INTRODUCTION

In the following work we will carry out a predictive analysis of credit defaults. In other words , we will try to predict which clients would be more susceptible to defaulting on their credit, based on information such as employment status, their bank account balance and the annual salary they receive. We will mainly use two models. One of decision trees and another of Discriminant Analysis. We will use the models after having done an EDA Exploratory Data Analysis.

EDA

We will define the EDA as the initial process of any Data Science project, the Exploration process. Classical statistics focused almost exclusively on inference, which is defined as a set of processes, sometimes very complex, designed to draw conclusions about large populations based on small samples. (Bruce, et al., 2020). Initially this process, conceived by John W. Tukey , consisted of presenting boxplots, scatterplots and a summary of important statistics such as the mean, median, quantiles , etc. (Tukey, 1977). With the recent availability of high computational power the EDA has also evolved. The drivers for such advances were the rapid development of new technology and access to more variety and greater amounts of data. (Bruce, et al, 2020)

Decision Trees

Decision trees, also called classification trees or regression trees, are a popular and effective method of classification. They were initially developed by Leo Breiman and others in 1984. Decision trees and their descendants Random Forests and Boosted trees are the basis of models widely used in Data Science. We can think of decision trees as a group of if-then-else rules , which are very easy to implement. (Bruce, et al, 2020). In contrast to linear and logistic regression, decision trees have the ability to find hidden patterns in complex data interactions.

Discriminant Analysis

Discriminant analysis is the oldest statistical classifier; It was introduced by RA Fisher in 1936 in an article published in the *Annals of Eugenics.* journal . Although discriminant analysis involves many techniques, the most widely used is Linear Discriminant Analysis LDA. Today it is not used as much, due to the creation of more sophisticated techniques such as tree models and logistic regression. However, the LDA is very useful in certain cases and is linked to other more used methods such as Principal Components. To understand this method it is necessary to understand the covariance matrix.

THE DATASET

For our analysis we will work with the dataset Default_Fin , pulled from the Kaggle platform .

i. Dimensions:

- 10000 records.
- Index, mark of employee or unemployed, bank balance, client salary, mark of delinquency or no delinquency.

variables

- Index : identifier number
- Employed : Boolean flag, where:
  1 = employee
  0 = unemployed.
- Bank balance: Customer's bank balance.
- annual salary : Annual salary of the client.
- Defaulted : boolean flag, where:
  1 = in arrears
  0 = no default

It is important to note that the label of our dataset , also known as the dependent variable, is not balanced, that is, there are many more observations of cases without default than of cases with default. A process for dealing with this will be described in the Procedure and Results section.

METHODOLOGY

We will use all the statistical techniques included in the R language for the EDA processes and the Decision Tree and LDA modeling, Linear Discriminant Analysis. We will interpret the results after each line of code we use. We will run tests to determine which is the best model for our dataset for our conclusion.

PROCEDURE AND RESULTS

We import our dataset and do some preliminary analysis.

```
> datos_cred <- read.csv(file = 'Default_Fin.csv',header= TRUE)
> str(datos_cred)
'data.frame':   10000 obs. of  5 variables:
 $ Index         : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Employed      : int  1 0 1 1 1 0 1 0 1 1 ...
 $ Bank.Balance : num  8754 9806 12883 6351 9428 ...
 $ Annual.Salary: num  532340 145274 381206 428454 461562 ...
 $ Defaulted.    : int  0 0 0 0 0 0 0 0 0 0 ...
```

We have 10,000 observations of 5 variables. Our dependent variable is Defaulted , which in our case means 0 = you are not in arrears in the payment of your credits, and 1 = If you are in arrears.

```
> head(datos_cred, 10)
   Index Employed Bank.Balance Annual.Salary Defaulted.
1      1        1      8754.36     532339.56          0
2      2        0      9806.16     145273.56          0
3      3        1     12882.60     381205.68          0
4      4        1      6351.00     428453.88          0
5      5        1      9427.92     461562.00          0
6      6        0     11035.08      89898.72          0
7      7        1      9906.12     298862.76          0
8      8        0      9704.04     211205.40          0
9      9        1     13932.72     449622.36          0
10    10        1         0.00     351303.24          0
```

In the summary above we see the structure of our dataframe . We realize that the Index column will not help us.

Next we verify that the column we want to predict (Default) is balanced .

```
> prop.table(table(datos_cred$Defaulted))

     0      1
0.9667 0.0333
```

As we can see, there is a much greater number of 0s than 1s, therefore we conclude that our dataset is not balanced.

We will balance our data through undersampling , as we see below.

```
> library(ROSE)
> data_cred_bal <- ovun.sample(Defaulted. ~ ., data = datos_cred,
+                              method = "under", N = 666, seed = 1)$data
> table(data_cred_bal$Defaulted.)

  0   1
333 333
```

As we can see, the number of clients without arrears "0" is the same as the number of clients with arrears "1".

SELECTION OF THE BEST VARIABLES

Next; We make the selection of what would be the best variables for the prediction.

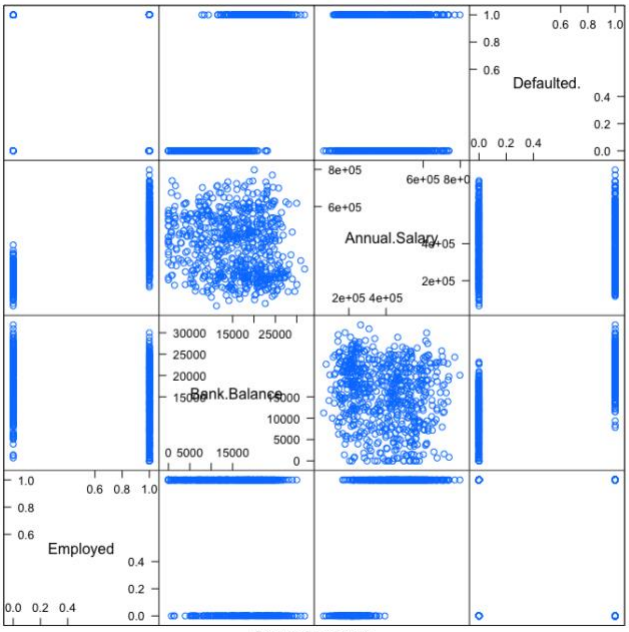
```
Best_Subset <-
  regsubsets(Defaulted.~.,
             data =data_cred_bal,
             nbest = 1,      # 1 best model for each number of predictors
             nvmax = NULL,    # NULL for no limit on number of variables
             force.in = NULL, force.out = NULL,
             method = "exhaustive")
summary_best_subset <- summary(Best_Subset)
as.data.frame(summary_best_subset$outmat)
         Employed Bank.Balance Annual.Salary
 ( 1 )                       *
 ( 1 )         *             *
 ( 1 )         *             *              *
```

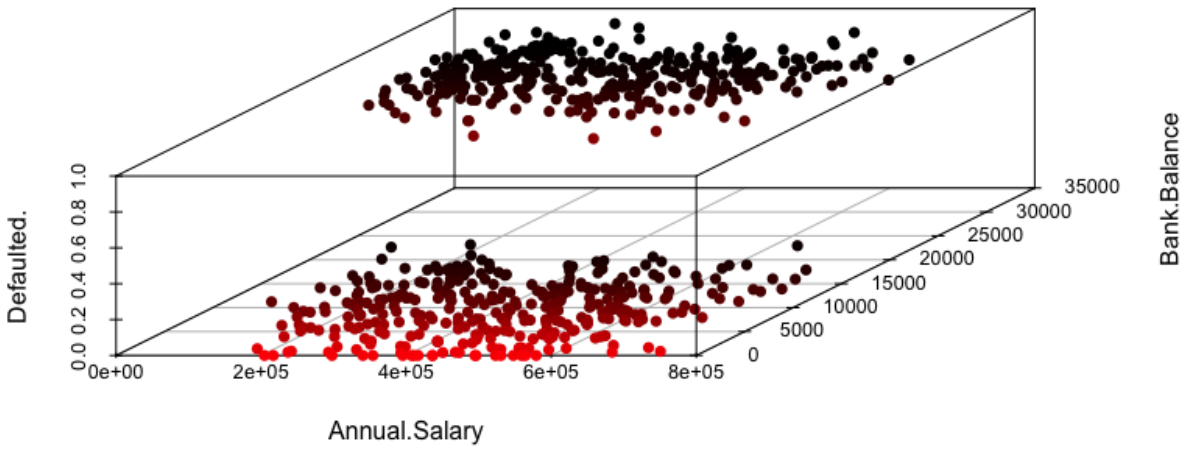
As we can see in this small ranking presented in the form of a DataFrame (above). The best predictor of credit delinquency would be the Account Balance (Bank.Balance). In other words, the amount of money that the client has deposited in his account. The second best predictor is employment status and the third is annual salary.

Below we see that all three variables are good attributes for predicting credit delinquency.

```
> which.max(summary_best_subset$adjr2)
[1] 2
> summary_best_subset$which[3,]
  (Intercept)      Employed  Bank.Balance Annual.Salary
         TRUE          TRUE          TRUE          TRUE
```

Scatter Plots





In this 3-dimensional graph we see that there is not much difference in the Account Balance of the people who earn more with the people who earn less. We can also observe that it has values close to 0 in its Account Balance, it is not in arrears. This is because surely people who do not have money in the bank cannot access credit.

summarization

```
> summary(fit_mora)

Call:
lm(formula = Defaulted. ~ Employed + Bank.Balance + Annual.Salary,
    data = data_cred_bal)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9207 -0.2248  0.0438  0.2395  0.8657

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.652e-01  4.600e-02  -7.939 8.72e-15 ***
Employed       1.185e-01  4.302e-02   2.753  0.00606 **
Bank.Balance   5.231e-05  1.784e-06  29.318  < 2e-16 ***
Annual.Salary -5.132e-08  1.226e-07  -0.419  0.67567
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3285 on 662 degrees of freedom
Multiple R-squared:  0.5708,    Adjusted R-squared:  0.5689
F-statistic: 293.5 on 3 and 662 DF,  p-value: < 2.2e-16
```

estimators _

$$Defaulted = 0.01185 Employed + 0.0000523 Bank.Balance - 5.132 * 10^{-8} Annual.Salary$$

The determination coefficient R^2 is 0.57, which means that 57% of the variation in default (Defaulted) is explained by the employment status, the Account Balance and the Annual Salary.

The p- values of the employment status and the account balance are less than 0.05, so we can say that they influence the default. The p- value of the annual salary is not less than 0.05, that is, it does not significantly influence the default.

Covariance Matrix

```
> cov(data_cred_bal)
                  Employed  Bank.Balance Annual.Salary    Defaulted.
Employed       2.220529e-01 -9.202918e+02  5.803949e+04 -2.481203e-02
Bank.Balance  -9.202918e+02  5.514588e+07 -1.800204e+08  2.784711e+03
Annual.Salary  5.803949e+04 -1.800204e+08  2.603994e+10 -3.877390e+03
Defaulted.    -2.481203e-02  2.784711e+03 -3.877390e+03  2.503759e-01
```

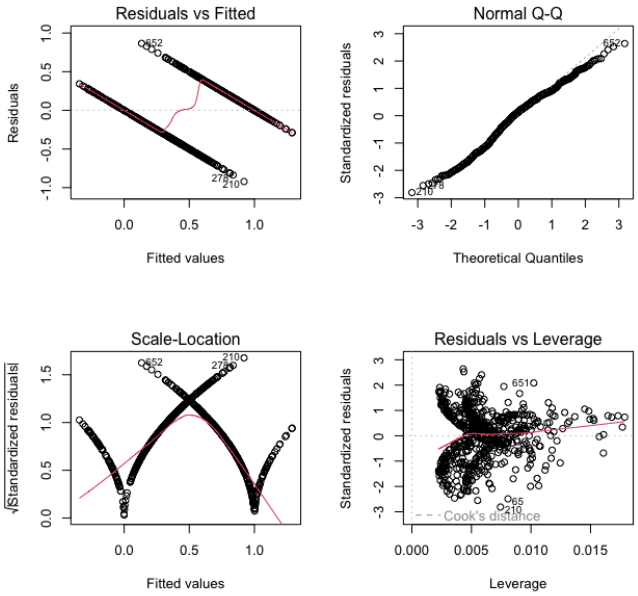
Correlations Matrix

```
> corr <- round(cor(data_cred_bal),2)
> corr
              Employed Bank.Balance Annual.Salary Defaulted.
Employed          1.00        -0.26          0.76      -0.11
Bank.Balance     -0.26         1.00         -0.15       0.75
Annual.Salary     0.76        -0.15          1.00      -0.05
Defaulted.       -0.11         0.75         -0.05       1.00
>
```

Color correlation matrix

Analyzing the correlation matrix we can see that there is a strong correlation between the Account Balance (Bank.Balance) and Default (Defaulted). Obviously there is also a strong correlation between Annual Salary (Annual.Salary) and employment status (Employed).

Residue Analysis



Normality test

Below we can see the different normality tests that we apply to our fit_mora fit .
We see that in the Anderson- Darling, Kolomogorov -Smirnov , Shapiro -France, Shapiro-Wilks and Pearson Chi- Squares tests that all p- values are less than 0.05. Therefore we REJECT the null Hypothesis that says that there is normality in the residuals .

```
> #Test de Normalidad
> # f) TEst de noramlidad
> library(nortest)
> res<-fit_mora$resid                    # saving the residuals from the regression as "res"
> ad.test(res)                          # Anderson-Darling test of normality

        Anderson-Darling normality test

data:  res
A = 3.6561, p-value = 3.934e-09

> lillie.test(res)                      # Kolomogorov-Smirnov test of normality

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  res
D = 0.060479, p-value = 4.734e-06

> sf.test(res)                          # Shapiro-Francia test of normality

        Shapiro-Francia normality test

data:  res
W = 0.9864, p-value = 1.868e-05

> shapiro.test(res)                     # Shapiro test of normalty

        Shapiro-Wilk normality test

data:  res
W = 0.98571, p-value = 4.241e-06

> pearson.test(res)

        Pearson chi-square normality test

data:  res
P = 59.919, p-value = 6.558e-05
```

Homoscedasticity Test

```
> bptest(fit_mora)

        studentized Breusch-Pagan test

data:  fit_mora
BP = 0.4598, df = 3, p-value = 0.9276
```

p- value is not less than alpha 0.05 therefore NO We reject the null hypothesis, that is, the errors have constant variance.

ANOVA

```
> ava1 <- aov(fit_mora)
> summary(ava1)
               Df Sum Sq Mean Sq F value   Pr(>F)
Employed        1   1.84    1.84  17.080 4.04e-05 ***
Bank.Balance    1  93.18   93.18 863.202  < 2e-16 ***
Annual.Salary   1   0.02    0.02   0.175    0.676
Residuals     662  71.46    0.11
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The p- values of the employment status and the Account Balance are less than alpha=0.05, so we REJECT the Null Hypothesis, that is, there are significant differences in the means of the groups. The Annual Salary has a p- value of 0.68 which is not less than alpha 0.05, which indicates that there is not much variance in arrears that can be explained by the Annual Salary.

Division of our data into training data and testing data.

```
> # Division de la data
> set.seed(1234)
> ind<-sample(2,nrow(data_cred_bal), replace=T,prob=c(0.7,0.3))
> trainData<-data_cred_bal[ind==1,]
> testData<-data_cred_bal[ind==2,]
> head(data3,10)
Error in head(data3, 10) : object 'data3' not found
> head(data_cred_bal,10)
   Employed Bank.Balance Annual.Salary Defaulted.
1         1      9946.32      552730.2          0
2         0     20925.60      210492.0          0
3         1      7414.32      602133.2          0
4         0     11731.08      211652.8          0
5         1      8485.32      224111.0          0
6         0     19160.40      258975.2          0
7         0     19964.28      238172.4          0
8         1         0.00      496318.7          0
9         0      8085.72      212067.5          0
10        1     10763.76      429069.0          0
>
```

K-means (K-means)

In the graph above we see the dispersions of each variable and their relationships. But this time, in the relationship between the Account Balance and the annual salary, we see two tables that are divided into two clusters . In one we see that the cluster is divided into those that do not have arrears (black) and those that have arrears (red). In the other table of this relationship we see that the clusters are divided into employee or non-employee.

Linear regression test

```
> summary(fit_reg)

Call:
lm(formula = myFormula, data = trainData)

Residuals:
     Min       1Q   Median       3Q      Max
-0.82566 -0.22083  0.03901  0.24085  0.85671

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.716e-01  5.400e-02  -6.881 1.96e-11 ***
Employed       1.021e-01  5.123e-02   1.994   0.0468 *
Bank.Balance   5.159e-05  2.078e-06  24.821  < 2e-16 ***
Annual.Salary  1.608e-08  1.447e-07   0.111   0.9116
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3258 on 458 degrees of freedom
Multiple R-squared:  0.5787,    Adjusted R-squared:  0.5759
F-statistic: 209.7 on 3 and 458 DF,  p-value: < 2.2e-16
```

```
> traindat_mse
[1] 0.10521
> test_mse <- mean((predict_reg - testData$Defaulted.)^2)
> test_mse
[1] 0.112276
```

There is not much difference between the MSE (above) of the training data and the test data, so we can conclude that the model will not overfit .

```
> actuals_preds <- data.frame(cbind(actuals=testData$Defaulted., predicteds=predict_reg))
> correlation_accuracy <- cor(actuals_preds)
> correlation_accuracy
             actuals predicteds
actuals    1.0000000  0.7411466
predicteds 0.7411466  1.0000000
```

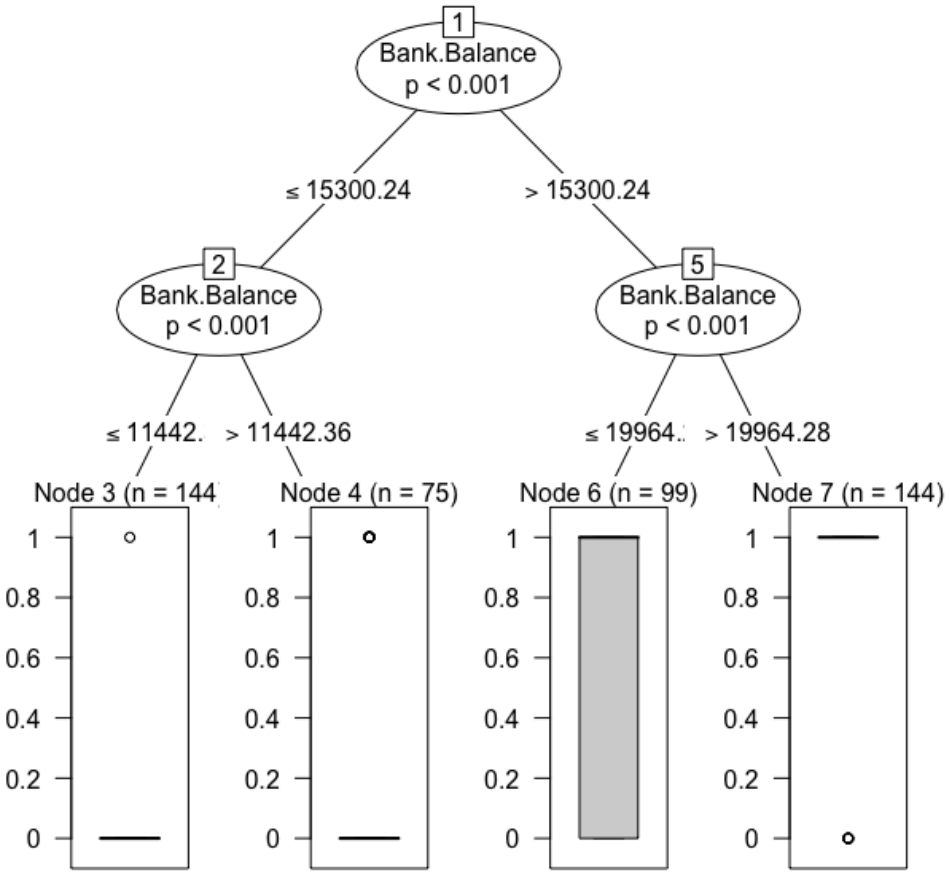
We see that there is a high correlation between the predicted values and the actual values in the regression, so we can say that the model would perform relatively well.

Model I) Decision Trees

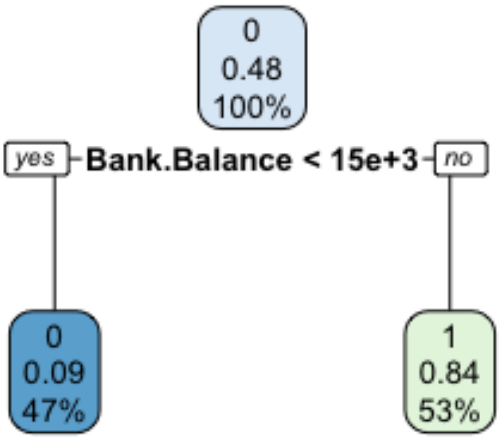
```
myFormula <- Defaulted.~ + Employed + Bank.Balance + Annual.Salary
heart_ctree<-ctree(myFormula,data=trainData)
table(predict(heart_ctree),trainData$Defaulted.)
```

```
                        0   1
 0.00694444444444444 143   1
 0.24                 57  18
 0.656565656565657    34  65
 0.965277777777778     5 139
```

As we see in the graph below, the most predictive variable of default is the account balance at the bank.



Below we see the graph of another Decision Tree model



Now we make the prediction with the test data

```
prediccion <-predict(fit, testData, type = 'class')
table_mat <- table(testData$Defaulted., prediccion)
table_mat
 prediccion
    0   1
0  75  19
1   9 101
```

The model predicted 75 cases of no default (“0”) correctly, but 19 false positives. The model predicted 101 delinquent cases correctly and 9 false negatives.

Precision Test (Accuracy)

```
> accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
> accuracy_Test
[1] 0.8627451
```

The prediction accuracy of our test data is 0.86 or 86%. Not bad for this model.

Model II) Discriminant Analysis

We fit the model and below we see our matrix with the test data

```
> fit_lda <- lda(Defaulted.~ Employed+Bank.Balance+Annual.Salary, data=trainData)
> fit_lda
Call:
lda(Defaulted. ~ Employed + Bank.Balance + Annual.Salary, data = trainData)

Prior probabilities of groups:
       0        1
0.517316 0.482684

Group means:
   Employed Bank.Balance Annual.Salary
0 0.7196653     9595.852      402594.5
1 0.6322870    20989.859      391291.0

Coefficients of linear discriminants:
                      LD1
Employed      4.130490e-01
Bank.Balance  2.086142e-04
Annual.Salary 6.501138e-08
```

```
> table_mat2 <- table(testData$Defaulted., predict_lda$class, dnn = c("Clase real",
 "Clase predicha"))
> table_mat2
          Clase predicha
Clase real  0  1
         0 78 16
         1 11 99
```

We see that with our test data the model correctly predicted 78 cases that are not in default and 99 cases that are in default. It incorrectly predicted 16 cases as delinquent but not really, and 11 cases not delinquent but actually.

Below We will do the accuracy test of this model

```
> accuracy_Test <- sum(diag(table_mat2)) / sum(table_mat2)
> accuracy_Test
[1] 0.8676471
```

Rounding to 0.87, we can say that this model has an accuracy of 87% (Accuracy) to predict cases of credit default or non-default. It is a more precise point than our previous model

CONCLUSION

Decision Tree and Linear Discriminant Analysis models have a fairly high precision, so we can say that: Both models can predict a customer's credit default, with 86% and 87% accuracy respectively. For this, it is necessary to previously know the employment status, the account balance and the client's annual salary. It should be noted that for this study the observations have been anonymized following good ethical practices in data science.