

Prediction of credit card defaulters with machine learning techniques

Dmytro Volynskyy, Ph.D

**Table of Contents**

[Abstract 3](#_Toc57373772)

[Introduction 4](#_Toc57373773)

[Literature Review 5](#_Toc57373774)

[Dataset 6](#_Toc57373775)

[Approach 8](#_Toc57373776)

[Initial Dataset Analysis 10](#_Toc57373777)

[Exploratory Data Analysis 12](#_Toc57373778)

[Pre-Processing and Feature Selection 16](#_Toc57373779)

[Model Development 19](#_Toc57373780)

[Predictions 20](#_Toc57373781)

[Conclusions 22](#_Toc57373782)

[References 22](#_Toc57373783)

# Abstract

The aim of this scientific project is to find whether a credit card client will have a default next month or no. Here, a large dataset of credit card clients in Taiwan from April 2005 to September 2005 is presented. Each customer in this set is described by twenty-four different features and labeled with a binary value to show if she/he paid or did not to a bank. The main research question and goal is to assess clients based on their demographic factors, credit data, history of payments, and bill statements, and to classify them into two categories of who is likely to have a default payment next month and who won’t. The probability of default payment will be learned and compared for people of various sex and education. Furthermore, the strongest predictors of default payment will be searched and weaker ones will be eliminated such that credit card companies highlight them and can decrease their own risks when approving clients for a new credit card or credit limit increase. In addition, analysis of group segmentation by clients’ demographic factors and education is of great interest in order to improve marketing among those which are presented in less quantity. For these tasks, based on customer’s attributes given in a dataset, well-known supervised machine learning classification algorithms like Random Forest and Naïve Bayes will be used along with R programming language. Then a comparison of their performance parameters will be provided, analyzed and the one with the highest ones will be recommended for practical use.

# Introduction

Outstanding credit payments and consumer defaults are the risks for the macroeconomic conditions as well as for the profitability of financial institutions all over the world. [1] Since they have millions of clients, there is a necessity for them to rely not on someone’s personal discretion but on a reliable algorithm when approving the customers for either a new credit card or a credit limit increase. Therefore, the main goal of this project will be creating a model that can assess, classify, and predict if the credit card user will able to pay their dues next month. For that purpose, an open-source dataset will be used along with the data analysis software RStudio and the machine learning technique will be applied. A data corpus itself presents a list of credit card clients in Taiwan from April 2005 to September 2005, where each of them is described by twenty-four different attributes and labeled with a class variable, which shows if she/he will be able to pay next month or no. [2] The classification algorithms like Random Forest and Naïve Bayes will be used in order to develop models of predicting the credit worthiness of customers based on their demographic factors, credit data, history of payments, and bill statements, and recommend them for practical use in accordance to their performance metrics as well.

# Literature Review

Assessing customer’s delinquency risks in the credit card and loan industry has become a rather important question for banks and other financial institutions in recent decades. [1] Such financial crises as one of 2007-2009 and the worldwide lockdown of 2020 definitely affect them by losing millions of dollars because of people becoming defaulting debtors. [1] With the development of machine learning, this task can be solved much faster and easier however at present a lot of methodologies were suggested which can provide various results for different datasets. Therefore, the performance metrics of those techniques, and their comparison, are still a widespread topic among researchers and professionals.

In their study, Butaru et al. (2016) [1] used models of the decision tree, random forest, and logistic regression for a dependent variable which was delinquency status over different quarters. For assessing the performance of those precision and recall were used as well as two statistics like F-measure and kappa. According to authors, logistic regression shows worse prediction in out-of-sample and out-of-time forecasts in comparison to another two techniques (i.e. for one bank and two-quarter horizon the values of F-measure are 66.1%, 51.7%, and 66.4% for Decision Tree, Logistic Regression and Random Forests respectively as well as kappa statistic equals to 68.4%, 59.5%, 68.5% accordingly). Shetty and Manoj R [3] in their article emphasize on the importance of dealing with imbalanced dataset. For that reason, they use Synthetic Minority oversampling approach (SMOTE) and Adaptive Synthetic Minority oversampling technique (ADASYN). They also demonstrate that Support Vector Machine classificatory performs better on balanced dataset than other algorithms as well as Recall and ROC values increased after balancing of data. Neema and Soibam [4] were comparing the performance of different methods in terms of cost-effectiveness and discovered that Random forest and artificial neural networks (ANN) are the most efficient in this regard. Moreover, they show non-linearity in incurred cost per customer. On the other hand, Random forest takes much time for modeling and if one wants better Matthew’s Correlation Coefficient (MCC), ANN should be preferred.

To summarize, in the forecasting of customer’s default payment it is common to use Random Forest, Decision tree, or ANN algorithms, which proved their accuracy and cost-effectiveness for this purpose. In addition, dealing with imbalanced datasets which are usual for this industry is quite important and should be done before applying the techniques. Thus, in this project in order to receive the results with the highest possible performance statistics, the emphasis will be on the Random Forest technique application, however, the Naïve Bayes’s one will also be used for comparison reasons.

# Dataset

Given dataset is an open-source data and taken from kaggle.com website (<https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset>). It contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. There are thirty thousand records with twenty-five features for each of them. The default.payment.next.month variable is a dependent class variable and can have two labels either 1 or 0. One means that the client will have a default payment next month, zero – no respectively. All attributes are labeled data and are described in the following table.

Table 1 – List of attributes with their types

|  |  |  |  |
| --- | --- | --- | --- |
| # | Column | Variable type | Details |
| 1 | ID | Numeric discrete | ID of each client |
| 2 | LIMIT\_BAL | Numeric continuous | Amount of given credit in NT dollars (includes individual and family/supplementary credit) |
| 3 | SEX | Qualitative binary | Gender (1=male, 2=female) |
| 4 | EDUCATION | Qualitative nominal | 1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown |
| 5 | MARRIAGE | Qualitative nominal | 1=married, 2=single, 3 =others |
| 6 | AGE | Numeric continuous | Age in years |
| 7 | PAY\_0 | Qualitative nominal | Repayment status in September, 2005  (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above |
| 8 | PAY\_2 | Qualitative nominal | Repayment status in August, 2005  Scale same as above |
| 9 | PAY\_3 | Qualitative nominal | Repayment status in July, 2005  Scale same as above |
| 10 | PAY\_4 | Qualitative nominal | Repayment status in June, 2005  Scale same as above |
| 11 | PAY\_5 | Qualitative nominal | Repayment status in May, 2005  Scale same as above |
| 12 | PAY\_6 | Qualitative nominal | Repayment status in April, 2005  Scale same as above |
| 13 | BILL\_AMT1 | Numeric continuous | Amount of bill statement in September, 2005 (NT dollars) |
| 14 | BILL\_AMT2 | Numeric continuous | Amount of bill statement in August, 2005 (NT dollars) |
| 15 | BILL\_AMT3 | Numeric continuous | Amount of bill statement in July, 2005 (NT dollars) |
| 16 | BILL\_AMT4 | Numeric continuous | Amount of bill statement in June, 2005 (NT dollars) |
| 17 | BILL\_AMT5 | Numeric continuous | Amount of bill statement in May, 2005 (NT dollars) |
| 18 | BILL\_AMT6 | Numeric continuous | Amount of bill statement in April, 2005 (NT dollars) |
| 19 | PAY\_AMT1 | Numeric continuous | Amount of previous payment in September, 2005 (NT dollars) |
| 20 | PAY\_AMT2 | Numeric continuous | Amount of previous payment in August, 2005 (NT dollars) |
| 21 | PAY\_AMT3 | Numeric continuous | Amount of previous payment in July, 2005 (NT dollars) |
| 22 | PAY\_AMT4 | Numeric continuous | Amount of previous payment in June, 2005 (NT dollars) |
| 23 | PAY\_AMT5 | Numeric continuous | Amount of previous payment in May, 2005 (NT dollars) |
| 24 | PAY\_AMT6 | Numeric continuous | Amount of previous payment in April, 2005 (NT dollars) |
| 25 | Default.payment.next.month | Qualitative binary | Default payment (class 0=no, 1=yes) |

# Approach

Initial dataset analysis

Exploratory data analysis

Pre-processing and feature selection

Model development

Predictions and conclusions

**Step 1**: Initial dataset analysis

At first, the data will be investigated. Its cleaning and removing of missing values will be performed.

**Step 2**: Exploratory data analysis

In the second step, descriptive statistics of data will be prepared. The univariate and multivariate analysis of variables will be performed with the following visualizations of their results using histograms, boxplots and heat map in order to get better understanding of them. Such graphics will help to see how each variable is distributed among each client, to find outliers and to figure out the correlation between different attributes.

**Step 3**: Pre-processing and feature selection

Further, data is being prepared for machine learning as well as feature selection is done having the results of multivariate analysis. That is to exclude attributes that have little or no effect on the dependent variable and thus to make the model less complicated and robust. Particular methods to deal with imbalanced datasets are applied for that purpose as well.

**Step 4**: Model development

Next, the given dataset is divided into sets – one for training using *k*-fold cross validation technique and another one for validation of the model. Machine learning Random Forest and Naïve Bayes algorithms will be applied to develop models and then to make predictions with them.

**Step 5**: Predictions and conclusions

Lastly, predictions will be done using validation dataset and the results will be compared with the original labels with the help of confusion matrix, F-score and area under the ROC curve. Furthermore, conclusions containing a thorough analysis of the study will be presented.

# Initial Dataset Analysis

The very first and important step of the machine learning process while performing any data analysis is to prepare data for its proper future interpretation. The source code for the project is presented in Rmd file at GitHub link (<https://github.com/dvolynskyy/CIND-820>). Clean data can notably increase the accuracy of any model. Usually we start that in R with reading a dataset file using a command ‘read.csv’. We want to have the headers as in original file and do not have factor values. For this ‘header’ argument is set to ‘True’ and ‘stringAsFactors’ – to ‘False’. Now we can start investigating our data. We would need to learn the shape, size, type and general layout of the data that we have. A command ‘head’ shows us the first six rows and gives us understanding how the dataset looks like. As it was mentioned earlier in this project, there are twenty-five columns describing attributes of each credit card customer in terms of his/her sex, education, age, payment history, bill amount and status of payments in different months (see table 1). To avoid any complications further, our target variable ‘default.payment.next.month’ is going to be renamed to simpler name ‘DEFAULT\_PAYMENT’ using the following code:

colnames(projectData)[colnames(projectData)=="default.payment.next.month"] <- "DEFAULT\_PAYMENT".

Then with ‘dim’ function we can easily check that the given data set has 30000 records in it. Using a standard command ‘str’ we may find out of what type is each variable in a data frame as it shown in the following screen capture:

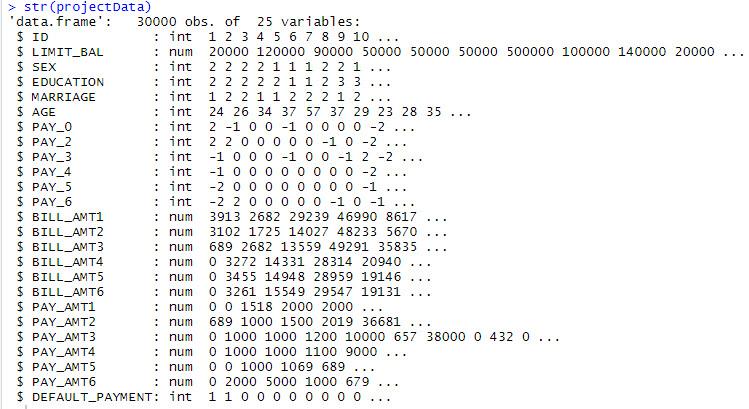


Figure 1 – results of str() function

Since we have a classification problem in this project, there is a necessity to know the proportion of instances that belong to each class label. This is important because it may highlight an imbalance in the data, that if severe may need to be addressed with rebalancing techniques.

y <- projectData$DEFAULT\_PAYMENT

cbind(freq=table(y), percentage=prop.table(table(y))\*100)

The code above creates a useful table showing the number of instances that belong to each class as well as the percentage that this represents from the entire dataset:



Figure 2 – Frequency of class instances in a dataset

As it is seen, the given dataset has rather imbalanced data where class ‘0’ is presented triple as much as class ‘1’. Due to this we will need to address this issue with specific methods.

# Exploratory Data Analysis

Applying the summary() function to a data frame will return the summary (Fig. 3) showing main descriptive statistics (min, 25 percentile, median, mean, 75 percentile, max) for all numeric values. It also indicates, if applicable, the number of missing values for an attribute (marked NA). In this case there no missing values at all.

Comparing details of data description from table 1 and the results of summary() function, it is seen that two attributes, Education and Marriage, have categories either not included in the dataset description or are meaningless (Fig. 4).

From the data description received in table 1 we can see that the Education varible has the following categories: 1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown. In addition, we can observe that this attribute holds 0 number as its value, which is not described in the dataset. We may assume that these 0 values along with 5 and 6 can be categorized under value 4 (other).

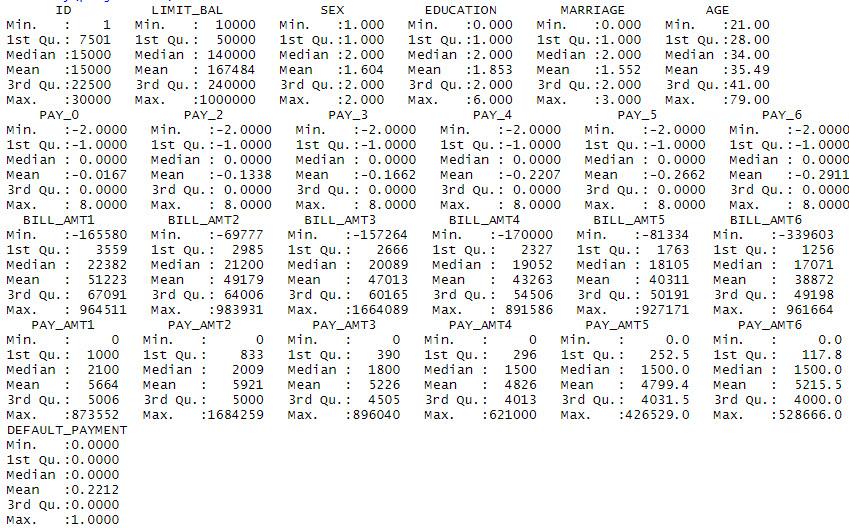


Figure 3 – Results of summary() function

A similar situation is for Marriage variable which holds values like 1=married, 2=single, 3 =others. Since the category 0 hasn’t been defined anywhere, we will include it in the ‘others’ category marked as 3.

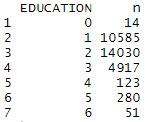
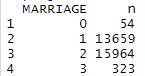
 

Figure 4 – Frequency of Education and Marriage variable categories

The following code do the actions mentioned above:

projectData$EDUCATION[projectData$EDUCATION == 0] <- 4

projectData$EDUCATION[projectData$EDUCATION == 5] <- 4

projectData$EDUCATION[projectData$EDUCATION == 6] <- 4

projectData$MARRIAGE[projectData$MARRIAGE == 0] <- 3

projectData %>% count(EDUCATION, sort = FALSE)

projectData %>% count(MARRIAGE, sort = FALSE)

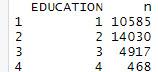
 

Figure 5 – Frequency of Education and Marriage variable categories after recategorizing

Another important aspect is to explore how the attributes related between each other. For this purpose, the correlation of each pair of numeric attributes will be considered.

correlations <- cor(projectData[,2:25])

print(correlations)

plot\_correlation(na.omit(projectData), maxcat = 5L)

The code above creates the correlation table between all pairs of attributes. Correlation values closer to one or minus one show stronger positive or negative correlations. This code generates a following heat map (Fig. 6) of variables correlations as well.

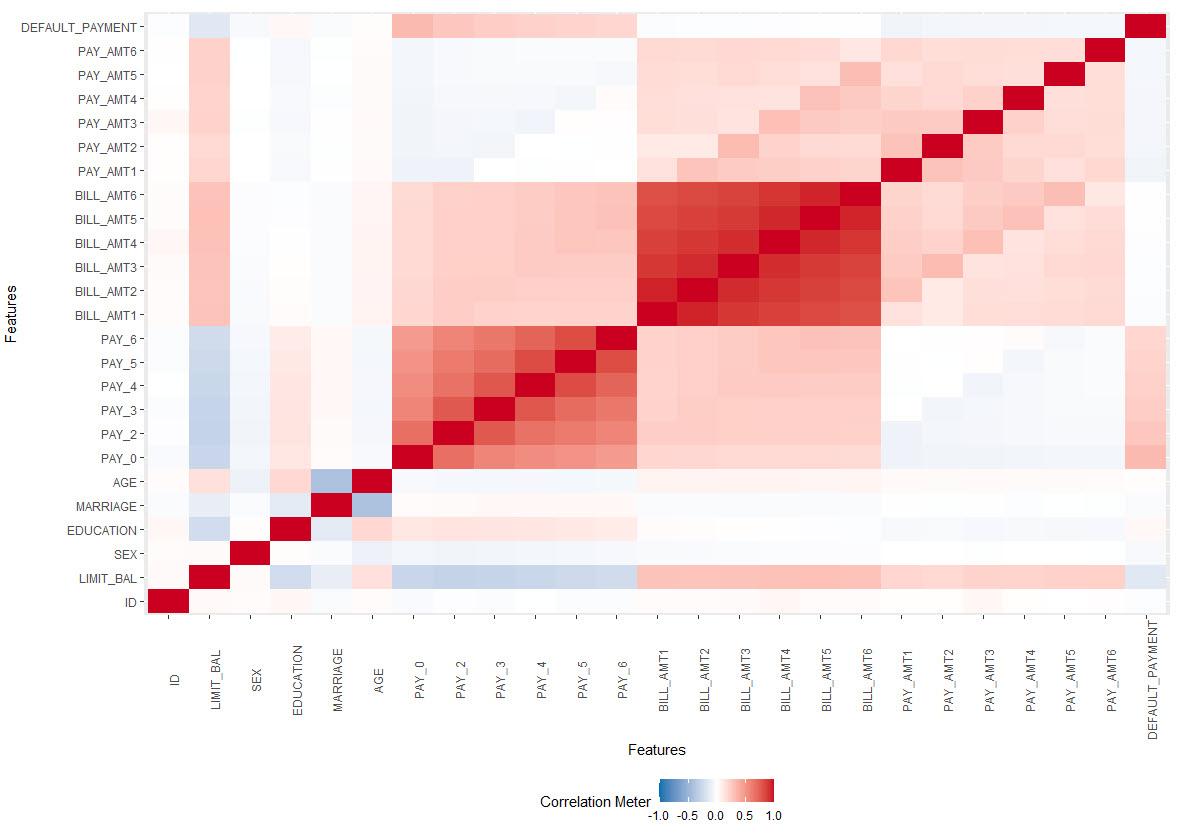
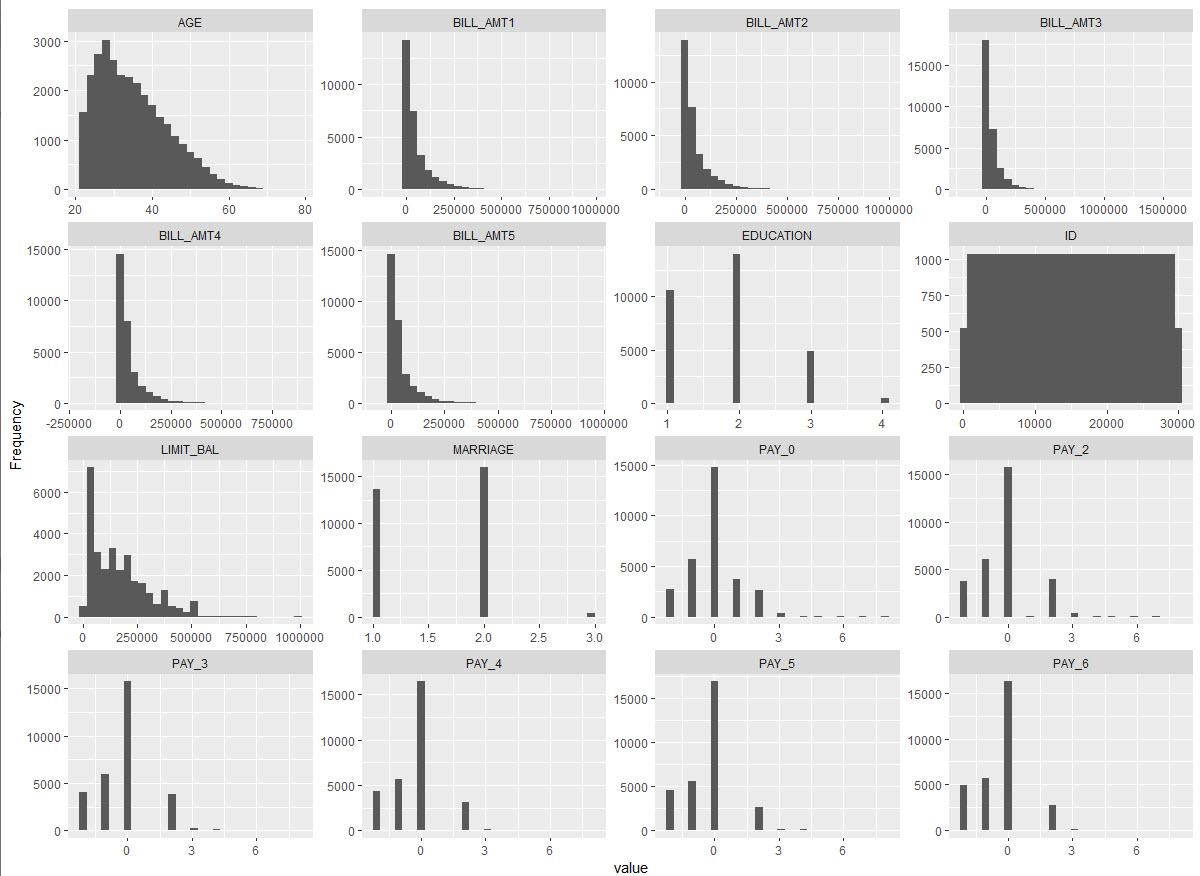


Figure 6 – Heat map of correlations between dataset attributes

From this heat map it is seen that particular variables like BILL\_AMT1, BILL\_AMT2, BILL\_AMT3, BILL\_AMT4, BILL\_AMT5, BILL\_AMT6 have a very weak correlation with a class attribute and thus have no impact on a final outcome. Due to this we will not include them in machine learning models.

With a following piece of code, we create a histogram for each attribute (Fig. 7) and thus performing a univariate analysis of them:

plot\_histogram(projectData)



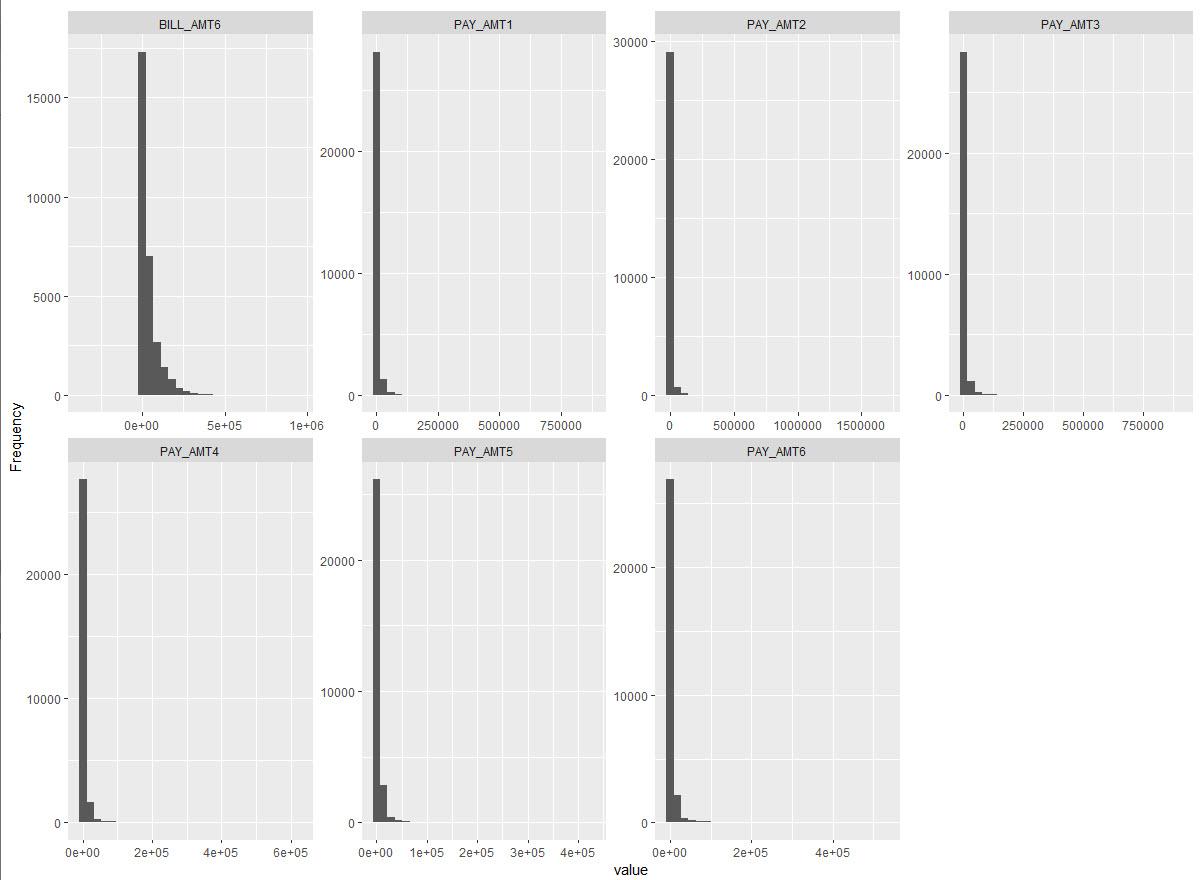


Figure 7 – Histograms for dataset attributes

The above graphs show that all PAY variables are skewed to the right. Otherwise there are no specific dependencies to highlight.

# Pre-Processing and Feature Selection

Next step is to prepare dataset for modelling. As it was mentioned above BILL\_AMT1, BILL\_AMT2, BILL\_AMT3, BILL\_AMT4, BILL\_AMT5, BILL\_AMT6 attributes have no impact on a class variable and should be deleted from it. That is done with the following code

modelData <- select(projectData, !c('ID', 'BILL\_AMT1','BILL\_AMT2','BILL\_AMT3','BILL\_AMT4','BILL\_AMT5','BILL\_AMT6'))

Then we plot a new heat map of correlations after removing attributes from the dataset.

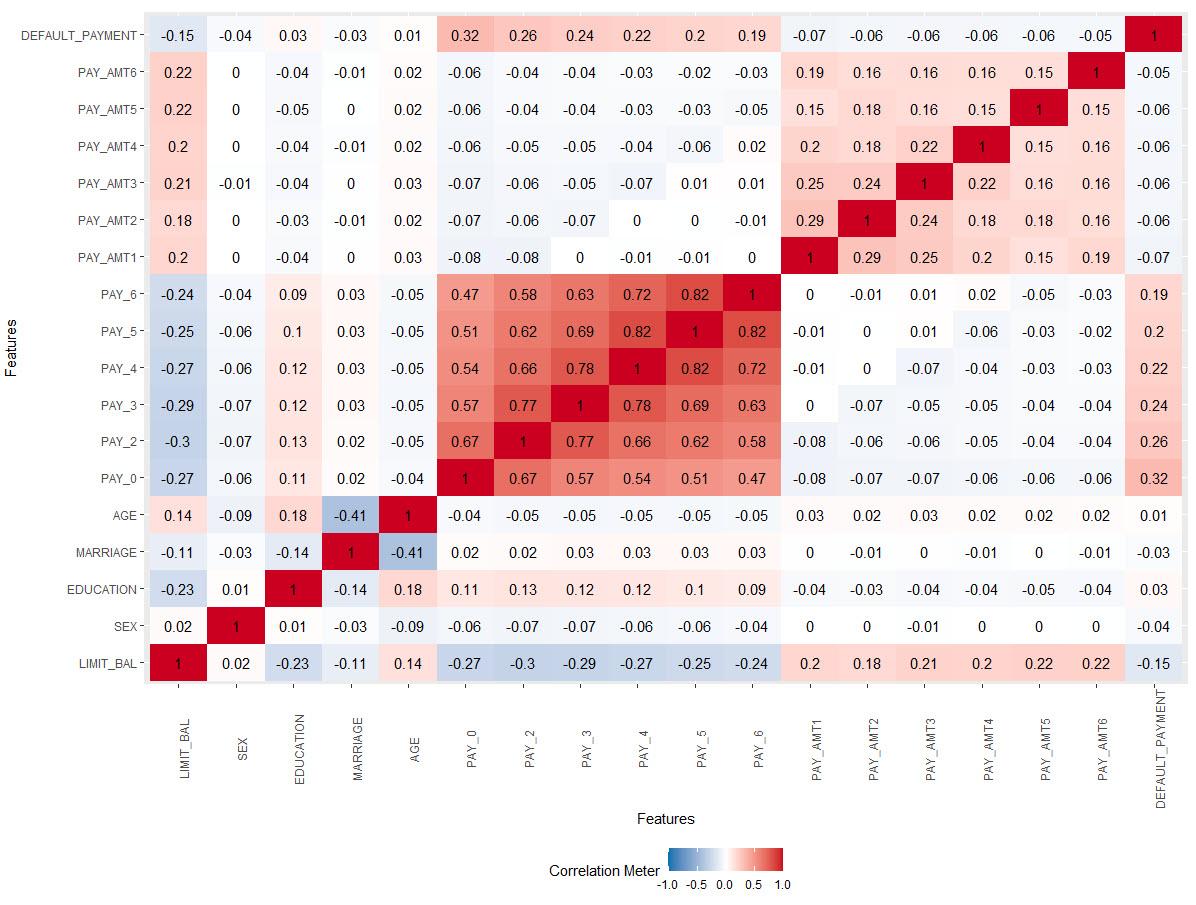


Figure 8 – Heat map of correlations between dataset attributes after removing features

Taking into consideration the heat map (Fig. 8) it is seen that even after removing specific attributes, all others have not so high correlation with the class variable.

Further we need to deal with class imbalance. For this reason, a CRAN package ‘Imbalance’ is installed. Specifically, a majority weighted minority oversampling technique (MWMOTE) for imbalance dataset will be used. It is a modification for SMOTE technique which overcomes some of the problems of the SMOTE technique when there are noisy instances, in which case SMOTE would generate more noisy instances out of them. After creating new artificial records as a separate data frame, we use function ‘rbind’ to merge two datasets and then we shuffle it in order not to have only the records of the same outcome class in each fold of the future *k*-fold cross validation modelling. Another correlation heat map (Fig. 9) is created to see how new instances in a dataset affected dependencies between each other. It is vividly seen that the values for PAY variables decreased and PAY\_AMT increased a little bit in comparison with previous heat map. Nevertheless, we assume that this change is not critical and will not biased the model itself very much.

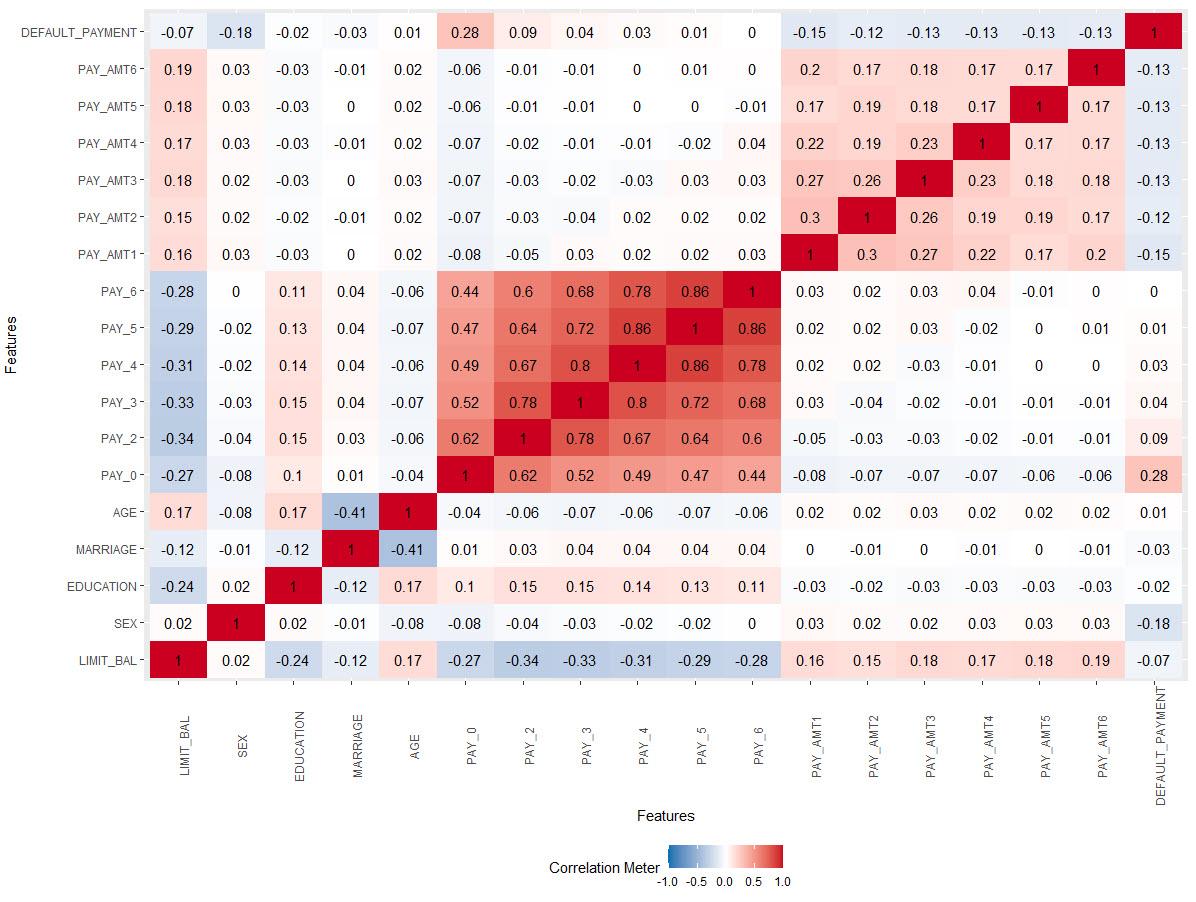


Figure 9 – Heat map of correlations between dataset attributes after data balanced with using MWMOTE() function

Then we check again in what proportion the records with different class values are presented. Execution of this code

z <- newData$DEFAULT\_PAYMENT

cbind(freq=table(z), percentage=prop.table(table(z))\*100)

shows that there is a ratio of 58.41% to 41.59% of instances that belong to class ‘0’ and class ‘1’ respectively. Due to this we assume that our dataset is balanced now and we can proceed to the machine learning modelling.

# Model Development

Now we move on to developing of models. For this purpose, a CRAN package ‘caret’ is going to be used. At first, the original dataset is split into training and validation sets in proportion 70% to 30% respectively. Then the models are to be trained on the test set using 10-fold cross-validation technique and then validated using a validation set of data.

We start from developing Random Forest and Naïve Bayes models without pre-processing of the data with standardization. Then another two models are created using the same machine learning supervised techniques however with centering and scaling. After developing their performance metrics are printed out, the fitted results and the graphs of importance of variables are plotted.

At start, a trControl argument is defined. The method is set to cross-validation with *k* equal to 10, classProbs – to TRUE. It is required to return probability of outcome. In this case it is ‘default’ or ‘not’. And summaryFunction is set to return outcome as a set of binary classification results.

First the models without pre-processing are being developed. As a method in train() function we set ‘rf’ for the Random Forest and ‘nb’ for the Naive Bayes algorithms, metric is set to ‘ROC’ so that the output data is suitable for generating an ROC curve

Then the same modelling is performed, but with setting additional argument preProc=c(‘center’, ‘scale’) in order to perform data standardization.

# Predictions

As long as all models are developed, now we proceed to their validation using a respective part of dataset. After each model returns the outcomes, a confusion matrix is created to get the performance metrics of each classifying algorithm. The plots of ROC curves for each method are generated (Fig. 10) and the area under the ROC curve is calculated.

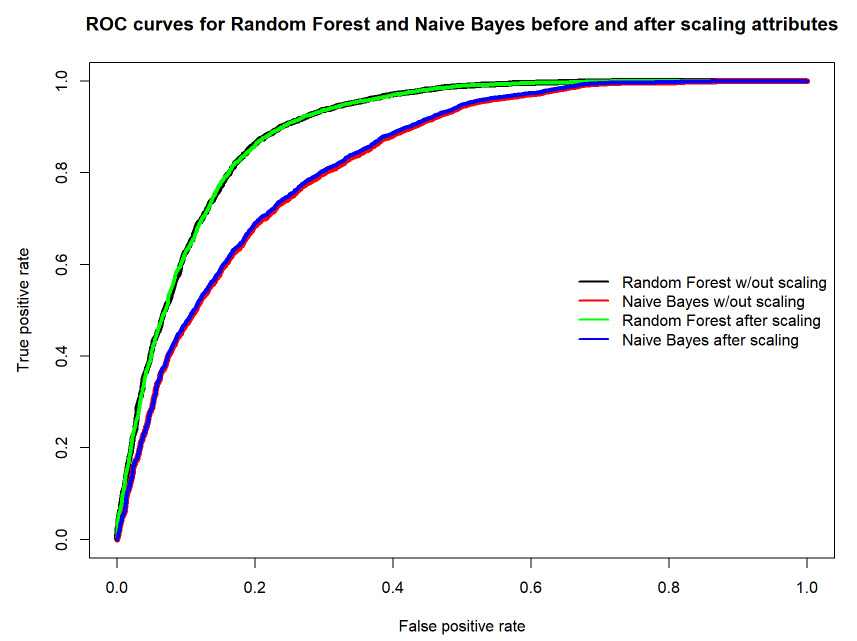


Figure 10 – ROC curves for Random Forest and Naïve Bayes before and after standardization

In the end we create a summary table (Table 2) with performance metrics for each developed model in order to compare them and to decide which is the best in our case. For the purpose of comparison of the performance of our models the following parameters from a confusion matrix are going to be used: Accuracy, Precision, Recall, F1 measure and Area under ROC curve (AUC).

Table 2 – Summary table of the developed models’ performance metrics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ML Technique | Accuracy | Precision | Recall | F1 Measure | AUC |
| Random Forest | 0.845 | 0.837 | 0.914 | 0.874 | 0.898 |
| Naïve Bayes | 0.717 | 0.836 | 0.645 | 0.728 | 0.829 |
| Random Forest after standardization | 0.844 | 0.837 | 0.911 | 0.873 | 0.897 |
| Naïve Bayes after standardization | 0.720 | 0.837 | 0.650 | 0.732 | 0.831 |

Accuracy represents the percentage of correct predictions. It is calculated by dividing the number of correct predictions by the number of total predictions. Precision shows the fraction of relevant examples (true positives) among all of the instances which were predicted to belong to a certain class. And recall demonstrates the percentage of total relevant results correctly classified by the algorithm. ROC graphs show us true positive rates plotted against false positive rates. The value of the area under the curve refers to the ability of the classifier to correctly classify true or false case of an attribute. The closer the area under the curve is to 100% the better the classifier.

The first thing that we can state looking at the summary table is that pre-processing of data (standardization) merely does not affect the model accuracy and other metrics. The difference between values of each parameter is less than 0.5%. On the other hand, it is vividly seen, that the Random Forest model greatly overperform the Naïve Bayes method by all parameters. This concludes the Random Forest is a better classifier for predicting clients who will be in default next month and will not be able to pay the credit card bill.

In comparison to the results described in [3], where the authors after balancing the dataset with the SMOTE technique got much smaller values of the accuracy, recall, precision and area under ROC curve for the Random Forest classifier (Accuracy – 0.77, Precision – 0.48, Recall – 0.43, AUCROC – 0.65), we can say that the Random Forest models developed in this project are rather accurate and reliable for their practical application.

# Conclusions

The objective of this project was to devise an effective classifier for a credit card company to predict clients who can be in default and will not pay their credit card bills. First of all, the data was cleaned with removing undocumented and mislabeled categories of variables. Then it was explored that among 23 attributes for each client six of them do not affect the outcome class. They were BILL\_AMT1, BILL\_AMT2, BILL\_AMT3, BILL\_AMT4, BILL\_AMT5, BILL\_AMT6. Therefore, they were excluded from the dataset. Secondly, the oversampling technique such as MWMOTE was used to balance the data. Then predictive analysis of data corpus was performed with the help of two classifiers, particularly the Random Forest and the Naïve Bayes for the classification purpose. One thing worth to mention is that in these models the attributes PAY\_AMT1 and PAY\_AMT2 have the biggest impact on the prediction of who will have a default payment. As a result of their F1 Measure and other metrics comparison, it was concluded that for this data the Random Forest classifier works best having the highest values of F1 Measure (0.87) along with accuracy (0.84) and recall (0.91). Therefore, it can be used by the credit card company to predict results of worthiness of its customers.

In regards of perspective, this work can be improved by adding other machine learning classification techniques for comparison. Moreover, it would be beneficial to use hyperparameters in order to tune each model for getting more accurate results when predicting default customers. Defining a classifier which takes less time and gives more reliable prediction is of great interest as well since running the model on a big dataset may take a long time and can pose a problem of trade-off between performance and time consuming.

# References

1. F. Butaru, Q. Chen, B. Clark, S. Das, A. W. Lo, and A. Siddique, “Risk and risk management in the credit card industry,” *Journal of Banking and Finance*, no. 72, pp. 218–239, 2016.
2. Lichman, M. (2013). UCI Machine Learning Repository [[http://archive.ics.uci.edu/ml]](http://archive.ics.uci.edu/ml%5d). Irvine, CA: University of California, School of Information and Computer Science.
3. A. S. Shetty and Manoj R, “Prediction of Default Credit Card Users Using Data Mining Techniques,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 7, pp. 816–821, May 2019.
4. S. Neema and B. Soibam, “The comparison of machine learning methods to achieve most cost-effective prediction for credit card default,” *Journal of Management Science and Business Intelligence*, vol. 2, no. 2, pp. 36–41, Aug. 2017.