Statiscical Learning project:

- **Country Segmentation based on Pollution and Energy Production: An Unsupervised Analysis**
- **Predictive Modeling of Loan Approval: A Supervised Learning Approach Using Customer Data**

Fabio Casalingo 28865A

Abstract

In the first part we will analyse with unsupervised algorithms how 67 states are distributed and grouped together in the field of pollution and energy production. While in the second part we will use supervised models to try to predict the approval of a loan based on customer characteristics.

Index

1. Unsupervised Learning

The project is focused on doing an unsupervised and a supervised analysis, in this first part we will discuss only about unsupervised part.

Stating with a definition, unsupervised learning is a branch of machine/statistical learning in which the algorithms will learn only from unlabelled data without any human implication.

In particular, I will demonstrate the application of Principal component analysis (PCA), a linear dimensionality reduction technique.

The data are linearly transformed into a new coordinate system, in order to clearly identify the directions (major components) that capture the largest differences in the data.

In a real coordinate system, a collection of p points is represented by a series of unit vectors. The i-th vector represents the direction of the line that best fits the data if it is orthogonal to the first i - 1 vector. The best fit line is defined if it does not give the distance between the points and the line it reduces the squared vertical These directions form an orthonormal basis in which the individual dimensions of the data are linearly uncorrelated.

Next, let's look at K-means clustering, which is a vector quantization approach derived from signal processing that attempts to divide n observations into k clusters, with each observation assigned to the cluster with the nearest mean (cluster centres or cluster centroid), which serves as the cluster prototype. This divides the data space into Voronoi cells. K-means clustering decreases within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which are the more difficult Weber problem. The mean optimises squared errors, but only the geometric median minimises Euclidean distances. And finally, hierarchical clustering is another unsupervised machine learning technique that groups unlabelled data into clusters. This is also known as hierarchical cluster analysis or HCA.

This method produces hierarchical tree-like clusters, called dendrograms.

Sometimes K-means clustering, and hierarchical clustering can produce similar results, but they work differently.

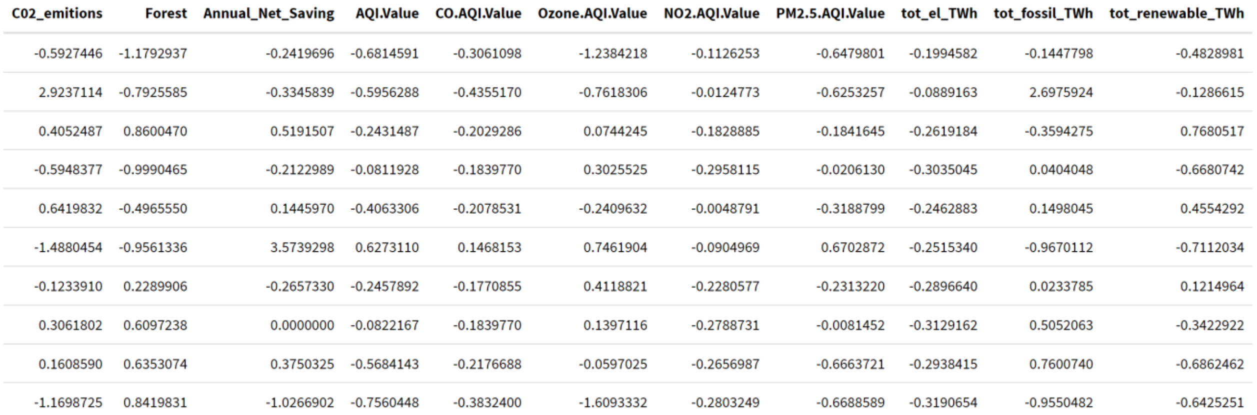
1.1 Dataset Exploratory Data Analysis (EDA)

Before starting let's have a look to the dataset of unsupervised part.

The dataset contains the data of 67 country and for them is stored data like emissions of CO2, electricity production from renewable sources and from non-renewable sources and some other variable that could help to actual understand better the possible similarities among countries.

The dataset contained NA values which were replaced with the average of the column values, and indices were created which, starting from the value of non-renewable or renewable energy production by individual factor, resulted in an overall index for renewable energy as our interest is to understand how this type of variable helps us to better understand the similarities between different countries.

The result is shown in Fig 1.1, in which we can see the numerical variables of interest in our dataset after they have been properly scaled where necessary and the missing values handled as mentioned.



C02_emitions	Forest	Annual_Net_Saving	AQI.Value	CO.AQI.Value	Ozone.AQI.Value	NO2.AQI.Value	PM2.5.AQI.Value	tot_el_TWh	tot_fossil_TWh	tot_renewable_TWh
-0.5927446	-1.1792937	-0.2419696	-0.6814591	-0.3061098	-1.2384218	-0.1126253	-0.6479801	-0.1994582	-0.1447798	-0.4828981
2.9237114	-0.7925585	-0.3345839	-0.5956288	-0.4355170	-0.7618306	-0.0124773	-0.6253257	-0.0889163	2.6975924	-0.1286615
0.4052487	0.8600470	0.5191507	-0.2431487	-0.2029286	0.0744245	-0.1828885	-0.1841645	-0.2619184	-0.3594275	0.7680517
-0.5948377	-0.9990465	-0.2122989	-0.0811928	-0.1839770	0.3025525	-0.2958115	-0.0206130	-0.3035045	0.0404048	-0.6680742
0.6419832	-0.4965550	0.1445970	-0.4063306	-0.2078531	-0.2409632	-0.0048791	-0.3188799	-0.2462883	0.1498045	0.4554292
-1.4880454	-0.9561336	3.5739298	0.6273110	0.1468153	0.7461904	-0.0904969	0.6702872	-0.2515340	-0.9670112	-0.7112034
-0.1233910	0.2289906	-0.2657330	-0.2457892	-0.1770855	0.4118821	-0.2280577	-0.2313220	-0.2896640	0.0233785	0.1214964
0.3061802	0.6097238	0.0000000	-0.0822167	-0.1839770	0.1397116	-0.2788731	-0.0081452	-0.3129162	0.5052063	-0.3422922
0.1608590	0.6353074	0.3750325	-0.5684143	-0.2176688	-0.0597025	-0.2656987	-0.6663721	-0.2938415	0.7600740	-0.6862462
-1.1698725	0.8419831	-1.0266902	-0.7560448	-0.3832400	-1.6093332	-0.2803249	-0.6688589	-0.3190654	-0.9550482	-0.6425251

Fig1.1 Snapshot of the first 10 rows from the final dataset created.

For example, we find CO2_emitions, which represents emissions in metric tons per capita; Forest, which represents the % of forests in the entire state; as for the other variables relating to consumption or particles in the air, the unit follows the one explained for CO2_emitions, while for Annual_Net_Saving the value is per 100,000 inhabitants.

The correlation matrix, shown in Fig 1.2, provides a detailed explanation of the relationship between environmental and socioeconomic factors. Several excellent patterns emerge from the examination. First, CO2 emissions demonstrate significant connections with air pollution indices inclusive of CO and NO2 AQI values, in comparison to the correlations between carbon emissions and air pollution stages, which suggest that major savings regions prioritise clean power resources or help in performance emissions. Reducing forest area cover has been shown to be relationships

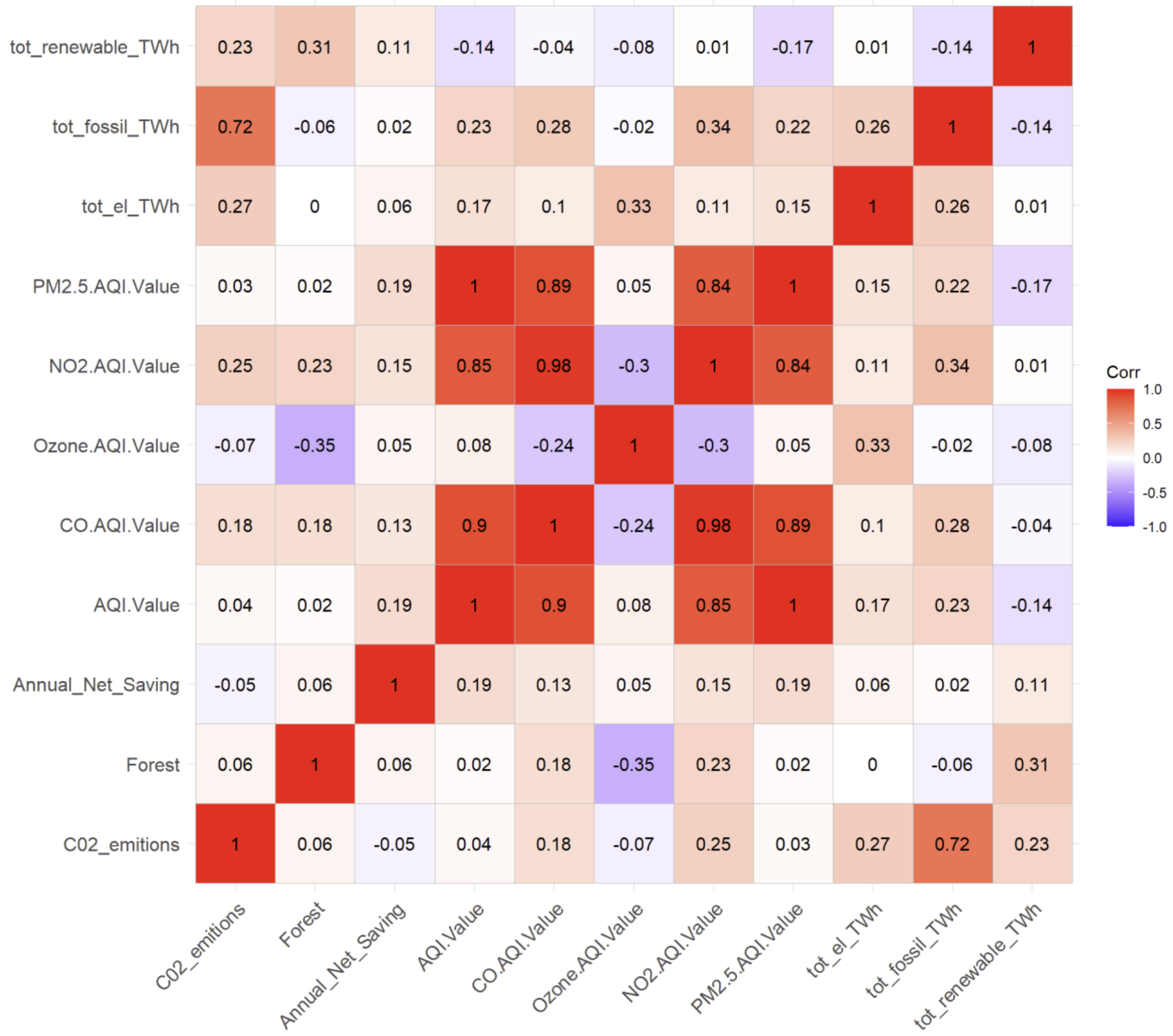


Fig 1.2 Correlation matrix plot

with air quality indices, suggesting that there is a substantial association between environmental health and pollutant levels. Furthermore, the high positive correlation between AQI values lends evidence to the association between air pollutants. These findings emphasise the multifaceted character of environmental development and the significance of comprehensive methods to tackling sustainability issues. Such insights are critical for policymakers and stakeholders looking to design effective measures to combat climate change and enhance air quality.

1.2 PCA Analysis

The first step for a good PCA analysis is to choose the right number of components, and for doing that we will use the scree plot which show us the eigenvalues of each component.

In PCA, a component represents another axis or direction in the data set space. These features are created from the original variables in a way that captures the maximum variation in the data. Each factor represents a linear combination of the original variables.

PCA transforms the original variables into additional uncorrelated variables, placing most of the variables in the data These principle components are ordered according to the number of variables they explain, the first explaining more variables in, the second explains the greater difference.

Thus, when we use the term component in PCA, we mean one of these axes or other directions that define the dimensions or structures of the data. These features are necessary to reduce the dimensionality of the data set and to preserve its core information. They help us analyse and visualize higher levels more effectively.

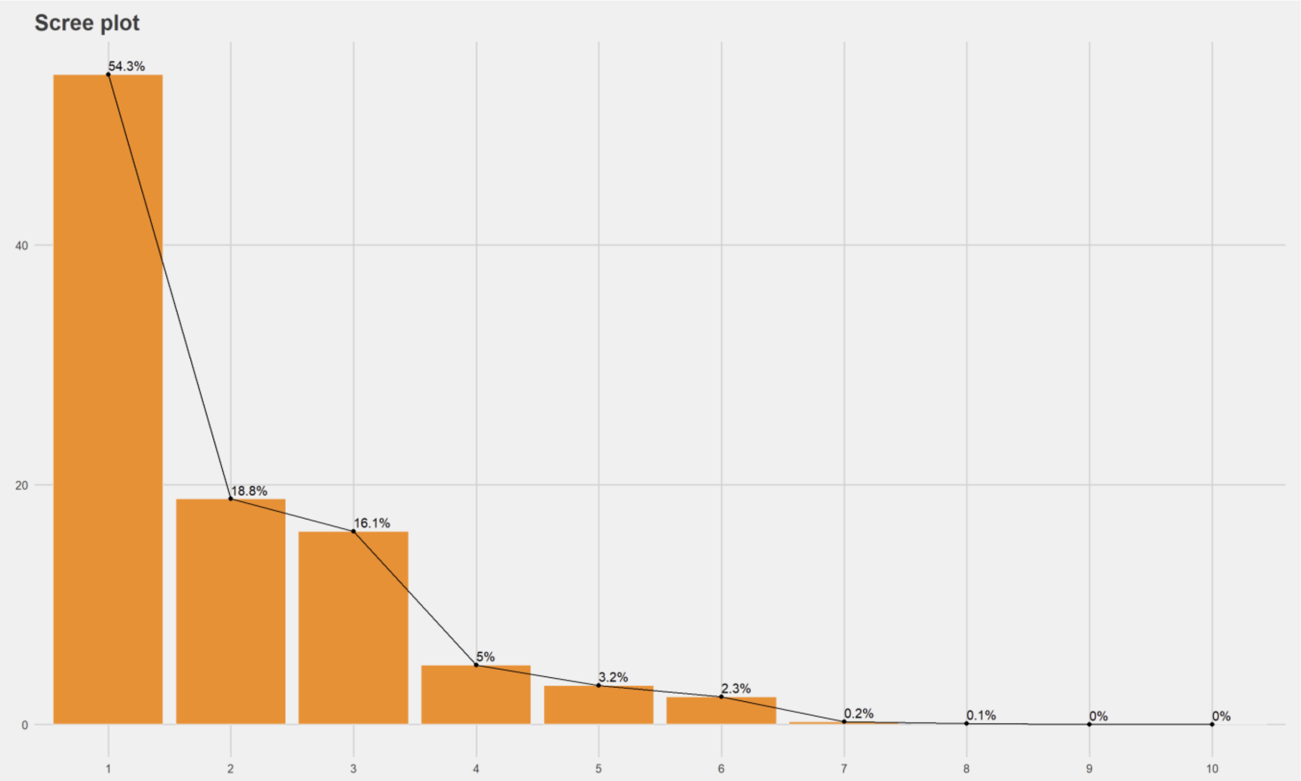
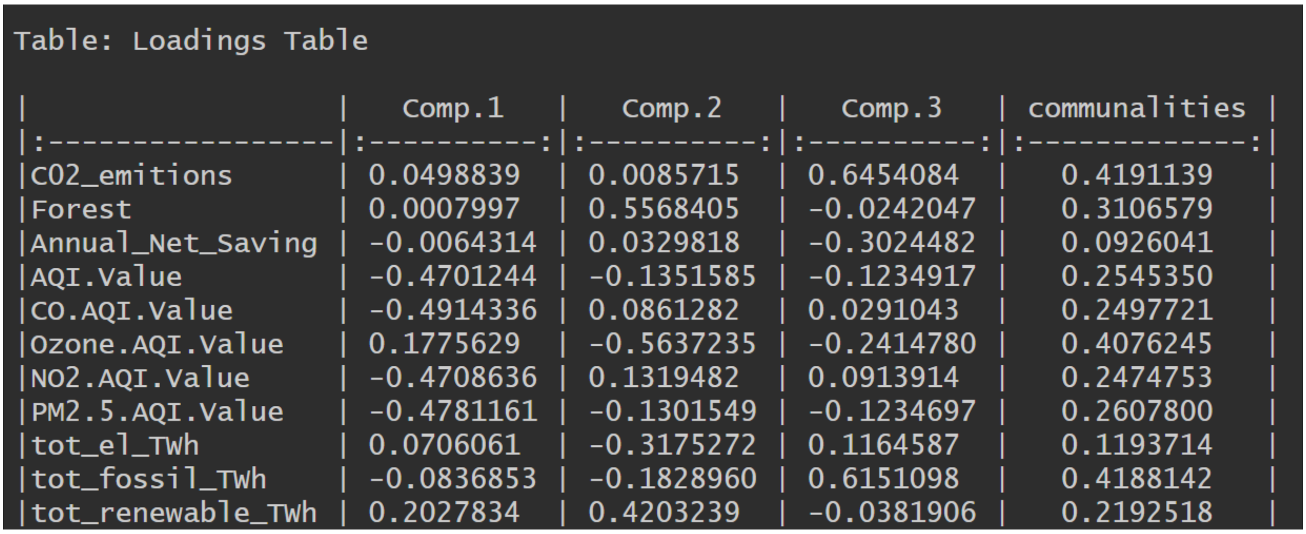


Fig 1.3 Scree Plot

Thanks to the scree plot, Fig 1.3, we can analyse the explained variance, which shows us that the sum of the first 3 components explains 89.2% of the total variance of our dataset, which is sufficient to conduct our analysis, so we choose the first 3 components.



```
Table: Loadings Table

|                  |   Comp.1   |   Comp.2   |   Comp.3   | communalities |
|:-----------------|:----------:|:----------:|:----------:|:-------------:|
|CO2_emitions      | 0.0498839  | 0.0085715  | 0.6454084  |   0.4191139   |
|Forest            | 0.0007997  | 0.5568405  | -0.0242047 |   0.3106579   |
|Annual_Net_Saving | -0.0064314 | 0.0329818  | -0.3024482 |   0.0926041   |
|AQI.Value         | -0.4701244 | -0.1351585 | -0.1234917 |   0.2545350   |
|CO.AQI.Value      | -0.4914336 | 0.0861282  | 0.0291043  |   0.2497721   |
|Ozone.AQI.Value   | 0.1775629  | -0.5637235 | -0.2414780 |   0.4076245   |
|NO2.AQI.Value     | -0.4708636 | 0.1319482  | 0.0913914  |   0.2474753   |
|PM2.5.AQI.Value   | -0.4781161 | -0.1301549 | -0.1234697 |   0.2607800   |
|tot_el_TWh        | 0.0706061  | -0.3175272 | 0.1164587  |   0.1193714   |
|tot_fossil_TWh    | -0.0836853 | -0.1828960 | 0.6151098  |   0.4188142   |
|tot_renewable_TWh | 0.2027834  | 0.4203239  | -0.0381906 |   0.2192518   |
```

Fig 1.4 Loadings Table

The first component has substantial loadings for air quality variables such as AQI values and pollutant concentrations, including CO.AQI.Value, NO2.AQI.Value, and PM2.5.AQI.Value. These factors are highly associated with one another and with Component 1. It appears that Component 1 mostly represents variables related to air pollution.

Forest and total_renewable_TWh are two variables with high loadings in Component 2. These factors have a pretty high positive connection with Component 2. This component appears to incorporate environmental sustainability issues, as both forest cover and renewable energy output have a favourable impact.

Component 3 is defined by loadings from variables such as CO2_emissions and total_fossil_TWh. These factors show strong relationships with Component 3. It shows that Component 3 might reflect variables related to carbon emissions and dependency on fossil fuels.

The communalities column displays the percentage of variance in each variable explained by the retrieved components. Notably, variables related to air quality and carbon emissions have quite high communalities, implying that the extracted components account for a considerable portion of their fluctuation. This demonstrates an excellent model-data fit for these variables.

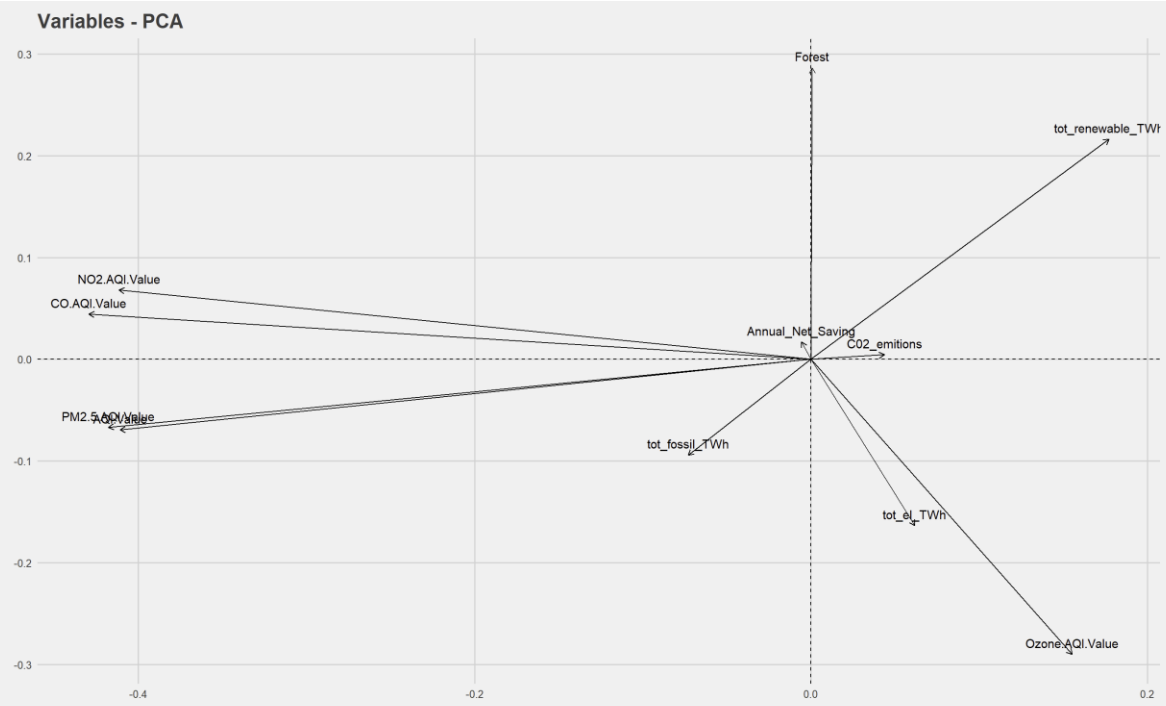


Fig 1.5 Representation of Loading

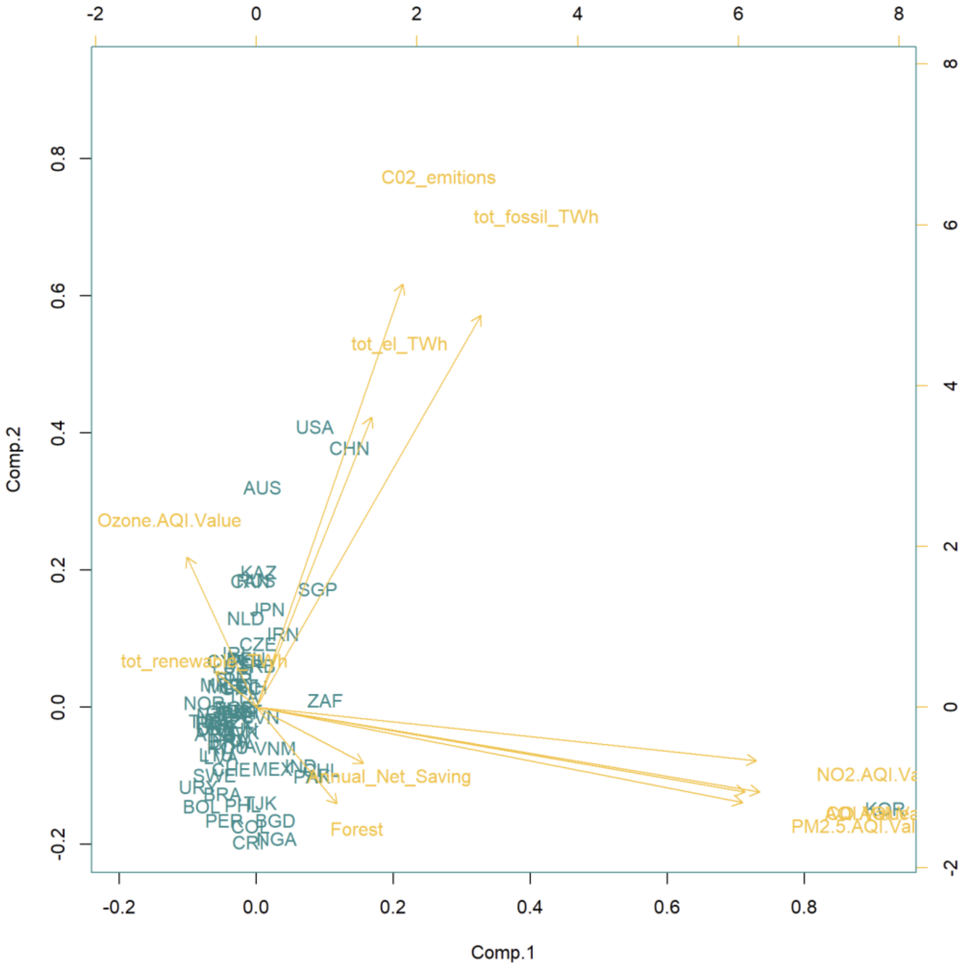


Fig 1.6 Country in PCA plot

Finally in Fig 1. 6 we see the distribution of the countries in the PCA space, and we can see there are states that stand out significantly and we have more that do not show much of a difference, clearly states like the USA and China are closely followed also by Australia, we see a very strong presence of polluting factors in Korea. We can although with difficulty observe that northern European countries tend to distribute themselves on the left side of the graph, representing the massive presence of renewable resources naturally present in the territory, exploited by these states.

Thanks to this analysis we have shown the differences between the various countries, the results obtained are consistent with what we might have expected, clearly, we do not see all the states those we have at our disposal are those that disseminate the most data, one reliable and two verified.

1.3 K-Means

K-means is an unsupervised machine and statistical learning technique that divides data into groups based on similarities. Its goal is to split n samples into k groups, with each observation assigned to the closest mean, known as the centroid.
Begin by randomly picking the initialization from the data points alone.
These focus areas act as initiation team centres.
Move each data point to the nearest centre. This is usually done by computing the Euclidean distance between each data point and each centre and assigning the data point to the groups with the nearest centres, take the average of all the data points assigned to each group and recalculate the focal points of the clusters.
Then just repeat the steps until consistency. Convergence occurs when the focal points do not change significantly or when a specified number of iterations is reached.

Letting X={x1,x2,...,xn}X={x1 ,x2 ,...,xn } be n data points in a dd-dimensional space, and let K be the number of clusters.
Select X to K random data points as initial midpoints: C={c1 ,c2 ,…,cK }.
For each data point xi, calculate its distance from each centroid cj using the Euclidean distance.
Assign each data point xi to a nearby central group, take the average of all the data points assigned to each group and recalculate the focal points, than repeat steps until the assembly occurs.

1.3.1 K-means clustering

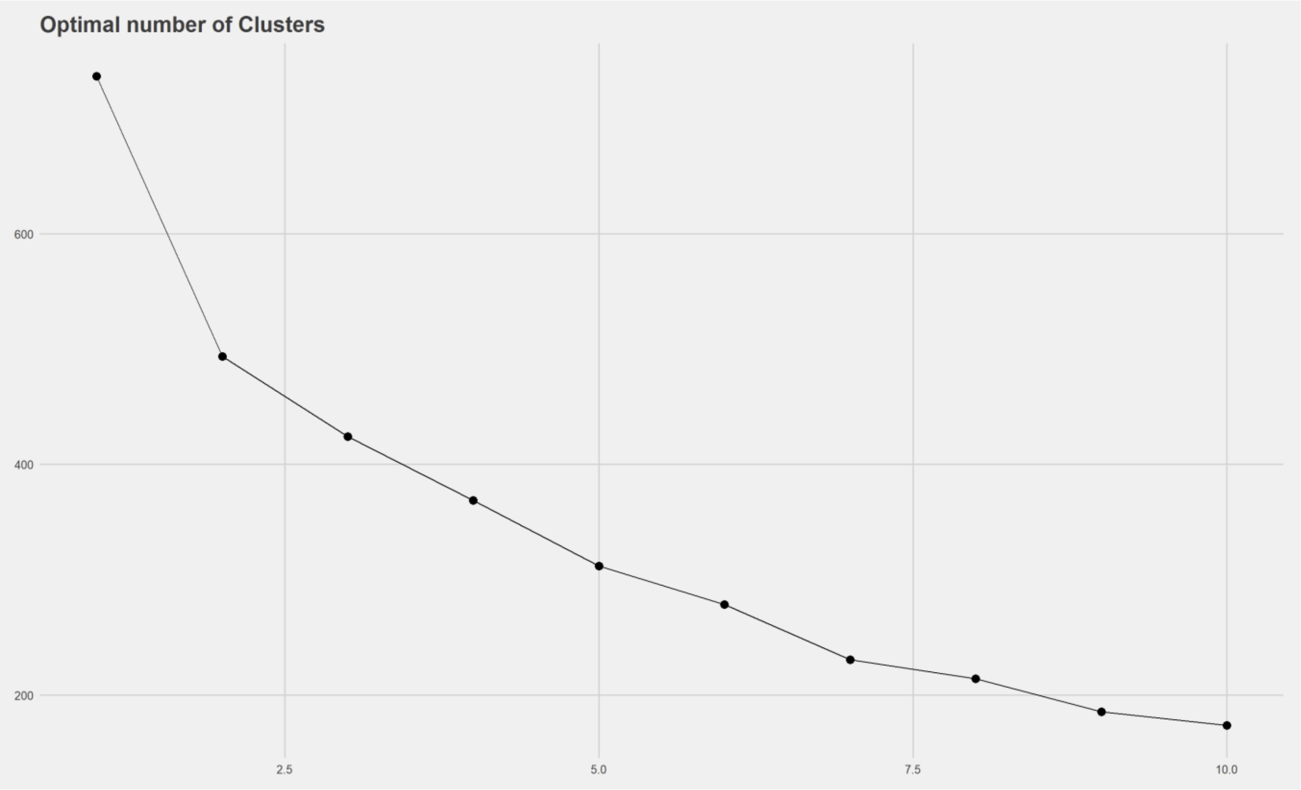


Fig 1.7 Optimal cluster

To start the clustering process, one must first define the optimal number of clusters to be selected and used; the graph in Fig 1.7 represents the Within Deviance. One way to find the correct number of clusters is the so-called elbow method, which consists of finding the point at which the 'curve' of the graph slows down. In this case, it is not easy to determine the correct number of clusters, but we will choose 7 which will allow us to frame the clusters in the best possible way.

After running the algorithm, the Fig 1.8 show the results, and since from this type of graph could be hard to correctly read the output the Fig 1.9 gives a clear visualization of the clusters.

Looking at the results obtained in these clusters, and you have results already analysed in the PCA we can begin to get a completer and more accurate picture.

We see Korea carving out its own space exactly as in the PCA by having many components in the air that put it in a situation that makes it unlike any other state.

We can confirm the similarity seen before in continental and southern Europe, while we can see how the northern European nations have clustered with Canada, in fact we have already mentioned the importance of the natural resources that distinguish these countries.

We can also observe a difference in the treatment of the U.S. and China that although in the PCA they were very close we can see that Socio economic differences led China to create a cluster with the other Asian regions such as India, Pakistan etc., while U.S. was divided with Russia and Australia.

Finally, we note how Latin America consistent with what we might expect has been assigned to a cluster that includes other nations as one of the few on that we have data from africa, however it is certainly stronger and more significant the similarity between the countries of central and south America.

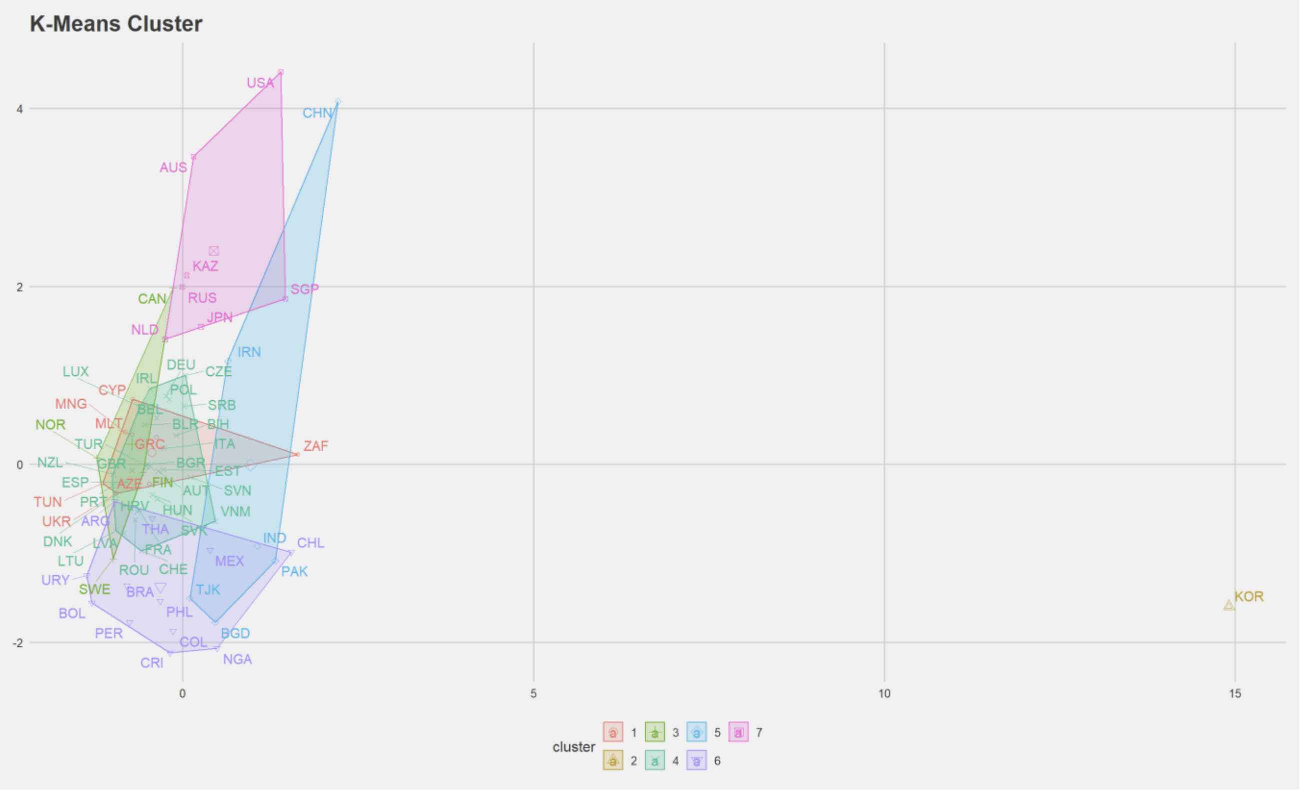


Fig 1.8 K-Means Cluster

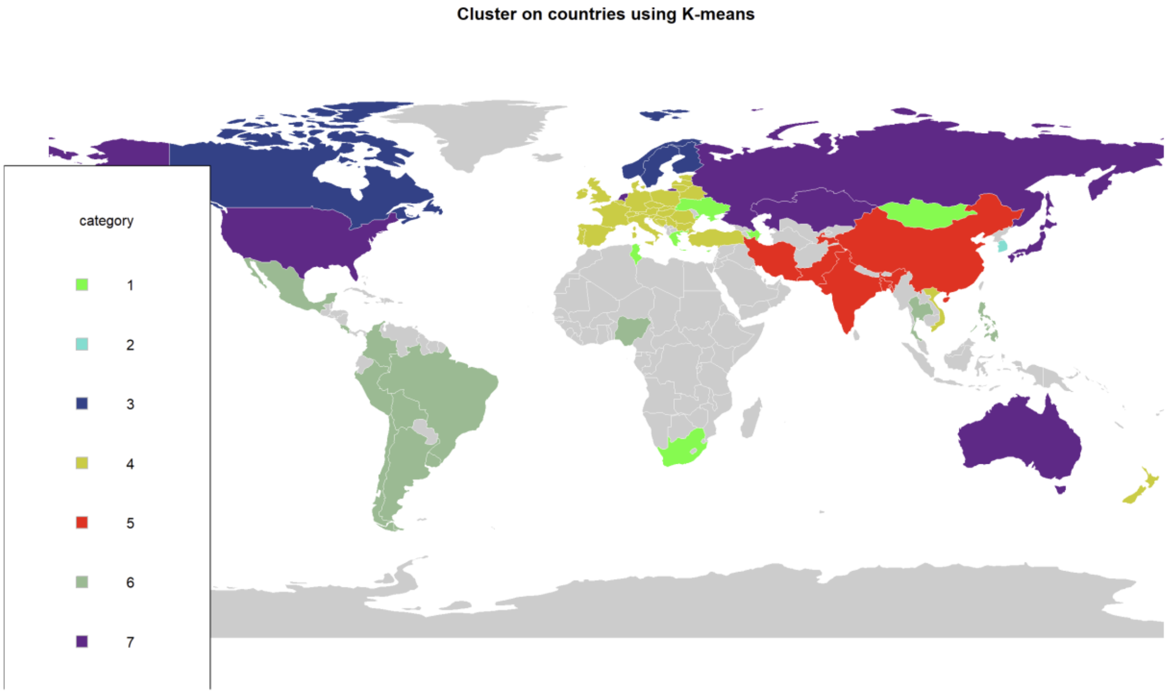


Fig 1.9 K-Means Cluster in 2d map

1.4 Hieratical Clustering

Hierarchical clustering is a method of grouping comparable data points according to their distance from each other. It provides a hierarchical structure of the groups, each containing subgroups or individual data points. There are two types of classifiers: convergent and divergent.

Aggregate clustering begins by treating each data point as a single group. The two adjacent clusters are then merged repeatedly until only one cluster remains. This process results in the creation of a dendrogram, a tree-like structure that shows the hierarchy of the clusters.

Partial clustering begins with all of the data points in the cluster, then separates them into smaller groups until each group has just one data point.

1.4.1 Single Linkage

One technique for making connections is to put jigsaw pieces together. Suppose you have several jigsaw pieces scattered around your desk, and you start collecting adjacent pieces until you get larger groups. It's an easy way to collect similar items.

But if you have parts that are far apart and still connected through a connection, they will end up in the same group, which can get confusing. However, it is a convenient tool for identifying patterns in data without having to set up several clusters in advance. That the facts make his story not tell pieces.

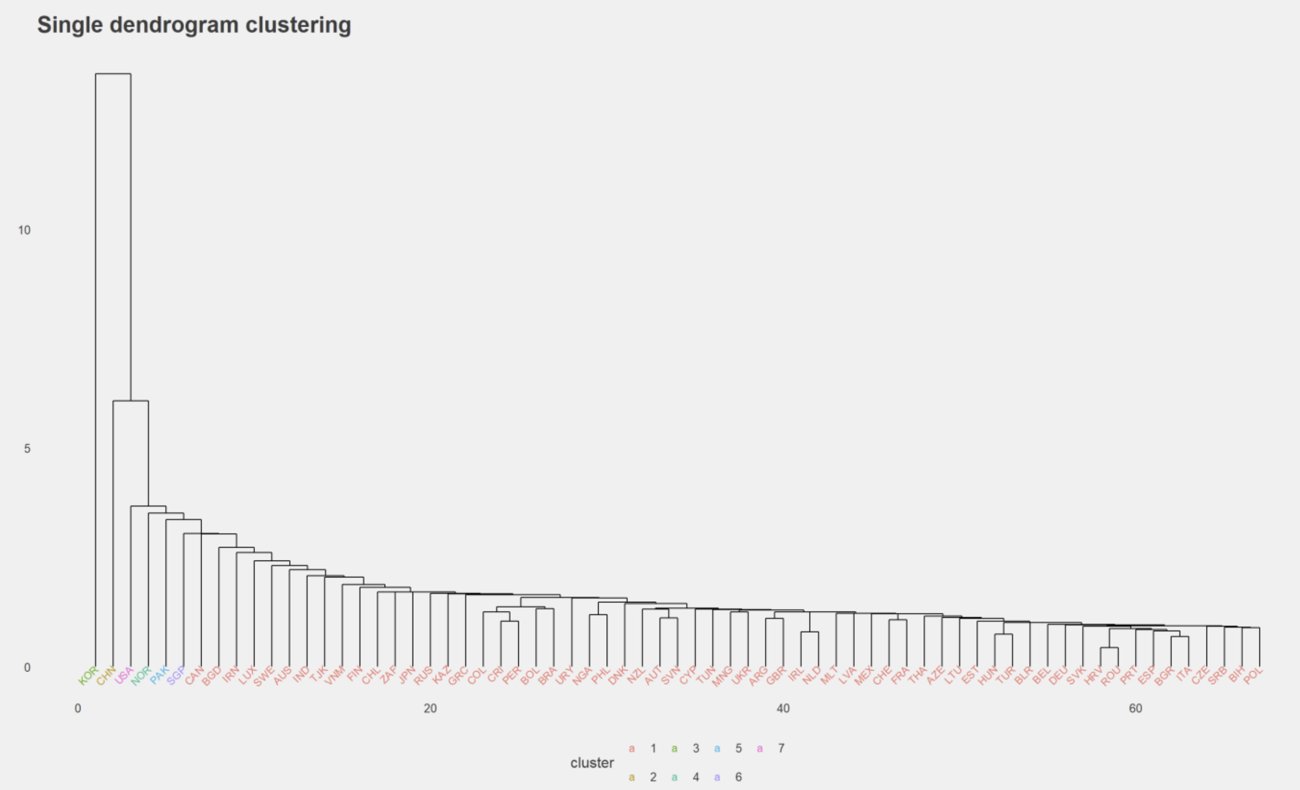


Fig 2.0 Single Linkage Dendrogram

As we can see from fig 2.0 the single linkage does not show an optimal result, we notice as always that Korea which has a separate branch that connects only with the branch coming from all the others, China and the United States have a behaviour that follows the expected one, but in many other cases we notice important divergences from the methods tested so far.

Let us continue with the Complete Linkage method to see if it offers better results.

1.4.2 Complete Linkage

Complete linkage clustering, also known as correlation cluster size, is the best way to enhance the distinction between objects in clusters. This method generates cluster formation based on the two most distant locations and causes tightly coupled clusters with little differentiation occurring. Unlike single link clusters, which connect clusters primarily based on the shortest distance among any factors, whole hyperlink clusters rarely cause chain consequences. When distant gadgets enjoy cluster formation, they are more likely to provide smaller, spherical clusters which are much less at risk of distant factors.

However, full linkage clustering may result in irregularly shaped clusters or clusters with different densities. Because it considers the maximum distance between any two points, it can prioritise large, simple clusters over small, complex ones, and can ignore finer details in the data.

Technically, the main difference between full and single networks lies in their distance scales. Single linkage clustering uses the minimum distance between any two points, whereas full linkage clustering uses the maximum distance. These basic differences drive the cluster structure, resulting in distinctive group structures and characteristics.

Because it considers the maximum distance between any two points, it can and will prioritize large, simple clusters over small, complex ones, and can and will ignore finer details in the data.

Technically, the main difference between full and single networks lies in their distance scales. These differences drive the cluster structure, resulting in distinctive group structures and characteristics.

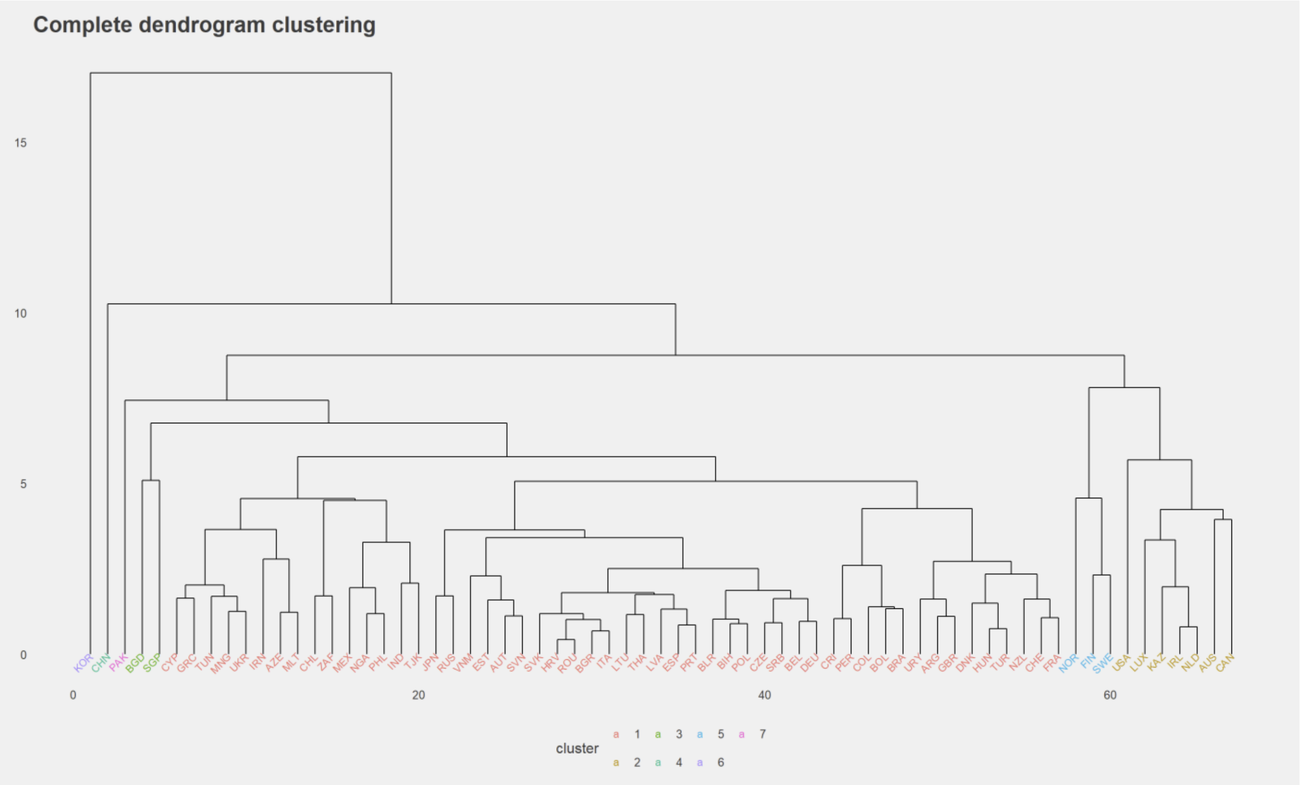


Fig 2.1 Complete Linkage Dendrogram

After applying the complete linkage algorithm we can see, Fig 2.1, the result, which returns a picture more similar and closer to the expected one, we can see that compared to the Single Link now the Northern European regions are framed as similar falling in the same cluster and with similar levels.

1.4.3 Average Linkage

Average linkage clustering calculates the distance between all pairs of points in different clusters. This method aims to balance the effect of excessive distance and the tendency to form clusters with more homogeneous shape.

Compared to a full aggregate correlation that considers the maximum distance between points, the averaging of clusters of correlations provides a more balanced approach. Differences among the networks lessen the impact of outliers and heterogeneously formed clusters, resulting in a greater uniform distribution of clusters.

However, average linkage clustering has several limitations. He may additionally battle with agencies of various sizes or sorts due to the fact he desires to breed companies of the equal size.

Furthermore, it could fail to capture well-described training if the facts comprise complicated or overlapping styles.

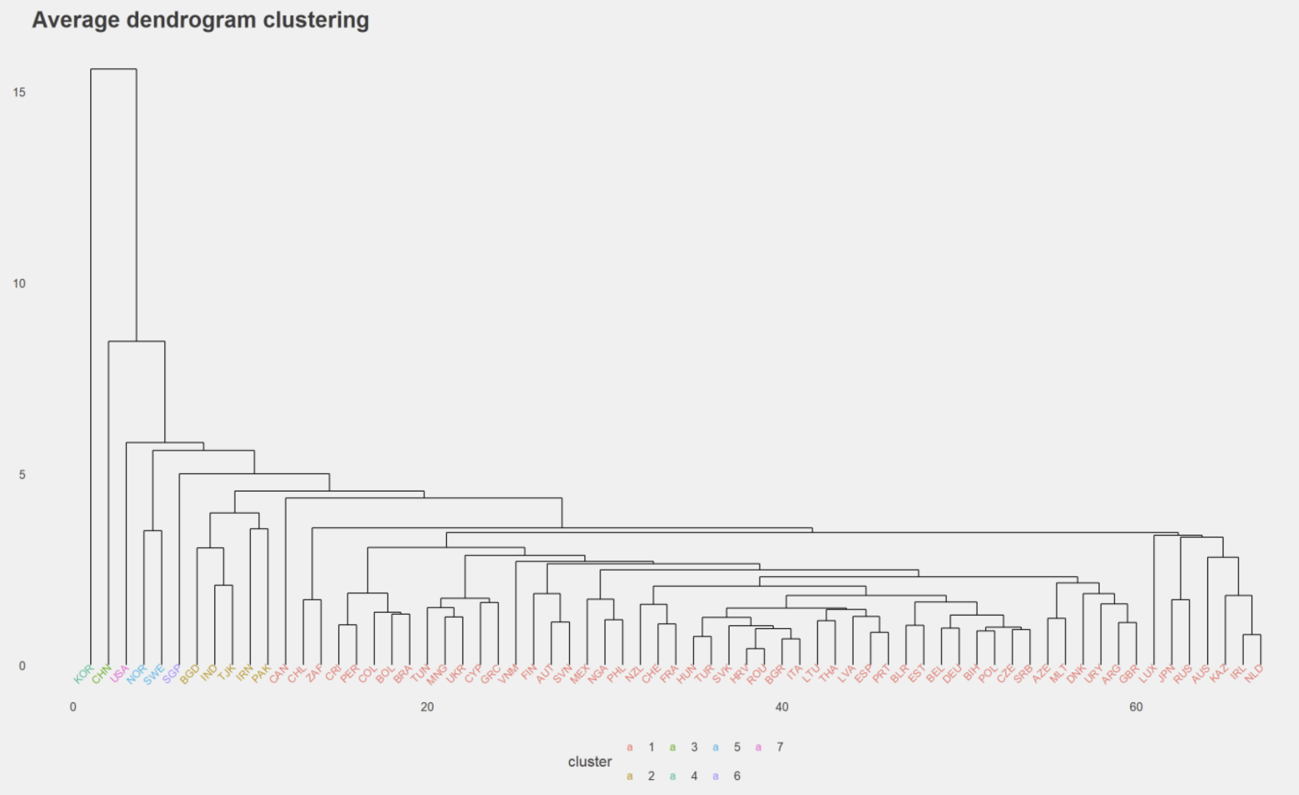


Fig 2.2 Average Linkage Dendrogram

The result of the Average Linkage method as we might have expected is somewhere in between the result of the Complete and Single Linkage method, we see that the links of Sweden and Norway have been identified, but Finlandia which was previously captured is missing and a cluster was created with Australia, Canada, the United States, and others which was not represented in this case.

1.4.4 Ward Linkage

Ward-linkage clustering, additionally known as the minimal variance technique, reduces the version in each cluster as its miles joined. The purpose of this approach is to offer extra correct, constant clustering while reducing the good-sized version carried with the aid of clustering.

In contrast to Average and Complete, which normally focus on distance metrics, Ward correlation groups optimise group differentiation at the same time. Ward correlations, which prioritise variation reduction, often create groups with the same size, making them useful for well-defined dataset cluster arrangements.

The major difference between Ward linkage and different strategies lies of their optimization targets. While Ward linkage minimizes the growth in variance, average linkage calculates the average distance, and complete linkage considers the most distance. This distinction results in varying cluster systems and characteristics.

However, Ward linkage clustering is computationally extensive and can war with big datasets because of its complexity. Additionally, it could no longer carry out nicely with datasets containing outliers or non-linear cluster structures.

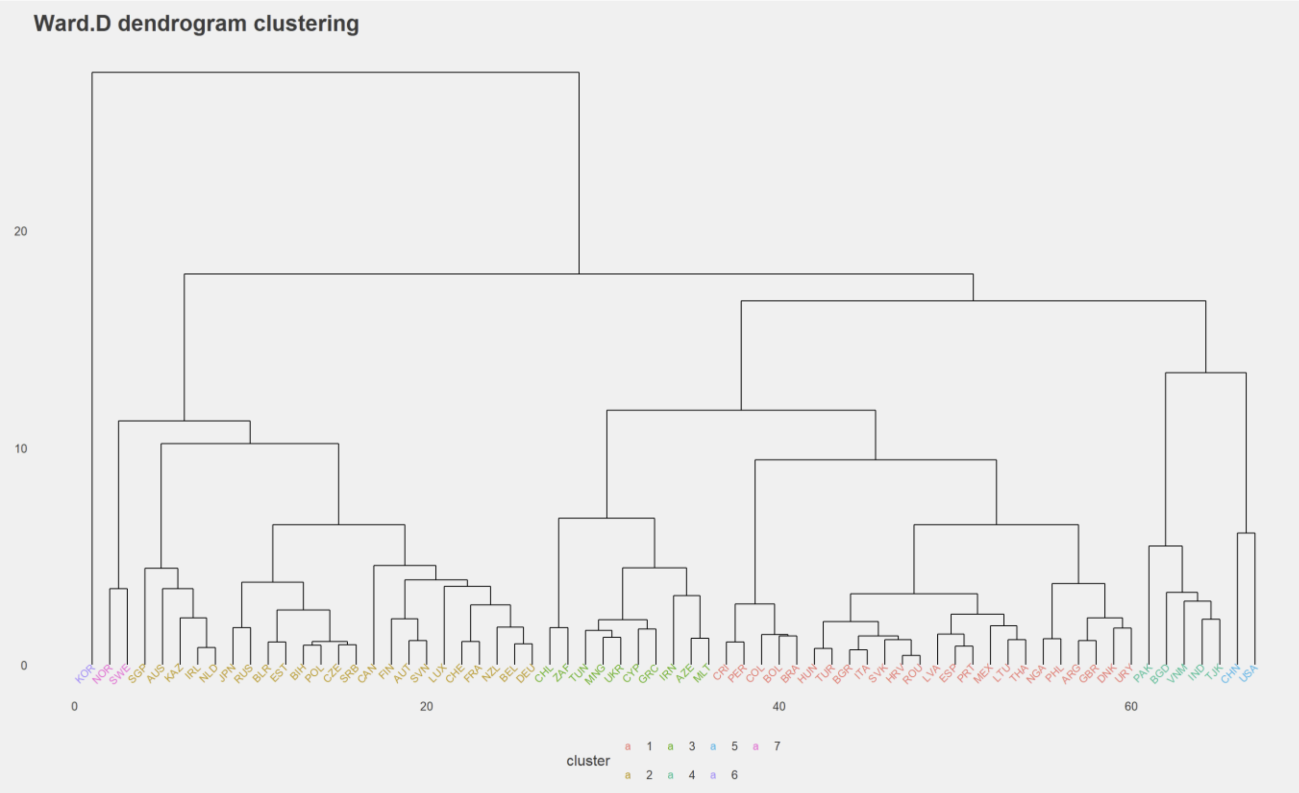


Fig 2.3 Ward Linkage Dendrogram

As can be seen in Fig 2.3, thanks to the Ward Linkage method we have obtained clearer clusters consistent both with what might be expected and with previous PCA and K-Means analyses. However, we find differences with the K-Means i.e., the United States forms a subgroup with China as suggested in the PCA analysis. Another difference is Finland and Canada that are not in the same cluster of Norway and Sweden, and even continental Europe and south are in different cluster unlike K-Means.

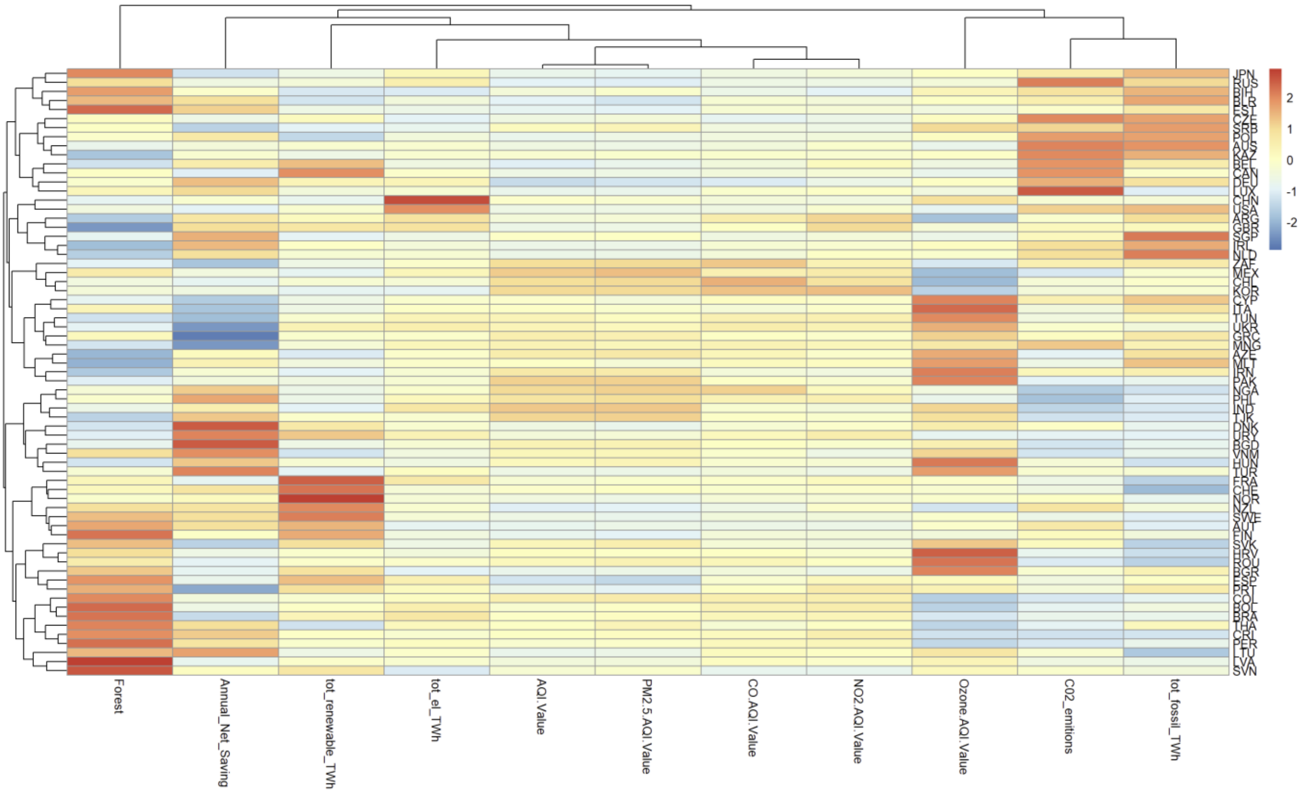


Fig 2.4 Correlation Heatmap

To conclude the unsupervised analysis, in regard to the data contained in the dataset we can state that the K-Means algorithm offers the best results in terms of clustering, followed then by the Ward method, while the others offer results that are not very satisfactory, while the PCA algorithm for size reduction generated a positive if difficult to read result due to Korea being very distant makes the other states appear very concentrated.

2.0 Supervised Learning

Supervised learning is an efficient branch of programme learning in which an algorithm learns to map inputs to verified outputs while generating predictions or special selections based on its known time. In this architecture, the algorithm is given labelled data, which indicates that each input is associated with an acceptable output. This allows the algorithm to analyse mappings between inputs and outputs, in order to predict unseen events.

Regression is an important technique for obtaining supervised knowledge aimed at predicting statistically persistent values. This is particularly useful when solving problems where the output variable is a real or constant price, such as forecasting household costs, inventory prices, temperature forecasting. In regression the rule set requires a property that maps the potential to a non-discontinuous location. Popular regression algorithms include linear regression, polynomial regression, and guide vector regression, among others.

Decision trees are a dynamic and adaptable model commonly used in classification and regression applications. They function by partitioning the feature space into regions, with each zone representing a distinct prediction. At each level, the algorithm selects the component that best splits the data, using a criterion like information gain or Gini inequality. Decision trees are straightforward to analyse and grasp, making them useful for understanding a model's decision-making process. Random forest is a cluster learning technique based on the idea of decision trees.

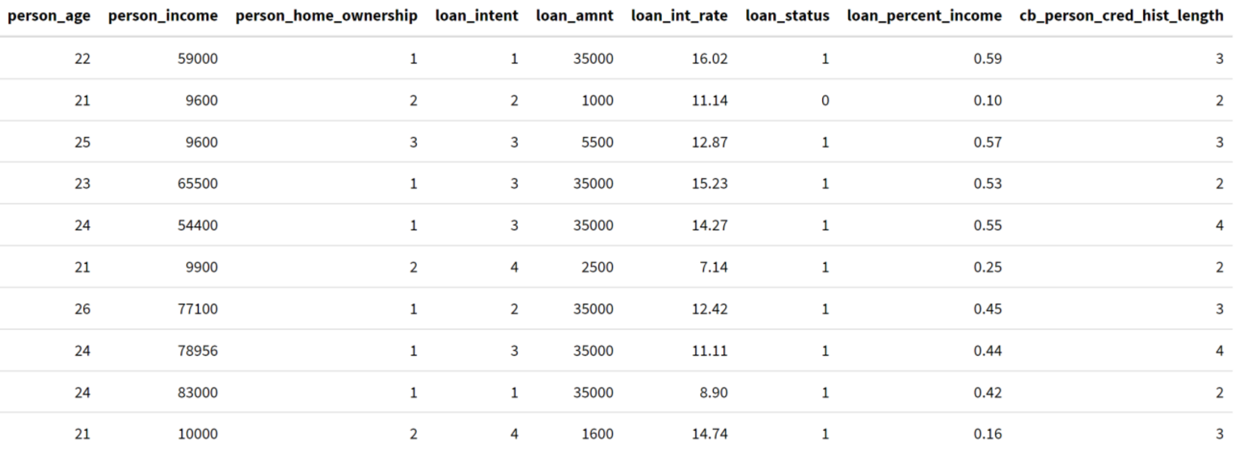
During training, they build a large number of decision trees based on the mode of classes (classification) or average prediction (regression) of individual trees. Random forests are well-known for their capacity to cope with high-dimensional data, as each tree is trained on a random piece of training data and a random selection of affected characteristics.

To summarise, supervised learning offers excellent tools for a variety of predictive modelling problems. These approaches, whether used to estimate stock prices, diagnose illnesses, or forecast demand, give significant insights and precise projections, resulting in industry-wide advances.

2.1 The Dataset

For this new analysis since the previous dataset was not congenial with this type of analysis, I chose to use a different one.

Again, had to handle missing values, and I chose to average the column in which these missing values were found.



person_age	person_income	person_home_ownership	loan_intent	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb_person_cred_hist_length
22	59000	1	1	35000	16.02	1	0.59	3
21	9600	2	2	1000	11.14	0	0.10	2
25	9600	3	3	5500	12.87	1	0.57	3
23	65500	1	3	35000	15.23	1	0.53	2
24	54400	1	3	35000	14.27	1	0.55	4
21	9900	2	4	2500	7.14	1	0.25	2
26	77100	1	2	35000	12.42	1	0.45	3
24	78956	1	3	35000	11.11	1	0.44	4
24	83000	1	1	35000	8.90	1	0.42	2
21	10000	2	4	1600	14.74	1	0.16	3

Fig 2.5 Head of Dataset

In Fig 2.5 we can see the first 10 rows of our dataset, the column “person_home_ownership” represents a column of categorical values that was converted to numeric, it included values such as Rent or Own.

The same treatment came for “loan_intent” which contains the reason for here it was requested for example personal, medical, or school reasons etc.

The variable we want to try to predict with the supervised methods, specifically we will focus on logistic regression, decision trees and random forest, is the variable "loan_status" which represent whether the loan was confirmed or not, it can take the values "0" or "1", where "0" represents a rejected loan and "1" confirmed.

The dataset in its entirety contains 32581 rows, but after removing the outliers the dataset has 29736 rows.

In Fig 2.6 we can observe the boxplot of the dataset with the outliers, presented clear problems represented by the outliers, and in Fig 2.7 we can see instead how the overview improved after the removal of the outliers.

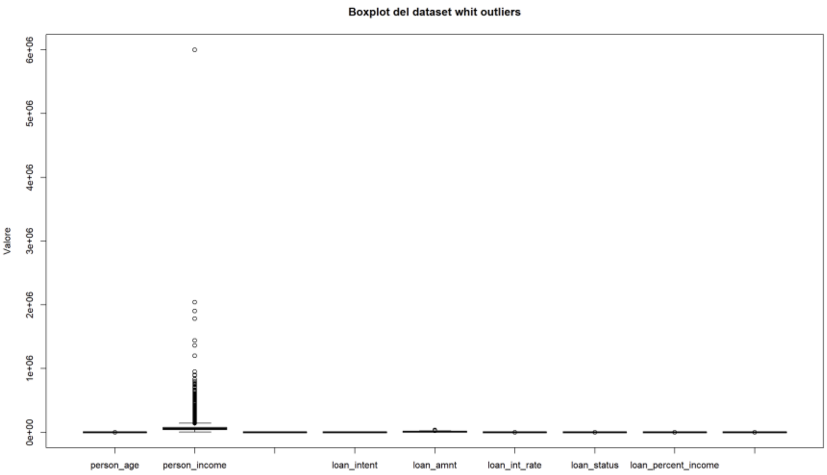


Fig 2.6 Boxplot with outliers

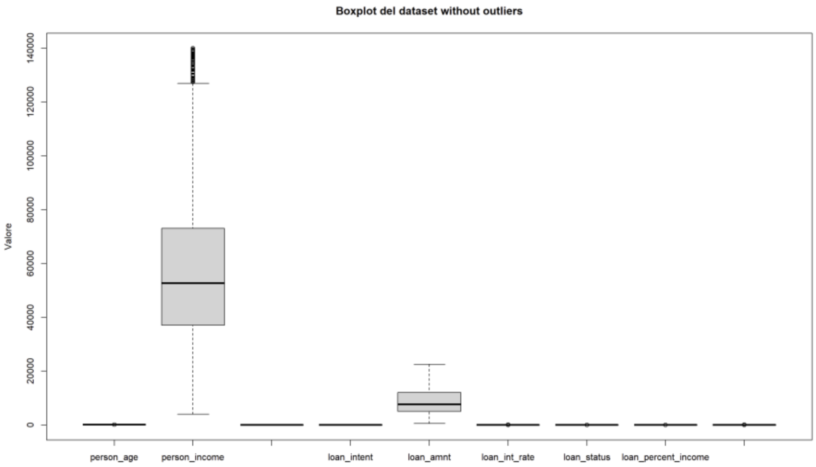


Fig 2.7 Boxplot without outliers

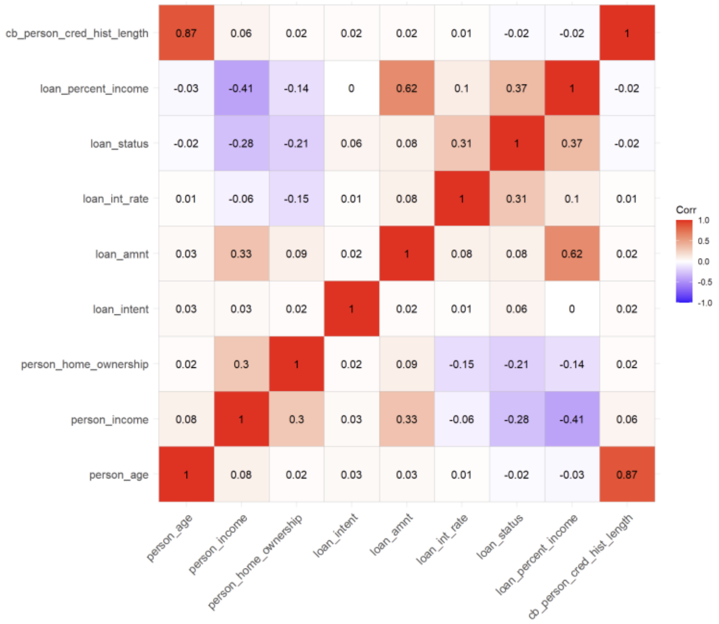


Fig 2.8 Correlation Matrix

2.2 Logistic Regression

Logistic regression is a fundamental statistical approach that is frequently used in machine learning, particularly for binary classification issues. Unlike linear regression, which predicts a continuous numerical value, logistic regression is intended to estimate the likelihood of a categorical result.

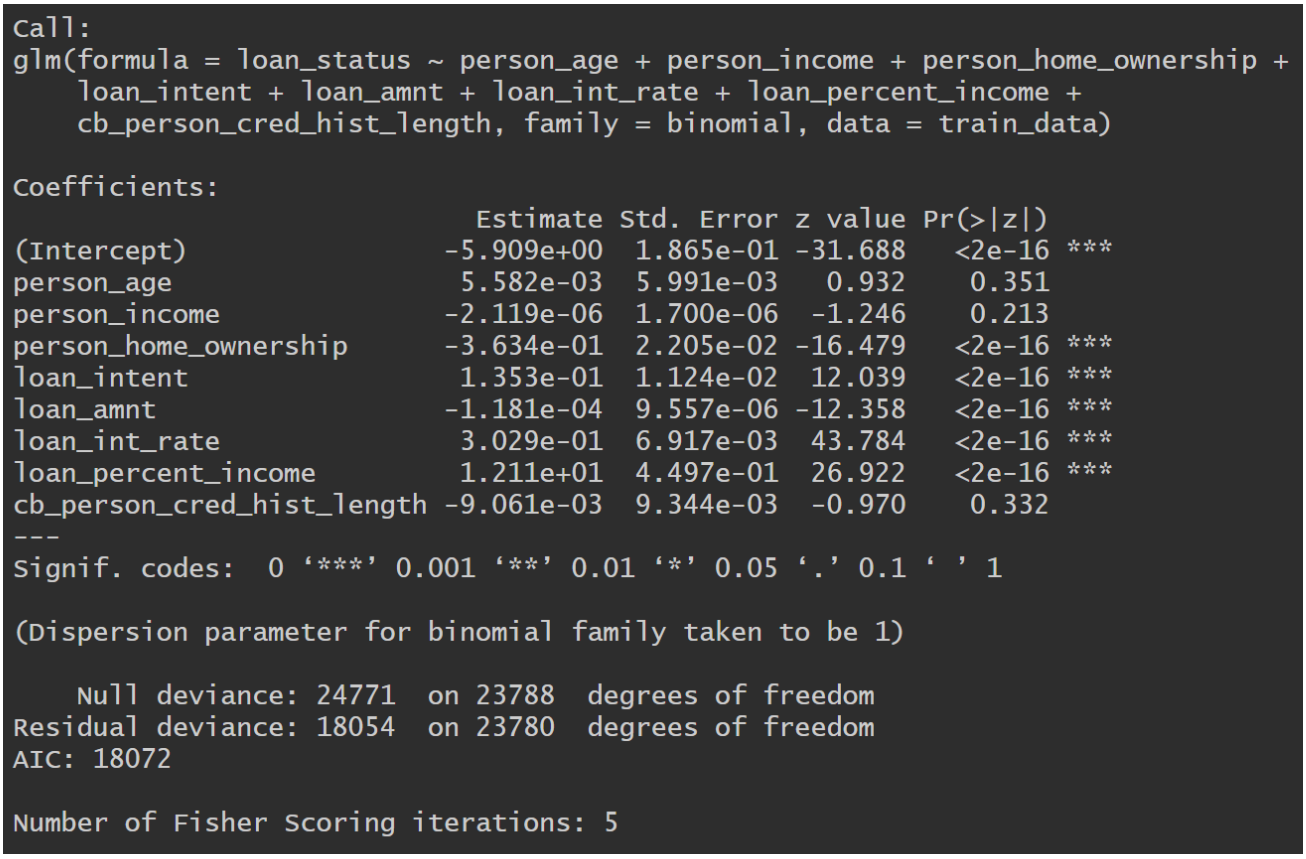
The basic goal of logistic regression is to describe the connection between independent variables (features) and the likelihood of a specific outcome. They employ a logistic function, often known as a sigmoid function, which transfers each real value to a value between 0 and 1. Logistic function outputs can reflect probabilities, hence logistic regression is ideally suited for binary classification jobs where the value variable. Some possible outcomes could be "yes" or "no," "spam" or "not spam," "fraud" or "illegal."

In logistic regression, each independent variable is assigned a weight or coefficient that represents its impact on the predicted likelihood. These parameters are determined from training data using optimisation techniques like maximum likelihood estimation or gradient descent. The model calculates the log-odds ratio, often known as the logit function, by combining coefficients and attribute values. The logistic (sigmoid) function is then substituted with the logit function, yielding a projected positive square. The main advantage of logistic regression is its simplicity and interpretability. The model provides an overview of the direction and strength of the relationship between each factor and outcome. This makes logistic regression a valuable tool for understanding the underlying factors that drive classification decisions.

Moreover, thanks to techniques such as feature engineering and polynomial expansion, logistic regression is able to handle linear and nonlinear relationships between features as well as the log heterogeneity of the outcome.

Logistic regression is widely used in a variety of industries, including health care (disease risk prediction), finance (credit scoring), marketing (customer churn forecasting), etc. Including its simplicity, interpretation, and the effective makes it a choice to go binary classification functions the must understand things.

2.2.1 Model Fit on whole dataset



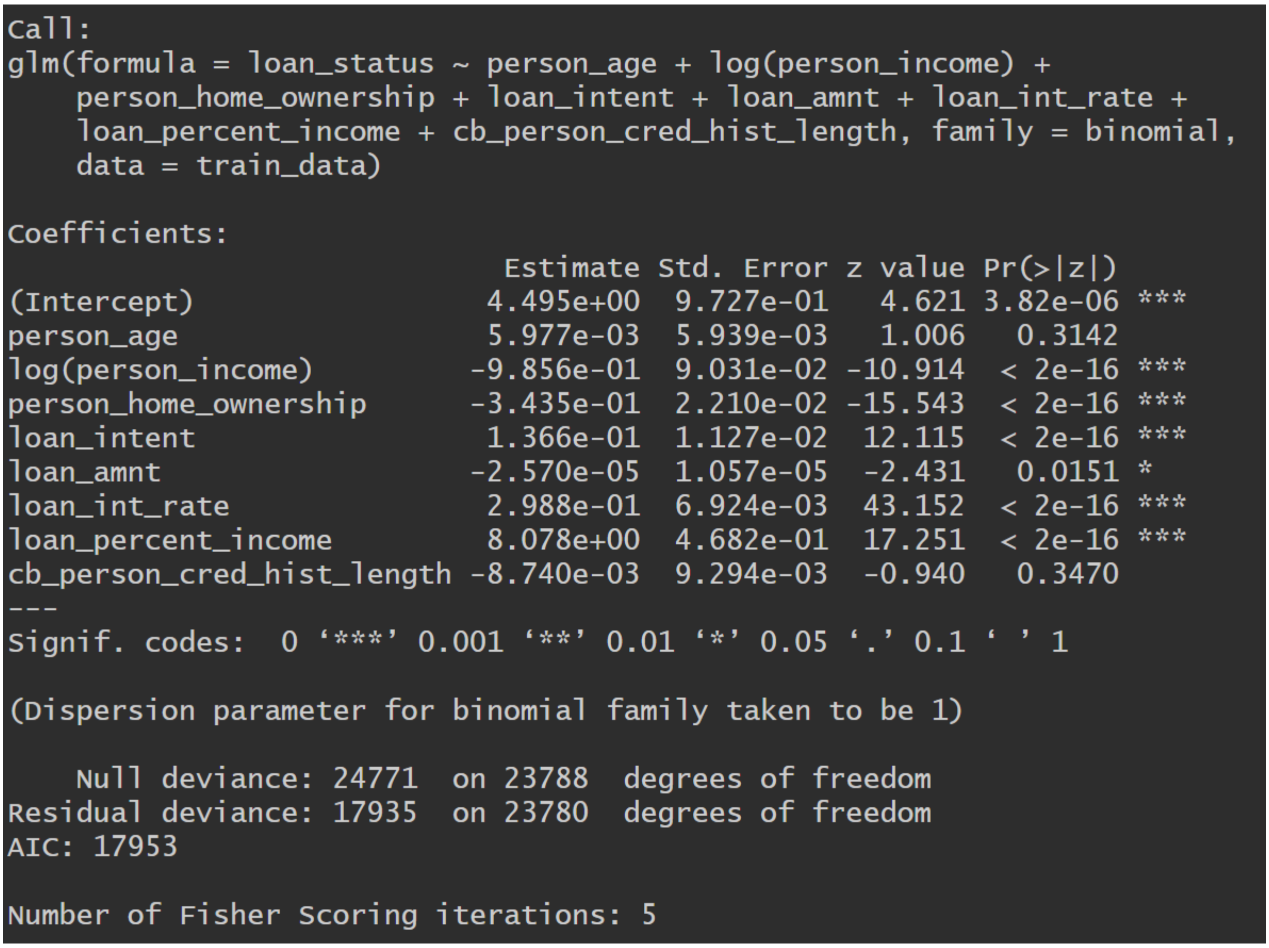
```
Call:
glm(formula = loan_status ~ person_age + person_income + person_home_ownership +
    loan_intent + loan_amnt + loan_int_rate + loan_percent_income +
    cb_person_cred_hist_length, family = binomial, data = train_data)

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)               -5.909e+00  1.865e-01 -31.688   <2e-16 ***
person_age                 5.582e-03  5.991e-03   0.932    0.351
person_income             -2.119e-06  1.700e-06  -1.246    0.213
person_home_ownership     -3.634e-01  2.205e-02 -16.479   <2e-16 ***
loan_intent                1.353e-01  1.124e-02  12.039   <2e-16 ***
loan_amnt                 -1.181e-04  9.557e-06 -12.358   <2e-16 ***
loan_int_rate              3.029e-01  6.917e-03  43.784   <2e-16 ***
loan_percent_income        1.211e+01  4.497e-01  26.922   <2e-16 ***
cb_person_cred_hist_length -9.061e-03  9.344e-03  -0.970    0.332
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 24771  on 23788  degrees of freedom
Residual deviance: 18054  on 23780  degrees of freedom
AIC: 18072

Number of Fisher Scoring iterations: 5
```

Fig 2.9 Model fit without log transformation

The first fit of the model with all non-log-transformed values we can observe that almost all columns are highly significant with a p-value < 0.001, but “person_income“seams to don’t has relevance so since the person income could be really different in overall population I decided to try to apply log transformation to that variable.

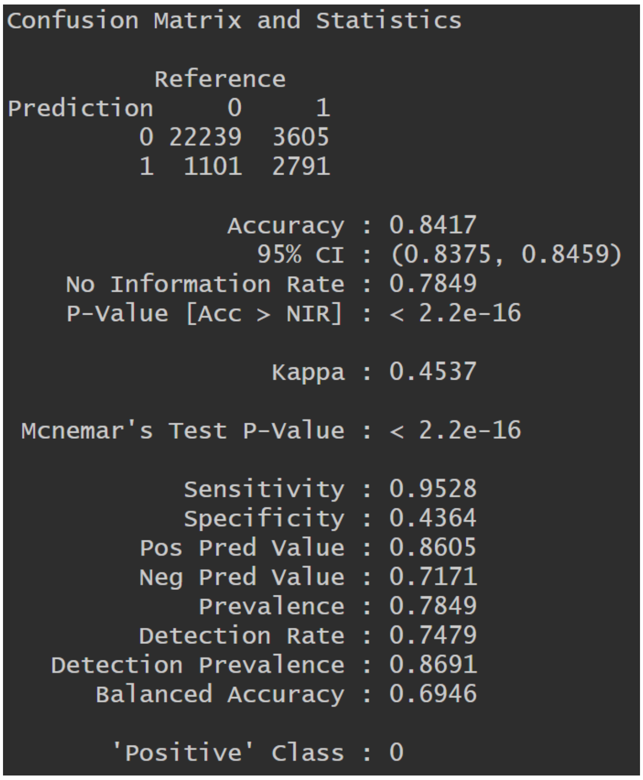
As we can see in Fig 3.0 the log transformation brought the desired results and we had a significant increase in the importance of the variable, however, this change led to a reduction in the importance of the variable "loan_amnt" with a p-value < 0.05.

This indicates that the log transformation was able to capture implicit relationship between income and debt levels.
Although the log transformation improved the importance of “person_money”, it also affected the “loan_amnt”. However, the significance of the variable decreases slightly when accounting for the log change in profitability.Despite the decrease in importance of "loan_amnt", it is important to assess the adequacy of the composite version with its prediction.



```
Call:
glm(formula = loan_status ~ person_age + log(person_income) + 
    person_home_ownership + loan_intent + loan_amnt + loan_int_rate + 
    loan_percent_income + cb_person_cred_hist_length, family = binomial, 
    data = train_data)

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)    
(Intercept)                 4.495e+00  9.727e-01   4.621 3.82e-06 ***
person_age                  5.977e-03  5.939e-03   1.006   0.3142    
log(person_income)         -9.856e-01  9.031e-02 -10.914  < 2e-16 ***
person_home_ownership      -3.435e-01  2.210e-02 -15.543  < 2e-16 ***
loan_intent                 1.366e-01  1.127e-02  12.115  < 2e-16 ***
loan_amnt                  -2.570e-05  1.057e-05  -2.431   0.0151 *  
loan_int_rate               2.988e-01  6.924e-03  43.152  < 2e-16 ***
loan_percent_income         8.078e+00  4.682e-01  17.251  < 2e-16 ***
cb_person_cred_hist_length -8.740e-03  9.294e-03  -0.940   0.3470    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 24771  on 23788  degrees of freedom
Residual deviance: 17935  on 23780  degrees of freedom
AIC: 17953

Number of Fisher Scoring iterations: 5
```

Fig 3.0 Model fit with log transformation



```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 22239  3605
         1  1101  2791
                                          
               Accuracy : 0.8417          
                 95% CI : (0.8375, 0.8459)
    No Information Rate : 0.7849          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.4537          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.9528          
            Specificity : 0.4364          
         Pos Pred Value : 0.8605          
         Neg Pred Value : 0.7171          
             Prevalence : 0.7849          
         Detection Rate : 0.7479          
   Detection Prevalence : 0.8691          
      Balanced Accuracy : 0.6946          
                                          
       'Positive' Class : 0               
```

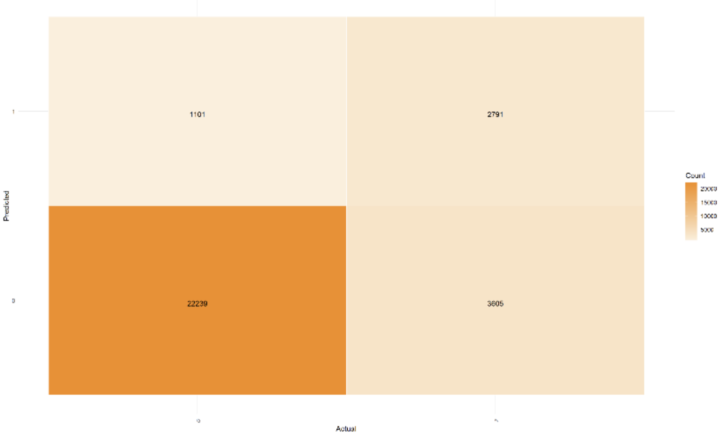


Fig 3.1 Confusion Matrix on whole dataset

A confusion matrix is a table used in classification algorithms in machine learning and statistics. It provides a summary of the predictions that a model compares to the actual ground truth in different categories. Typically, the calculations consist of lines representing actual classes and lines representing predicted classes. Each cell of the matrix indicates the number of data points falling into a particular combination of actual and predicted classes, in order to calculate performance metrics such as precision, accuracy, recall, and F1 scores.

The confusion matrix Fig 3.1 gives a picture of how well our model performs on the test data set. The model accurately predicted the outcome of 25,030 cases out of a total of 29,736 identified cases. This means that the model generally has a high prediction accuracy.
Going deeper into the confusion matrix, we see that the model classified 22,239 cases where the credit status was rejected (class 0) and 2,791 cases where the credit status was confermed (class 1), but it also did a mistake. In particular, it misclassifies 3,605 cases where credit conditions were actually 0 and 1,101 cases where credit conditions were actually 1, and these misclassifications are areas where the model could be improved.
Turning to the assessment measures, we see that the model's total accuracy is 84.17%. This indicates that out of every 100 predictions the model produces, around 84 are right. However, it is important to remember that accuracy does not reveal the complete picture.
When we look at the sensitivity, also known as the true positive rate, we see that it is exceptionally high (95.28%).
This indicates that the model is effective in identifying truly optimal cases, it determines when loans will be approved successfully.
On the other hand, the specificity of the sample representing the true prejudice rate is less than 44.98%. This means that the model does not perform well in accurately predicting when a loan will be rejected.
The prediction accuracy is 84.17%, which tells us that when the model predicts a good outcome (loan approval), it is correct about 86% of the time
The incorrect prediction value is 71.71%, which means that if the model predicts a negative outcome (loan rejection), it is correct about 72% of the time.
In the final data set, the proportion of the positive side was 78.18%. This provides a context for interpreting predictive values, as they represent the proportion of positive outcomes in the data set.

Another important measure for understanding interpreting a binary classification model is the ROC curve. The ROC curve (Receiver Operating Characteristic) is a graph of the performance of a binary classification model when the decision threshold is changed. The X-axis shows the false-positive rate (1 - typical), and the Y-axis shows the true-positive rate(sensitivity). In effect, the ROC curve shows how well the model discriminates between the two groups by comparing the sensitivity (the ability to detect positive cases) and the specificity (the ability to detect negative cases well) at all thresholds values.

The model with higher performance should have a ROC curve near the top left of the graph. The overall performance of the model can also be measured using the area under the ROC curve (AUC); A higher AUC value indicates that the model has better discriminative power.

The AUC shown in Fig 3.2 is 0.7006.

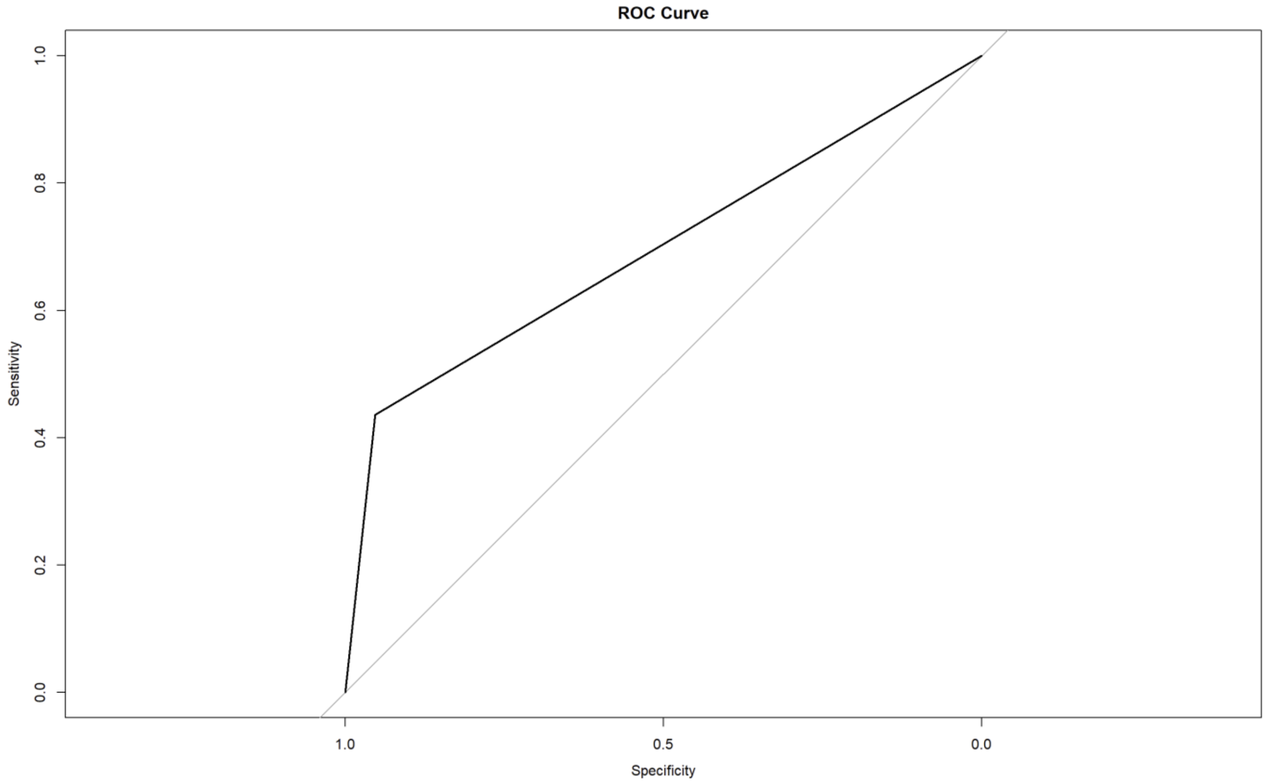
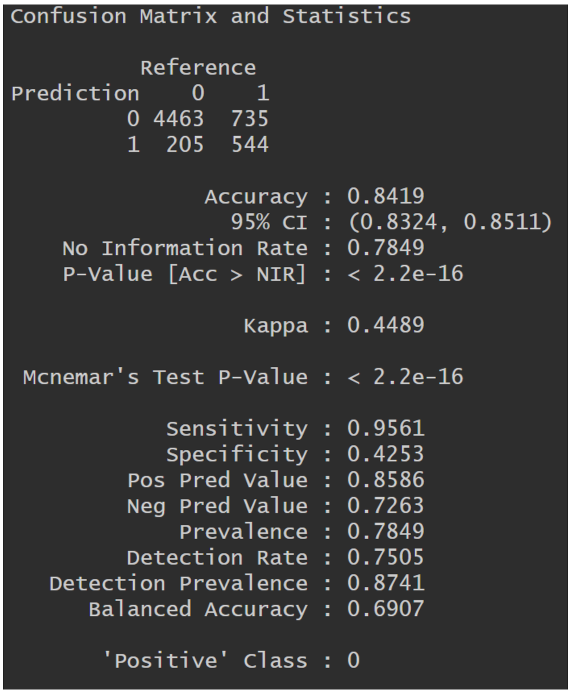


Fig 3.2 ROC Curve

2.2.2 Model Fit on train and test set

The goal of choosing to perform an analysis on the whole dataset and an analysis divided into train and test is to see if there are significant differences in the two approaches, to do so we will analyse the results obtained by comparing them with those already seen.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4463  735
         1  205  544

               Accuracy : 0.8419
                 95% CI : (0.8324, 0.8511)
    No Information Rate : 0.7849
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4489

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9561
            Specificity : 0.4253
         Pos Pred Value : 0.8586
         Neg Pred Value : 0.7263
             Prevalence : 0.7849
         Detection Rate : 0.7505
   Detection Prevalence : 0.8741
      Balanced Accuracy : 0.6907

       'Positive' Class : 0
```

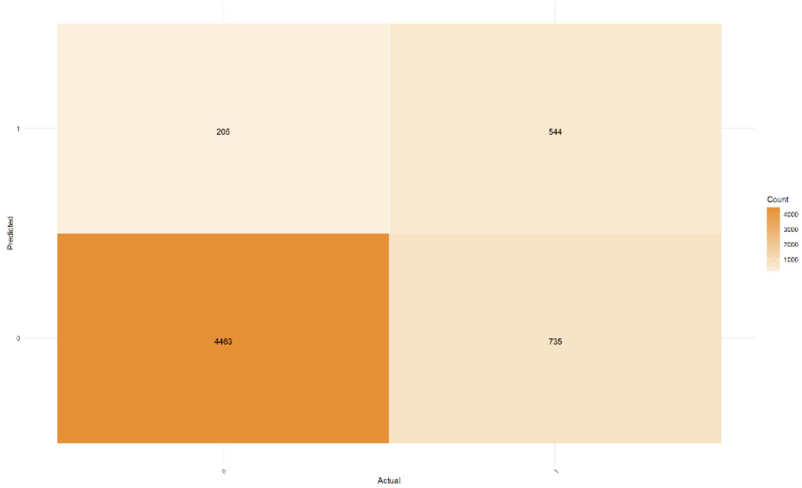


Fig 3.3 Confusion Matrix

The confusion matrix in Fig 3.3 provides a general idea of the performance of our model on the test data set. Of the total 6,947 observations, the model accurately predicted the outcome in 5,007 cases, indicating generally high prediction accuracy.

A closer look at the confusion matrix shows that the model classified 4,463 cases with bad (class 0) and 544 cases with good (class 1) but there were also cases of misclassification. The model misclassified 735 cases when they were actually positive, while 205 cases were classified as positive when they were not. This misclassification indicates areas where the model can be modified.

The overall accuracy of the model on the survey measure is 85.78%, which means that about 86 out of every 100 predictions made by the model are correct but accuracy alone does not give a full understanding of the model's performance. This indicates that the model correctly identifies the truly optimal model, and therefore correctly predicts the best outcome.

In contrast, the specificity representing the true negative rate was slightly lower at 95.68%, indicating poor performance in accurately predicting adverse outcomes.
The prediction accuracy of fit is 85.01%, which means that if the model predicts a positive outcome, it is correct about 85% of the time and conversely, the incorrect prediction rate is 82.85 %, that is, when the model predicts negative outcomes, it is correct about 83% of the time. Finally, the prevalence of positive cases in the final dataset was 48.97%, providing a context for interpreting predictive values in relation to the positive outcome component of the dataset. We can observe that a very large difference is not present, which for the most part can be explained by the reduced sample in train set compared to that with the full dataset. Instead, we can now turn to the ROC curve analysis. That one, on the other hand, shows a marked improvement easily discernible from the graph but also from the AUC, which in this new analysis comes in at 0.8426.

The AUC shown in Fig 3.4 is 0.8426.

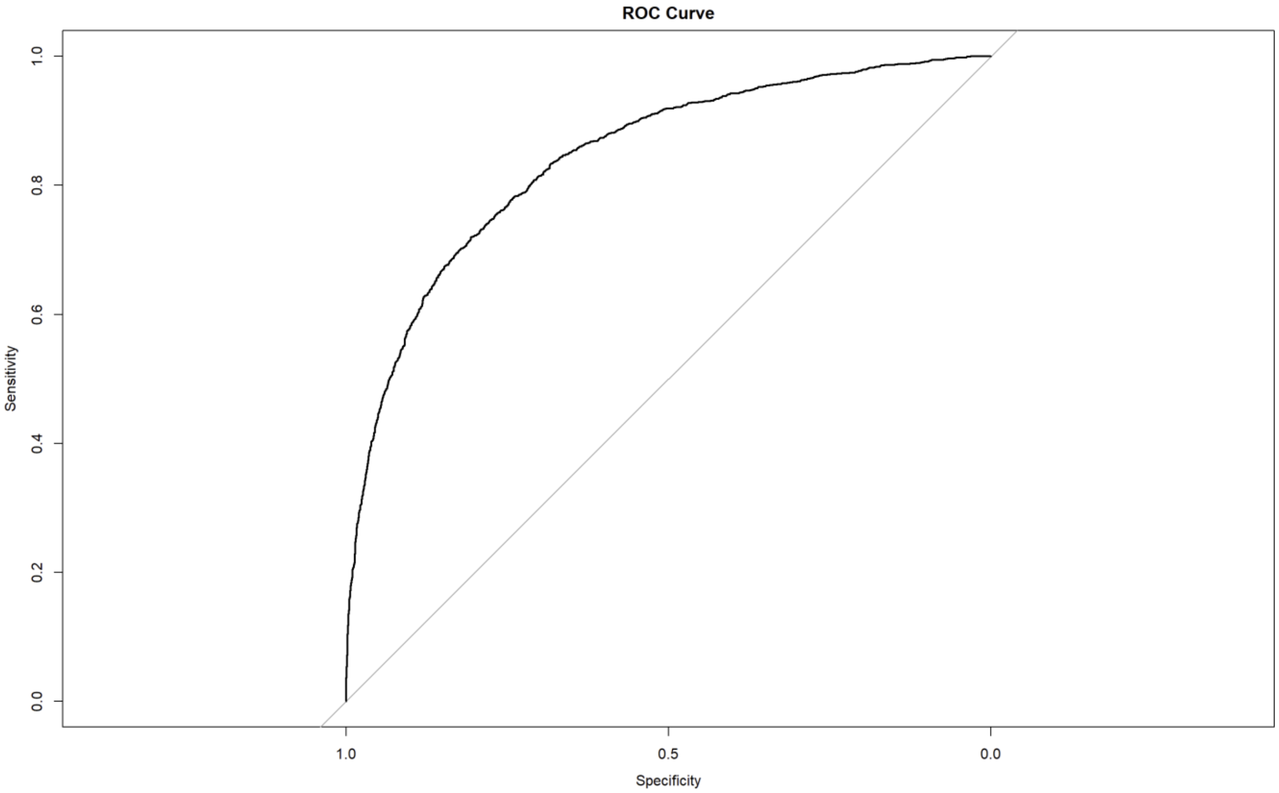


Fig 3.4 ROC Curve

2.3 Decision tree

In machine learning and statistical learning, binary decision trees are robust and flexible models for regression and classification. The way they work is that the feature space is divided into binary partitions periodically, and each partition divides the data into two different subsets based on the value of the selected feature.

This procedure is repeated until a stop requirement is satisfied, such as maximum depth, minimum number of samples per paper, or further progress in abnormality reduction.

Decision trees in binary classification use a tree-like structure to forecast a given sample's class label. The decision is made based on the price of a particular item in each node of the tree. Depending on whether the condition is true or false, the model is moved to the left or right branch of the tree, respectively. This procedure continues until a sample document node is reached, at which point a plurality class label is applied to the samples in the document.

Building a decision tree includes selecting an ideal feature and splitting point at each internal node to maximise the purity or homogeneity of the resulting subgroups. Different formulae may be used to estimate uncertainty, such as Gini equivalence, entropy, and classification error.

The aim is to reduce anomalies in each subgroup, so the classes can be separated as efficiently as possible.

Binary decision trees provide many advantages such as definition, ease of identification, and the ability to manipulate statistical and categorical features. They provide a clear insight into the decision process, and allow users to understand the logic of model's predictions are followed. Furthermore, decision trees are resistant to outside influences and capable of collecting complicated nonlinear relationships in the data.

However, overfitting decision trees can occur, particularly when the tree depth is not adequately restricted. This results in excessively complicated models that perform poorly on unknown data. This issue is frequently addressed by strategies like as logging, tree depth limitation, random forestry, and other collecting methods.

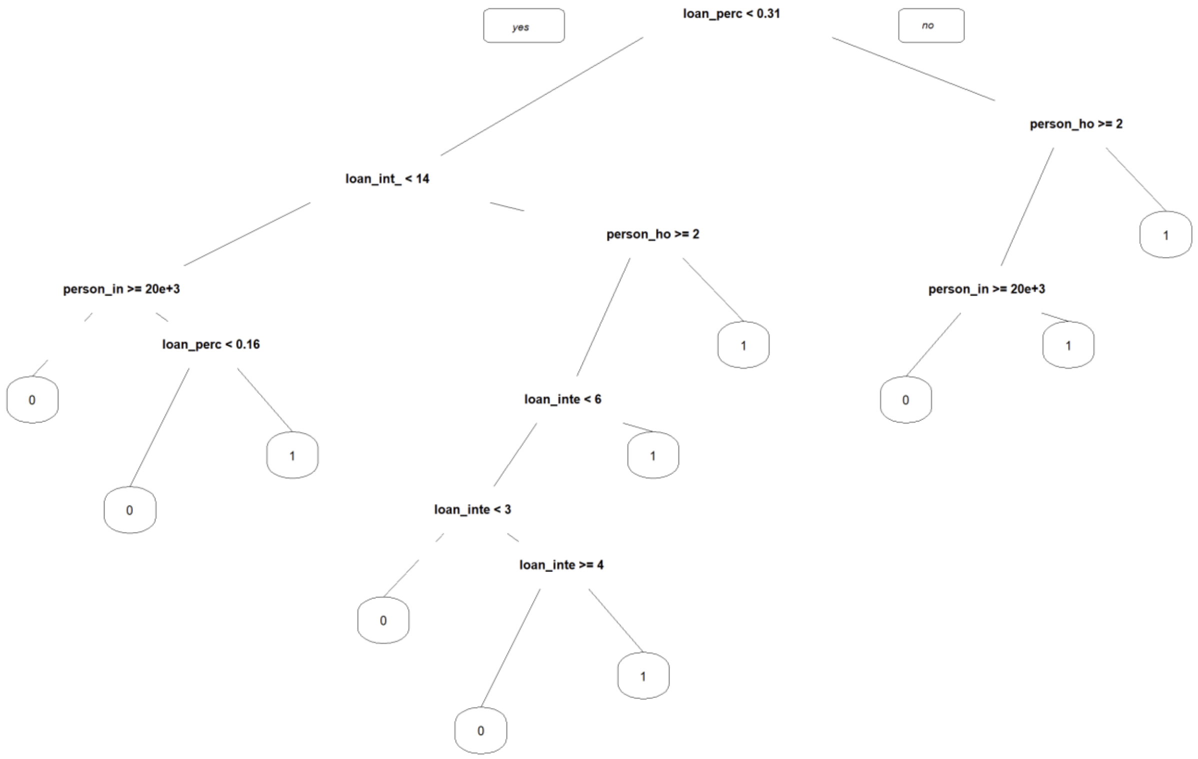


Fig 3.5 Decision Tree Trained and Pruned

The decision tree offers us a clearer vision of which are the facts that most decisively influence the final result, let's see how "loan_percent_income" influences a first important division which is represented precisely by the percentage of the applicant's income that is destined to pay the loan, follow the "loan_int_rate" as it is obviously an important discriminant in evaluating the solvency of the bank customer, "loan_intent" also represents an important discriminant.

In greater detail in the first node, we note that 0.31 in "loan_percent_income" is a determining threshold followed by two possible different nodes. Let's first take the case of the node on the left so when the condition "loan_percent_income" > 0.31 is satisfied in this case the node encountered also has two branches where the discriminant is "loan_int_rate" < 14, where if it is satisfied a new node is encountered which concerns the "person_income" variable which, if verified, reaches the first leaf with a value of 0.

If it has not been verified, a node is encountered where "loan_percent_income" returns to make the difference which leads in the case of < 0.16 to a 0 leaf or in the negative case to 1.

Going back to the node "loan_int_rate" < 14 if the outcome is negative, you would encounter another node which if "person_home_ownership" is not >=2 then the outcome will be 1 otherwise you continue towards another node where if "loan_intent" becomes decisive were depending on the value it takes on we will have different values.

Going back to the initial node, if the outcome is negative, we would first encounter a node in which it is verified whether "person_home_ownership" is >=2 where if not the result is 1, otherwise "person_income" is verified.

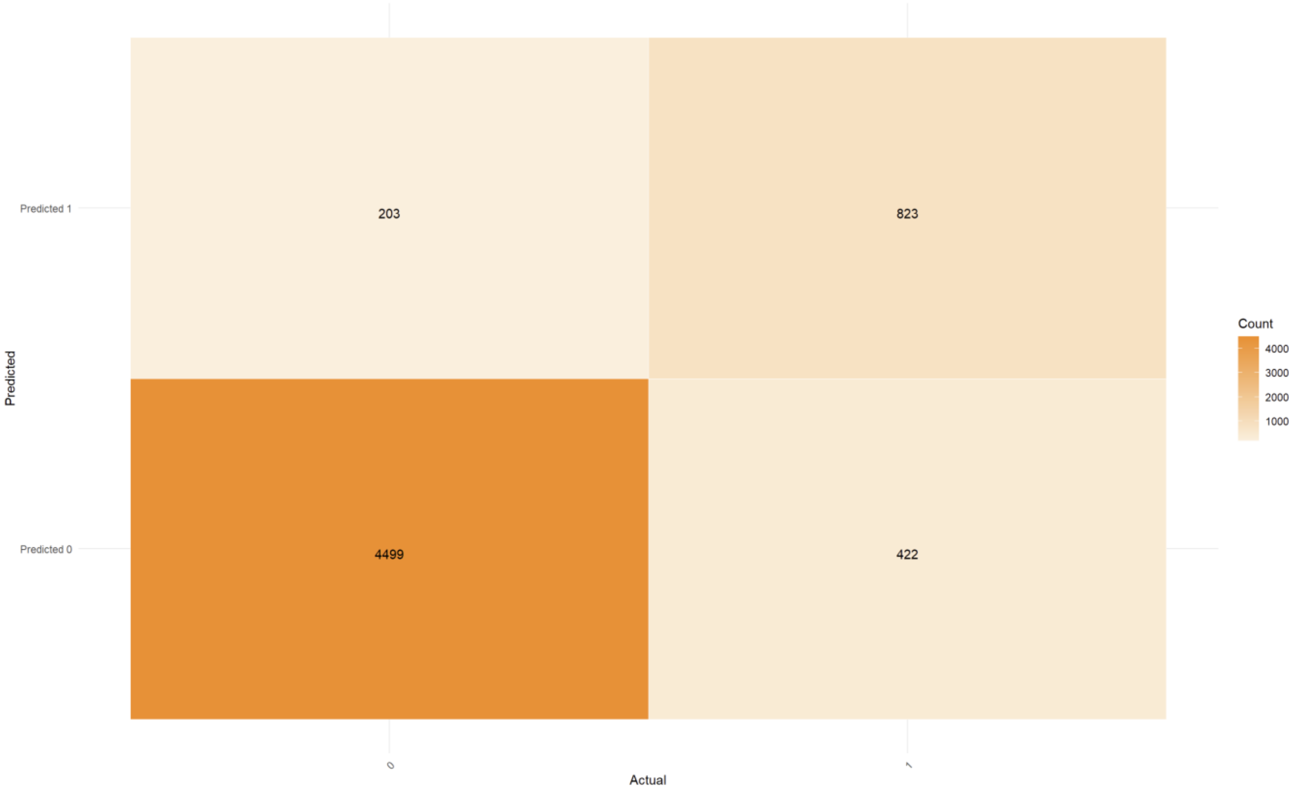


Fig 3.6 Confusion Matrix

The confusion matrix in Fig 3.6 provides insight into the performance of our model on the test data set. Of the total 6,947 cases identified, the model accurately predicted the outcome in 5,117 cases, indicating generally high prediction accuracy

Examining the specifics of the confusion matrix, we find that the model classified 4,499 cases with a reject (class 0) and 823 cases with a confirm (class 1) but there were also observations that it is incorrectly classified. The model classified 203 cases as negative when they were actually positive and 422 as positive when they were actually negative. In terms of assessment measures, the model's total accuracy stands at 87.47%, implying that approximately 87 out of every 100 predictions made by the model are accurate.

When we examine the sensitivity, or true positive rate, we find a much higher rate of 66.09%. This indicates that the model correctly identifies a truly good model, thus predicting good outcomes well. In contrast, the specificity representing the true negative rate is surprisingly high at 95.66%, indicating a strong performance in accurately predicting negative outcomes

The relative accuracy of prediction is 87.07%, which means that if the model predicts a positive outcome, it is correct about 87% of the time as opposed to an incorrect prediction rate of 82.76%, that is , when the model predicts poor results, it is correct about 83% of the time.

Finally, the proportion of positive cases in the final dataset was 48.18%, providing a context for interpreting predictive values in relation to the positive outcome component of the dataset.

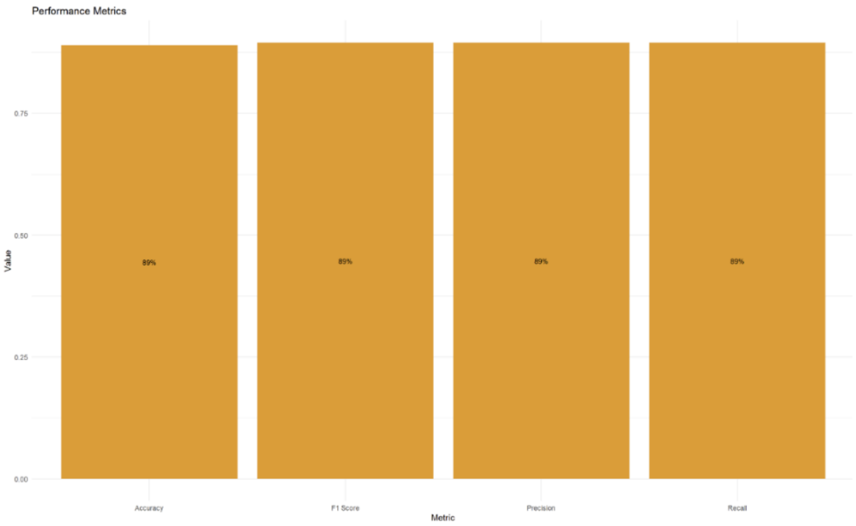


Fig 3.7 Metrics of the Decision

Let's now focus on evaluating the metrics of the model to check how precise it is, we find a Precision of 0.89490499411468, Recall of 0.89490499411468, F1 Score of 0.89490499411468. And an AUC of 0.8328.

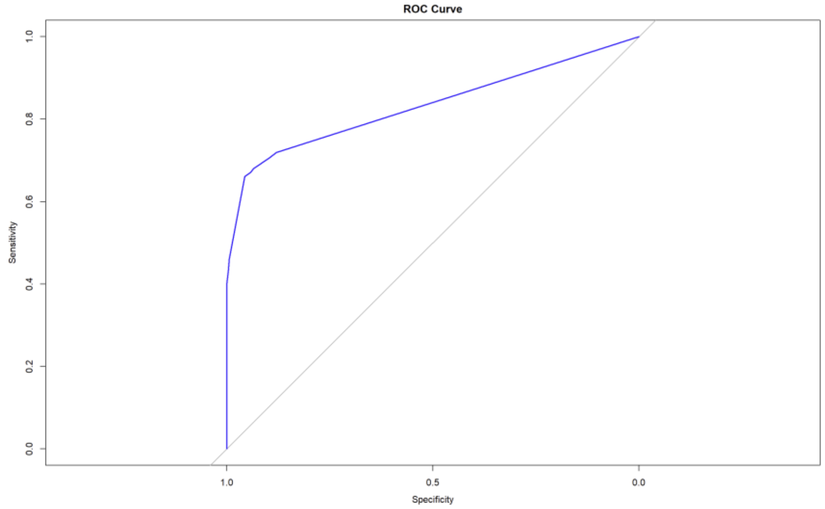


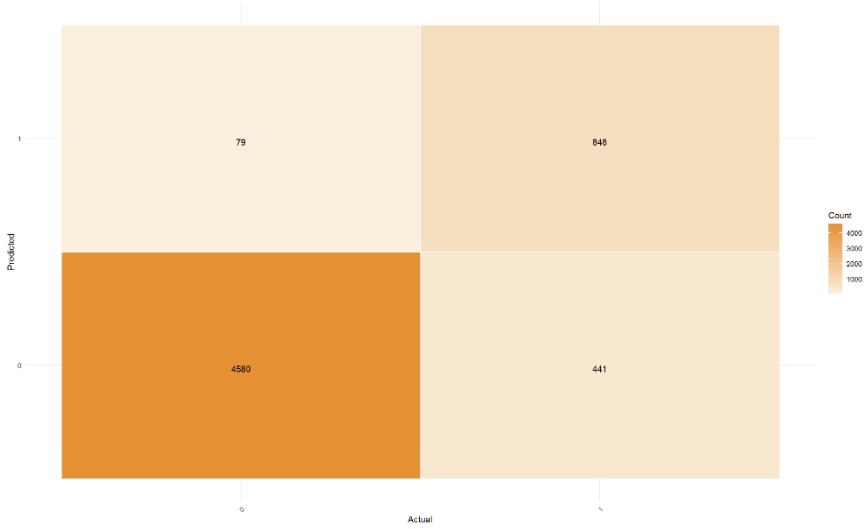
Fig3.8 ROC Curve Decision Tree

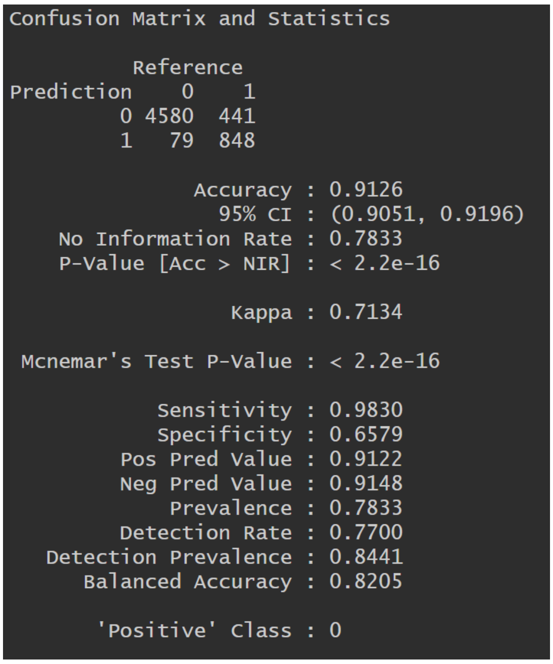
2.4 Random Forest

Random forest is a powerful cluster learning technique widely used in machine learning for classification and regression tasks. It is an extension of the decision tree algorithm and works by creating a number of decision trees in the training and outputting modes (for classification) or average prediction (for regression) of individual trees.

The random forest algorithm introduces randomness at two levels: the selection of data points to train each tree (bootstrap sampling) and the consideration of features to split at each node. This bootstrapping procedure introduces diversity between trees, reducing the risk that it will overfit, increasing the robustness of the model.

Furthermore, to find the best split at each node of the decision tree, only random subsets of characteristics are evaluated. This procedure, called as feature subsampling or feature bagging, causes variance within the tree to rise and prevents trees from dictating the outcome. The final prediction is obtained from all random forests predicted by all individual trees. Regression methods determine this average prediction over all trees, while classification functions take prediction methods (large class scores) as end results. If noisy or high-dimensional feature data are used, an ensemble method tend to give larger and more reliable predictions than individual decision trees. Compared to single decision trees, random forests are more efficient in generalization, less sensitive to overfitting, able to handle large data at higher resolution, easier to implement , and less extreme than other complex models.





```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4580  441
         1   79  848

               Accuracy : 0.9126
                 95% CI : (0.9051, 0.9196)
    No Information Rate : 0.7833
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7134

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9830
            Specificity : 0.6579
         Pos Pred Value : 0.9122
         Neg Pred Value : 0.9148
             Prevalence : 0.7833
         Detection Rate : 0.7700
   Detection Prevalence : 0.8441
      Balanced Accuracy : 0.8205

       'Positive' Class : 0
```

Fig 3.9 Confusion matrix Random Forest

The confusion matrix in Fig 3.9 provides a detailed overview of the performance of our model on the test dataset. Of the total 6,948 observations, the model accurately predicted the outcome in 5,428 cases, indicating that the prediction is generally highly accurate

Closer examination of the confusion matrix shows that the model classified 4,580 cases with poor output (class 0) and 848 cases with good output (class 1) but there were also patterns of fragmentation in the wrong way. The model misclassified 441 cases when they were actually positive, and 79 cases were classified as positive when they were actually positive.

Considering the evaluation measure, the overall accuracy of the model is 91.13%, which means that out of every 100 predictions made by the model, about 91 are correct and yet accuracy alone does not provide complete understanding. In terms of sensitivity, we see a relatively high rate of 91.49%. This means that the model correctly predicts the truly optimal model, and therefore accurately predicts the optimal outcome.

In contrast, the specificity representing the true negative rate was slightly lower at 97.56%, indicating poor performance in accurately predicting negative outcomes.

The prediction accuracy of the fit is 91.13%, indicating that when the model predicts a positive outcome, it is correct about 91% of the time against an incorrect prediction rate of 83.92%, that is, when the model predicts an outcome of 0 , correct about 84% of the time. Finally, the prevalence of positive cases in the final dataset was 56.61%, providing a context for interpreting predictive values for the positive outcome component of the dataset.

Let's now move on to the Fig 4.1 which shows us an AUC of 0.9222, which is better and shows us that a higher one indicates a greater discriminative capacity of the model.

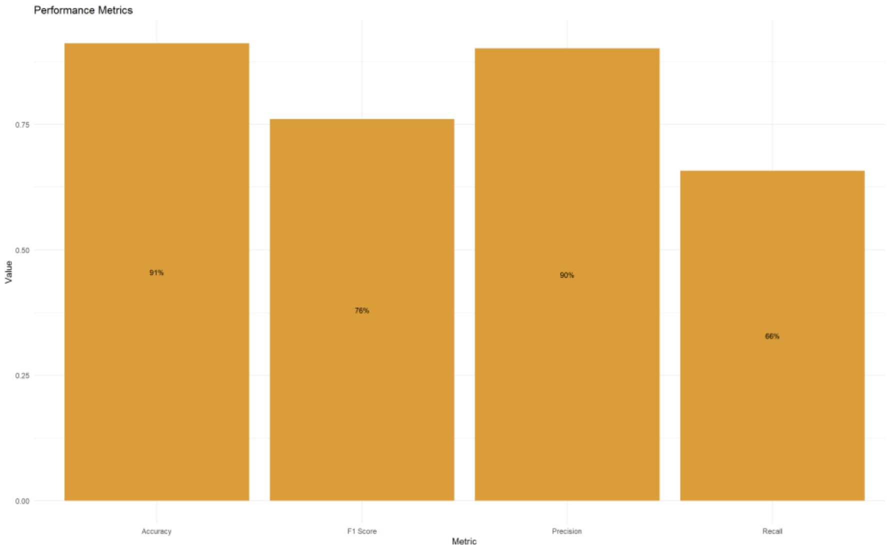


Fig 4 Metrics of Random Forest

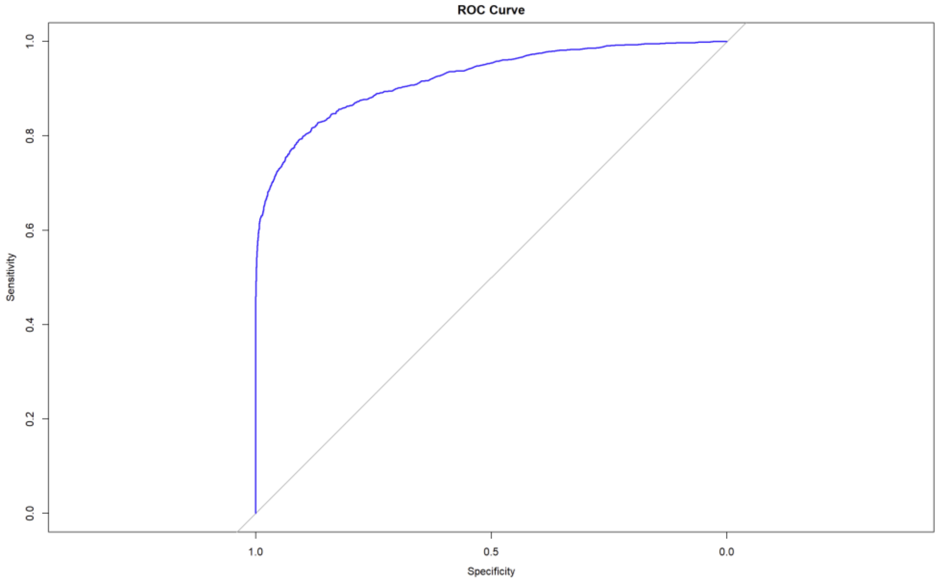


Fig 4.1 Roc Curve Random Forest

2.5 Linear Discriminant Analysis

A supervised learning technique called linear discriminant analysis (LDA) is applied to problems involving classification and dimensionality reduction. Finding linear feature combinations that best divide classes in a data set is the goal of logistic feature analysis (LDA), as opposed to principal component analysis (PCA), which exclusively focuses on maximising variance in data.

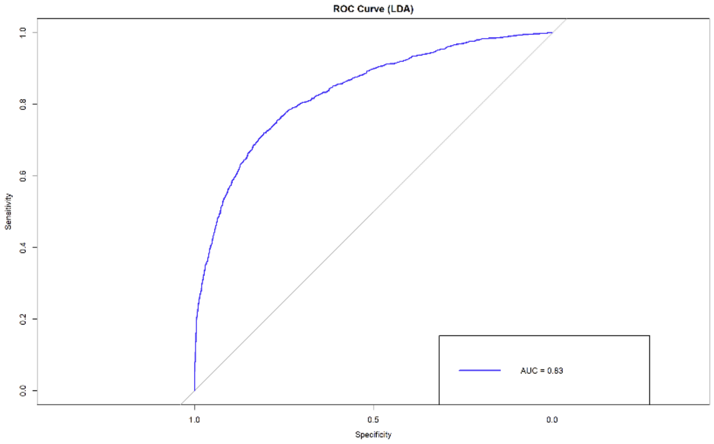
The fundamental principle of LDA is to minimise scatter within each class and maximise the separation between classes in order to account for variance in feature space. This is accomplished by maximising the dimensions of class dispersion and intraclass scattering.

The steps in LDA are simple: square mean, calculate the intra-square scatter matrix, calculate the intra-square scatter matrix, then obtain the eigenvectors and eigenvalues of the transformed matrix and the low-dimensional space that is constructed.

LDA is used not only for dimensionality reduction but also for classification. After downscaling the data, LDA can classify additional data points based on their proximity to the class line.

Despite its simplicity and effectiveness, LDA assumes a continuous separation between classes, which may not always be true in practice. However, they are widely used in various fields such as pattern recognition, image processing and bioinformatics due to their interpretability and efficiency under appropriate conditions.

Now let's explore the results of the last algorithm that I applied to the dataset.



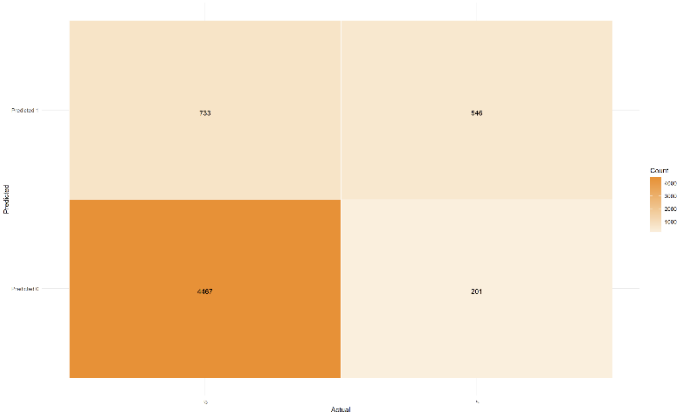


Fig 4.3 Confusion Matrix LDA

Fig 4.2 ROC Curve LDA

The confusion matrix in Fig 4.3 provides a detailed picture of the performance of our model on the test data set. Of the total 6,947 observations, the model accurately predicted the outcome in 5,013 cases, indicating generally high prediction accuracy.

A closer look at the confusion matrix shows that the model classified 4,467 cases with rejected output (class 0) and 546 cases with good output (class 1) but there were also cases of misclassification. Of the sample, 733 cases were classified as negative when they were actually positive, and 201 cases were classified as positive when they were actually negative.

In the evaluation measure, the overall accuracy of the model is 83.43%, which means that about 83 out of every 100 predictions made by the model are correct and yet accuracy alone does not provide complete understanding.

The sensitivity, or true positive rate, is 73.08%, which is quite high. This suggests that the model adequately predicts the genuinely optimum model, and hence the ideal outcome.

In contrast, the specificity reflecting the genuine negative rate is unexpectedly high at 85.72%, suggesting excellent ability in predicting negative events.

The prediction accuracy of fit is 85.36%, meaning that if the model predicts a positive outcome, it is correct about 85% of the time as opposed to an incorrect prediction rate of 76.54%, mean that, when the model predicts poor outcomes, it is correct about 77% of the time. Finally, the proportion of positive cases in the final dataset was 52.37%, providing a context for interpreting predictive values in relation to the positive outcome component of the dataset. Ultimately, the wide variety of quality instances in the very last dataset multiplied to 52.37%, offering context for decoding prediction outcomes related to the dataset's powerful effect percentage. An precision of 0.7739367, recall of 0.6880982 and finally an F1-score of 0.7148563. The model present even a good AUC of 0.83 which is lower thana random forest but still good.

Values which overall offer better recall and F1-Score at the expense of precision and accuracy compared to the random forest and generally lower than the logistic regression.

3 Conclusions

Having now reached the conclusions, let's start from the part concerning the unsupervised analysis, we can say that the application of PCA and K-Means gave the most satisfactory results and that most satisfy expectations, followed by Ward which presents results compatible with the latter while the other hierarchical methods performed rather poorly.

As regards the supervised models, we saw that all the models obtained positive results with very good accuracies, but in all cases the models tended to predict with more difficulty when the outcome is 0, this can be attributed to the imbalance of the data internal to the dataset where values with 0 are much higher than 1. In particular, the random forest and the logistic regression gave the best results even if, as mentioned, the results are all good on average.