STATISTICAL AND MACHINE LEARNING

INDIVIDUAL ASSIGNMENT

INTRODUCTION

This project aims to predict credit card defaults payment in a Taiwanese bank during the 2000s. The data table consists of 24 variables and a dummy target variable: default.payment.next.month. These variables describe each customer according to their demographic information such as gender, education, marital status, age and also their banking information: their payment status, their bill amount, and their previous payment amount from April to August 2005. We will then ask ourselves, what are the probabilities of default for the following month according to their profile or what are the strongest variables in terms of predicting default. We will answer these questions using 5 classification algorithms in Machine Learning.

1) THE DATA PROCESSING

Before using any algorithm, it is important to clean up the data paying attention in particular to missing values. To do this I have treated encoded variables differently from non-encoded ones.

- For encoded data, I replaced the missing values by the mode. (The most replicated one). These variables are gender, education and marriage.
- For non-encoded data, I replaced the values with the mean or median, depending on whether the variables are categorical or not. PAY_0 to PAY_6 being categorical but whose order was still meaningful, was replaced by the median and LIMIT_BAL also.

Age, BILL_AMT and PAY_AMT were replaced by the mean. The age was then rounded to an integer.

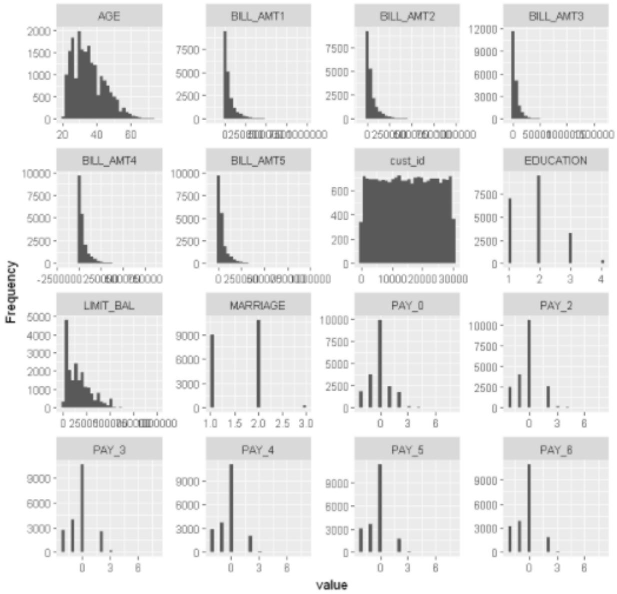
One of the great features of R is that we can use 'factors'. This avoids the need to apply dummy variables or to use age categories for example. Factoring will deal in the same way and include categorization. I have used it for the variables gender, marriage, education and default.payment.next.month.

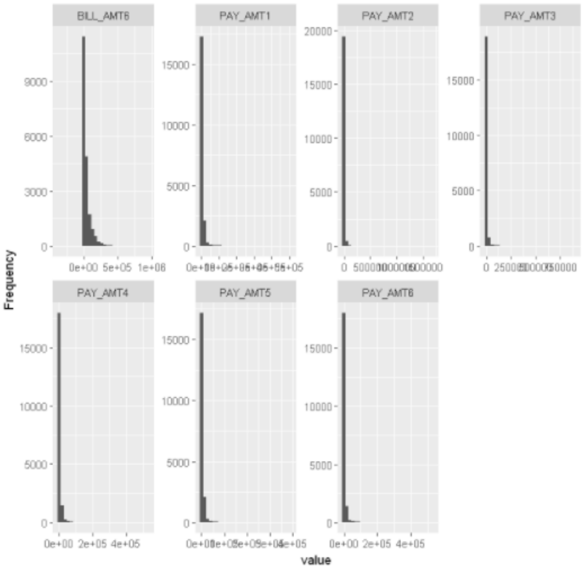
Finally, in the categorical data I sometimes saw the value zero which was an unattributed item according to our excel. I therefore replaced them with the expression "other" in the variables of marriage and education.

2) EXPLORATORY ANALYSES

After cleaning the data and before creating the models it is always interesting to have a look at exploratory analyses as Trevor Hastie and Robert Tibshirani would say in their book *"An introduction to statistical Learning" (2013).* For this we can start with a simple plot of our entire table.

By plotting the histograms of my data table, I was able to obtain the following graphs:





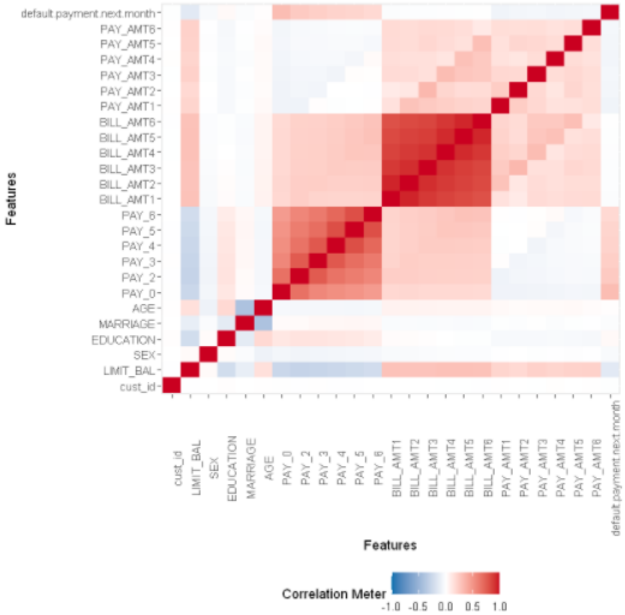
We can notice that most of our variables are positively skewed. This means that the mean will be larger than the median and the mode. An exception occurs for PAY_0 to PAY_6 which are negatively skewed.

By doing some statistical research we can already give the following observations:

- The average age of the clients is 35 years
- 61% of clients are women and 39% are men
- 53% of our clients are single and 45% are married (2% for the others)
- The average credit amount granted is equal to $166,432
- The average AMT of the bills is equal to 44 826$ and decreases month after month
- The average AMT PAID is equal to $5229.
- 22% of our customers pay the default payment the following month (Our target variable)

On the next research I decided to look at the correlations between my variables and also to draw statistics from them using p-values.

a. THE CORRELATION MATRIX



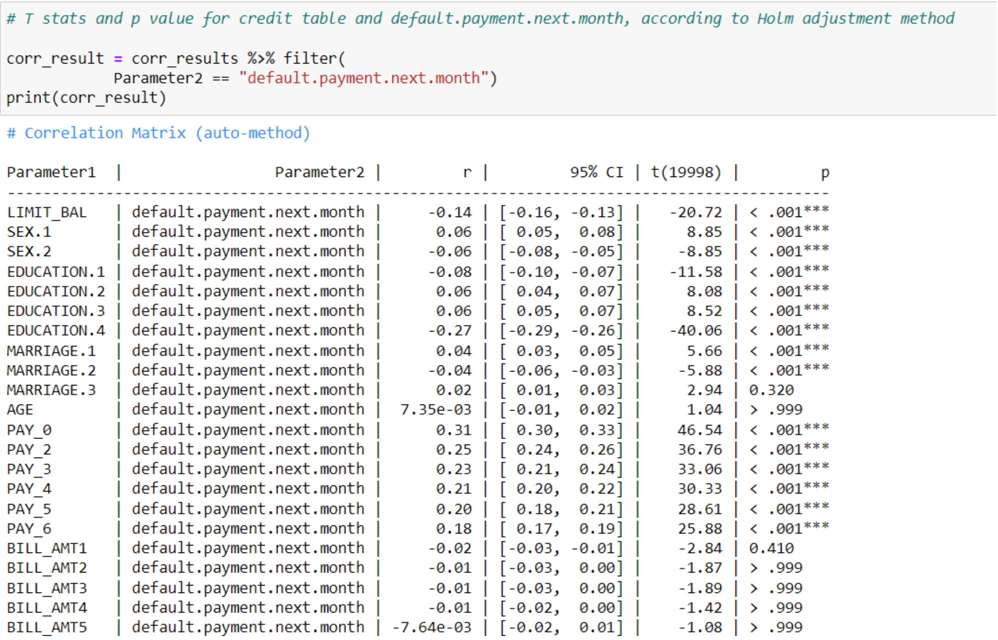
The correlation matrix will help us to choose which variables we should consider for the algorithms.

The most correlated variables to our target one (default_payment) are the payment ones (from PAY_0 to PAY_6). There is also a negative correlation between LIMIT_BAL and default_payment. PAY AMT only has a slight negative correlation but it is not obvious yet.

The correlation matrix can also help us to sort out the less important variables. In other words, those that are not significant in the p-value.

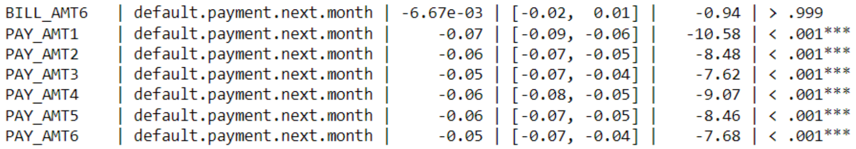
We will see this in depth in the next section.

b. STATISTICS AND P VALUE



```
# T stats and p value for credit table and default.payment.next.month, according to Holm adjustment method

corr_result = corr_results %>% filter(
            Parameter2 == "default.payment.next.month")
print(corr_result)
```

```
# Correlation Matrix (auto-method)

Parameter1   |                Parameter2 |         r |         95% CI | t(19998) |        p
--------------------------------------------------------------------------------------------
LIMIT_BAL    | default.payment.next.month |     -0.14 | [-0.16, -0.13] |   -20.72 | < .001***
SEX.1        | default.payment.next.month |      0.06 | [ 0.05,  0.08] |     8.85 | < .001***
SEX.2        | default.payment.next.month |     -0.06 | [-0.08, -0.05] |    -8.85 | < .001***
EDUCATION.1  | default.payment.next.month |     -0.08 | [-0.10, -0.07] |   -11.58 | < .001***
EDUCATION.2  | default.payment.next.month |      0.06 | [ 0.04,  0.07] |     8.08 | < .001***
EDUCATION.3  | default.payment.next.month |      0.06 | [ 0.05,  0.07] |     8.52 | < .001***
EDUCATION.4  | default.payment.next.month |     -0.27 | [-0.29, -0.26] |   -40.06 | < .001***
MARRIAGE.1   | default.payment.next.month |      0.04 | [ 0.03,  0.05] |     5.66 | < .001***
MARRIAGE.2   | default.payment.next.month |     -0.04 | [-0.06, -0.03] |    -5.88 | < .001***
MARRIAGE.3   | default.payment.next.month |      0.02 | [ 0.01,  0.03] |     2.94 | 0.320
AGE          | default.payment.next.month |  7.35e-03 | [-0.01,  0.02] |     1.04 | > .999
PAY_0        | default.payment.next.month |      0.31 | [ 0.30,  0.33] |    46.54 | < .001***
PAY_2        | default.payment.next.month |      0.25 | [ 0.24,  0.26] |    36.76 | < .001***
PAY_3        | default.payment.next.month |      0.23 | [ 0.21,  0.24] |    33.06 | < .001***
PAY_4        | default.payment.next.month |      0.21 | [ 0.20,  0.22] |    30.33 | < .001***
PAY_5        | default.payment.next.month |      0.20 | [ 0.18,  0.21] |    28.61 | < .001***
PAY_6        | default.payment.next.month |      0.18 | [ 0.17,  0.19] |    25.88 | < .001***
BILL_AMT1    | default.payment.next.month |     -0.02 | [-0.03, -0.01] |    -2.84 | 0.410
BILL_AMT2    | default.payment.next.month |     -0.01 | [-0.03,  0.00] |    -1.87 | > .999
BILL_AMT3    | default.payment.next.month |     -0.01 | [-0.03,  0.00] |    -1.89 | > .999
BILL_AMT4    | default.payment.next.month |     -0.01 | [-0.02,  0.00] |    -1.42 | > .999
BILL_AMT5    | default.payment.next.month | -7.64e-03 | [-0.02,  0.01] |    -1.08 | > .999
```

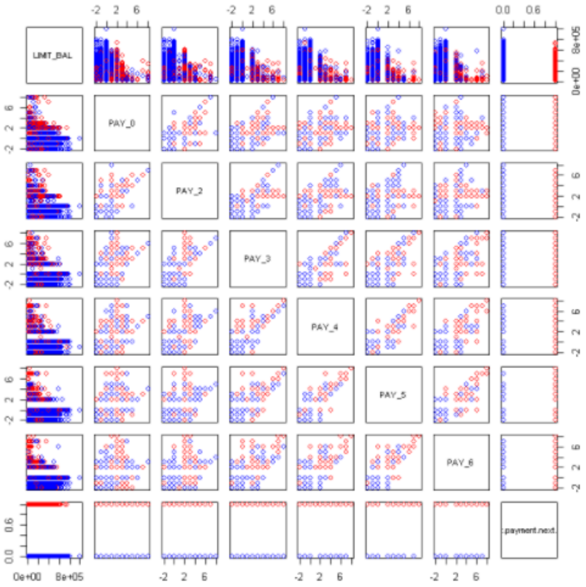


```
BILL_AMT6    | default.payment.next.month | -6.67e-03 | [-0.02,  0.01] |    -0.94 | > .999
PAY_AMT1     | default.payment.next.month |     -0.07 | [-0.09, -0.06] |   -10.58 | < .001***
PAY_AMT2     | default.payment.next.month |     -0.06 | [-0.07, -0.05] |    -8.48 | < .001***
PAY_AMT3     | default.payment.next.month |     -0.05 | [-0.07, -0.04] |    -7.62 | < .001***
PAY_AMT4     | default.payment.next.month |     -0.06 | [-0.08, -0.05] |    -9.07 | < .001***
PAY_AMT5     | default.payment.next.month |     -0.06 | [-0.07, -0.05] |    -8.46 | < .001***
PAY_AMT6     | default.payment.next.month |     -0.05 | [-0.07, -0.04] |    -7.68 | < .001***
```

By looking at the t statistics and p values my hypotheses become clearer. Each of the variables was compared to the target and I could draw the following conclusions: The marriage3, age, and all BILL_AMT variables are not significant. The reason why marriage 3 is not significant is probably due to the fact that many values were misclassified and thus had to be categorized as 'other' during the cleaning. From this table, it also appears that all other variables are highly significant with a p-value below 0.05 for most of them which is an excellent information!

The t statistic measures the effect of one variable on the other. In this case it is extremely significant for values such as pay and education. Usually, a t statistic of 2 is already considered as very meaningful, in our case it can reach 46 or -40. Negative t-statistics simply means a negative correlation, but which may still be very significant. To conclude, the effect of pay, the balance limit but also education play the most important roles.

c. PLOTTING THE MAIN DIFFERENCES



In order to observe the main differences between features I have based myself on the variables that are most correlated within each other. This allows me to observe which predictors are the highest by comparing clients who paid next month in red and clients who hasn't in blue.

In each monthly payment (from PAY_0 to PAY_6), we can see that the default values are generally higher than the others.

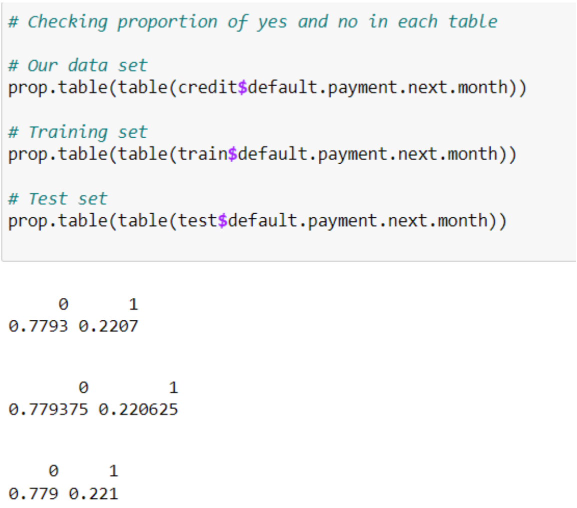
High rates of this PAY variable are synonymous with delay. This indicates that, in general, their late payment exacerbates other delays.

In the next step we will then predict which customers will be able to pay next month through the use of algorithms.

MACHINE LEARNING MODELS

I decided to use 5 Machine Learning algorithms corresponding to data classification namely: LDA, QDA, Logistic Regression, Decision Tree and Random Forest to predict the default payment.

In order to run our models, I split the data into train (80%) and test set (20%). For this, I removed all the unnecessary variables such as client_id. I checked if the number of 1's and 0's was equivalent in each table: train, test and start table.



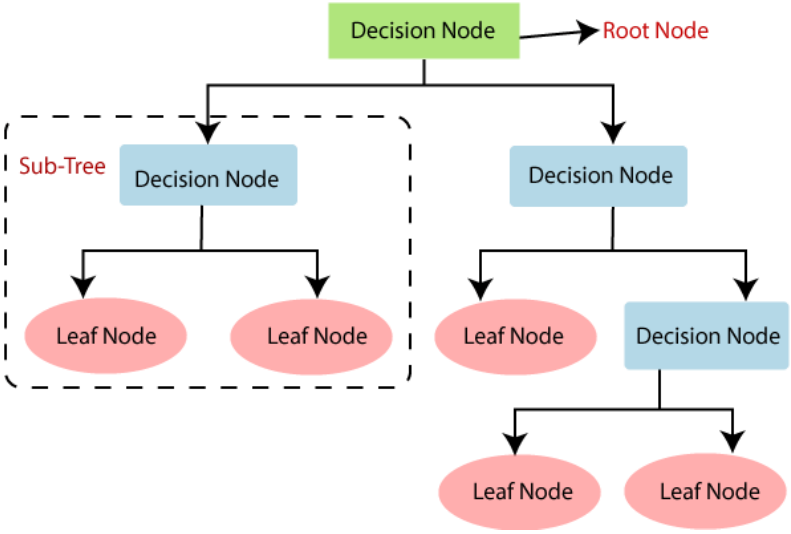
```
# Checking proportion of yes and no in each table

# Our data set
prop.table(table(credit$default.payment.next.month))

# Training set
prop.table(table(train$default.payment.next.month))

# Test set
prop.table(table(test$default.payment.next.month))
```

```
     0      1
0.7793 0.2207


       0        1
0.779375 0.220625


    0     1
0.779 0.221
```

The distribution is similar. We have about 22% paid versus 78% unpaid on each table, so we can start with the algorithms.

1. Decision Tree

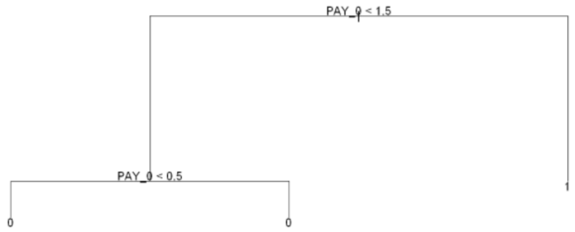
Decision Tree is part of the decision-making algorithms, used for classification or regression and known as a supervised learning algorithm in Machine Learning. Decision Tree as its name may indicates is composed of branches and nodes. Each node separates the branch into two splits corresponding to a criterion. The data is split from the most general idea up to the most specific one until it achieves a special unit. To make a prediction, the mean or the mode are used from the train data.



https://www.devops.ae/decision-tree-classification-algorithm/

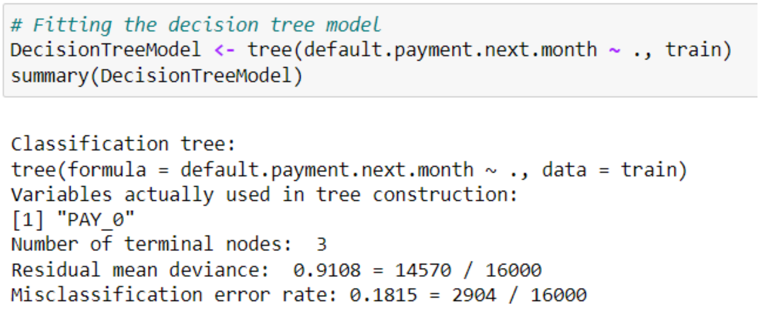
This algorithm is very easy to interpret and fast, however the accuracy may be lower than the other possibilities. Indeed, it is often criticized as not powerful enough for complex data.

In our case this is indeed the case, even if I had a good accuracy (0.81) the AUC is however very low and therefore consider as no discriminant (0.32).



We can see on the graphic above that our decision tree has 4 branches and 3 endpoints. The first decision node corresponds to the PAY_0 bigger than 1.5, in that case our target variable will be equal to 1. The second node, corresponding to the 'chance node' will be 0 even if PAY_0 is bigger or smaller than 0.5.

To fit the process, I used the train data thanks to the 'tree' library which gave me the following summary:

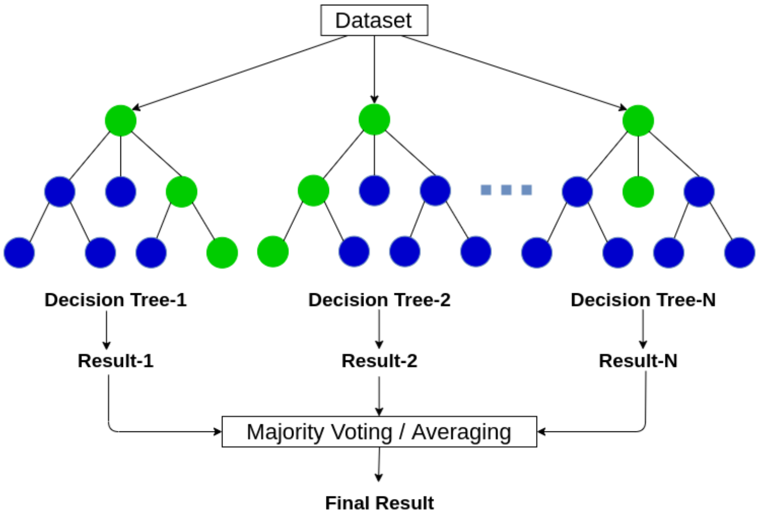


```
# Fitting the decision tree model
DecisionTreeModel <- tree(default.payment.next.month ~ ., train)
summary(DecisionTreeModel)
```

```
Classification tree:
tree(formula = default.payment.next.month ~ ., data = train)
Variables actually used in tree construction:
[1] "PAY_0"
Number of terminal nodes:  3
Residual mean deviance:  0.9108 = 14570 / 16000
Misclassification error rate: 0.1815 = 2904 / 16000
```

2. Random Forest

Random forest is also part of the decision-making algorithms. It is used for supervised learning and can handle both classification or regression target. Random Forest unlike Decision Tree makes plenty of decisions trees to create a final one based on the majority. Moreover, it does not select the best choice as DT and thanks to the diversity of decisions created the algorithm become smoother.

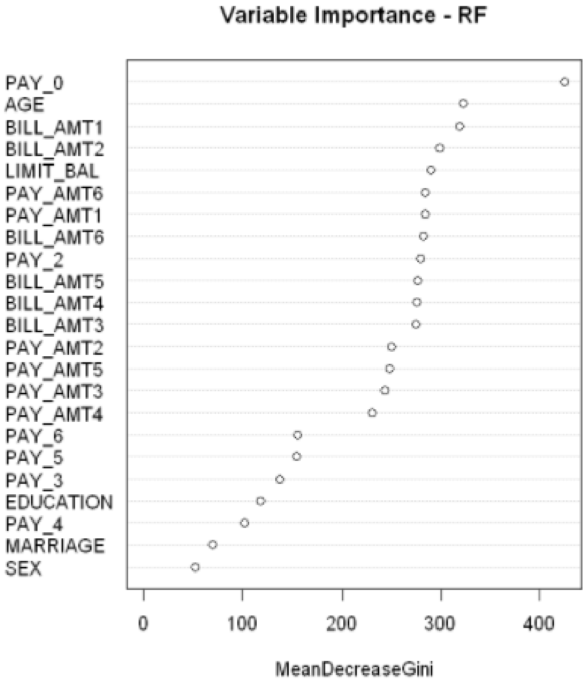


https://ai-pool.com/a/s/random-forests-understanding

The advantages of the random forest is that it is powerful, easy interpretable and accurate. However, as random forest run multiple trees at the same time the prediction can take longer and can be biased in terms of chosen variables.

For the fitting aspect, I run my model thanks to the randomForest library in R with my train data, targeting 'default.payment.next.month' with 10 trees. To get a better result, I can even increase these number of trees. From this, I predicted my model and got 80% of accuracy with a considered and acceptable AUC of 0.73. After K-fold cross validation I obtained a better accuracy with 81%.

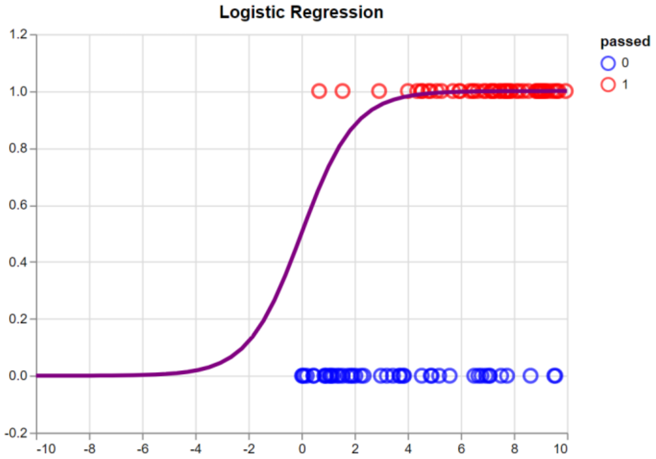
The variable importance below shows a high significance for PAY_0, Age, and BILL_AMT1.



3. Logistic Regression

The Logistic Regression is a supervised algorithm used in Machine Learning for classification problems. It is represented by a logistic function: p(X) = (e β0+β1X) / (1+e β0+β1X) in order to model the dependent variable. While using Binomial Logistic Regression you must pay attention to the number of classes as you can only have two of them, in other words: a binary target variable. Multinomial Logistic Regression is used when dealing with more than 2 classes and a quantitative problem. Finally, Ordinal Logistic Regression is made for categorical target variable. While using LR, the probabilities are predicted thanks to the sigmoid function.

Example:



https://sci2lab.github.io/ml_tutorial/logistic_regression/

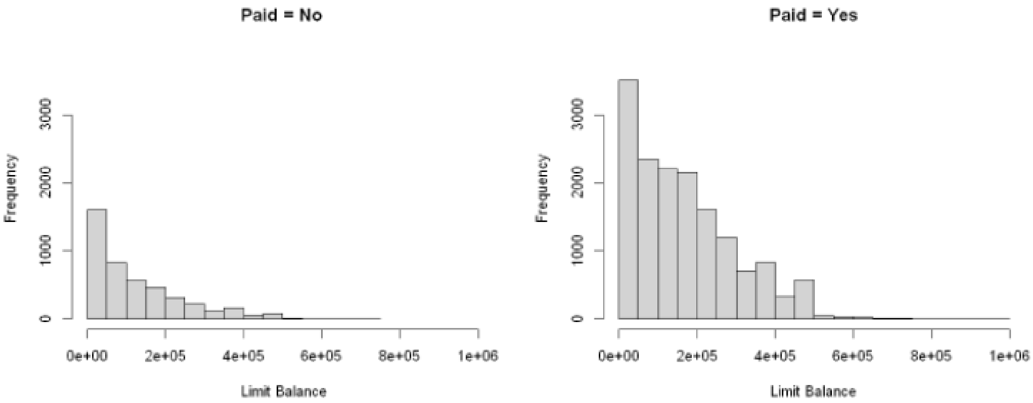
The violet curve represents the Sigmoid function.

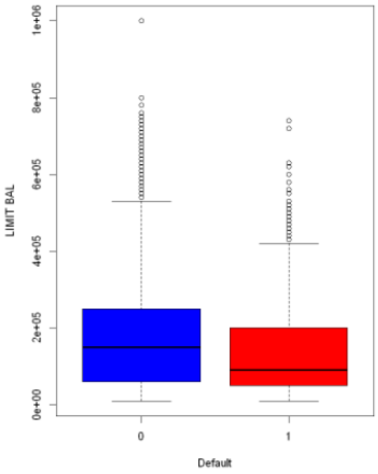
Logistic Regression is a simple algorithm, easy to interpret and that you can train fast. It also gives the direction of the association whether they are negative or not. It gives a good accuracy for simple data sets. However, you need to pay attention to the number of observations and features. When you have more features than observation it can lead to an overfitting. Moreover, LR is not a powerful algorithm especially when dealing with complex data and relationships.

To fit the process, I used the 'glm' library available on R and kept only the most relevant variables thanks to the stepwise process. While doing my predictions and evaluation I got my confusion matrix with 771 errors out of 3229 good results, which resulted on an accuracy of 81%. The AUC however is equal to .72 which is acceptable.

Explaining the limit balance of non-payers:

It can be seen from the distribution above that the limit balance of the non-payers is much lower than that of the payers.





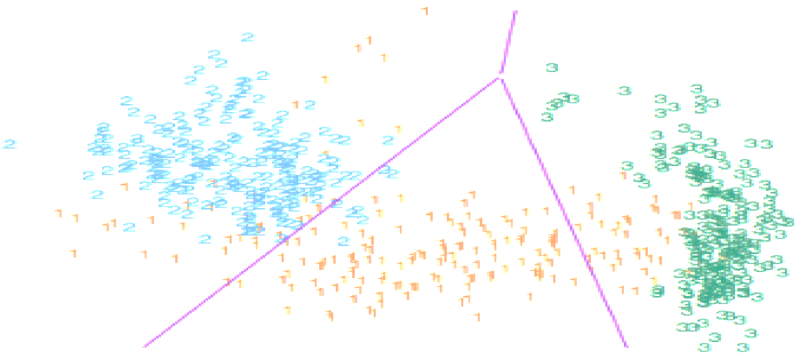
Relation between Limit Balance and default payment:

The relationship between the limit balance and the default payment is quite significant.

The average non-payer generally has a lower limit balance than the average payer.

4. LDA

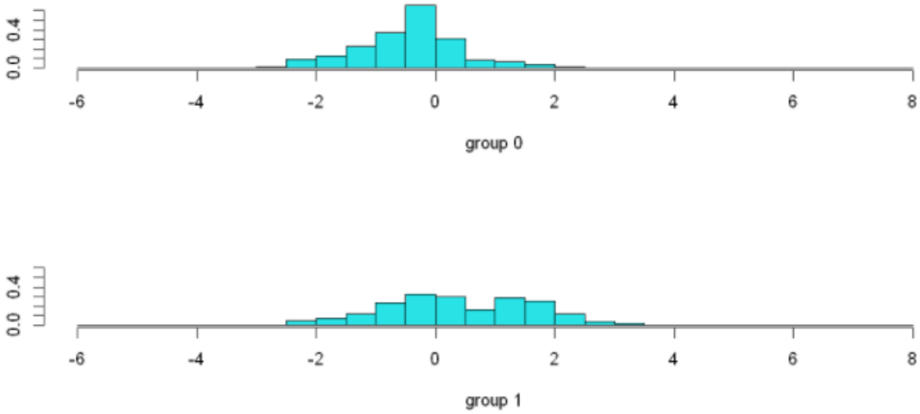
Linear Discriminant analysis is based on a dimensionality reduction technique for supervised learning. It can model the differences between groups by separating the data into classes. The algorithm can have some similarities with the logistic regression model. They share the same function, except that LDA consider X predictors separately in each class. It also uses a Baye's theorem to create the estimation. Their forms are also similar. Moreover, LDA assumes normally distributed data and a common covariance matrix (common to all classes in our data set).



LDA - An introduction to statistical Learning (2013)

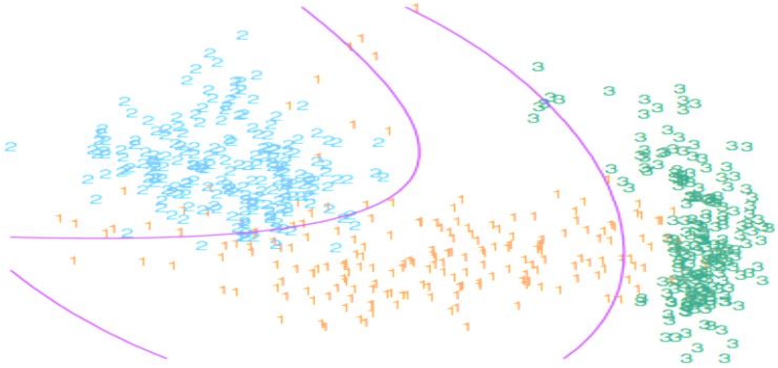
However, LDA can support well separated classes unlike the Logistic Regression. This algorithm cannot do any prediction for regression and may do some overlapping in case of small number of features. The power of LDA is also known as 'low' unlike Tree based methods. The advantage is that LDA tends to be more stable than LR and can also be used easily when dealing with multi class.

To fit my train data to the algorithm I used the 'lda' library available on R which gave me a great accuracy of 81%. The AUC equal to 0.71 is considered and acceptable. After k-fold cross validation I got almost the same accuracy equal to 0.811.



5. QDA

Quadratic Discriminant Analysis is an example of bayes classifier, just like LDA, QDA is also based on normal distribution for each class. However, the QDA considers the fact that each class has its own covariance matrix. It means that if the number of predictors is high, the number of parameters also need to highly increase (As we want to estimate a different covariance matrix for each group). Taking this into account it indicates that QDA take much more time to train unlike LDA. However, the classification will become more accurate and is more flexible as well!

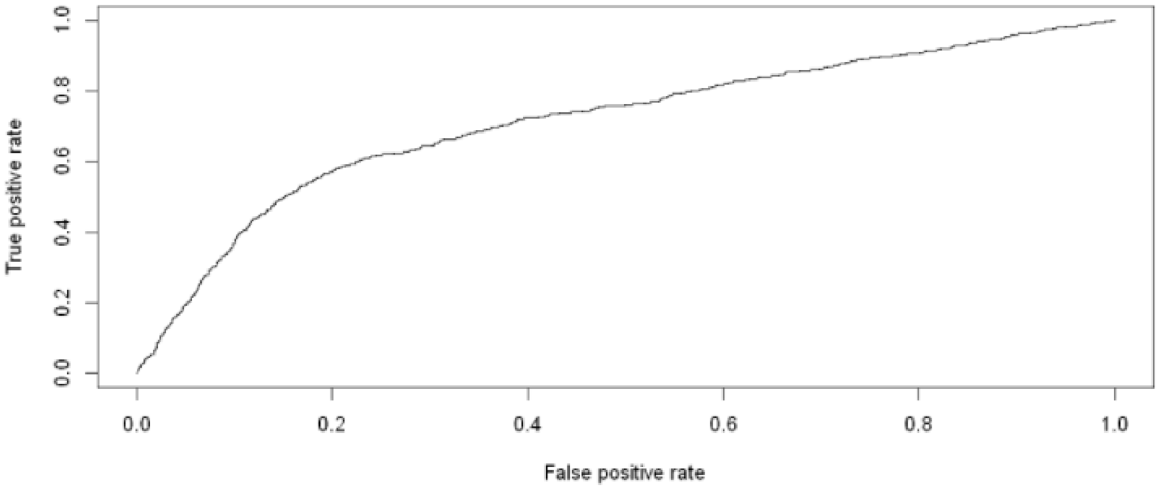


QDA - An introduction to statistical Learning (2013)

As LDA, QDA can be a good algorithm even with multi class unlike Logistic Regression. However, the size of the sample needs to be higher than the number of predictors, otherwise the performance will decline. Especially if the number of predictors is almost the same as the size of the sample. In conclusion, QDA should be prioritize over LDA if the train set is consequent. LDA is preferred when we have only a small train set.

To fit the process, I used the qda library in R, plus our train set. It results to a very low accuracy equal to 39% and an AUC of 0.71. In conclusion I can say that QDA is not the right algorithm for our data and target variable even after using cross validation.

QDA ROC Curve:



The Receiver Operating Characteristic Curve is another method to measure the performance et evaluate our algorithm. I use it for the QDA in order to see the difference between True and False positive rate. The ROC curve is not perfect, as the curve is far away from the ideal discriminator but still significant as our AUC is equal to 0.71.

CONCLUSION:

In conclusion I can say that the best post k fold algorithm was the Random Forest with also the highest AUC followed shortly by the LDA. Logistic Regression also performed well on average and came third. On the other hand, the Decision Tree and the QDA were not suitable for this type of variables and data.

I also tried the stepwise system for variable selection, but it turned out that it changed the final result very slightly or even lowered it. You can find an example of stepwise process with logistic regression.

SOURCES:

Minh Phan’s courses

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. Springer.

https://www.upgrad.com/blog/random-forest-vs-decision-tree/#:~:text=A%20decision%20tree%20combines%20some,forest%20model%20needs%20rigorous%20training

https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/

https://thatdatatho.com/linear-vs-quadratic-discriminant-analysis/

https://afit-r.github.io/discriminant_analysis