UNIVERSITÀ
DEGLI STUDI
DI MILANO

Statistical Learning Project

Application of supervised and unsupervised techniques on Pima Indians Diabetes Dataset

EMANUELE MORALES
941935

Data Science and Economics

September, 2020

List of Figures

Contents

Chapter 1

Introduction

1.1 First section: Unsupervised Methods

In the first section, two unsupervised methods are applied.

The first one is Principal component analysis, that allows to summarize the dataset with a smaller number of significant variables, that are able to explain the largest variability of the original set. The aim of this preliminary explorative analysis is to obtain principal components and exploit them to visualize data in a low-dimensional representation, that captures the largest amount of information. In the report the "eight dimensionality" of the dataset is reduced to a "three dimensionality" by applying PCA, and a 3d representation is given by using the "rgl" package on R.

Aim of the analysis:

- find the number of principal components that explains a sufficient variance of data;
- give a graphical representation of the data by using the components obtained.

The second method applied is K-means Clustering. Clustering is a technique that allows to partition the set of data into distinct subgroups, so that the observations inside each subgroup are similar each other and different from the observations of the other groups. In order to obtain a graphical representation of the clustering in the diabetes dataset, the clustering will be implemented starting from the principal components in output from the previous PCA.

Aim of the analysis:

- find the best number of clustering by using Elbow Method;
- observe if a binary clustering is able to partition diabetic and not diabetic patient (not unsupervised anymore).

1.2 Second section: Supervised Methods

In the second section, supervised techniques are applied to predict which category belongs to a patient (diabetic/non diabetic). Since the dataset has a qualitative outcome, classification problems are solved. After having split the dataset in training and test sets, different methods are applied.

The first one is logistic regression, that models the probability that a certain observation belongs to a particular category.

Then, classification trees are applied, to observe how the performance of the prediction changes applying the following tree and ensemble models:

- Unpruned Tree;
- Pruned Tree with cross validation method;

- Bagging;
- Random Forest;
- Boosting.

The goal of this section is to compare the different supervised techniques performances on the test set, and evaluate what is the best model that solves the diabetes classification problem, paying attention also to phenomena that could compromise the generalization of the model, like for example overfitting.

Chapter 2

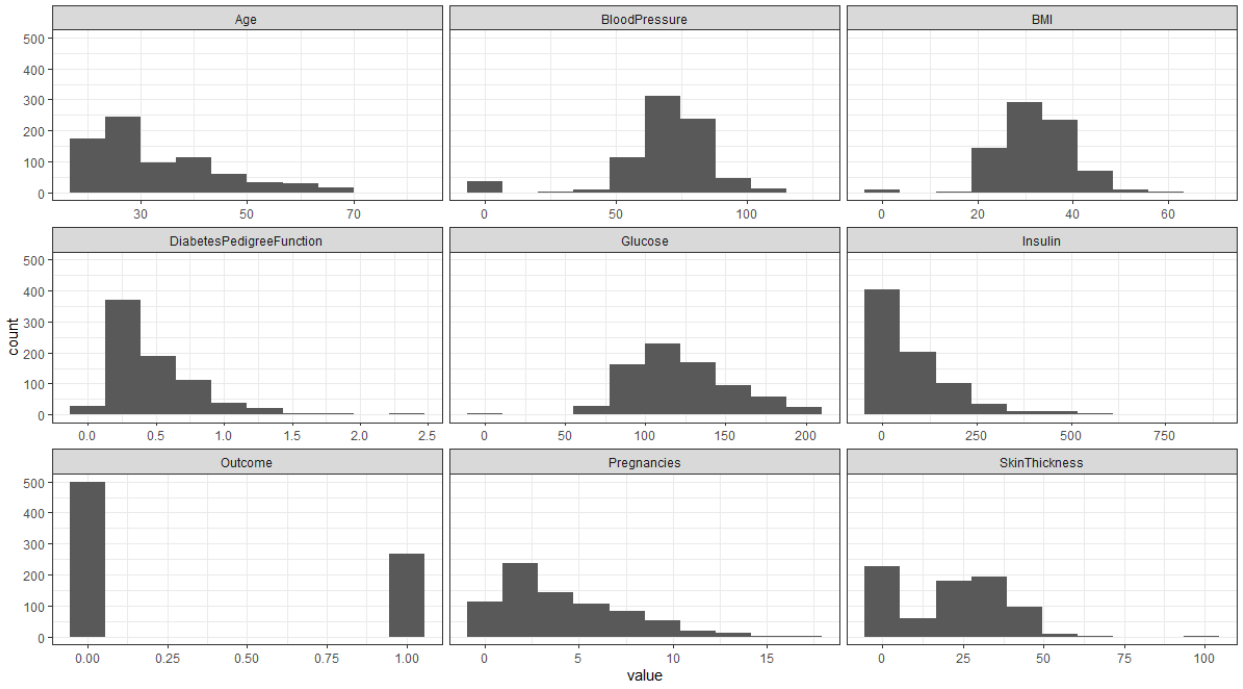
Dataset and data cleaning

2.1 Dataset structure

The dataset used in this analysis is the Pima Indians Diabetes Dataset, that involves predicting the onset of diabetes within 5 years in Pima Indians given some variables that represents medical details. The dataset consists of 768 observations and includes 8 features and 1 binary target response:

- Number of times pregnant;
- Plasma glucose concentration;
- Diastolic blood pressure (mm Hg);
- Triceps skin fold thickness (mm);
- 2-Hour Serum insulin;
- Body mass index;
- Diabetes Pedigree Function;
- Age;
- Class variable (0 = non diabetic in five years; 1 = diabetic in five years)

It follows a graphical representation of the distribution of the data:



It can be observed that some medical variables have missing values; for example there are about 50 patients with a Blood Pressure = 0, or Insulin with more than 300 values equal to zero. Since the number of observations is not so high, it is impossible to remove the rows containing the missing values.

For this reason it has been chosen to substitute the zero values in the variables Glucose, Blood Pressure, Diabetes Pedigree Function, Insulin and BMI with the median value of each variable calculated on the available data.

2.1.1 Data in unsupervised section

In the section where PCA and Clustering are applied, the column Outcome is dropped to apply the unsupervised techniques that do not expect target labels.

2.1.2 Data in supervised section

In the section dedicated to supervised methods, the outcome variable is reintroduced and the dataset is split into training and test set using a proportion of 70% for training and 30% for test set.

Moreover, since the medical variables have different scales, it is applied Min-Max normalization to scale the data in a range between 0 and 1. Normalization is not necessary for algorithms that are scale invariant, like for example tree predictors, but it can be useful for techniques like logistic regression, in which the scale of data matters.

2.1.3 Accuracy function

In this analysis the accuracy of an algorithm is defined as the number of correct observations classified in the test or training set over the total number of observations belonging to the set.

Chapter 3

Unsupervised Techniques: PCA and Clustering

3.1 Principal Component Analysis

3.1.1 PCA: Theoretical Background

Supposing to have a dataset containing n observation in a p-dimensional space, the aim of PCA is to look for the dimensions that explain the most variance in the observations.

The first principal component is the result of the linear combination of the features which has the largest variance.

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \ldots + \phi_{p1}X_p$$

The phi parameters are called loadings and they are normalized, that means that their squared sum all over the features is equal to one. Therefore, the maximization problem to be solved to find the component is:

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \text{ subject to } \sum_{j=1}^{p}\phi_{j1}^2 = 1.$$

The second principal component is found in a similar way: it is the linear combination of the features that has the largest variance and it must be uncorrelated with respect to the first PC. Uncorrelation means that the loading vector of the second component is orthogonal with respect to the direction of the first one.

Since the most important aspect in this analysis is the variance, data are standardized with 0 mean before the implementation.

3.1.2 PCA: Application on dataset

After having processed data like described in the chapter 2.1.1, PCA is applied on the Diabetes Dataset. It is useful to understand how the variables are correlated each other:

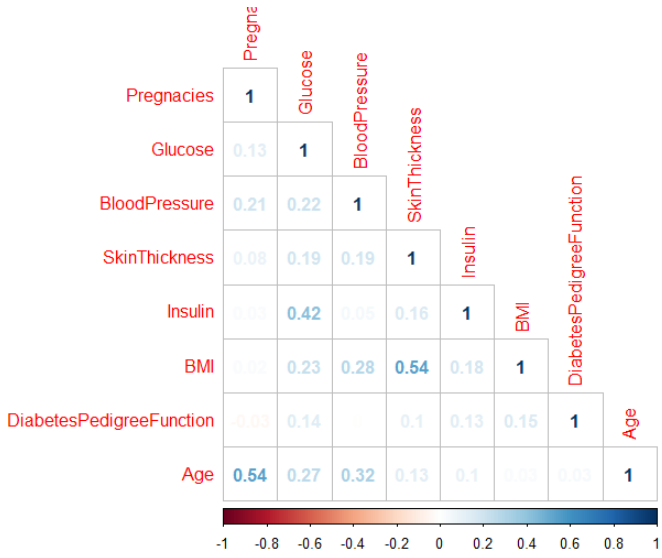


Figure 3.1: Correlation plot of the variables

The most correlated variables are Age-Pregnacies, Insuline-Glucose and BMI-SkinThickness.
PCA applied on dataset returns the following result:

```
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4     Comp.5     Comp.6     Comp.7    Comp.8
Standard deviation     1.5109551 1.2229382 1.0684121 0.9573535 0.87684860 0.73707242 0.68443405 0.6187225
Proportion of Variance 0.2853732 0.1869472 0.1426881 0.1145657 0.09610793 0.06790947 0.05855625 0.0478522
Cumulative Proportion  0.2853732 0.4723204 0.6150084 0.7295742 0.82568209 0.89359156 0.95214780 1.0000000
```

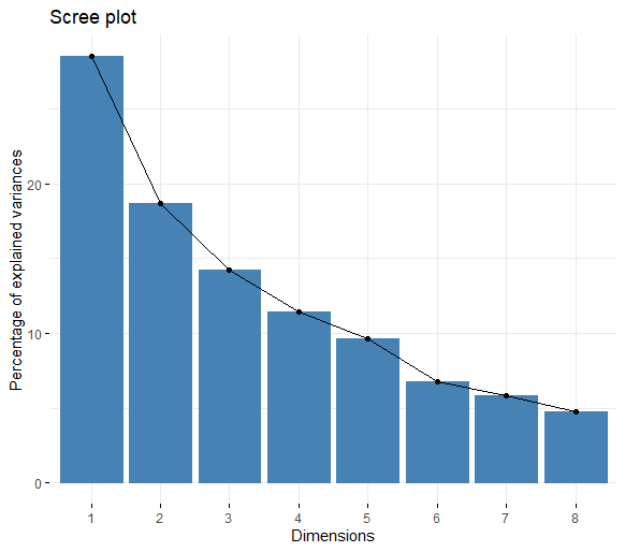


Figure 3.2: Screeplot:plots the variances against the number of the principal component

From the screeplot it is observed that the first component explains the 28% of the variance, and the cumulative variance explained together with the second is 47%, that it is not enough to have a good model. In order to have a more precise model it should be considered a number of PC equal to at least 5, but in

this case data could not be visualized. Hence the first three components (61% of variance explained) will be taken into account to obtain a three dimensional representation of the data, obtaining the following layout in the space:

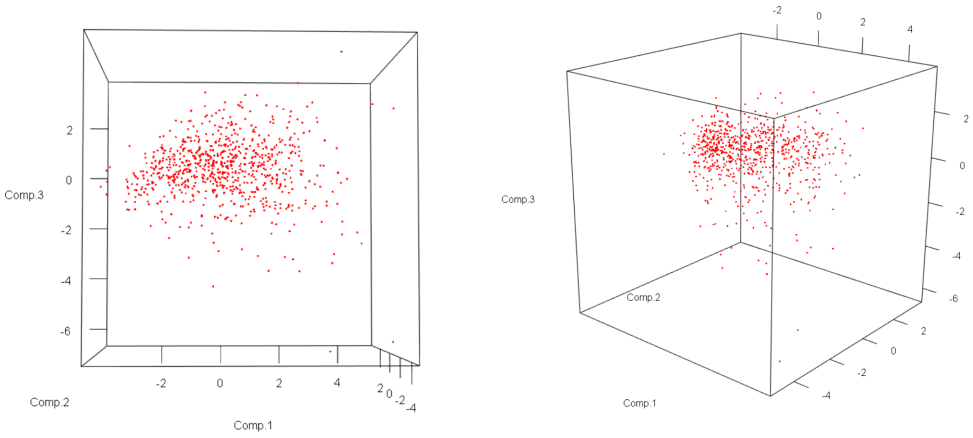


Figure 3.3: Plot of the observation along the first three principal components

Now it is possible also to analyze the biplot that represents the scores of principal components and the magnitude of the loading vectors.

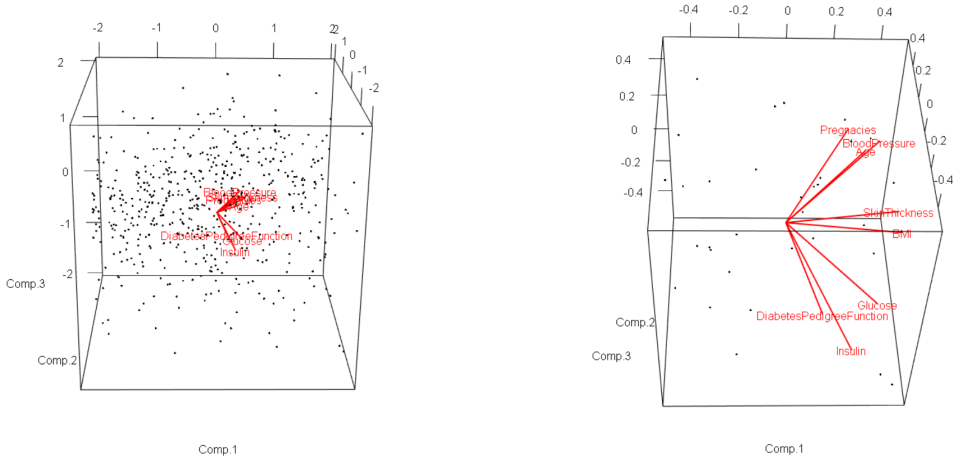


Figure 3.4: Biplot: original scale (left), in a reduced scale (right)

The first loading places a weight over 0.3 for all the variables except the DiabetesPedigreeFunction. The second loading gives both positive both negative weights to the variables with an high magnitude on Pregnancies and Age and the lowest to the BloodPressure. The third loading gives both positive both negative weights to the variables and gives the highest value to Insulin.

```
Loadings:
                        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
Pregnacies               0.301  0.558         0.160  0.387  0.155  0.509  0.372
Glucose                  0.424        -0.442 -0.229 -0.174 -0.697  0.173  0.143
BloodPressure            0.377  0.172  0.305 -0.111 -0.762  0.275         0.247
SkinThickness            0.397 -0.309  0.398         0.421 -0.119 -0.489  0.397
Insulin                  0.307 -0.236 -0.574 -0.319  0.148  0.622
BMI                      0.402 -0.398  0.381                       0.502 -0.525
DiabetesPedigreeFunction 0.157 -0.273 -0.270  0.887 -0.170
Age                      0.386  0.518         0.142                -0.451 -0.583

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.125  0.125  0.125  0.125  0.125  0.125  0.125  0.125
```

Moreover as observed previously from the correlation plot, we can observe that the most correlated variables are also represented by the closest arrows in the biplot (Pregnacies-Age, BMI-SkinThickness...)

3.2 K-Means Clustering

3.2.1 K-Means Clustering: theoretical background

K-Means is a method for clustering that separates data in K distinct, non-overlapping clusters. This type of clustering is based on the concept that within-cluster variation must be as small as possible, which means that the difference between the observations in a cluster must be minimal.

One of the most used within-cluster variation used is the squared Euclidean distance; it measures the pairwise squared Euclidean distance between observations in the kth cluster, divided by the total number of observations in the kth cluster. To obtain a good clustering, squared Euclidean distance must be minimized.

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}.$$

The starting point of k-Means algorithm is a random assignment of the observations to a cluster. Then for each of this cluster is calculated the centroid (a vector containing the means of each observation). Finally, observations are assigned to the cluster whose centroid is closest.

This algorithm implies that the number of the clusters must be given as input to the algorithm, and that since the algorithm starts randomly, it will reach a local (and not global) optimum.

To find the best number of clusters that partition the observation, it can be used the Elbow Method, that computes the sum of squared distances from each point to its centroid with respect to the number of clusters identified.

3.2.2 K-Means Clustering: application on dataset

Since this section of the report is dedicated to unsupervised methods, it will be used a generic approach to the clustering. It is supposed that the number of outcomes (2 outcomes:diabetic, non diabetic) is unknown. Hence it is applied the Elbow Method to discover the best number of clusters that minimize the sum of squared distances from the observations to centroids (distortion).

Elbow method gives as output the following plot:

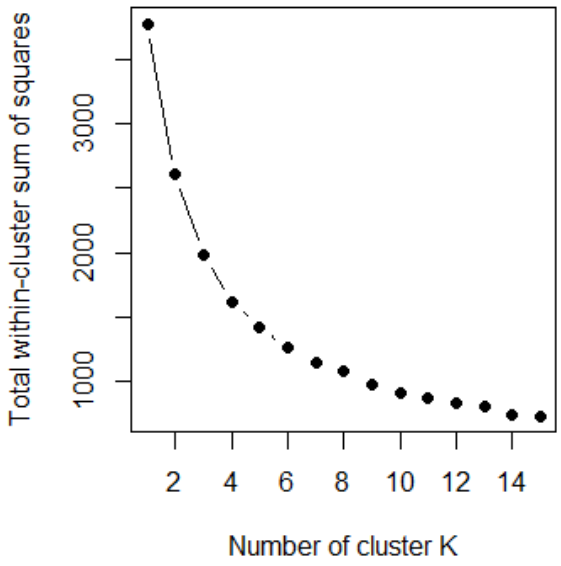


Figure 3.5: Elbow Plot: method to select the optimal number of clusters by fitting the model with a range of values for K

The "elbow" of the plot can be considered at K = 4.

Implementing the 4-clustering using the principal components previously calculated, the following 3D cluster representation is obtained. From this clustering, groups are not clearly separated each other and they are quite close each other.

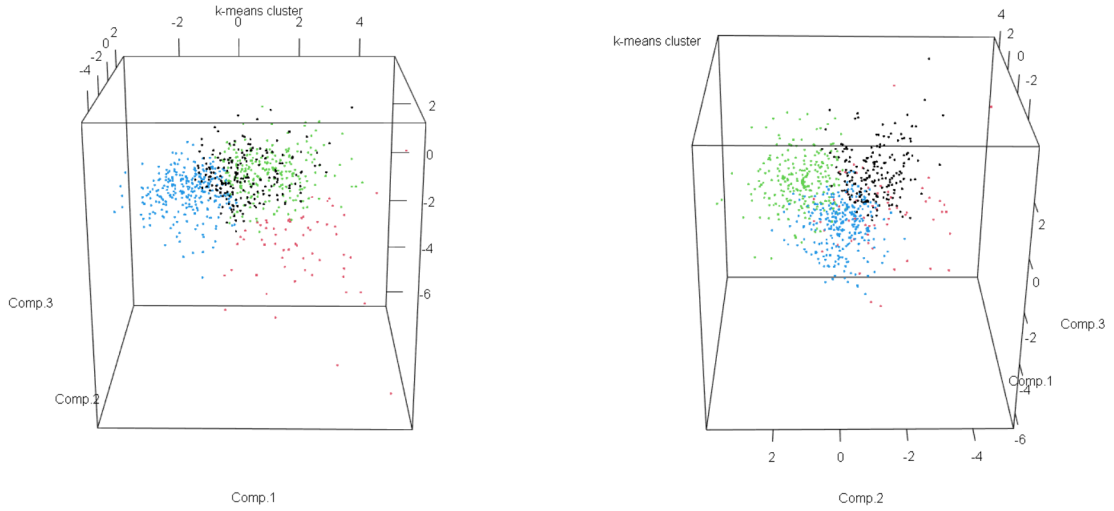


Figure 3.6: Visualization of the clustering with k = 4

It is possible now to take a step forward towards the supervised learning, by supposing to know the number of actual categories of the dataset (diabetic, non diabetic) and introducing the column Outcome of the dataset.

It could be interesting to observe if an unsupervised k-means clustering with k = 2 would be able to distinguish the healthy patients to the unhealthy ones.

It is implemented the 2-clustering, obtaining the following representation, that is compared to the actual classification of the dataset.

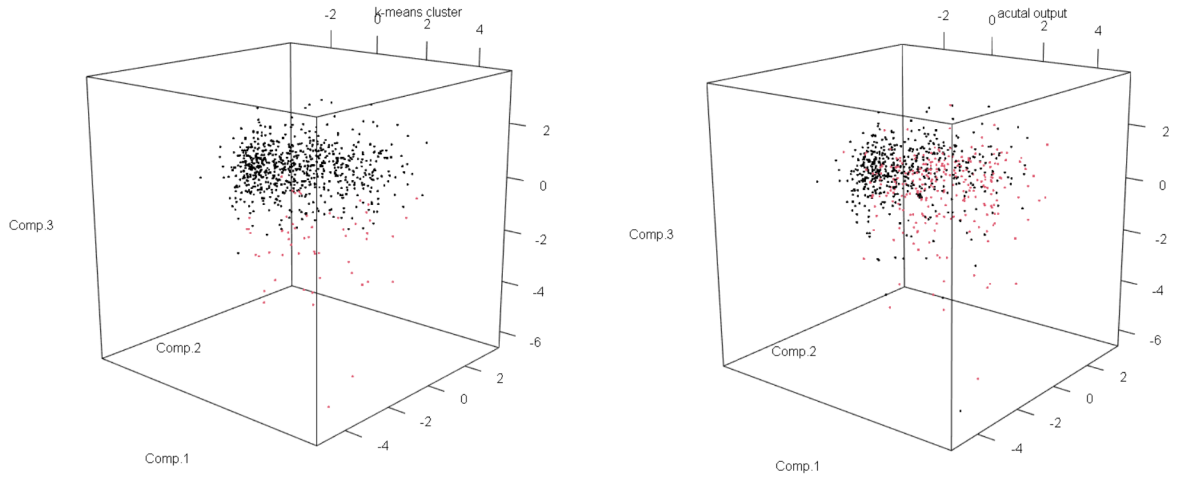


Figure 3.7: Clustering with k = 2 (on the left), actual binary clusters (on the right)

The confusion matrix shows how the clustering classifies correctly the healthy patient, but it is almost unable to identify the people with diabetes.

```
        result
cluster   1   2
      1 473 239
      2  27  29
```

This could happen because in order to obtain a graphical representation data, PCA used only the 61% of the variance of the data. Moreover from the Elbow Analysis it is observed that 2 clusters correspond to an high level of distortion of the observation.

Chapter 4

Supervised Learning: Logistic Regression and Classification Trees

4.1 Multiple Logistic Regression

4.1.1 Multiple Logistic Regression: theoretical background

In order to model the probability of an observation to belong to a category, relying on multiple predictors, it is used the logistic function:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

in which the coefficients β are estimated using the maximum likelihood method. MLR method looks for the values of beta that plugged in the p(X) model yields a number close to 0 for patient without diabetes and close to 1 for patient with diabetes. Likelihood function takes the form:

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

After the estimation of the coefficient, it is possible to calculate the probability of a patient to be diabetic by substituting to β the coefficients estimated and to X the values of predictors.

4.1.2 Multiple Logistic Regression: application on dataset

Once that dataset has been split in training and test set as shown in chapter 2.1.2, it is possible to define a Logistic Regression model and fit it to training set and then make prediction on the test data.

Fig. 4.1 shows the coefficients of the logistic regression that predicts the probability of having diabetes using all the variables available in the dataset, associated to the p-value.

```
Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)               -5.2757     0.6144  -8.587  < 2e-16 ***
Pregnacies                 1.7633     0.6567   2.685  0.00725 **
Glucose                    5.6573     0.6993   8.090 5.97e-16 ***
BloodPressure             -0.6382     0.8900  -0.717  0.47336
SkinThickness              0.8878     1.4516   0.612  0.54079
Insulin                   -1.4249     1.1136  -1.280  0.20068
BMI                        4.5675     1.0552   4.329 1.50e-05 ***
DiabetesPedigreeFunction   1.5661     0.8003   1.957  0.05036 .
Age                        0.7966     0.6835   1.166  0.24379
```

Figure 4.1: Logistic regression coefficients

The variables with a statistical significant p-value (less than 0.05) are pregnancies, glucose and BMI. The coefficients related to them are all positive, that means that when the values of these predictors increase, also the probability of incurring in diabetes increases.

Logistic regression on the training set returns the following confusion matrix, associated with a value of accuracy of: 0.777.

```
          actual
predicted   0   1
        0 310  79
        1  41 108
```

Once the model is fit on the training set it is possible to make prediction and evaluate the performances on the test set.

The confusion matrix resulting from the application on the test is the following, associated with a value of accuracy of 0.700.

```
          actual
predicted  0  1
        0 95 15
        1 54 66
```

The plots in figure 4.2 represent the observation for training and test set and their "position" in the fitted logistic curve. The blue points represent the observations whose actual outcome is 1 and the black one the observation whose actual outcome is 0. If a blue point is above 0.5 it means that it is correctly predicted and if a black point is under 0.5 means it is correctly predicted.

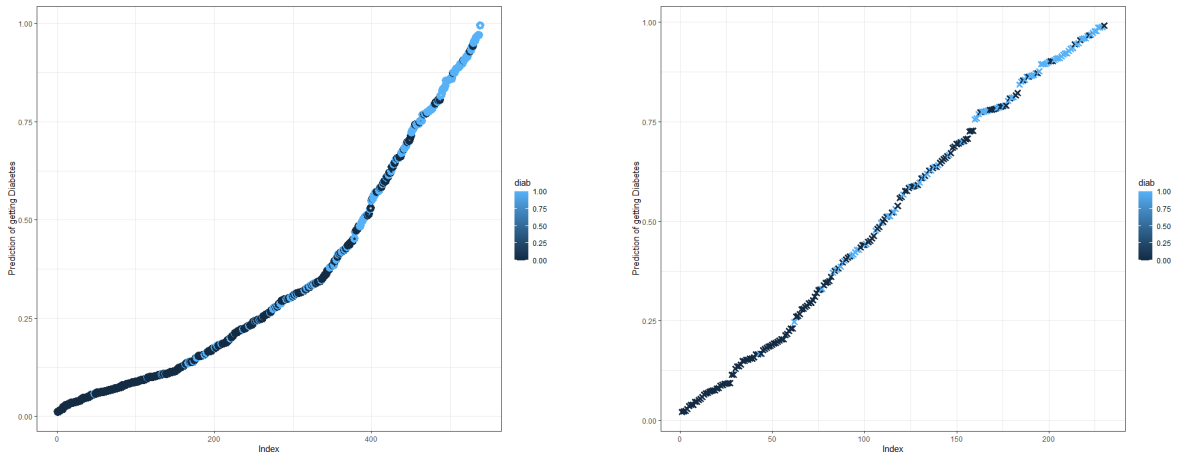


Figure 4.2: Observation on the fitted Logistic Curve: training set (left), test set (right

It is possible to build a new logistic regression considering only the statistically significant variables, to observe if a simplification of the model leads to a reducing in overfitting and to best performances. The variables considered are Pregnancies, Glucose and BMI and the coefficients obtained are in Fig. 4.3

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.1855     0.4421 -11.730  < 2e-16 ***
Pregnacies    2.0847     0.5484   3.801 0.000144 ***
Glucose       5.4644     0.6125   8.922  < 2e-16 ***
BMI           4.5860     0.8675   5.287 1.25e-07 ***
```

Figure 4.3: Coefficients estimated using three only variables

This new logistic regression on the training set returns the following confusion matrix, associated with a value of accuracy of: 0.775.

```
          actual
predicted   0   1
        0 311  81
        1  40 106
```

The confusion matrix resulting from the application on the test is the following, related to a value of accuracy of 0.7035.

```
          actual
predicted   0   1
        0 103  20
        1  46  61
```

It is observable a slight improvement in the test accuracy with respect to the complete-variables model and a reduction in overfitting.

4.2 Classification tree

4.2.1 Classification tree: theoretical background

Classification tree is a tree-based method that predicts the class of an observation basing on the most commonly occurring class of training observation in the region to which it belongs. The criteria that can be used to do the binary split in the tree are the classification error rate, Gini index, entropy.

In this specific case, it will be used the Gini index, a function that takes on a small value if all the proportion of training observation in the m-th region from the k-th class are close to zero or one.

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

To prevent overfitting, it could be useful to reduce the complexity of the tree by applying "Tree Pruning", that allows to obtain a subtree, reducing variance. The optimal subtree is the one that leads to the lowest test error rate.

Subtrees are obtained through Cost Complexity Pruning, a method that considers a sequence of trees indexed by a non negative tuning parameter alpha. This parameter controls the trade-off between the complexity and the fitting to training data of the subtree. With alpha = 0, the tree will be equal to the unpruned tree, with and high alpha the tree will tend to reduce dimensionality. For each value of alpha there correspond a subtree T (included in the unpruned tree T0) such that:

$$\sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

The value of alpha is selected using cross-validation.

4.2.2 Classification tree: application on dataset

First, it is fit an unpruned classification tree, that returns a 12 terminal-nodes-tree using the variables Glucose, BMI, AGE, Skin Thickness and DBF.

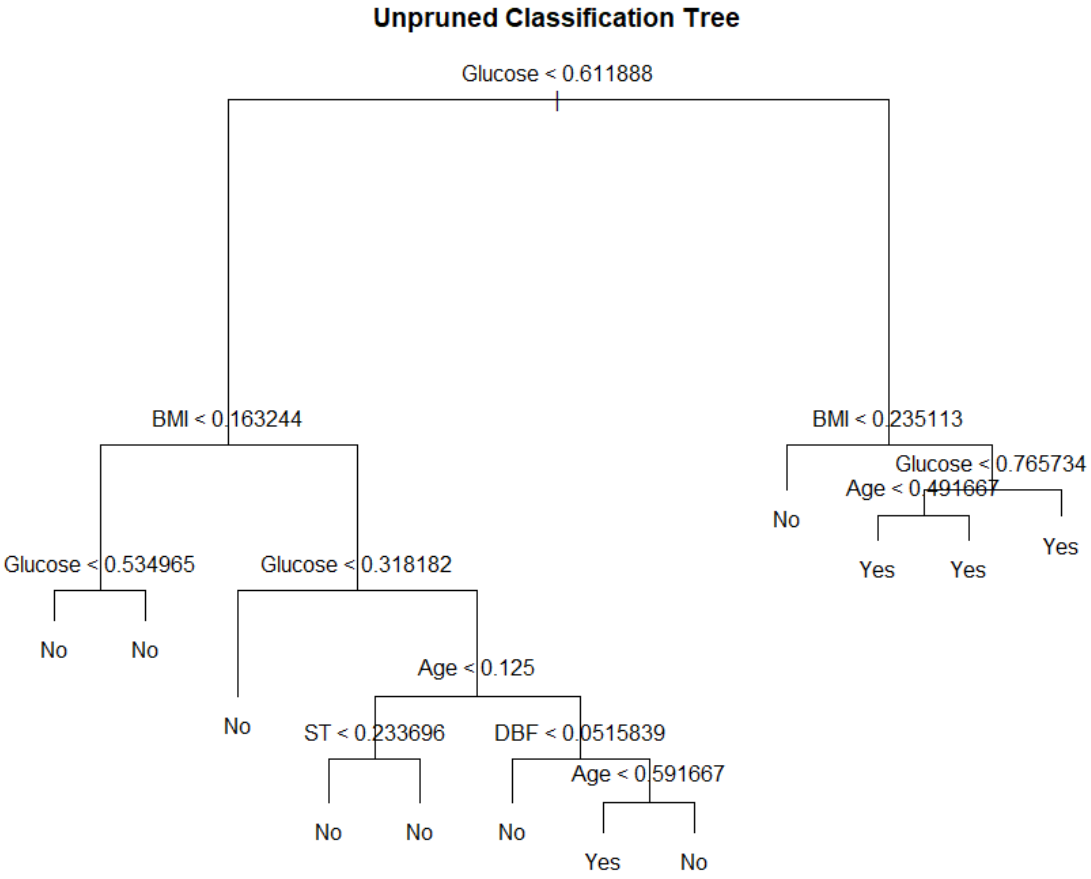


Figure 4.4: Single Unpruned tree

This tree is related to the following training confusion matrix and a training accuracy = 0.818.

predicted \ actual	No	Yes
No	303	50
Yes	48	137

To evaluate the goodness of the work of the tree, the model is applied on the test set obtaining the following confusion matrix and a test accuracy = 0.726, showing overfitting.

```
          actual
predicted  No Yes
      No  109  23
      Yes  40  58
```

To reduce the variance error, it is now used cross-validation to find subtrees pruned from the original one.

Plot in Fig.4.5 shows that the best subtree obtained from cross-validation is the one with 7 terminal nodes, that corresponds to a misclassification of about 0.24.

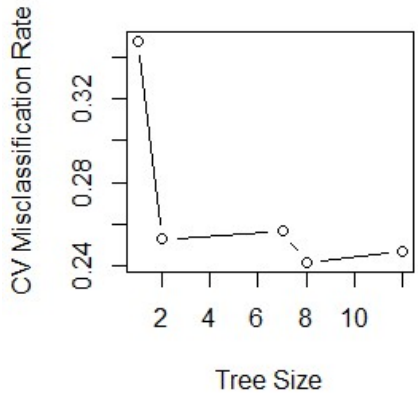


Figure 4.5: Cross Validation applied to find the best number of terminal nodes

It is possible also to compare the plot of the CV error respect to the training and test error, as shown in fig.4.6.

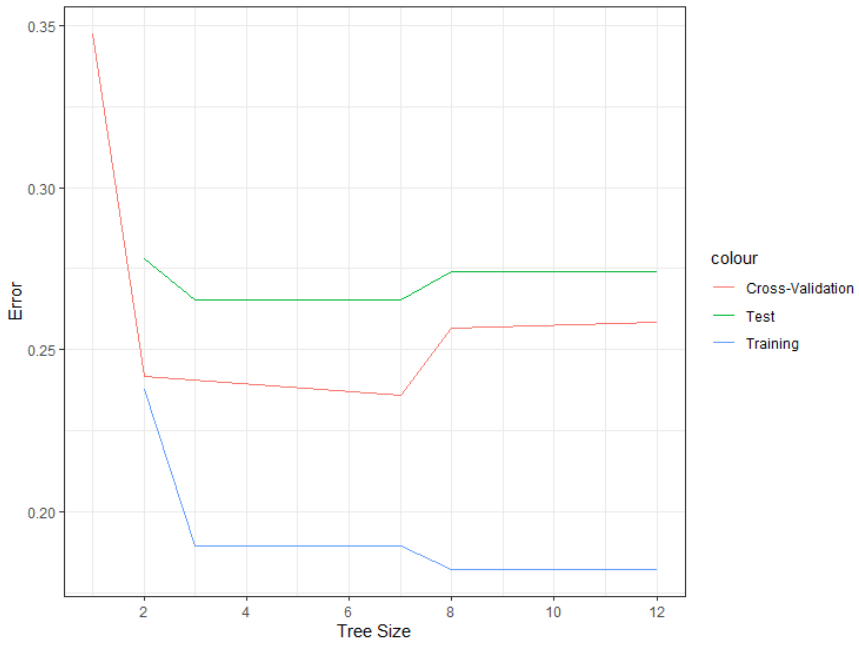
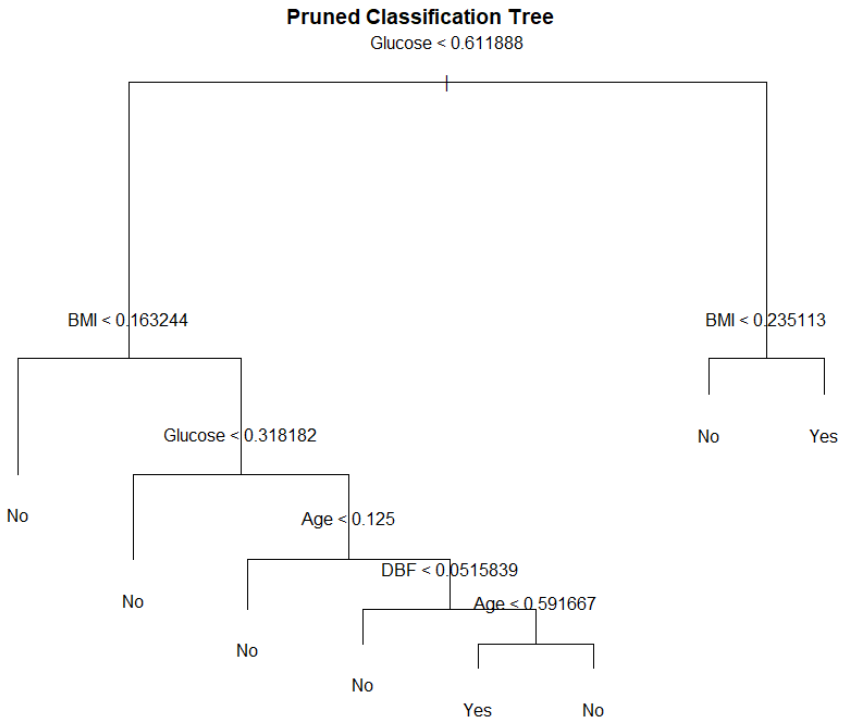


Figure 4.6: Comparing of training, test and cross validation errors

Hence the original tree is pruned and the following subtree with 7 terminal nodes is obtained:



This tree is related to the following training confusion matrix and the training accuracy is equal to 0.810.

	actual	
predicted	No	Yes
No	290	41
Yes	61	146

The application of this new tree on the test set gives as result the following confusion matrix, with a test accuracy of 0.735.

	actual	
predicted	No	Yes
No	104	16
Yes	45	65

It can be observed that the test performance of the test accuracy slightly improves and there is a reduction in overfitting.

4.2.3 Bagging and Random Forest: theoretical background

Bagging is a method used for reducing the variance of classification trees by averaging the set of observations, belonging to different training sets taken from the population. Since usually it is possible to access to only one training set, it is used the bootstrap techniques, that takes repeated samples from the single training data available, obtaining B training dataset. In the case of classification, for a given test observation it is possible to record the class predicted by each of the B trees and take the majority vote. The overall prediction is the most commonly occurring class among the B predictions.

Similarly, Random Forest methods build decision trees considering in each split of the trees a random sample of m predictors from the full set of p predictors.

Usually it is chosen a value of m = $\sqrt{p}$

This methods allows to decorrelate the different trees, that will not be prone to use the stronger predictors in the top split (leading to high correlation between trees) like in bagging, but will consider also the the weaker predictors, making the trees uncorrelated. Consequently, averaging on uncorrelated tree leads to a lower variance.

4.2.4 Bagging and Random Forest: application on dataset

The plot in fig.4.7 shows the results from bagging trees on the Diabetes dataset. The error is plotted as a function of the number of trees constructed using bootstrapped training data sets. It can be observed that a good value of the number of trees that can be used is 50, that corresponds to the point where error settles down. Anyway, improving the number of trees used for Boosting does not increase over-fitting. For this reason it is used the default value of the function RandomForest (K=500).

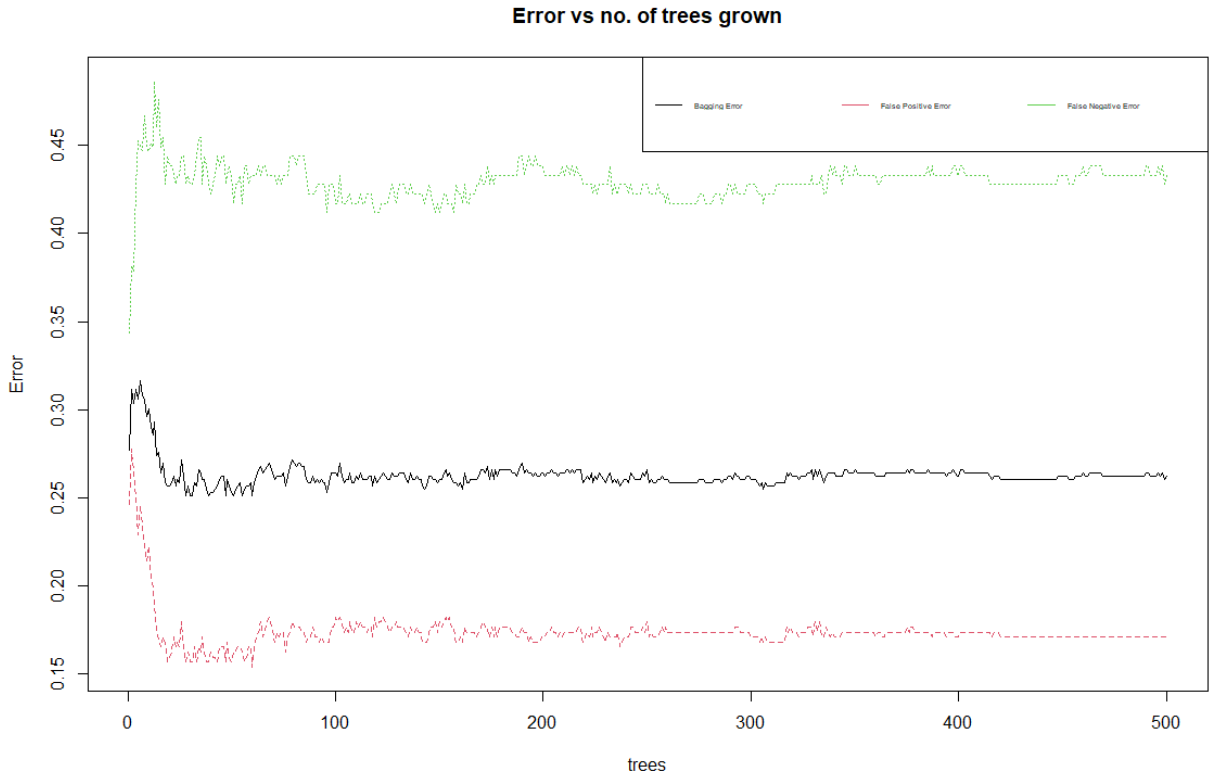


Figure 4.7: Errors related to misclassification, false positive and false negative applying Bagging technique

Applying boosting method with 500 trees on the test it is obtained the following confusion matrix, associated to a level of accuracy equal to 0.735.

	actual	
predicted	No	Yes
No	107	19
Yes	42	62

To give an interpretation of how bagging works, it can be useful to understand what are the most important variables considered in this model. This means to identify the amount of decreasing in Gini

index related to splits over a given predictor, averaged over all B trees. From the following plot it can be observed how the three most important variables in the model to reduce Gini index are Glucose, BMI and DiabetesPedigreeFunction.

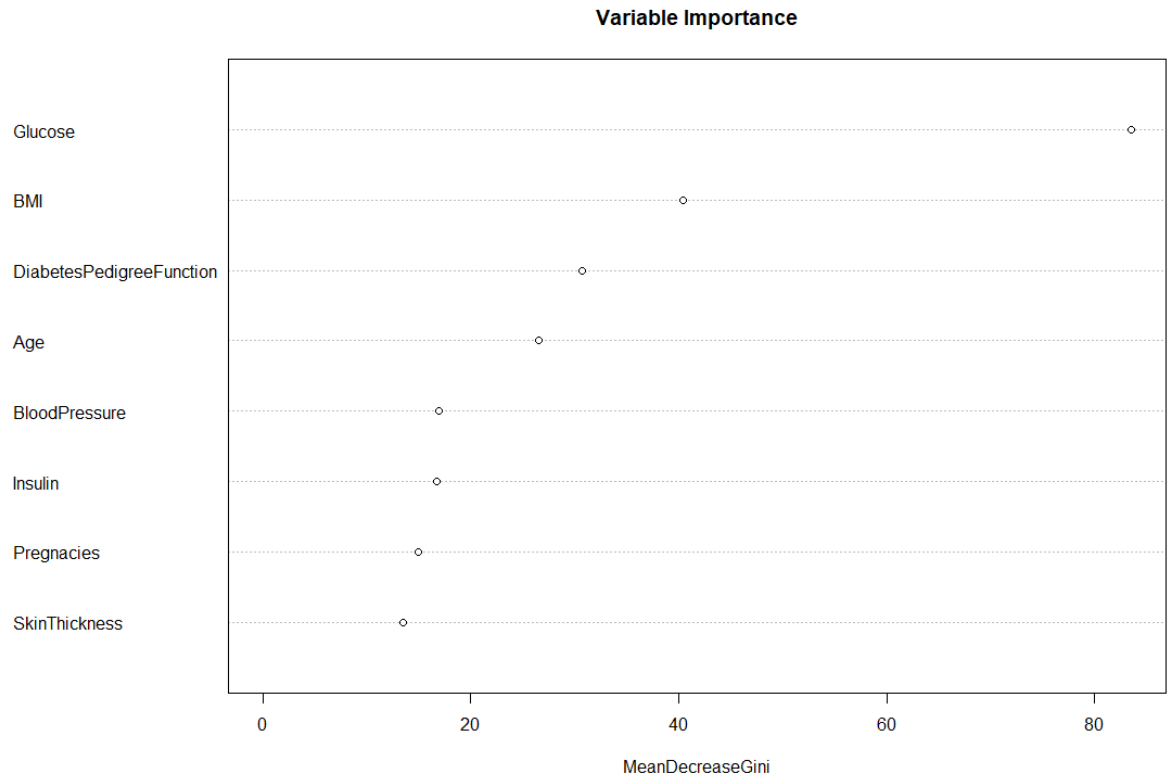


Figure 4.8: Importance of variables applying bagging method

Now Random Forest is applied, considering a subset of predictors m = 3 (about $\sqrt{8}$). Also in this case it is possible to plot the error as a function of the number of trees used in bootstrap. Error seems to stabilize after a value of K = 100. Since an increase in K does not imply an increase in overfitting, it is used the default value of the RandomForest function (K=500) to fit the model.

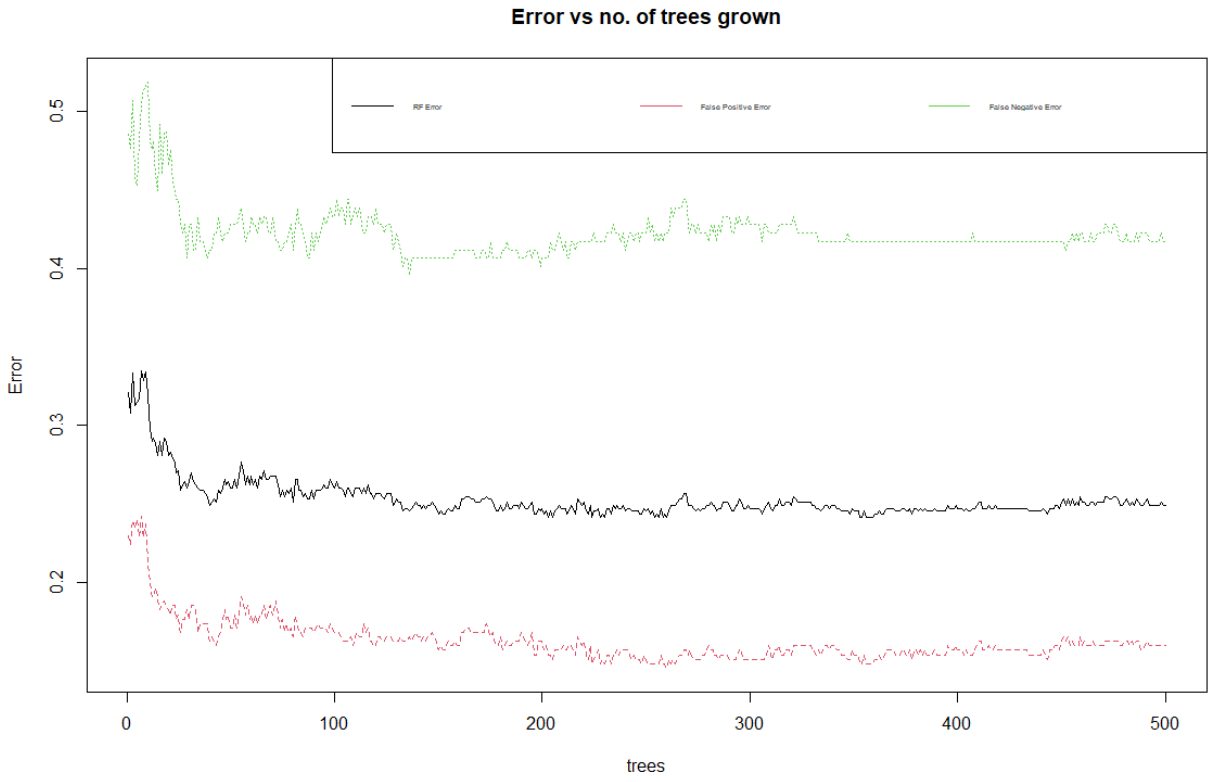


Figure 4.9: Errors related to misclassification, false positive and false negative applying Random Forest technique

The confusion matrix of the Random Forest applied on the test set is the following, related to a test accuracy equal to 0.752, that has improved with respect to the bagging test accuracy.

	actual	
predicted	No	Yes
No	106	14
Yes	43	67

Likewise for bagging, it is possible to identify the variable importance plot in fig.4.10. Also in this case the most important variables are Glucose, BMI and DiabetesPedigreeFunction.

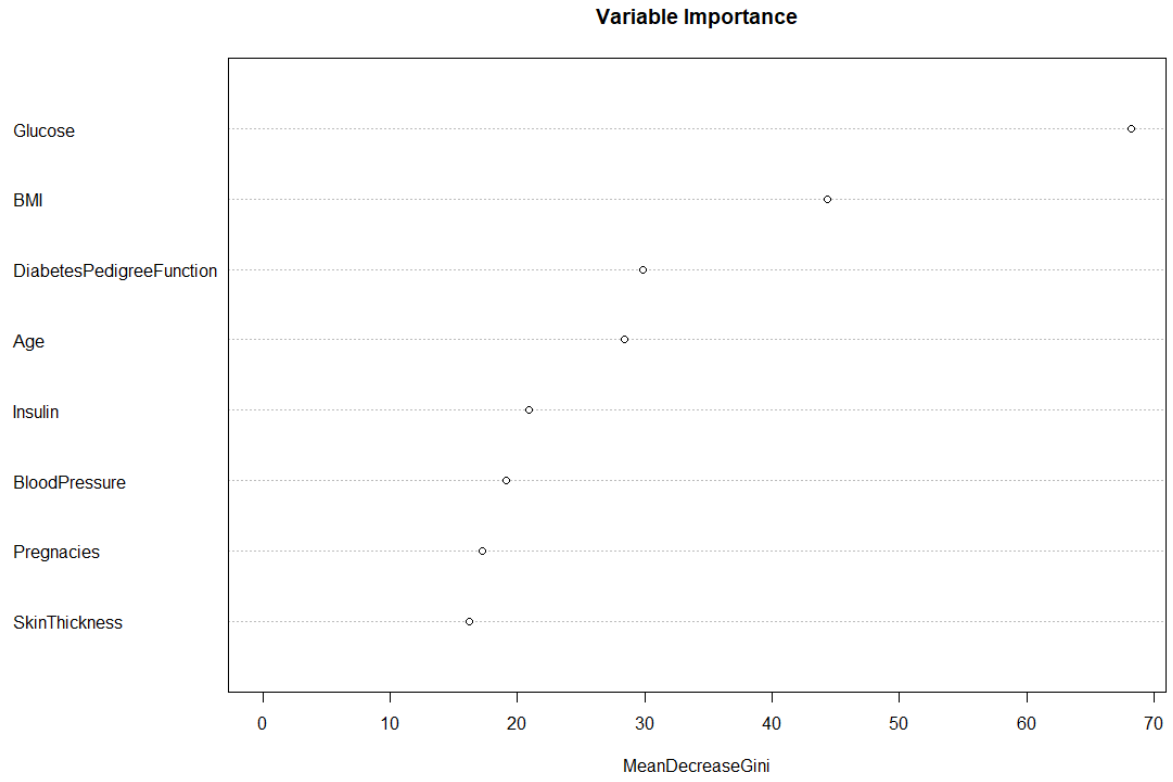


Figure 4.10: Importance of variables applying random forest method

4.2.5 Boosting: theoretical background

The last model applied is Boosting. In this approach, trees are grown sequentially using information from previously grown trees, without involving bootstrap sampling. Decision trees are grown sequentially; each tree is grown using information from previously trees. The parameters involved in Boosting are:

- B: number of trees to use.
- λ: shrinkage parameter that controls the rate of boosting learning. It controls how fast the method learns.
- d: interaction depth. It represents the number of splits in each tree.

4.2.6 Boosting: application on the dataset

The tuning grid used for tuning is set up as follows:

- Parameter B is chosen in a range between 500:3000;
- λ (shrinkage) is chosen between three standard values (0.001, 0.01, 0.1);
- parameter d is chosen in a range between 1:5;
- n.minobsinnode represents the minimum number of observations in a node of the tree and is set to 10.

It can be observed in the plot in fig.4.11 that summarizes the results:

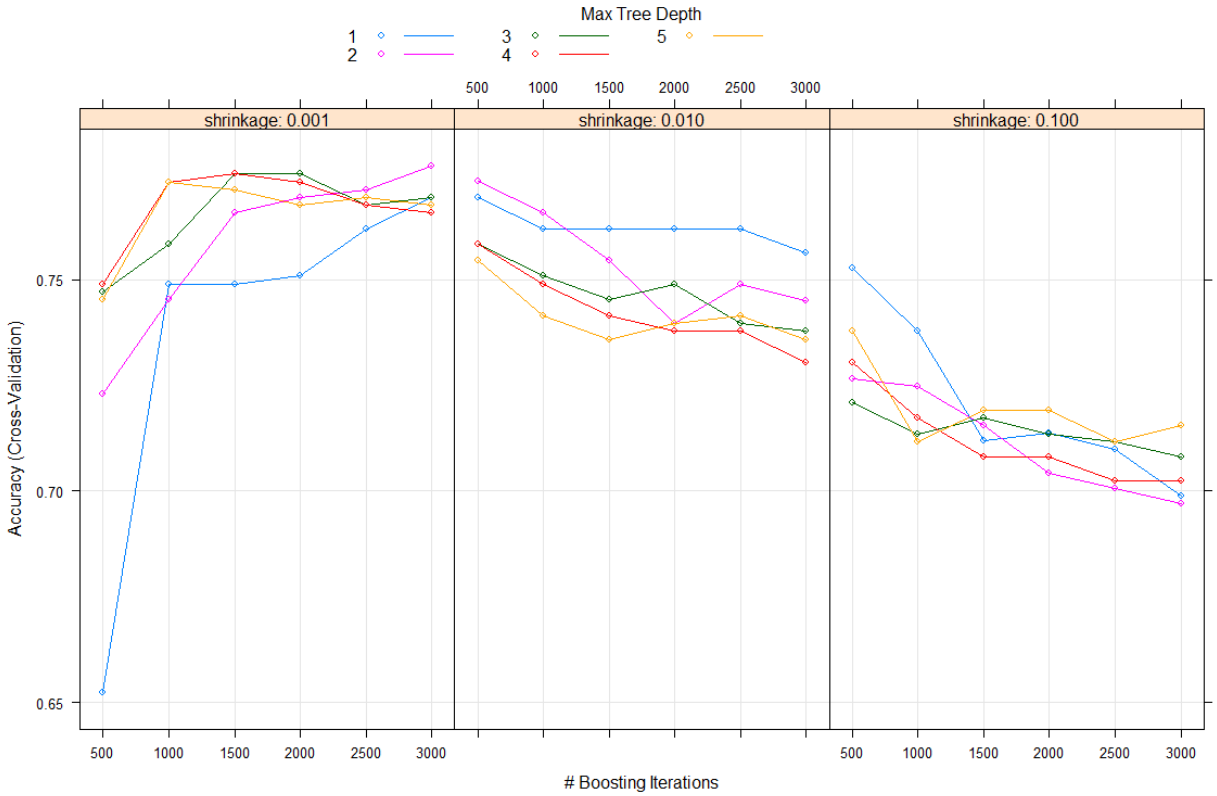


Figure 4.11: Summary of the results of tuning Boosting method

The best combination of parameters is:

```
n.trees interaction.depth shrinkage n.minobsinnode
   3000                 2     0.001             10
```

And applying Boosting with these parameters on the test test is obtained a value of accuracy equal to 0.761.

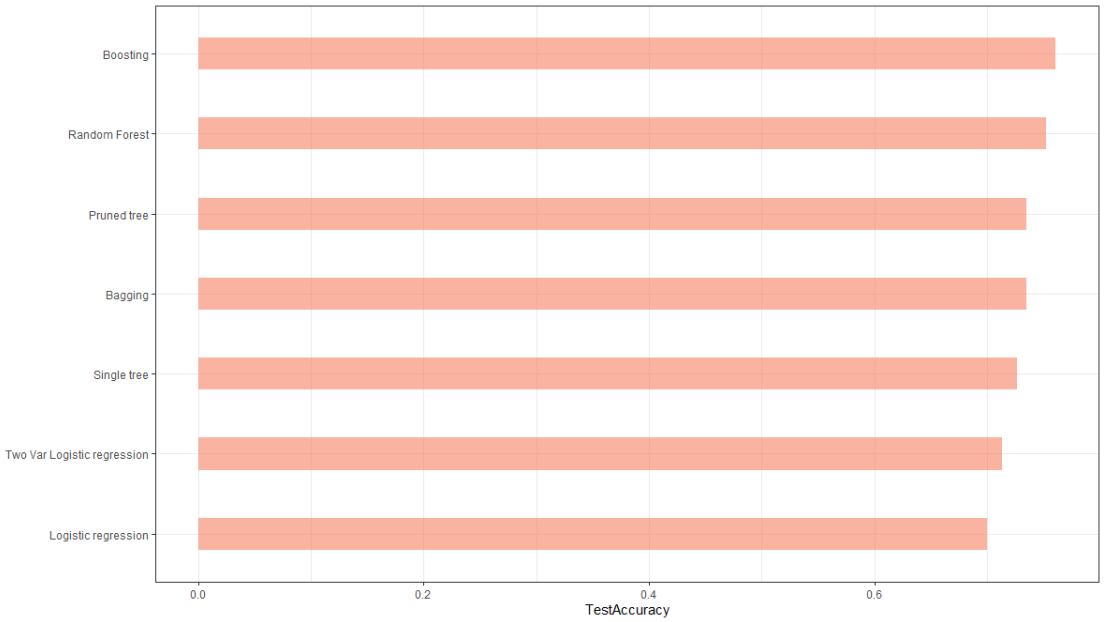
Chapter 5

Conclusion

Unsupervised learning has been useful to conduct an exploratory data analysis, to obtain an informative way to visualize data through PCA and to discover natural subgroups among the observations.

It has been achieved the goal of obtain a representation in three dimensions of the original eight dimensional data and, by applying elbow method, it has been found that the best number of clusters identified in the data is equal to four subgroups. Introducing the outcome variables, approaching in this way to supervised techniques, it has been discovered that clustering is not able to discriminate correctly diabetic and non diabetic people.

Regarding the supervised learning techniques applied, the whole methods have given as result a test accuracy of at least 70%. The next histogram gives a ranking of the models accuracy in prediction of potential diabetic/non diabetic patients in 5 years:



In this case, according to test accuracy, Boosting method is the best model to make prediction on this dataset, with a test accuracy equal to 0.761.