**Medical Data Analysis**

# DESCRIPTION:

Evaluation Techniques:

1. Loss function - A Metric are used to monitor and measure a model's performance. It is a function that computes the distance between the algorithm's actual output and the expected output. This is used for classification as well as regression. Different types of loss function include mae, mse and rmse.
2. Confusion Matrix – This is mainly used to evaluate the performance of the trained model and improve the overall predictive power of the model. We have to keep in mind of the appropriate data values that we are taking so that it does not produce improper results.

Dataset:

The medical dataset we have considered is the diabetes.csv dataset. Its constraints include,

* Pregnancies: Number of times pregnant
* Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
* BloodPressure: Diastolic blood pressure (mm Hg)
* SkinThickness: Triceps skin fold thickness (mm)
* Insulin: 2-Hour serum insulin (mu U/ml)
* BMI: Body mass index (weight in kg/(height in m)^2)
* DiabetesPedigreeFunction: Diabetes pedigree function
* Age: Age (years)
* Outcome: Class variable (0 or 1)

By performing a performance evaluation, the reader could get a clear idea the various packages that have been installed and the inbuilt functions present in them to make this evaluation possible. As usual we first import the data, then split it into training and testing data. We train the test data using a logistic regression model and then predict the results for the test data and evaluate the overall performance with the actual values and the predicted values.

# PROGRAM:

data <- read.csv(file.choose()) #choosing the diabetes dataset print(data)

str(data) library("Metrics") library("gmodels") library("caret")

View(data)

train\_index <- createDataPartition(data$Outcome, p = 0.7, list = FALSE) train\_data <- data[train\_index, ]

test\_data <- data[-train\_index, ]

View(train\_data)

# Create the logistic regression model

model <- glm(formula = Outcome ~ Glucose + BloodPressure +Age

+DiabetesPedigreeFunction, data = train\_data,

family = binomial(link = "logit")

)

predictions <- predict(model, newdata = test\_data, type = "response") mae(test$Outcome, predictions)

mse(test$Outcome, predictions) rmse(test$Outcome, predictions)

length(predictions)#230 nrow(test)#230

predicted\_labels <- ifelse(predictions >= 0.5, "0", "1") actual\_labels <- test$Outcome

confusion\_matrix <- table(actual\_labels, predicted\_labels) print(confusion\_matrix)

accuracy <- 91/230 #sum of diagonals / n accuracy

precesion <- 58/176 #tp/tp+fp precesion

recall <- 58/79 #tp/tp+fn recall

specificity <- 33/151 #tn/tn+fp specificity

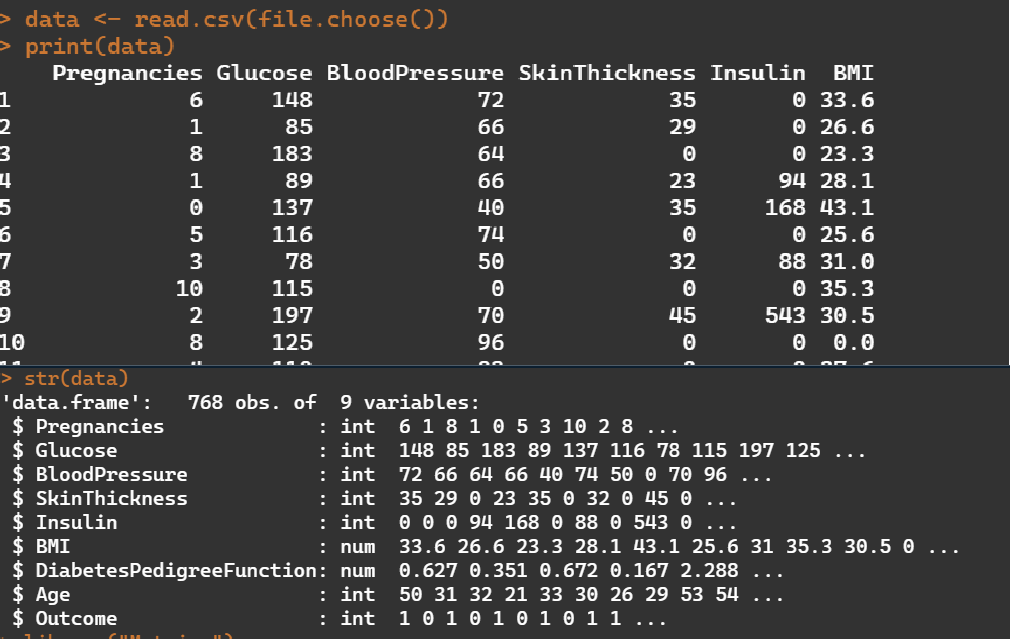
f1\_score = 2\*precesion\*recall/(precesion + recall) f1\_score

# OUTPUT:

Explanation:

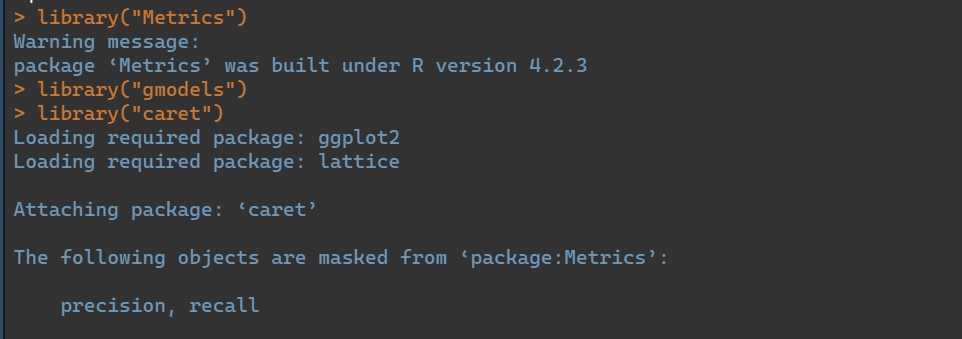
we choose the diabetes dataset and print the values and also get its structure. The structure of the dataset is important to note because it helps us know which variables are numeric and which are categorical in nature.

Output:



Explanation:

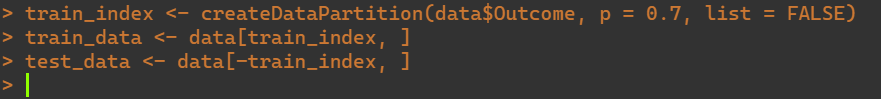
Importing the necessary libraries Output:



Explanation:

Here we are splitting the data into training and testing data. The train to test split is 70:30. That is if there are 768 attibutes it is split into 538:230.

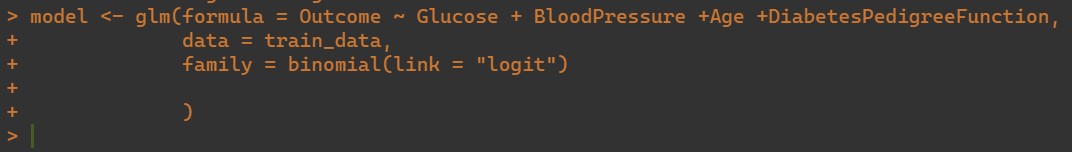
Output:



Explanation:

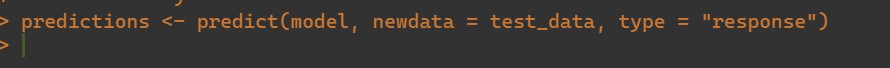
Here we are training our logistic regression model. We have conisered out dependent variable to be y, which is outcome and glucose, bloodpressure age and diabetesdegreefunction as our independent variables (x- values).

Output:

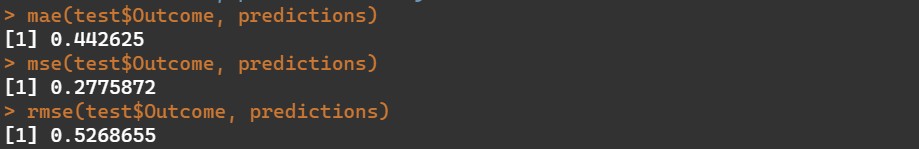


Explanation:

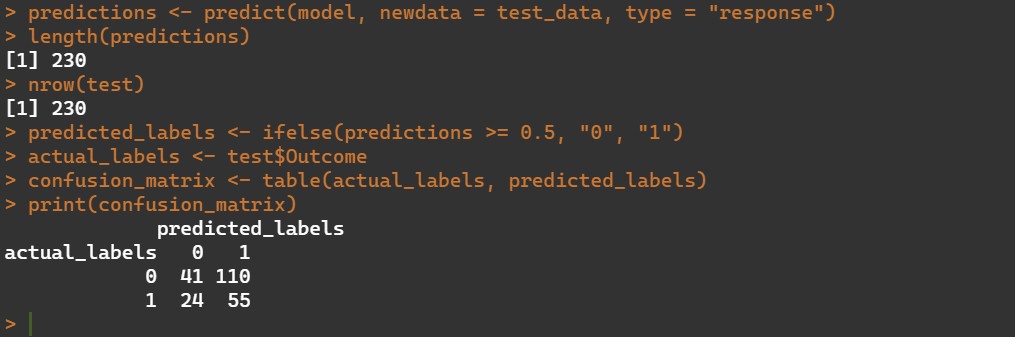
Making predictions for our test data. Output:



Explanation:

Getting the mean absolute error, mean square error and root mean square error. Output:

Explanation:

We label our predcited values and our actual values and create a confusion matrix out of it. Output:

Explanation:

From the matrix we try to get the accuracy, precesion, recall and specificity and finally get the f1\_score.

Output:

