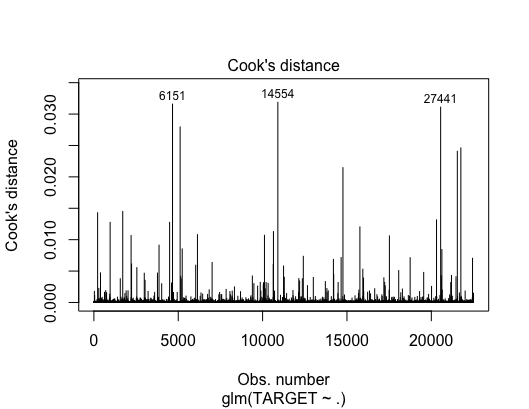
Code Photos 13: ROC curve for logistic regression (source: my R code)



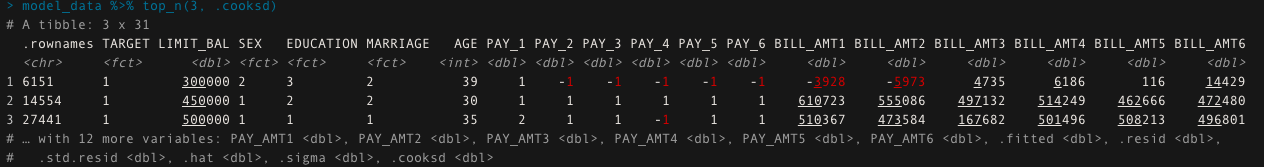
Code Photos 14: Recall and Precision Curve for logistic regression (source: my R code)



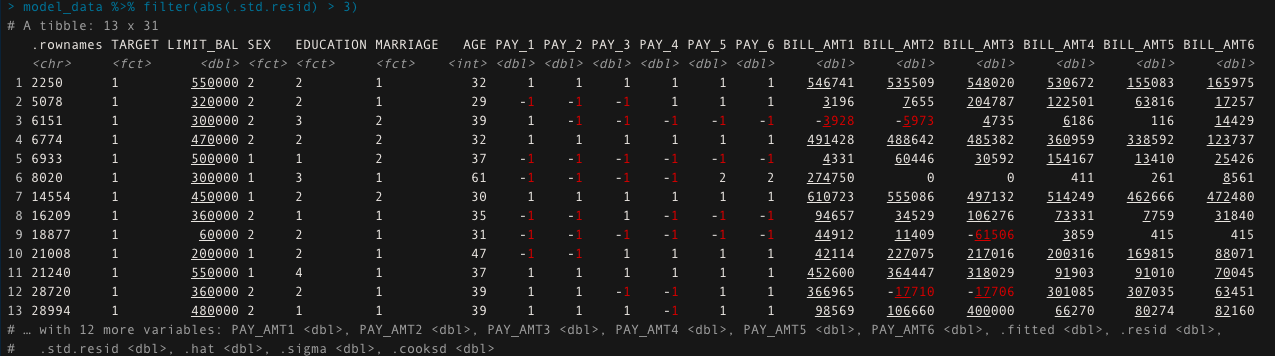
Code Photos 15: AUC value for logistic regression (source: my R code)



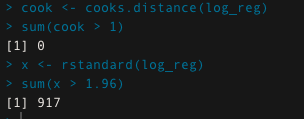
Code Photos 16: Cook's Distance(assumption) Logistic Regression (source: my R code)



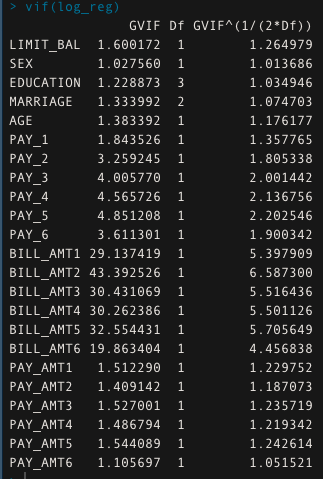
Code Photos 17: Top 3 values of cook's distance(assumption) logistic regression (source: my R code)



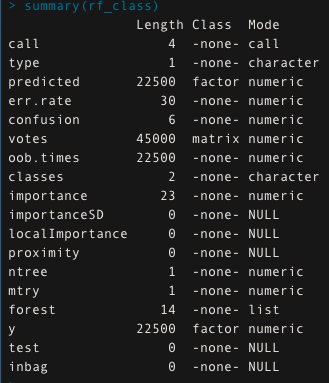
Code Photos 18: Potential Influence Variables(assumption) logistic regression(source: my R code)



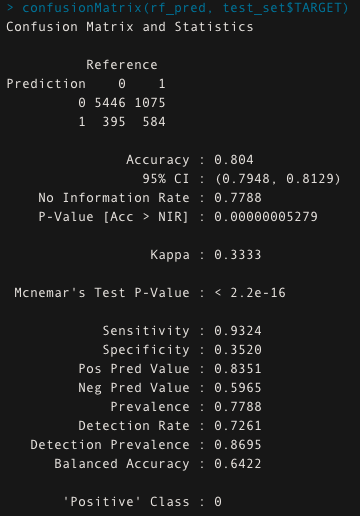
Code Photos 19: Residuals fit and cook(assumption) logistic regression(source: my R code)



Code Photos 20: VIF matrix (logistic regression) (assumption) (source: my R code)



Code Photos 21: Random Forest (summary of model) (source: my R code)



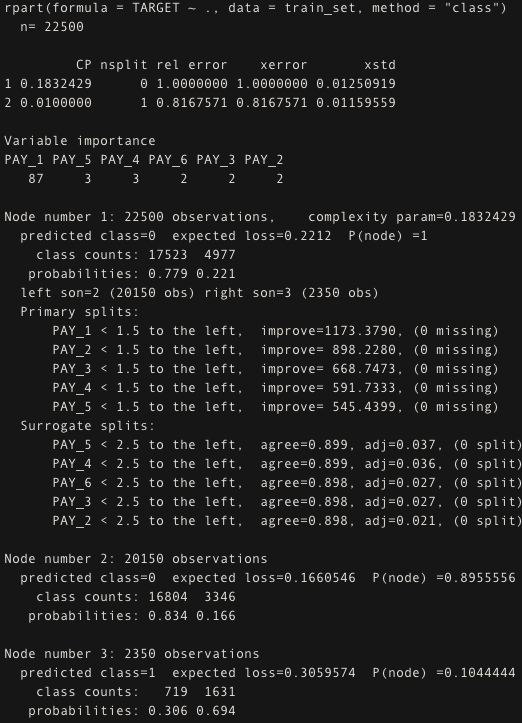
Code Photos 22: Confusion Matrix for Random Forest (source: my R code)



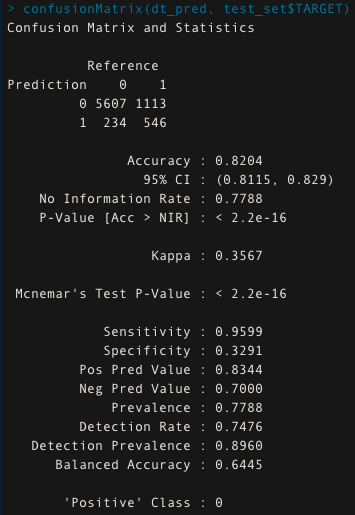
Code Photos 23: Accuracy after Cross Validation in Random Forest (k=10) (source: my R code)



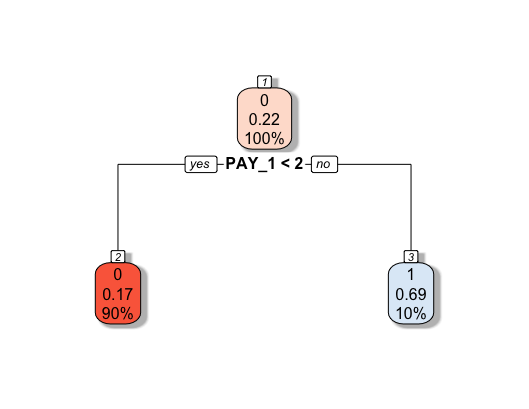
Code Photos 24: Accuracy after Cross Validation in. Random Forest (k=20) (source: my R code)



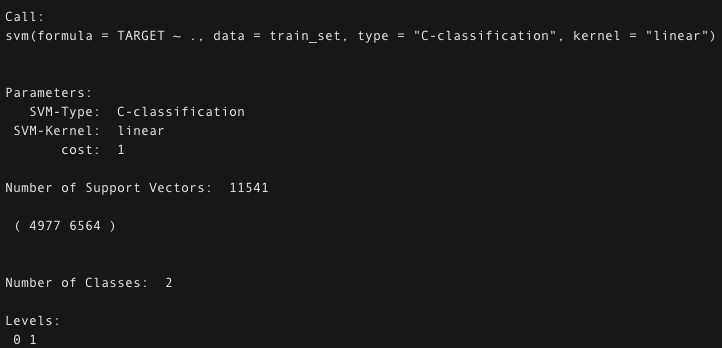
Code Photos 25: Decision Tree Model (source: my R code)



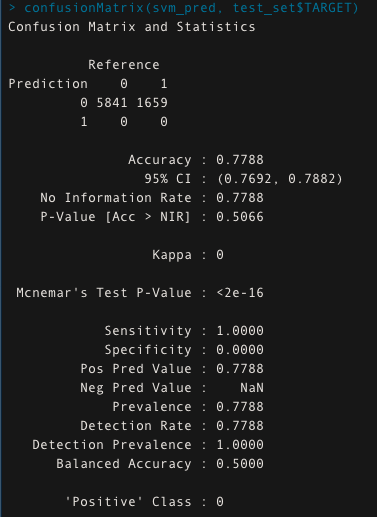
Code Photos 26: Confusion Matrix for Decision Tree (source: my R code)



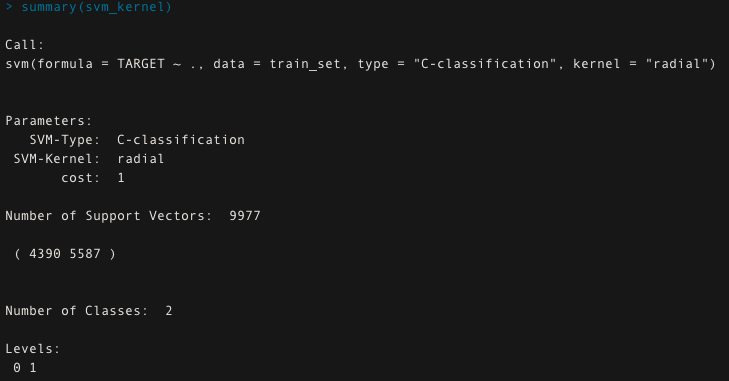
Code Photos 27: Decision Tree Graph (source: my R code)



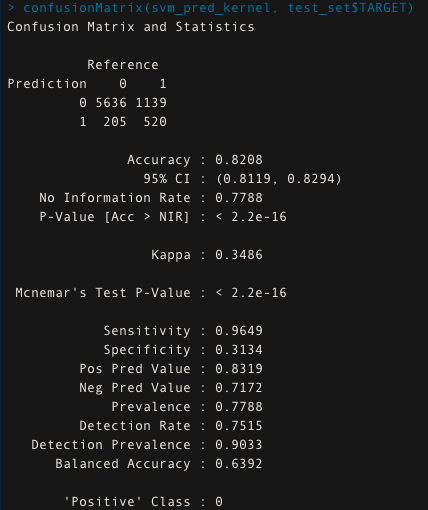
Code Photos 28: SVM model (linear) (source: my R code)



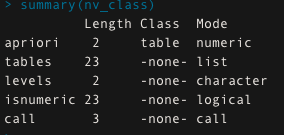
Code Photos 29: Confusion Matrix for SVM(linear) (source: my R code)



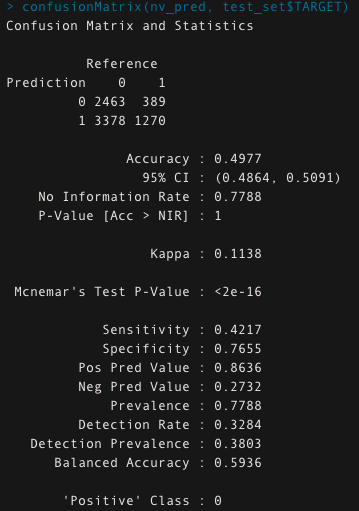
Code Photos 30: SVM model (kernel) (source: my R code)



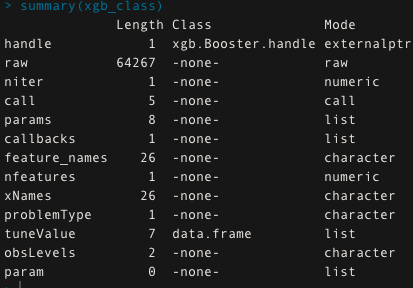
Code Photos 31: Confusion Matrix for SVM(kernel) (source: my R code)



Code Photos 32: Naive Bayes Model (source: my R code)



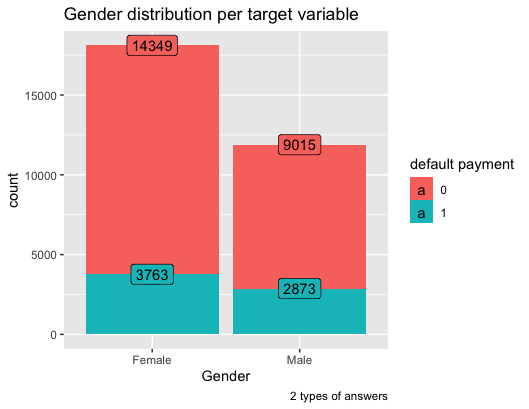
Code Photos 33: Confusion Matrix for Naive Bayes (source: my R code)



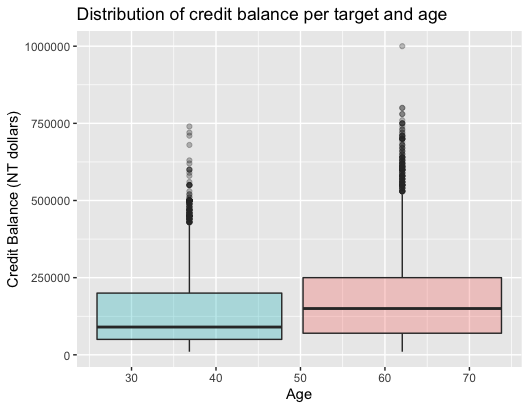
Code Photos 34: XGBoost Model (source: my R code)

# Final Visualisations

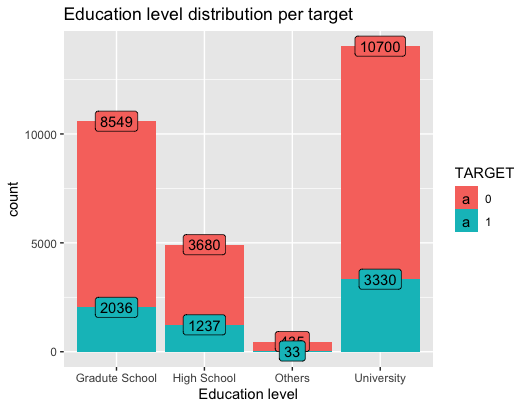
## Demographics



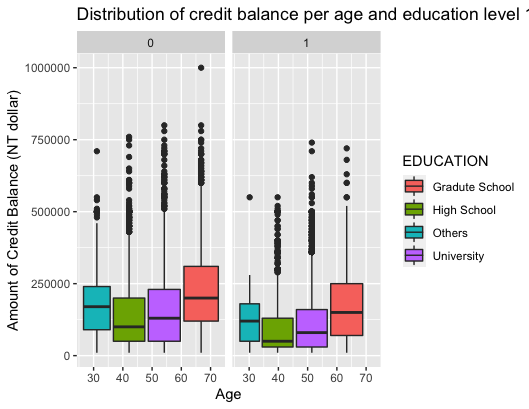
Graph 1: Gender Distribution per Target Variable (source: my R code)



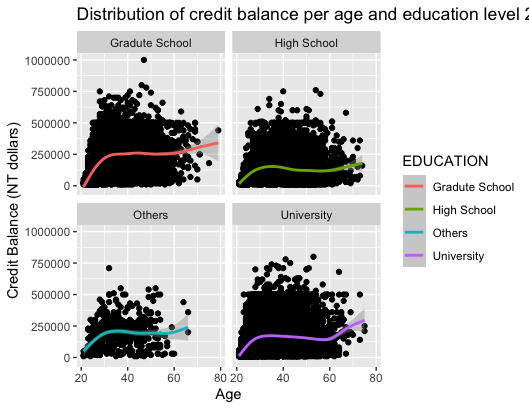
Graph 2: Distribution of credit balance per age and target (source: my R code)



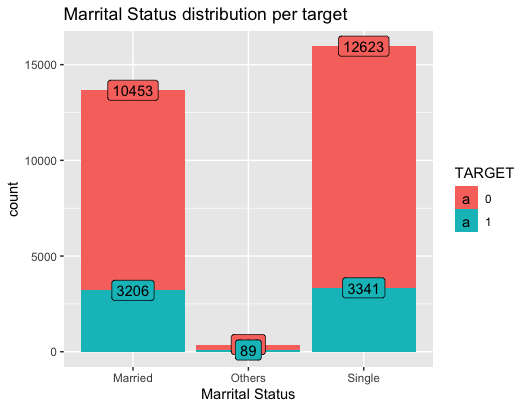
Graph 3: Education level distribution per target variable



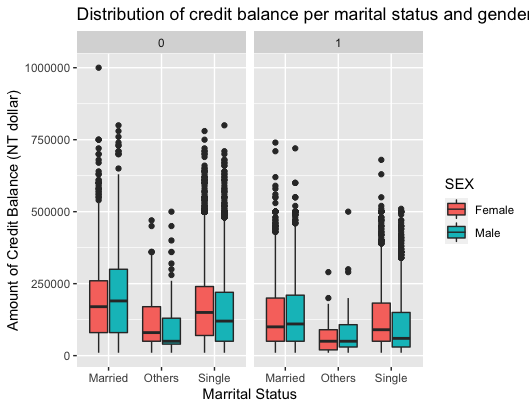
Graph 4: Credit Balance distribution per age and education level (source: my R code)



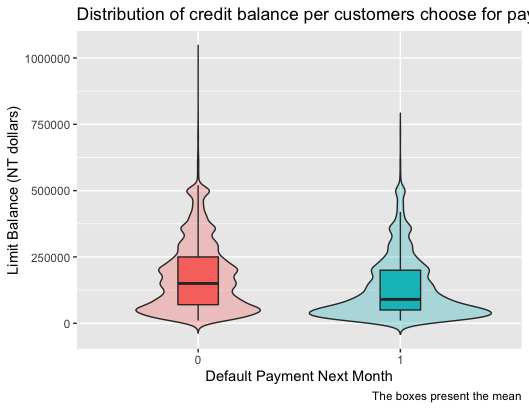
Graph 5: Credit Balance distribution per age and education level (source: my R code)



Graph 6: Marital Status distribution per target variable (source: my R code)

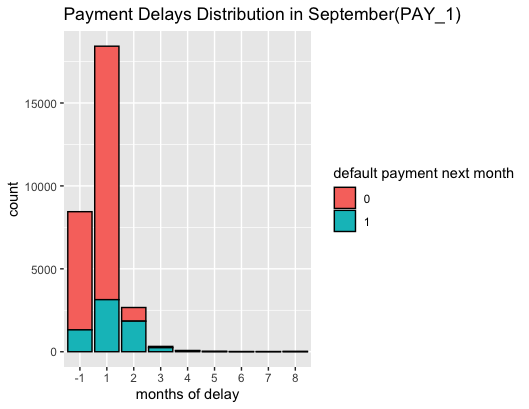


Graph 7: Credit Balance distribution per marital status and gender (source: my R code)

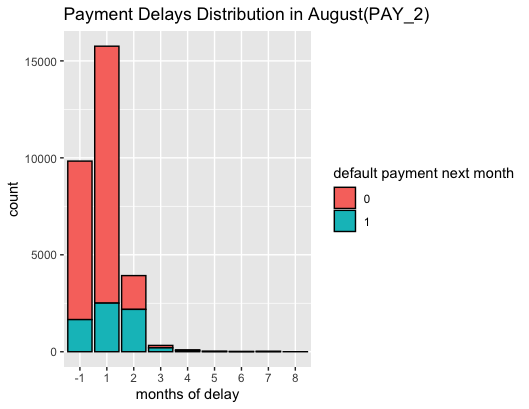


Graph 8: How people choose the next's month payment depends on their credit balance (source: my R code)

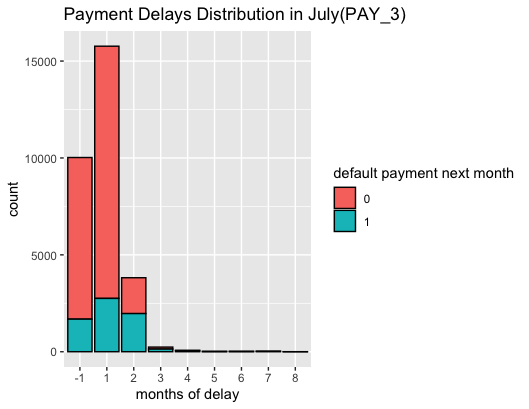
## Economic



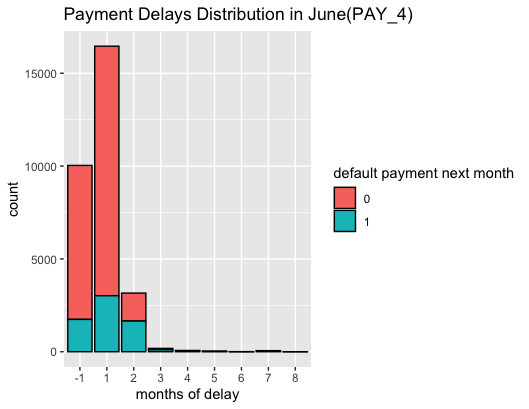
Graph 9: Payment Delays in September (source: my R code)



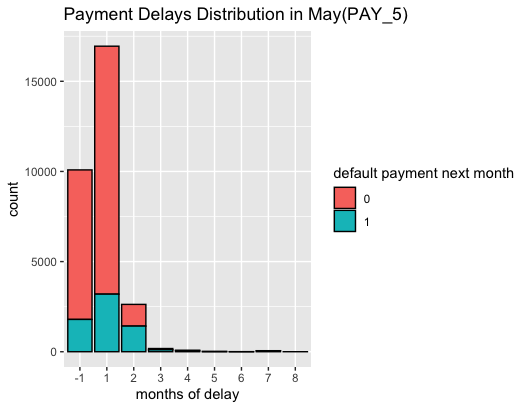
Graph 10: Payment Delays in August (source: my R code)



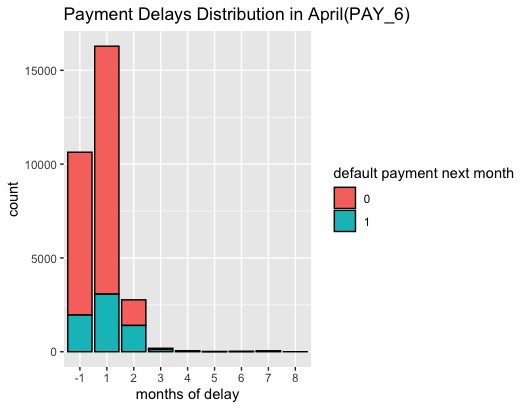
Graph 11: Payment Delays in July (source: my R code)



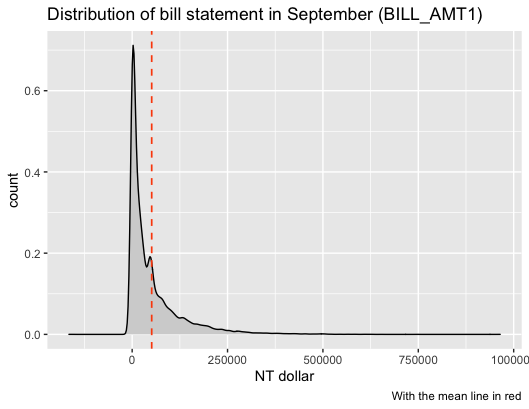
Graph 12: Payment Delays in June (source: my R code)



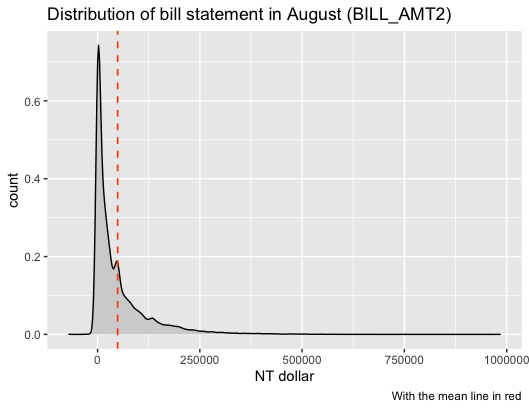
Graph 13: Payment Delays in May (source: my R code)



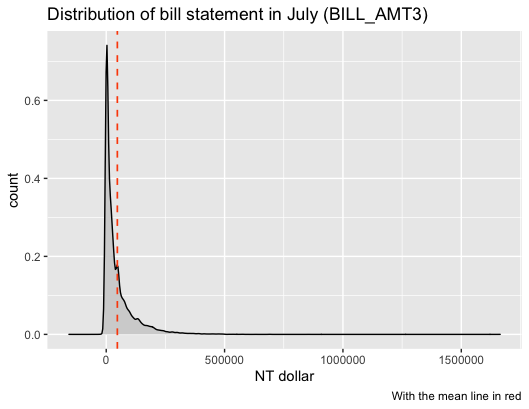
Graph 14: Payment Delays in April (source: my R code)



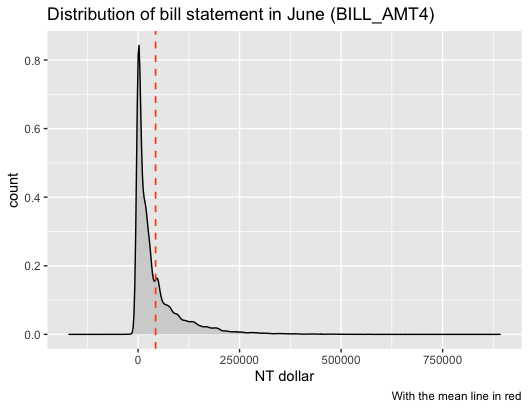
Graph 15: Bill Statement in September (source: my R code)



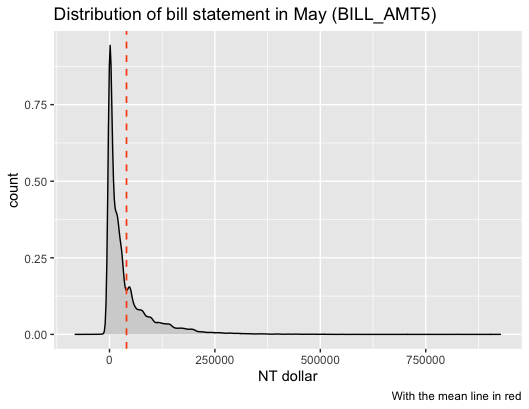
Graph 16: Bill Statement in August (source: my R code)



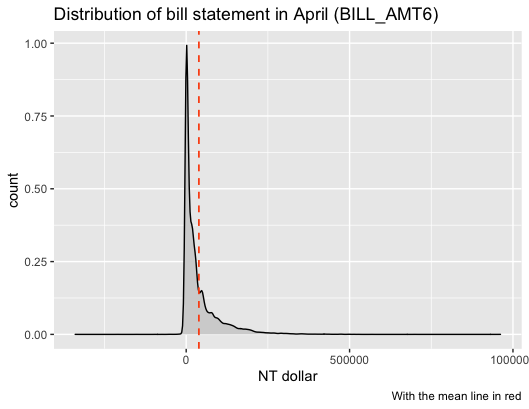
Graph 17: Bill Statement in July (source: my R code)



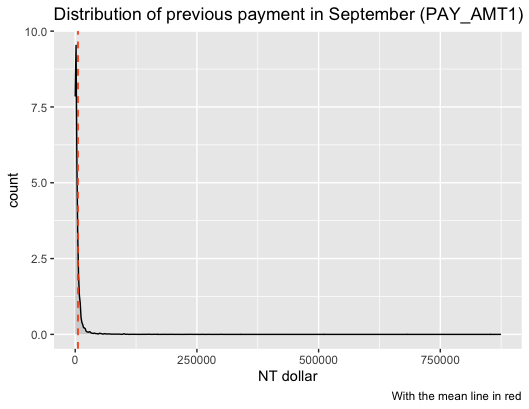
Graph 18: Bill Statement in June (source: my R code)



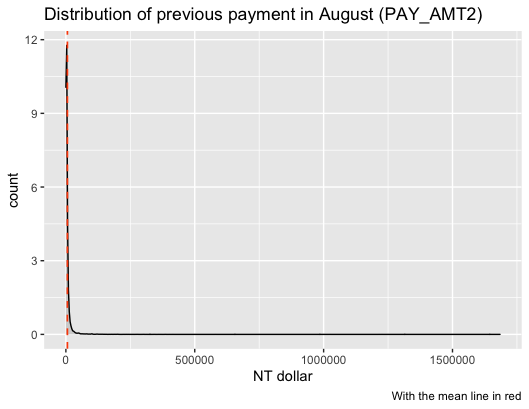
Graph 19: Bill Statement in May (source: my R code)



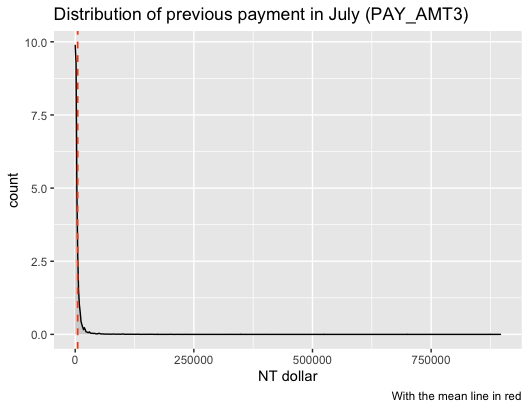
Graph 20: Bill Statement in April (source: my R code)



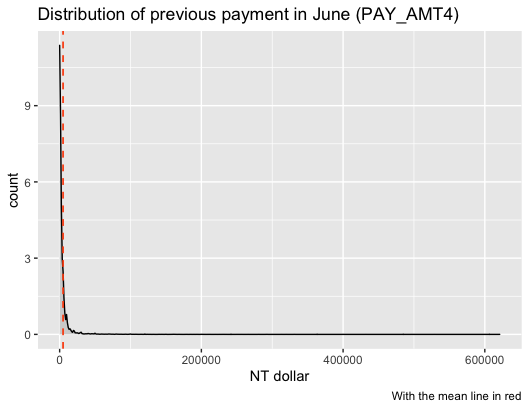
Graph 21: Previous Payment in September (source: my R code)



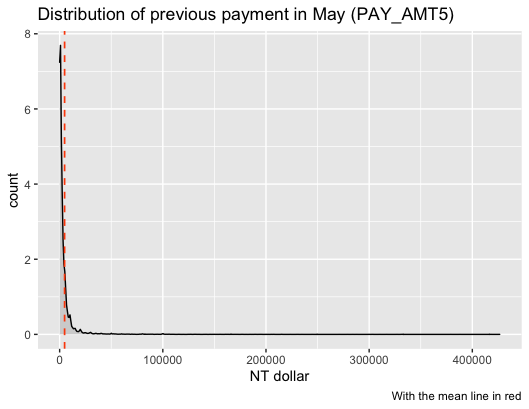
Graph 22: Previous Payment in August (source: my R code)



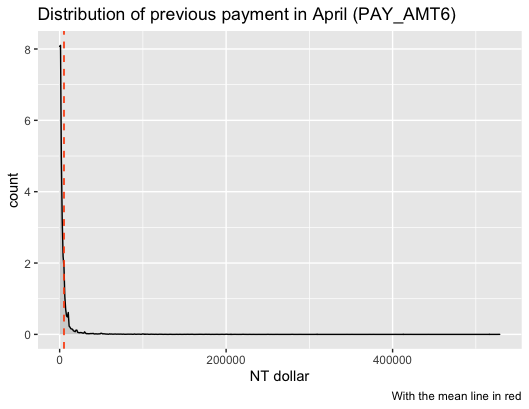
Graph 23: Previous Payment in July (source: my R code)



Graph 24: Previous Payment in June (source: my R code)



Graph 25: Previous Payment in May (source: my R code)



Graph 26: Previous Payment in April (source: my R code)

## R Code

#################################################

#Advanced Analytics & Machine Learning

#MGT7179

#Assignment 2 - Classification

#Vasileios Gounaris Bampaletsos-40314803

#Classification Problem

#Default Payments of Credit Card Clients in Taiwan

#The dataset contains information on default payment

#demographic factors, credit data, payments' history

#bill statements of credit clients in Taiwan from

#April 2005 to September 2006

#Analyze and predict the default payment for the next month

#Our clients(banks) will have a good tool to predict

#their clients behavior about payments

#methods which used:

#Logistic Regression

#Random Forest (with cross validation)

#Decision Tree

#SVM

#SVM Kernel

#Boosted Trees (XGBOOST method & cross validation)

#Naive Bayes

###################################################

# ~

#set the working directory

setwd("/Users/basilisgounarismpampaletsos/Desktop/PROJECTS 2/30:04 analytics")

options(scipen = 9)

#load the libraries

library(readxl)

library(psych)

library(ggplot2)

library(caTools)

library(statsr)

library(dplyr)

library(BAS)

library(car)

library(tidyr)

library(purrr)

library(gridExtra)

library(forcats)

library(corrplot)

library(magrittr)

library(caret)

library(Hmisc)

library(tidyverse)

library(ggpubr)

library(ROCR)

library(broom)

library(lubridate)

library(GGally)

library(ISLR)

library(hrbrthemes)

library(viridis)

library(e1071)

library(plyr)

library(readr)

library(repr)

library(glmnet)

library(ggthemes)

library(scales)

library(wesanderson)

library(styler)

library(xgboost)

library(randomForest)

library(rsample)

library(gbm)

library(h2o)

library(pdp)

library(lime)

library(naniar)

library(leaps)

library(tree)

library(MASS)

library(class)

library(data.table)

#library(gam)

library(sandwich)

library(rpart.plot)

library(lmtest)

library(ranger)

library(nnet)

library(pROC)

library(kernlab)

#load the data

data <- read.csv("Credit\_Card.csv")

###########################################################

#summarize the data for the first time

#check the data's distribution and descriptive measures

summary(data)

#check the structure of our dataset

str(data)

#check the dataset if it has missing values

vis\_miss(data)

sum(is.na(data))

#check some basic statistics in the dataset

#looking for the minimum, median and maximum values

#understand the data's distribution

#looking for outliers

summary(data$ID) #client ID number

summary(data$LIMIT\_BAL) #stats for amount of given credit in NT dollars

summary(data$SEX) #stats for GENDER(1=male, 2=female)

summary(data$EDUCATION) #stats for education level(1,2,3,4,5,6)

summary(data$MARRIAGE) #stats for marital status(1=married, 2=single,3=others)

summary(data$AGE) #stats for the ages

summary(data$PAY\_0) #stats for repayment status for September 2005(1 to 9 months)

summary(data$PAY\_2) #stats for repayment status for August 2005(1 to 9 months)

summary(data$PAY\_3) #stats for repayment status for July 2005(1 to 9 months)

summary(data$PAY\_4) #stats for repayment status for June 2005(1 to 9 months)

summary(data$PAY\_5) #stats for repayment status for May 2005(1 to 9 months)

summary(data$PAY\_6) #stats for repayment status for April 2005(1 to 9 months)

summary(data$BILL\_AMT1) #stats for bill statement for September 2005 (NT dollar)

summary(data$BILL\_AMT2) #stats for repayment status for August 2005 (NT dollar)

summary(data$BILL\_AMT3) #stats for repayment status for July 2005 (NT dollar)

summary(data$BILL\_AMT4) #stats for repayment status for June 2005 (NT dollar)

summary(data$BILL\_AMT5) #stats for repayment status for May 2005 (NT dollar)

summary(data$BILL\_AMT6) #stats for repayment status for April 2005 (NT dollar)

summary(data$PAY\_AMT1) #stats for repayment status for September 2005 (NT dollar)

summary(data$PAY\_AMT2) #stats for repayment status for August 2005 (NT dollar)

summary(data$PAY\_AMT3) #stats for repayment status for July 2005 (NT dollar)

summary(data$PAY\_AMT4) #stats for repayment status for June 2005 (NT dollar)

summary(data$PAY\_AMT5) #stats for repayment status for May 2005 (NT dollar)

summary(data$PAY\_AMT6) #stats for repayment status for April 2005 (NT dollar)

summary(data$default.payment.next.month) #default payment (1=yes, 0=no)

#make basic visualisations for better understanding

#make simple histograms for numerice variables

#histograms help to check the data quality

#find the outliers and problems in the dataset

#data at this stage are all intiger/numeric

#NUMERIC VARIABLES

hist(data$ID)

hist(data$LIMIT\_BAL)

hist(data$SEX)

hist(data$EDUCATION)

hist(data$MARRIAGE)

hist(data$AGE)

hist(data$PAY\_0)

hist(data$PAY\_2)

hist(data$PAY\_3)

hist(data$PAY\_4)

hist(data$PAY\_5)

hist(data$PAY\_6)

hist(data$BILL\_AMT1)

hist(data$BILL\_AMT2)

hist(data$BILL\_AMT3)

hist(data$BILL\_AMT4)

hist(data$BILL\_AMT5)

hist(data$BILL\_AMT6)

hist(data$PAY\_AMT1)

hist(data$PAY\_AMT2)

hist(data$PAY\_AMT3)

hist(data$PAY\_AMT4)

hist(data$PAY\_AMT5)

hist(data$PAY\_AMT6)

hist(data$default.payment.next.month)

#####################################################################

#CORRELATIONS

#####################################################################

#at this point we want to see the correlations how the variables connect each other

#check this connections/associations because they are important for models' building

#use 2 different ways to check the correlation

#very useful graphs

#full cor matrix, half cor matrix

#correlation matrix

#using pearson method

continuous\_var.cor = cor(na.omit(data), method = "pearson")

corrplot(continuous\_var.cor)

#half correlation matrix

#using pearson method

ggcorr(na.omit(data), method = c("everything", "pearson"))

#####################################################################

#FIX THE DATA

#####################################################################

#change the name of target variable because it is very big for our visualisations, correlation visualisations

#from default.payment.next.month to TARGET

names(data)[25] <- "TARGET"

#change the name PAY\_0 to PAY\_1 because we dont want a gap between our variables(0 to 2)

names(data)[7] <- "PAY\_1"

#fix pay1

data$PAY\_1[data$PAY\_1 == -2] <- -1

data$PAY\_1[data$PAY\_1 == 0] <- 1

#fix pay2

data$PAY\_2[data$PAY\_2 == -2] <- -1

data$PAY\_2[data$PAY\_2 == 0] <- 1

#fix pay3

data$PAY\_3[data$PAY\_3 == -2] <- -1

data$PAY\_3[data$PAY\_3 == 0] <- 1

#fix pay4

data$PAY\_4[data$PAY\_4 == -2] <- -1

data$PAY\_4[data$PAY\_4 == 0] <- 1

#fix pay5

data$PAY\_5[data$PAY\_5 == -2] <- -1

data$PAY\_5[data$PAY\_5 == 0] <- 1

#fix pay6

data$PAY\_6[data$PAY\_6 == -2] <- -1

data$PAY\_6[data$PAY\_6 == 0] <- 1

#data$TARGET[data$TARGET == "1"] <- "Yes"

#data$TARGET[data$TARGET == "0"] <- "No"

#change the values of variables

#from numbers to characters

#variable: sex

#check how many unique values it has

unique(data$SEX)

#rename the values

data$SEX[data$SEX == "1"] <- "Male"

data$SEX[data$SEX == "2"] <- "Female"

#variable: education

#check how many unique values it has

unique(data$EDUCATION)

#combine the 3 unknown values(0,5,6) with value others(4)

data$EDUCATION <- ifelse(data$EDUCATION == 0 | data$EDUCATION == 5 | data$EDUCATION == 6,4, data$EDUCATION)

#rename the values of education

data$EDUCATION[data$EDUCATION == "1"] <- "Gradute School"

data$EDUCATION[data$EDUCATION == "2"] <- "University"

data$EDUCATION[data$EDUCATION == "3"] <- "High School"

data$EDUCATION[data$EDUCATION == "4"] <- "Others"

#variable: marriage

#check how many unique values it has

unique(data$MARRIAGE)

#combine the 0 value with 3(others) value

data$MARRIAGE <- ifelse(data$MARRIAGE == 0, 3, data$MARRIAGE)

data$MARRIAGE[data$MARRIAGE == "1"] <- "Married"

data$MARRIAGE[data$MARRIAGE == "2"] <- "Single"

data$MARRIAGE[data$MARRIAGE == "3"] <- "Others"

#change the type of variables

#INTIGER TO FACTOR

data$PAY\_1 <- as.factor(data$PAY\_1)

data$PAY\_2 <- as.factor(data$PAY\_2)

data$PAY\_3 <- as.factor(data$PAY\_3)

data$PAY\_4 <- as.factor(data$PAY\_4)

data$PAY\_5 <- as.factor(data$PAY\_5)

data$PAY\_6 <- as.factor(data$PAY\_6)

data$SEX <- as.factor(data$SEX)

data$EDUCATION <- as.factor(data$EDUCATION)

data$MARRIAGE <- as.factor(data$MARRIAGE)

data$TARGET <- as.factor(data$TARGET)

#delete the ID variable because is useless

data$ID <- NULL

#check the distribution of clean data

#only the variables as factor (no the numeric)

table(data$TARGET)

table(data$SEX)

table(data$AGE)

table(data$MARRIAGE)

table(data$EDUCATION)

table(data$PAY\_1)

table(data$PAY\_2)

table(data$PAY\_3)

table(data$PAY\_4)

table(data$PAY\_5)

table(data$PAY\_6)

#####################################################################

#FINAL VISUALISATIONS

#VISUALISE THE DATA FOR BETTER UNDERSTANDING

#####################################################################

#create some demographics visualisations

#include education, age, marital status, limit balance and target

#help the audiance(clients) to understand the data

#1 - Gender(SEX) combining with default payment(target variable)

#see how many people choose which option

ggplot(data = data, mapping = aes(x = SEX, fill = TARGET)) +

geom\_bar() +

labs(fill="default payment", x="Gender", y= "count",

title="Gender distribution per target variable",

caption="2 types of answers") +

stat\_count(aes(label = ..count..), geom = "label")

#Distribution of credit balance per target and age

ggplot(data = data, aes(x= AGE, y = LIMIT\_BAL, fill=TARGET)) +

geom\_boxplot(alpha=0.3) +

theme(legend.position="none") +

labs(x="Age", y= "Credit Balance (NT dollars)",

title="Distribution of credit balance per target and age")

#boxplot to check the education level of the customers

ggplot(data = data, mapping = aes(x = EDUCATION, fill = TARGET)) +

geom\_bar() +

ggtitle("EDUCATION") +

stat\_count(aes(label = ..count..), geom = "label") +

labs(x="Education level", y= "count",

title="Education level distribution per target")

#Distribution of credit balance per age and education level

ggplot(data = data, mapping = aes(x=AGE, y=LIMIT\_BAL))+

geom\_boxplot(aes(fill = EDUCATION))+

facet\_wrap(~TARGET) +

labs(x="Age", y= "Amount of Credit Balance (NT dollar)",

title="Distribution of credit balance per age and education level 1")

#graph to understand better the distribution of credit balance

#in ages and education level

ggplot(data = data, mapping = aes(x=AGE, y=LIMIT\_BAL)) +

geom\_point()+

geom\_smooth(aes(color = EDUCATION))+

facet\_wrap(~EDUCATION) +

labs(x="Age", y= "Credit Balance (NT dollars)",

title="Distribution of credit balance per age and education level 2")

#marital status

ggplot(data = data, mapping = aes(x = MARRIAGE, fill = TARGET)) +

geom\_bar() +

stat\_count(aes(label = ..count..), geom = "label") +

labs(x="Marrital Status", y= "count",

title="Marrital Status distribution per target")

#martial status and balance

ggplot(data = data, mapping = aes(x=MARRIAGE, y=LIMIT\_BAL))+

geom\_boxplot(aes(fill = SEX))+

facet\_wrap(~TARGET) +

labs(x="Marrital Status", y= "Amount of Credit Balance (NT dollar)",

title="Distribution of credit balance per marital status and gender")

#CONTINUOUS VARIABLES

#CHECK the economy variables

#limit balance (LIMIT\_BAL)

#payment delays (PAY\_1-6)

#bill statement (BILL\_AMT1-6)

#payments (PAY\_AMT1-6)

#BILL\_AMT1

#check the distribution of the amount of bills in dataset

#how much money customers need to pay

ggplot(data, aes(x = BILL\_AMT1))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(BILL\_AMT1)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of bill statement in September (BILL\_AMT1)",

caption="With the mean line in red")

#BILL\_AMT2

ggplot(data, aes(x = BILL\_AMT2))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(BILL\_AMT2)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of bill statement in August (BILL\_AMT2)",

caption="With the mean line in red")

#BILL\_AMT3

ggplot(data, aes(x = BILL\_AMT3))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(BILL\_AMT3)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of bill statement in July (BILL\_AMT3)",

caption="With the mean line in red")

#BILL\_AMT4

ggplot(data, aes(x = BILL\_AMT4))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(BILL\_AMT4)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of bill statement in June (BILL\_AMT4)",

caption="With the mean line in red")

#BILL\_AMT5

ggplot(data, aes(x = BILL\_AMT5))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(BILL\_AMT5)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of bill statement in May (BILL\_AMT5)",

caption="With the mean line in red")

#BILL\_AMT6

ggplot(data, aes(x = BILL\_AMT6))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(BILL\_AMT6)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of bill statement in April (BILL\_AMT6)",

caption="With the mean line in red")

#PAY\_AMT1

#check the payment statement

ggplot(data, aes(x = PAY\_AMT1))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(PAY\_AMT1)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of previous payment in September (PAY\_AMT1)",

caption="With the mean line in red")

#PAY\_AMT2

ggplot(data, aes(x = PAY\_AMT2))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(PAY\_AMT2)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of previous payment in August (PAY\_AMT2)",

caption="With the mean line in red")

#PAY\_AMT3

ggplot(data, aes(x = PAY\_AMT3))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(PAY\_AMT3)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of previous payment in July (PAY\_AMT3)",

caption="With the mean line in red")

#PAY\_AMT4

ggplot(data, aes(x = PAY\_AMT4))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(PAY\_AMT4)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of previous payment in June (PAY\_AMT4)",

caption="With the mean line in red")

#PAY\_AMT5

ggplot(data, aes(x = PAY\_AMT5))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(PAY\_AMT5)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of previous payment in May (PAY\_AMT5)",

caption="With the mean line in red")

#PAY\_AMT6

ggplot(data, aes(x = PAY\_AMT6))+

geom\_density(aes(y = ..count..), fill = "lightgray") +

geom\_vline(aes(xintercept = mean(PAY\_AMT6)), linetype = "dashed", size = 0.6, color = "#FC4E07") +

labs(x="NT dollar", y= "count",

title="Distribution of previous payment in April (PAY\_AMT6)",

caption="With the mean line in red")

#PAY\_1-6

#check the distribution of delays

#how people payed their bills(no delay, 1month, 2,.....)

#and also how they choose to pay next month's payment

#PAY\_1

ggplot(data, aes(PAY\_1)) +

geom\_bar(colour="black", mapping = aes(fill = TARGET)) +

labs(x="months of delay", y= "count", fill="default payment next month",

title="Payment Delays Distribution in September(PAY\_1)")

#PAY\_2

ggplot(data, aes(PAY\_2)) +

geom\_bar(colour="black", mapping = aes(fill = TARGET)) +

labs(x="months of delay", y= "count", fill="default payment next month",

title="Payment Delays Distribution in August(PAY\_2)")

#PAY\_3

ggplot(data, aes(PAY\_3)) +

geom\_bar(colour="black", mapping = aes(fill = TARGET)) +

labs(x="months of delay", y= "count", fill="default payment next month",

title="Payment Delays Distribution in July(PAY\_3)")

#PAY\_4

ggplot(data, aes(PAY\_4)) +

geom\_bar(colour="black", mapping = aes(fill = TARGET)) +

labs(x="months of delay", y= "count", fill="default payment next month",

title="Payment Delays Distribution in June(PAY\_4)")

#PAY\_5

ggplot(data, aes(PAY\_5)) +

geom\_bar(colour="black", mapping = aes(fill = TARGET)) +

labs(x="months of delay", y= "count", fill="default payment next month",

title="Payment Delays Distribution in May(PAY\_5)")

#PAY\_6

ggplot(data, aes(PAY\_6)) +

geom\_bar(colour="black", mapping = aes(fill = TARGET)) +

labs(x="months of delay", y= "count", fill="default payment next month",

title="Payment Delays Distribution in April(PAY\_6)")

#violin graph

#check the distribution of target variable in the scale of limit balance variable

#how balance(NT dollars) they have and how they pay(default or not)

ggplot(data = data, aes(x = TARGET, y = LIMIT\_BAL)) +

geom\_violin(aes(fill = TARGET), trim = FALSE, alpha = 0.3) +

geom\_boxplot(aes(fill = TARGET), width = 0.2, outlier.colour = NA) +

theme(legend.position = "NA") +

labs(fill="Payment next month", x="Default Payment Next Month", y= "Limit Balance (NT dollars)",

title="Distribution of credit balance per customers choose for payment",

caption="The boxes present the mean")

###############################################################

#Prepare the data for the methods - SPLIT THE DATA

###############################################################

#split the data

#train\_set 75%

#test\_set 25%

set.seed(123)

split = sample.split(data$TARGET, SplitRatio = 0.75)

train\_set = subset(data, split == TRUE)

test\_set = subset(data, split == FALSE)

#####################################################################

#Find the best variables for our models

#Use Bayesian Information Criterion (BIC) to find the best variables

#using forward selection

best\_sel = regsubsets(TARGET ~. , data = train\_set, method = "forward",

nvmax = length(data)-1)

best\_sel\_sum = summary(best\_sel)

plot(best\_sel\_sum$bic, type = 'b', col = "blue", pch = 19,

xlab = "Number of Variables",

ylab = "Cross-Validated Prediction Error",

main = "Forward Stepwise Selection using BIC")

points(which.min(best\_sel\_sum$bic), best\_sel\_sum$bic[which.min(best\_sel\_sum$bic)],

col = "red", pch = 19)

#list all variable which are important to use

final = t(best\_sel\_sum$which)[,which.min(best\_sel\_sum$bic)]

final\_names = names(data)[-24]

final\_names[final[24:length(data)]]

final\_names

#####################################################################

#lOGISTIC REGRESSION

#####################################################################

#create the logistic regression model

log\_reg <- glm(formula = TARGET ~.,

family = binomial(link="logit"),

data = train\_set)

summary(log\_reg)

#make predictions

log\_pred <- predict(log\_reg, newdata = test\_set[-24], type = 'response')

log\_pred

#set the probabilities >0.5 = yes and <0.5 = no

class\_pred <- as.factor(ifelse(log\_pred > 0.5, 1, 0))

#build the odds for the final model's variables

exp(log\_reg$coefficients)

#confusion matrix and other statistics

confusionMatrix(class\_pred, test\_set$TARGET)

#R^2

logisticPseudoR2s <- function(LogModel) {

dev <- LogModel$deviance

nullDev <- LogModel$null.deviance

modelN <- length(LogModel$fitted.values)

R.l <- 1 - dev / nullDev

R.cs <- 1- exp ( -(nullDev - dev) / modelN)

R.n <- R.cs / ( 1 - ( exp (-(nullDev / modelN))))

cat("Pseudo R^2 for logistic regression\n")

cat("Hosmer and Lemeshow R^2 ", round(R.l, 3), "\n")

cat("Cox and Snell R^2 ", round(R.cs, 3), "\n")

cat("Nagelkerke R^2 ", round(R.n, 3), "\n")

}

logisticPseudoR2s(log\_reg)

head(log\_pred)

class(log\_pred)

#ROC

#transform the input data into a standardized format.

pr <- prediction(log\_pred, test\_set$TARGET)

#All kinds of predictor evaluations are performed using this function.

prf <- performance(pr, measure = "tpr", x.measure = "fpr")

prf

plot(prf)

#AUC

auc <- performance(pr, measure = "auc")

auc <- auc@y.values[[1]]

auc

#Precision and Recall

precision\_recall <- performance(pr, "prec", "rec")

plot(precision\_recall)

#####################################################################

#CHECK ASSUMPTIONS

#####################################################################

#assumption 1

#linearity

train\_set$LIMIT\_BAL\_log <- log(train\_set$LIMIT\_BAL)\*train\_set$LIMIT\_BAL

train\_set$AGE\_log <- log(train\_set$AGE)\*train\_set$AGE

train\_set$BILL\_AMT1\_log <- log(train\_set$BILL\_AMT1)\*train\_set$BILL\_AMT1

train\_set$BILL\_AMT2\_log <- log(train\_set$BILL\_AMT2)\*train\_set$BILL\_AMT2

train\_set$BILL\_AMT3\_log <- log(train\_set$BILL\_AMT3)\*train\_set$BILL\_AMT3

train\_set$BILL\_AMT4\_log <- log(train\_set$BILL\_AMT4)\*train\_set$BILL\_AMT4

train\_set$BILL\_AMT5\_log <- log(train\_set$BILL\_AMT5)\*train\_set$BILL\_AMT5

train\_set$BILL\_AMT6\_log <- log(train\_set$BILL\_AMT6)\*train\_set$BILL\_AMT6

train\_set$PAY\_AMT1\_log <- log(train\_set$PAY\_AMT1)\*train\_set$PAY\_AMT1

train\_set$PAY\_AMT2\_log<- log(train\_set$PAY\_AMT2)\*train\_set$PAY\_AMT2

train\_set$PAY\_AMT3\_log<- log(train\_set$PAY\_AMT3)\*train\_set$PAY\_AMT3

train\_set$PAY\_AMT4\_log<- log(train\_set$PAY\_AMT4)\*train\_set$PAY\_AMT4

train\_set$PAY\_AMT5\_log<- log(train\_set$PAY\_AMT5)\*train\_set$PAY\_AMT5

train\_set$PAY\_AMT6\_log<- log(train\_set$PAY\_AMT6)\*train\_set$PAY\_AMT6

train\_set$PAY\_AMT1<- log(train\_set$PAY\_AMT1)\*train\_set$PAY\_AMT1

formula <- TARGET ~.

model <- glm(formula, family = "binomial", data = train\_set)

summary(model)

#assumption 2

#infuential values

#cook's distance graph

plot(log\_reg, which = 4, id.n = 3)

# Extract model results

model\_data <- augment(log\_reg) %>% mutate(index = 1:n())

#the data for the top 3 values as we can see in the cook's graph

model\_data %>% top\_n(3, .cooksd)

#plot the standardised Residuals

ggplot(model\_data, aes(index, .std.resid)) +

geom\_point(aes(color = TARGET), alpha = .5) +

theme\_bw()

#filter potential infuential data points

model\_data %>% filter(abs(.std.resid) > 3)

#how many variables violate the cook's distance

cook <- cooks.distance(log\_reg)

sum(cook > 1)

#residuals fit

x <- rstandard(log\_reg)

sum(x > 1.96)

#assumption 3

#multicollinearity

vif(log\_reg)

#####################################################################

#Random Forest x2 with tuning (cross validation k-folds = 10)

#####################################################################

#create the random forest model

rf\_class <- randomForest(x = train\_set[-24],

y = train\_set$TARGET,

ntree = 10)

summary(rf\_class)

#make predictions in the test set

rf\_pred <- predict(rf\_class, newdata = test\_set[-24])

rf\_pred

#create the confusion matrix to see the model's performance

confusionMatrix(rf\_pred, test\_set$TARGET)

#evaluate the model using k-fold cross-validation

#search if we can boost the model's performance

fold = createFolds(y = train\_set$TARGET, k=10)

rf\_cross = lapply(fold, function(x){

train\_fold = train\_set[-x, ]

test\_fold = train\_set[x, ]

rf\_class\_cross = randomForest(formula = TARGET ~. ,

data = train\_fold, ntree = 20)

rf\_pred\_cross = predict(rf\_class\_cross, newdata = test\_fold,

type = "class")

rf\_conf = confusionMatrix(rf\_pred\_cross, test\_fold$TARGET)

rf\_conf

accuracy = rf\_conf$overall[1]

return(accuracy)

})

#check the tuning model's accuracy

mean(as.numeric(rf\_cross))

#####################################################################

#Decision Tree Classification

#####################################################################

#create the decision tree model

dt\_class <- rpart(formula = TARGET ~. , data = train\_set, method = 'class')

summary(dt\_class)

#plot the decision tree

plot(dt\_class)

text(dt\_class)

rpart.plot(dt\_class, box.palette = "RdBu", shadow.col = "gray", nn = TRUE)

#make predictions to test set

dt\_pred <- predict(dt\_class, newdata = test\_set, type = 'class')

dt\_pred

#confusion matrix

confusionMatrix(dt\_pred, test\_set$TARGET)

#####################################################################

#SUPPORT VECTOR MACHINE (SVM)

#####################################################################

#create the svm model

svm\_class <- svm(formula = TARGET ~. ,

data = train\_set,

type = 'C-classification',

kernel = 'linear')

summary(svm\_class)

#make predictions to test set

svm\_pred <- predict(svm\_class, newdata = test\_set[-24])

svm\_pred

#confusion matrix

confusionMatrix(svm\_pred, test\_set$TARGET)

#####################################################################

#KERNEL SUPPORT VECTOR MACHINE (KERNEL SVM) (gaussian)

#####################################################################

#create the svm kernel model

svm\_kernel <- svm(formula = TARGET ~. ,

data = train\_set,

type = 'C-classification',

kernel = 'radial')

summary(svm\_kernel)

#make predictions to test set

svm\_pred\_kernel <- predict(svm\_kernel, newdata = test\_set[-24])

svm\_pred\_kernel

#confusion matrix for svm kernel(gaussian)

confusionMatrix(svm\_pred\_kernel, test\_set$TARGET)

#####################################################################

#NAIVE BAYES

#####################################################################

#create the naive bayes model

nv\_class <- naiveBayes(x = train\_set[-24],

y = train\_set$TARGET)

summary(nv\_class)

#make predictions to test set

nv\_pred <- predict(nv\_class, newdata = test\_set[-24], type = "class")

nv\_pred

#confusion matrix for naive bayes

confusionMatrix(nv\_pred, test\_set$TARGET)

#####################################################################

#XGBOOST DART (cross validation k-folds = 5)

#####################################################################

#create the xgboost model

xgb\_class <- train(TARGET ~.,

data = train\_set,

method = "xgbTree", trControl = trainControl("cv", number = 5))

summary(xgb\_class)

# Best tuning parameter mtry

xgb\_class$bestTune

# Make predictions on the test data

xgb\_pred <- predict(xgb\_class, newdata = test\_set$TARGET)

xgb\_pred

head(xgb\_pred)

confusionMatrix(xgb\_pred, test\_set$TARGET)