FACULTY OF COMPUTING
& INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY

FCIT
KAU

كلية الحاسبات
وتقنية المعلومات
جامعة الملك عبدالعزيز

Artificial Intelligence

Final project

[CPCS 331]

Instructor:

Dr. Ali Almashaike

Team members:

ID	Name
1846612	Omar Zainalabdeen
1845792	Adel Alharthi
1741886	Abdullah Alqhtani
1846409	Khalid Alghamdi
1845646	Albaraa Baatiyyah

Decision Tree and Naïve Bayes classifiers

This is the Dataset of diabetes, downloaded from the website of the kaggle, taken from the hospital Frankfurt, Germany. The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

There are total 768 observations and with nine columns. Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, and Outcome.The dataframe of the data set is shown below

```
> str(diab)
'data.frame':	768 obs. of  9 variables:
 $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
 $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
 $ BloodPressure           : int  72 66 64 66 40 74 50 68 70 96 ...
 $ SkinThickness           : int  35 29 26 23 35 24 32 32 45 32 ...
 $ Insulin                 : int  160 116 175 94 168 112 88 210 543 402 ...
 $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 29.8 ...
 $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
 $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
 $ Outcome                 : Factor w/ 2 levels "No","Yes": 2 1 2 1 2 1 2 1 2 2 ...
```

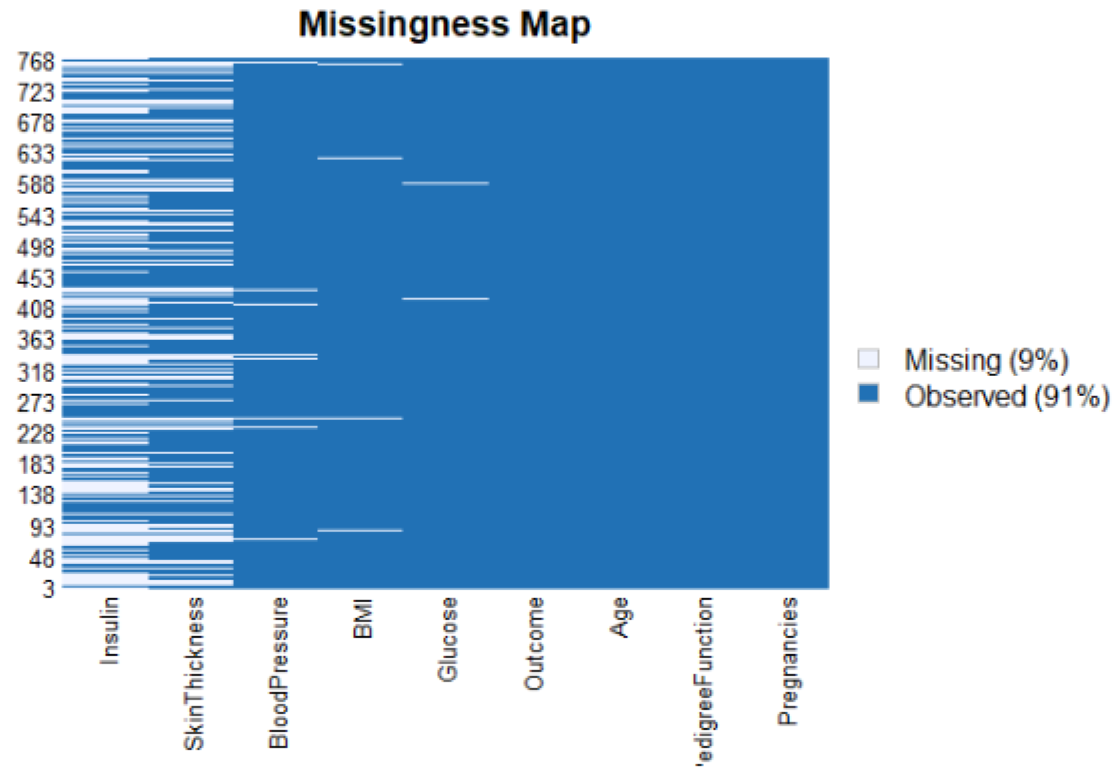
Data preprocessing

Before we study the data set let’s convert the output variable (‘Outcome’) into a categorical variable. This is necessary because our output will be in the form of 2 classes, True or False. Where true, will denote that a patient has diabetes, and false denotes that a person is diabetes free. First of all we transform the outcomes of the diabetes data set into Yes or No from 1 and 0 to describe as the patient have diabetes or not.

While analyzing the structure of the data set, we can see that the minimum values for Glucose, Bloodpressure, Skinthickness, Insulin, and BMI are all zero. This is not ideal since no one can have a value of zero for Glucose, blood pressure, etc. Therefore, such values are treated as missing observations.

In the below code snippet, we’re setting the zero values to NA’s:

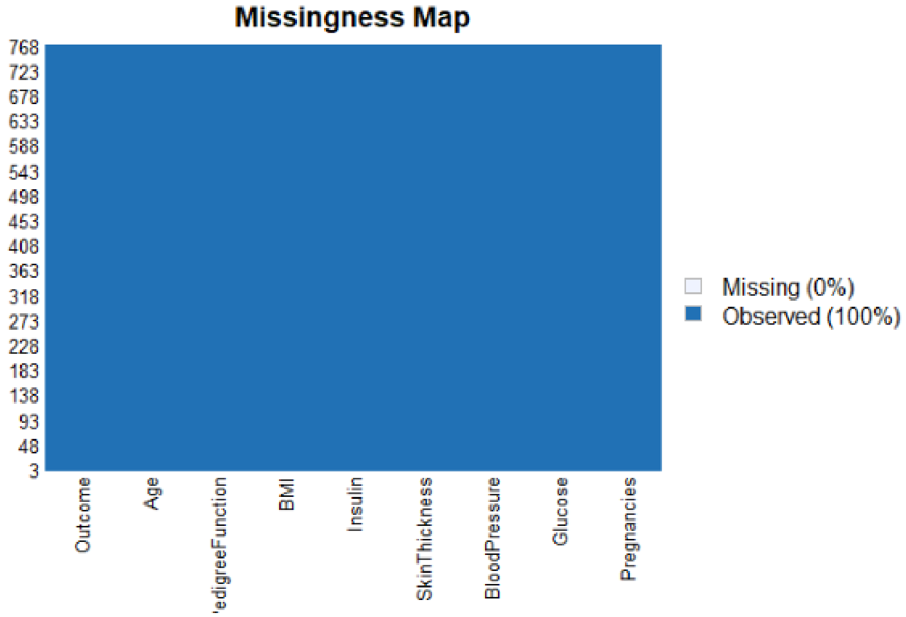
To check how many missing values we have now, let’s visualize the data:



The above illustrations show that our data set has plenty missing values and removing all of them will leave us with an even smaller data set, therefore, we can perform imputations by using the mice package in R.

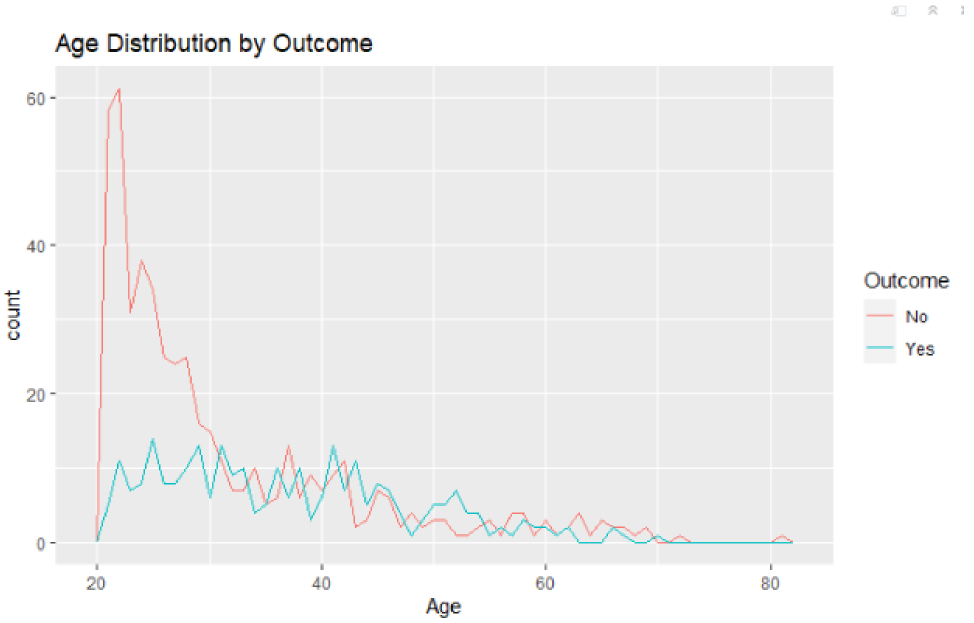
The output looks good, there is no missing data.

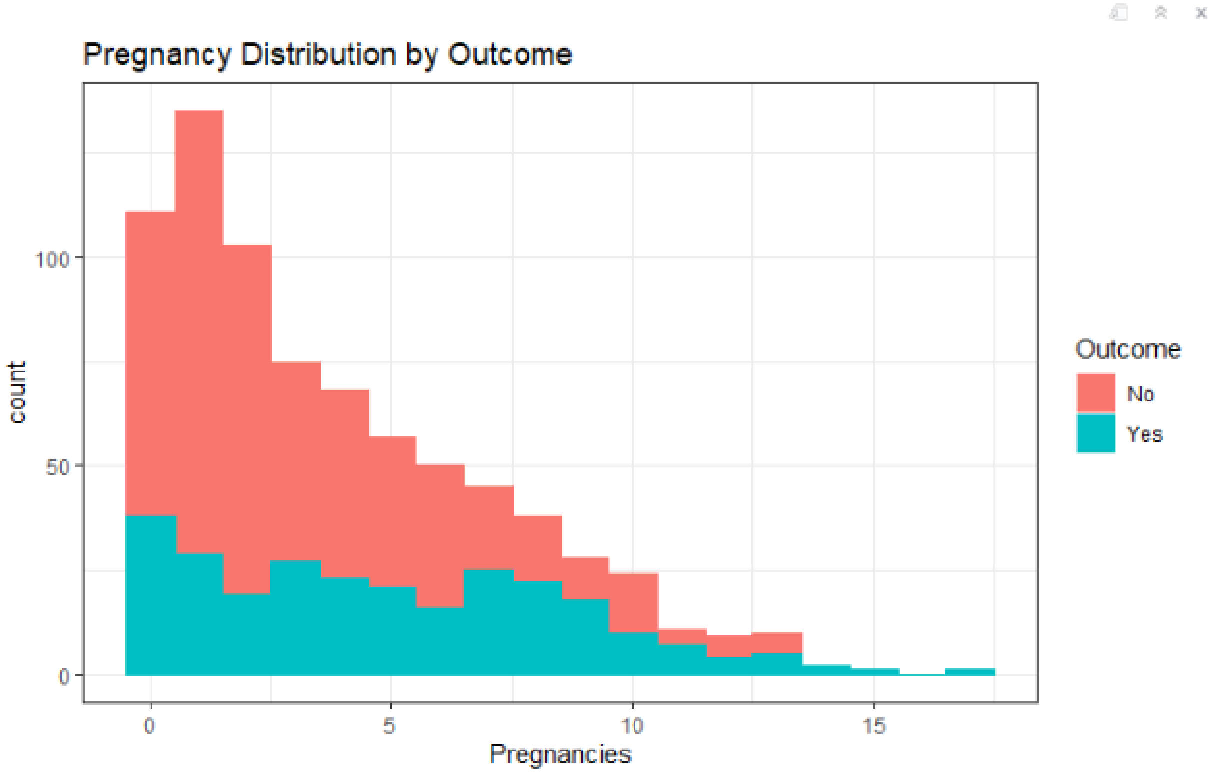
Now let's perform a couple of visualizations to take a better look at each variable, this stage is essential to understand the significance of each predictor variable.

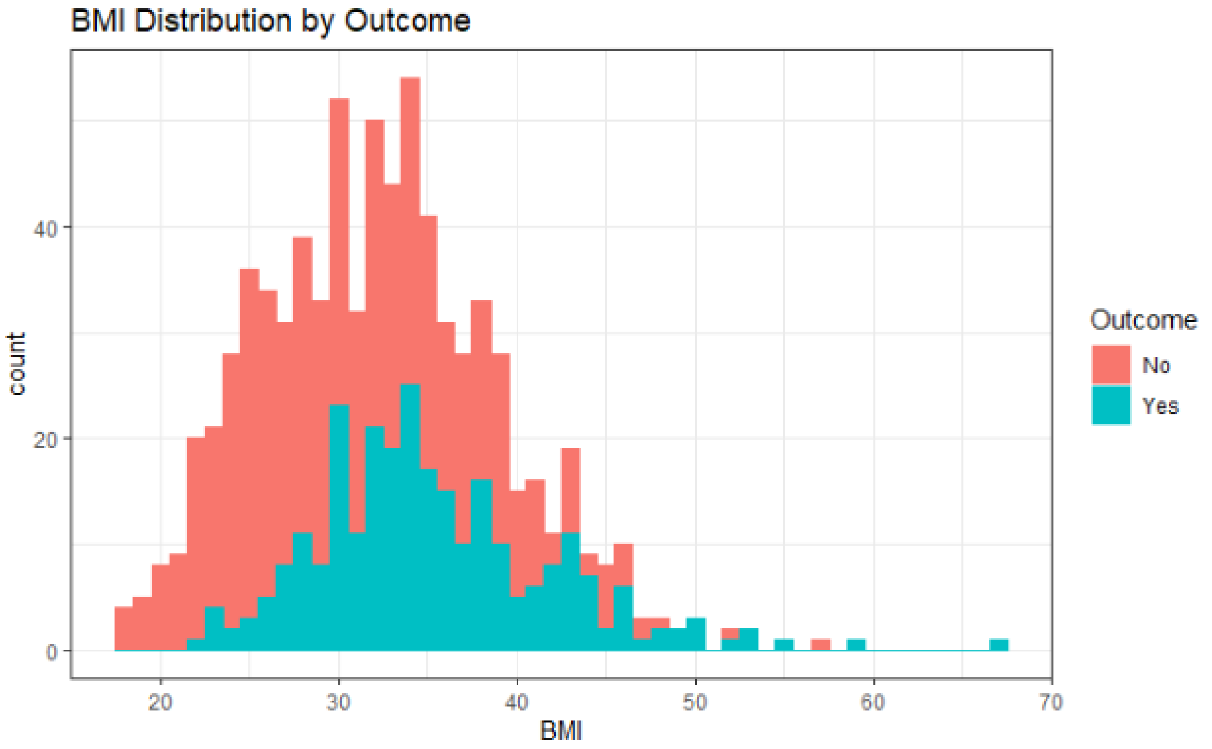


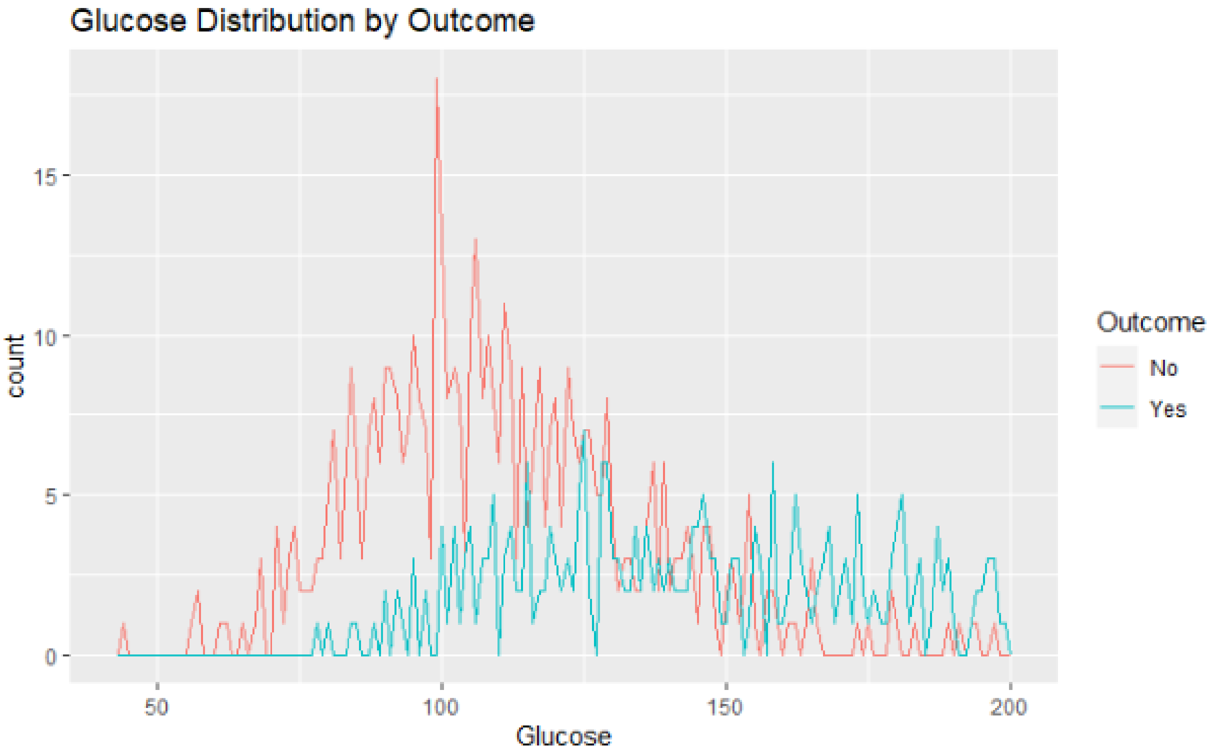
The output looks good, there is no missing data.

Now let’s perform a couple of visualizations to take a better look at each variable, this stage is essential to understand the significance of each predictor variable.

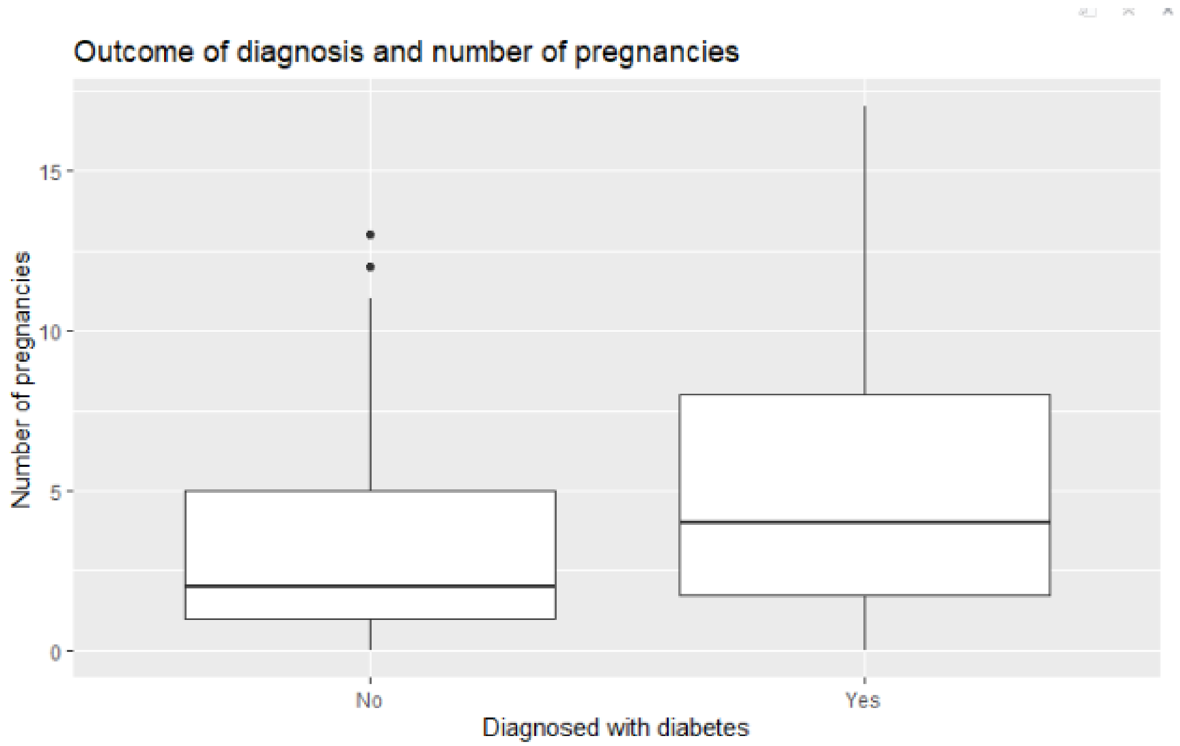




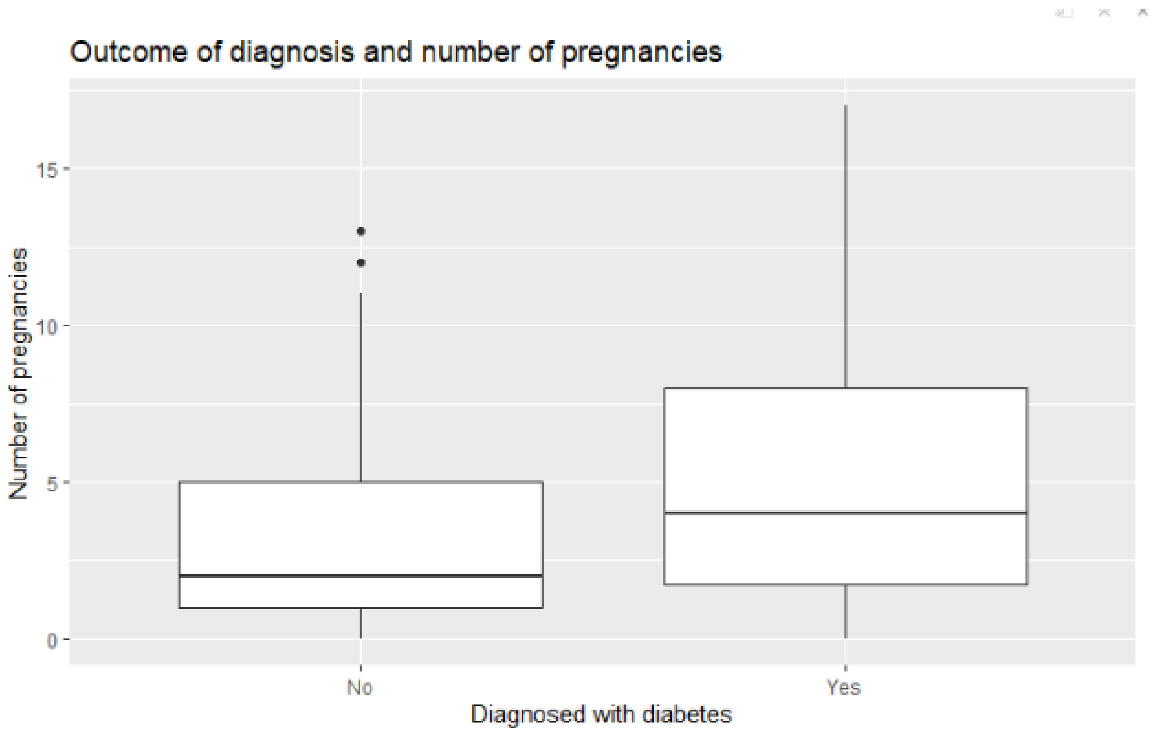




There is boxplot for the Outcome of diagnosis and number of pregnancies and Correlation of number of pregnancies and BMI.



obvious that the more are the numbers of pregnancies the higher is the positive diagnosis rate. It is also worth to note that the range of number pregnancies is quite big for those diagnosed with diabetes, compared to those who were not diagnosed.



There is an obvious correlation of high BMI and positive diagnosis of diabetes.

Classification

Data partitioning

This stage begins with a process called Data Splicing, wherein the data set is split into two parts:
Training set: This part of the data set is used to build and train the Machine Learning model.
Testing set: This part of the data set is used to evaluate the efficiency of the model.

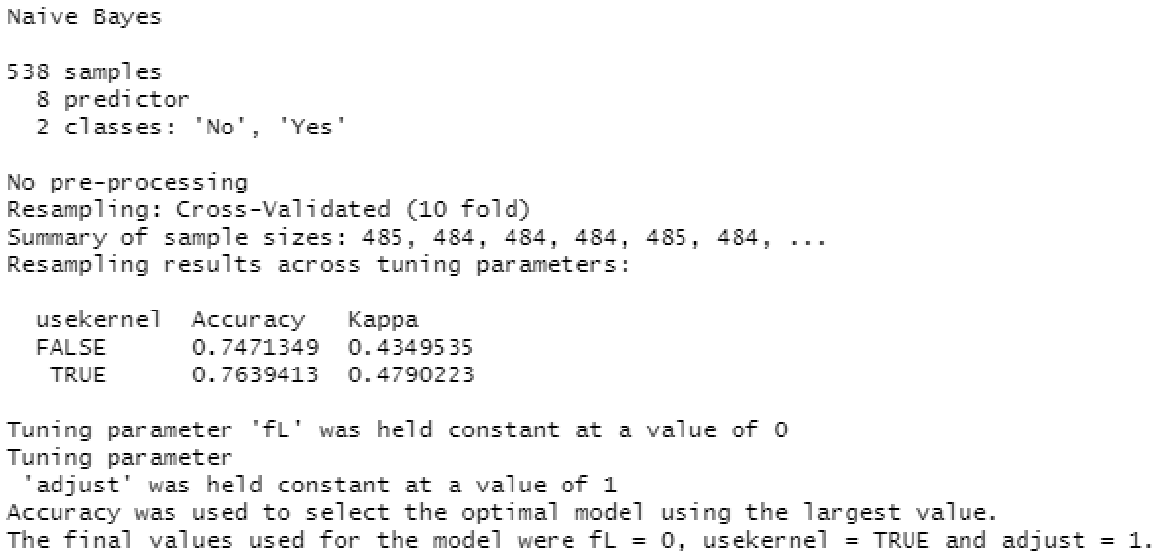
prop.table(table(diab$Outcome))*100
No   Yes
65.10417 34.89583

For comparing the outcome of the training and testing phase let’s create separate variables that store the value of the response variable:
create objects x which holds the predictor variables and y which holds the response variables.

Naïve Bayes

Now it’s time to load the e1071 package that holds the Naive Bayes function. This is an in-built function provided by R.

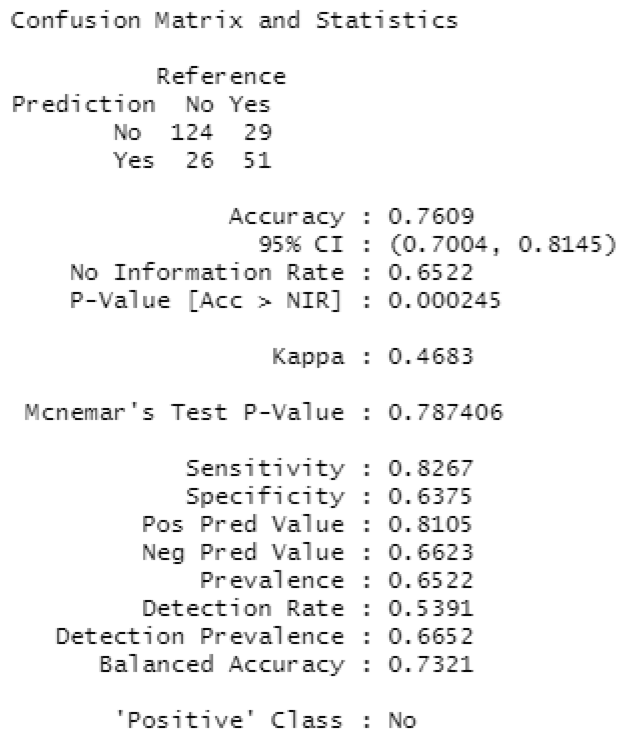
After loading the package, the below code snippet will create Naive Bayes model by using the training data set:



```
Naive Bayes

538 samples
  8 predictor
  2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 485, 484, 484, 484, 485, 484, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.7471349  0.4349535
   TRUE      0.7639413  0.4790223

Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter
 'adjust' was held constant at a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1.
```

We thus created a predictive model by using the Naive Bayes Classifier.

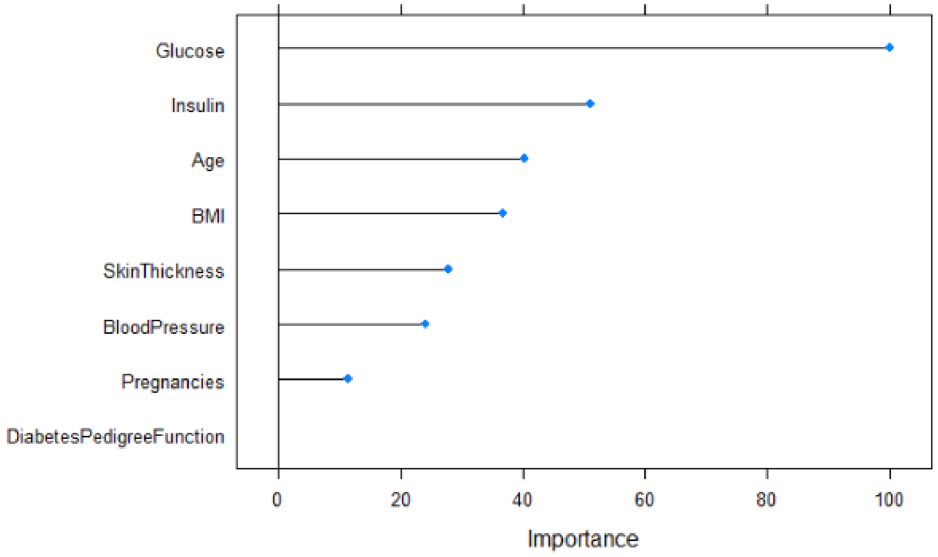
To check the efficiency of the model, we are now going to run the testing data set on the model, after which we will evaluate the accuracy of the model by using a Confusion matrix.



```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  124  29
       Yes  26  51

               Accuracy : 0.7609
                 95% CI : (0.7004, 0.8145)
    No Information Rate : 0.6522
    P-Value [Acc > NIR] : 0.000245

                  Kappa : 0.4683

 Mcnemar's Test P-Value : 0.787406

            Sensitivity : 0.8267
            Specificity : 0.6375
         Pos Pred Value : 0.8105
         Neg Pred Value : 0.6623
             Prevalence : 0.6522
         Detection Rate : 0.5391
   Detection Prevalence : 0.6652
      Balanced Accuracy : 0.7321

       'Positive' Class : No
```

The final output shows that we built a Naive Bayes classifier that can predict whether a person is diabetic or not, with an accuracy of approximately 76%.

To summaries the demo, let’s draw a plot that shows how each predictor variable is independently responsible for predicting the outcome.



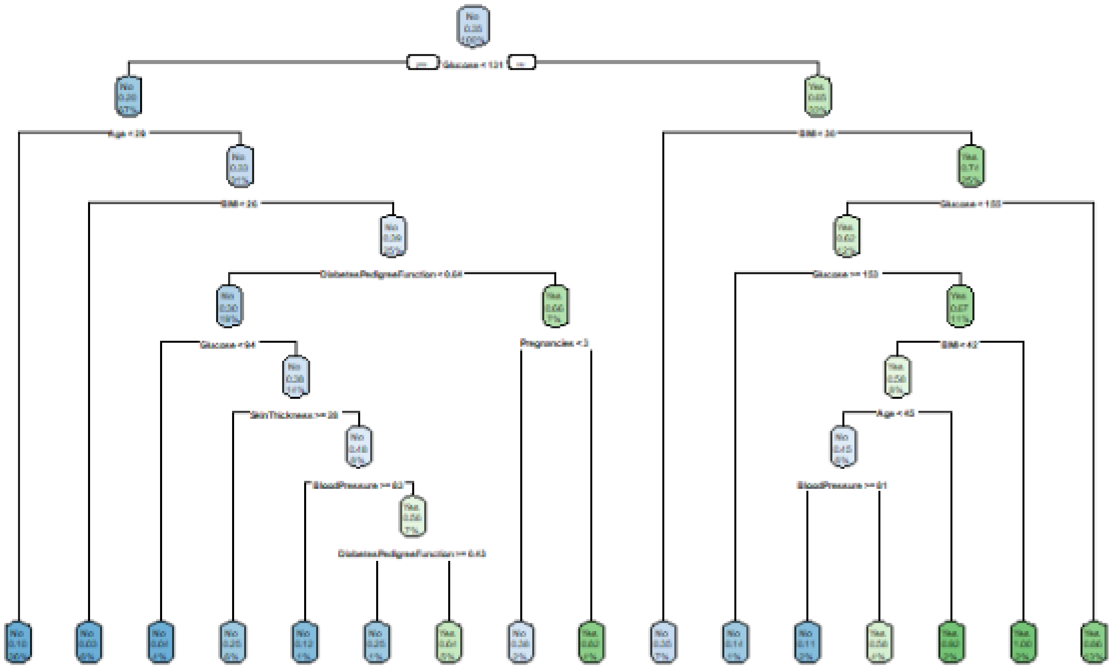
From the above illustration, it is clear that ‘Glucose’ is the most significant variable for predicting the outcome.

Decision Tree

Decision Trees are versatile Machine Learning algorithm that can perform both classification and regression tasks. They are very powerful algorithms, capable of fitting complex datasets. Besides, decision trees are fundamental components of random forests, which are among the most potent Machine Learning algorithms available today.

First step is to distribute the data set into training and the testing with ration of 70 and 30.

Then the model is run on the data set for the decision tree and here is the tree below



Below are the rules which are derived from the decision tree.

Rule number: 59 [Outcome=Yes cover=13 (2%) prob=1.00]
Glucose>=130.5
BMI>=29.6
Glucose< 154.5
Glucose< 152.5
BMI>=41.55

Rule number: 117 [Outcome=Yes cover=12 (2%) prob=0.92]
Glucose>=130.5
BMI>=29.6
Glucose< 154.5
Glucose< 152.5

```
 BMI< 41.55
 Age>=44.5

Rule number: 15 [Outcome=Yes cover=71 (13%) prob=0.86]
 Glucose>=130.5
 BMI>=29.6
 Glucose>=154.5

Rule number: 47 [Outcome=Yes cover=22 (4%) prob=0.82]
 Glucose< 130.5
 Age>=28.5
 BMI>=26.35
 DiabetesPedigreeFunction>=0.6375
 Pregnancies>=2.5

Rule number: 367 [Outcome=Yes cover=28 (5%) prob=0.64]
 Glucose< 130.5
 Age>=28.5
 BMI>=26.35
 DiabetesPedigreeFunction< 0.6375
 Glucose>=93.5
 SkinThickness< 27.5
 BloodPressure< 83
 DiabetesPedigreeFunction< 0.4255

Rule number: 233 [Outcome=Yes cover=24 (4%) prob=0.58]
 Glucose>=130.5
 BMI>=29.6
 Glucose< 154.5
 Glucose< 152.5
 BMI< 41.55
 Age< 44.5
 BloodPressure< 81

Rule number: 46 [Outcome=No cover=13 (2%) prob=0.38]
 Glucose< 130.5
 Age>=28.5
 BMI>=26.35
 DiabetesPedigreeFunction>=0.6375
 Pregnancies< 2.5

Rule number: 6 [Outcome=No cover=40 (7%) prob=0.35]
 Glucose>=130.5
 BMI< 29.6

Rule number: 366 [Outcome=No cover=8 (1%) prob=0.25]
 Glucose< 130.5
 Age>=28.5
```

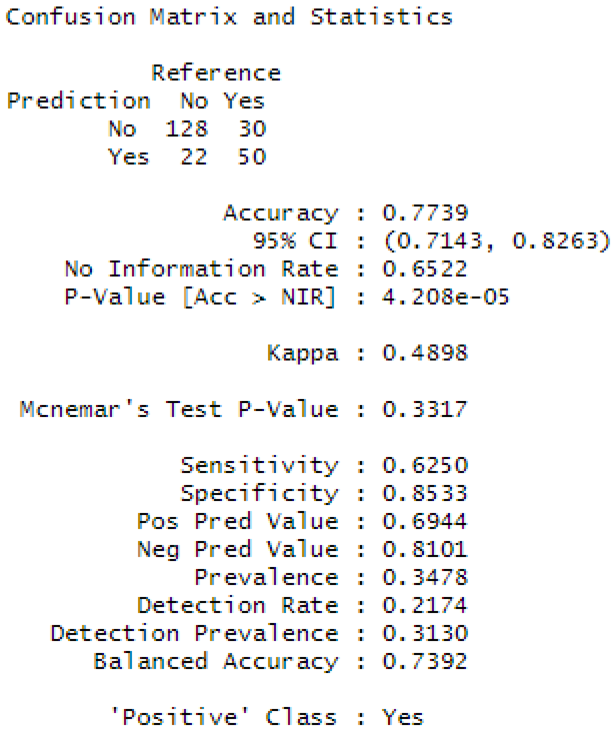
```
  BMI>=26.35
  DiabetesPedigreeFunction< 0.6375
  Glucose>=93.5
  SkinThickness< 27.5
  BloodPressure< 83
  DiabetesPedigreeFunction>=0.4255

Rule number: 90 [Outcome=No cover=32 (6%) prob=0.25]
  Glucose< 130.5
  Age>=28.5
  BMI>=26.35
  DiabetesPedigreeFunction< 0.6375
  Glucose>=93.5
  SkinThickness>=27.5

Rule number: 28 [Outcome=No cover=7 (1%) prob=0.14]
  Glucose>=130.5
  BMI>=29.6
  Glucose< 154.5
  Glucose>=152.5

Rule number: 182 [Outcome=No cover=8 (1%) prob=0.12]
  Glucose< 130.5
  Age>=28.5
  BMI>=26.35
  DiabetesPedigreeFunction< 0.6375
  Glucose>=93.5
  SkinThickness< 27.5
  BloodPressure>=83

Rule number: 232 [Outcome=No cover=9 (2%) prob=0.11]
  Glucose>=130.5
  BMI>=29.6
  Glucose< 154.5
  Glucose< 152.5
  BMI< 41.55
  Age< 44.5
  BloodPressure>=81

Rule number: 4 [Outcome=No cover=196 (36%) prob=0.10]
  Glucose< 130.5
  Age< 28.5

Rule number: 44 [Outcome=No cover=24 (4%) prob=0.04]
  Glucose< 130.5
  Age>=28.5
  BMI>=26.35
  DiabetesPedigreeFunction< 0.6375
```

Glucose< 93.5

Rule number: 10 [Outcome=No cover=31 (6%) prob=0.03]
Glucose< 130.5
Age>=28.5
BMI< 26.35

Rule #117: Has the largest cover of 13%. So if Glucose>=130.5, the person is not going to be diagnosed with diabetes with a probability of 0.86. Here are conditions below.

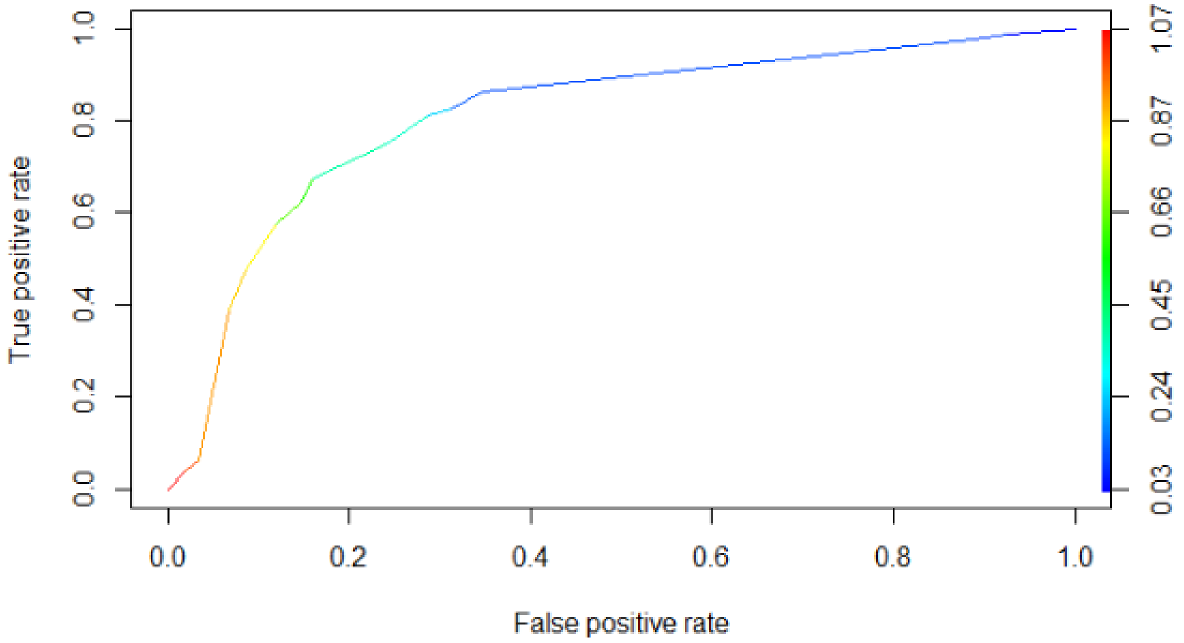
Rule number: 15 [Outcome=Yes cover=71 (13%) prob=0.86]
Glucose>=130.5
BMI>=29.6
Glucose>=154.5



```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  128  30
       Yes  22  50

               Accuracy : 0.7739
                 95% CI : (0.7143, 0.8263)
    No Information Rate : 0.6522
    P-Value [Acc > NIR] : 4.208e-05

                  Kappa : 0.4898

 Mcnemar's Test P-Value : 0.3317

            Sensitivity : 0.6250
            Specificity : 0.8533
         Pos Pred Value : 0.6944
         Neg Pred Value : 0.8101
             Prevalence : 0.3478
         Detection Rate : 0.2174
   Detection Prevalence : 0.3130
      Balanced Accuracy : 0.7392

       'Positive' Class : Yes
```

Accuracy of the model is 0.7739, which seems a fair measure, however, p-value is a low number, and this means that the model is performing very well. Sensitivity of 0.625 indicated that only 62.5% of people with diabetes were predicted to be sick, and specificity of 0.85 indicates that 85% of people who were not sick, were correctly predicted to be so.

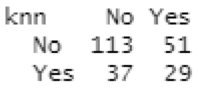
ROC curve



Area under the curve is only 0.811, which is not much more than 0.5 of no information rate.

KNN

From the table it is obvious that specificity, sensitivity, ppv and npv are low, so the model is not performing very well.



```
knn     No Yes
  No   113  51
  Yes   37  29
```

Conclusion: Neither Decision tree, nor KNN are performing properly. So it is better not to use any of those models. We have used the Naive Bayes which performed very well