**MACHINE LEARNING MAJOR PROJECT**

**Name :** M Girish

Email : mgirish2221@gmail.com

Every machine learning project begins by understanding what the data and drawing the objectives. While applying machine learning algorithms to your data set, you are understanding, building and analyzing the data as to get the end result.

Following are the steps involved in creating a well-defined ML project:

1] Understand and define the problem

2] Prepare the data

3] Explore and Analyse the data

4] Apply the algorithms

5] Reduce the errors

6] Predict the result

**PROBLEM STATEMENT :-**

In this **Diabetes Prediction using Machine Learning Project Code**, the objective is to predict whether the person has Diabetes or not based on various features like Number of Pregnancies, Insulin Level, Age, BMI. The data set that has used in this project is taken from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage."

**TOOLS AND LIBRARIES :-**

* **Jupyter Notebook** : The Jupyter Notebook is an open-source web application that allows us to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.
* **Python3** : Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation.
* **Pandas** : Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.
* **Scikit-Learn Library** : Scikit-learn is an open source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities.
* **NumPy** : NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

**DAIBETES DATASET DESCRIPTION :-**

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. It is provided courtesy of the Pima Indians Diabetes Database and is available on Kaggle. Here is the link to the dataset. It consists of several medical predictor variables and one target variable, Outcome. Predictor variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on. The dataset has 9 columns as shown below;

• Pregnancies – Number of times pregnant

• Glucose – Plasma glucose concentration a 2 hours in an oral glucose tolerance test

• Blood Pressure – Diastolic blood pressure (mm Hg)

• Skin Thickness – Triceps skinfold thickness (mm)

• Insulin – 2-Hour serum insulin (mu U/ml)

• BMI – Body mass index (weight in kg/(height in m)^2)

• Diabetes Pedigree Function – Diabetes pedigree function

• Age – Age (years)

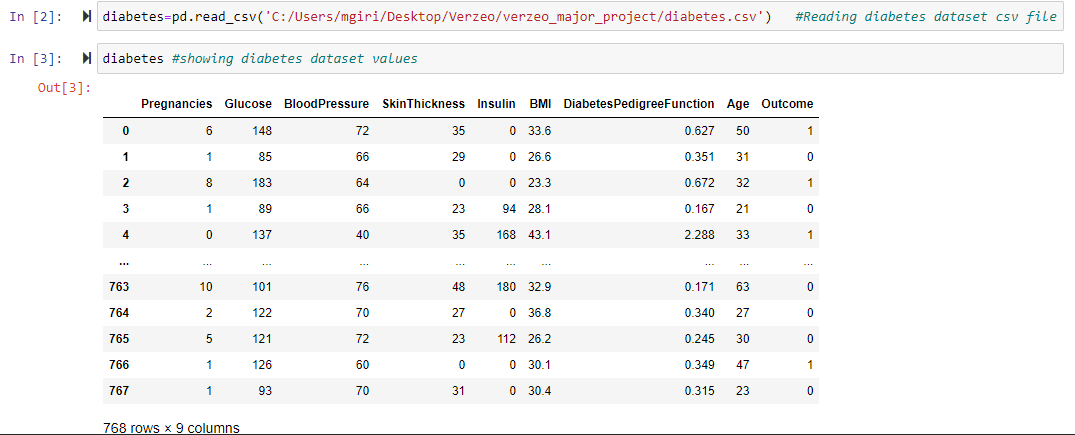
• Outcome – Class variable (0 or 1) 268 of 768 are 1, the others are 0

**DATA ANALYSIS AND PREPROCESSING :-**

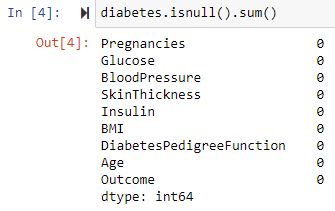
Data analysis is a big task in machine learning most of the time in building a model is taken by this process. Because, for machine learning models training and testing of data used should be clean and filtered according to model criteria.

Generally, cleaning and filtering of data is done by using Numpy , Pandas and Matplotlib libraries.

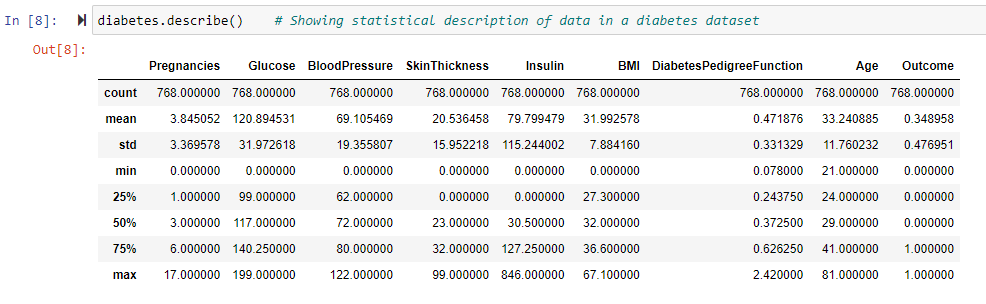
1. READING DATA :



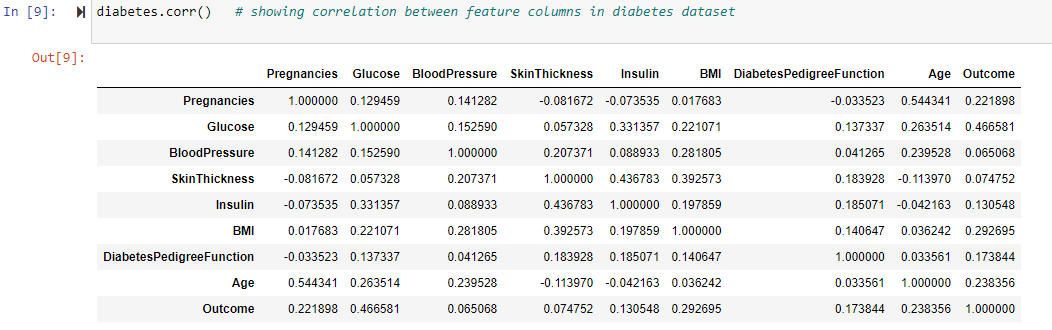
1. CHECKING NULL VALUES IN FEATURES :



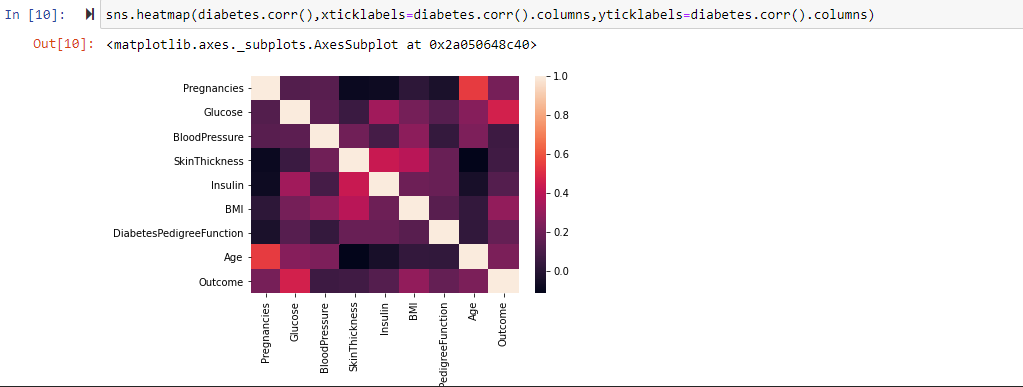
1. STATISTICAL DESCRIPTION OF DATA :



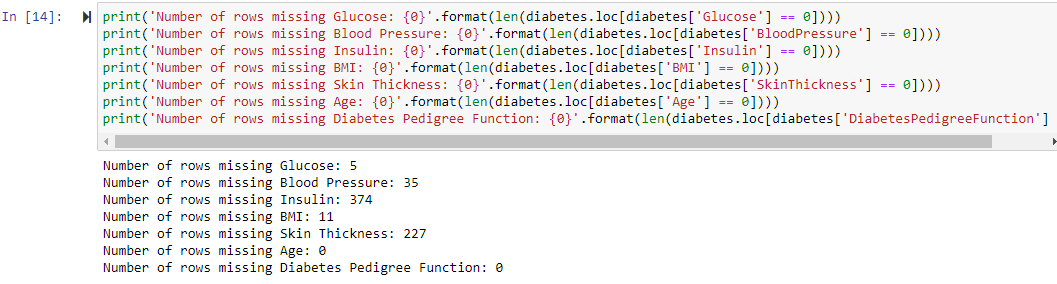
1. CORRELATION BETWEEN THE FEATURES :



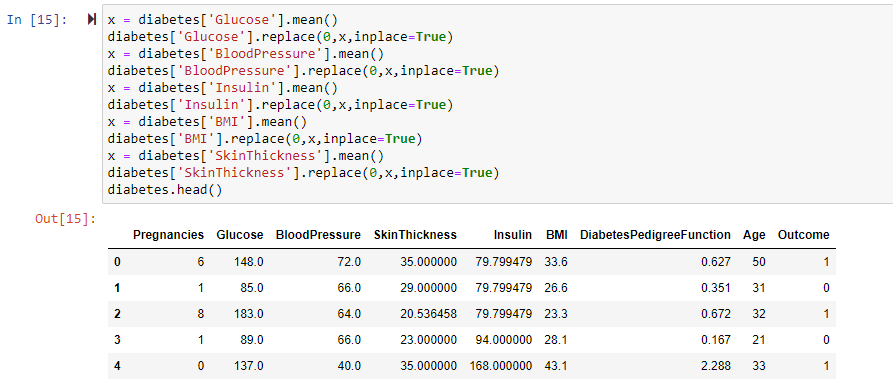
1. HEATMAP OF FEATUTES:



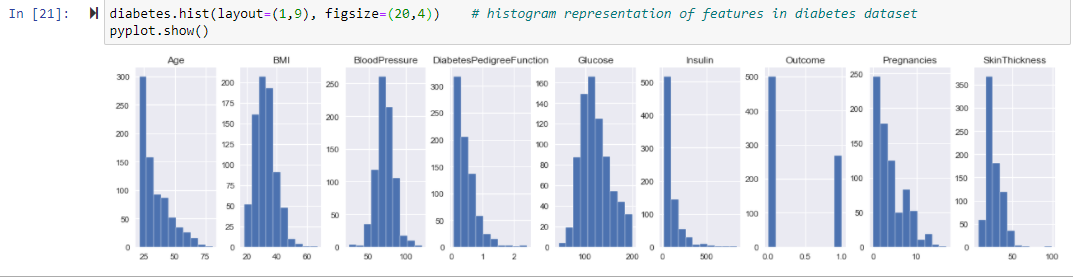
1. FINDING MISSING VALUES IN FEATURES :



1. FILLING MISSING VALUES:



1. HISTOGRAM OF FEATURE VALUES :



**MODEL SELECTION :-**

For given Diabetes dataset the class/label values are discrete and characterised so its better to use classification models to achieve better performance of a model. I have selected a KNN model and Decision Tree model.

1. **KNN Model :-**

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
* It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
* **How it works:-**

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbours

Step-2: Calculate the Euclidean distance of K number of neighbours

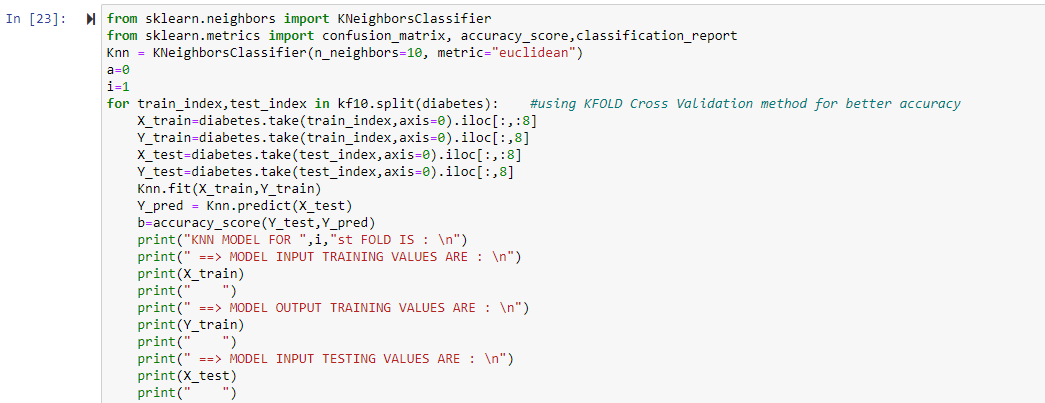
Step-3: Take the K nearest neighbours as per the calculated Euclidean distance.

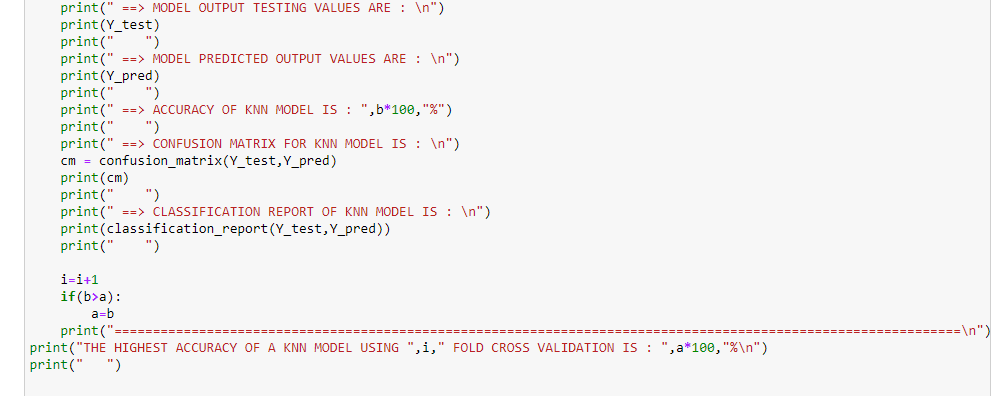
Step-4: Among these k neighbours, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbour is maximum.

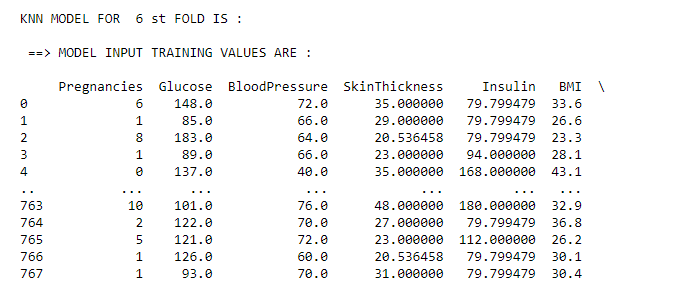
Step-6: Our model is ready.

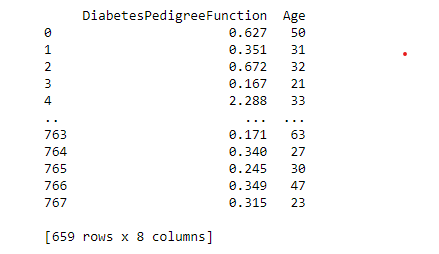


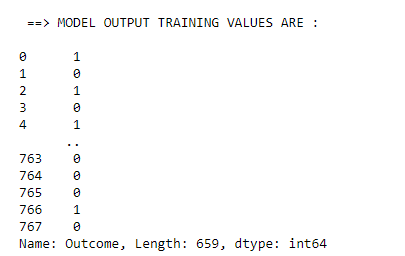


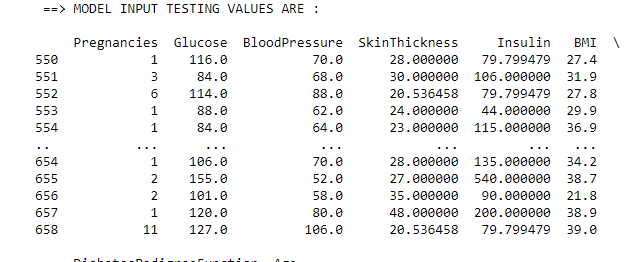
****

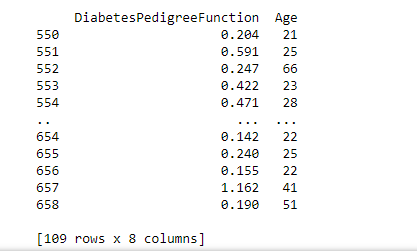
OUTPUT:

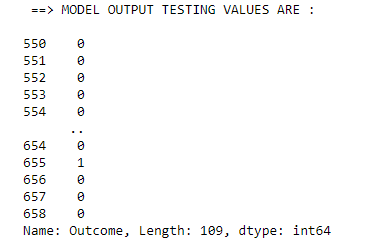


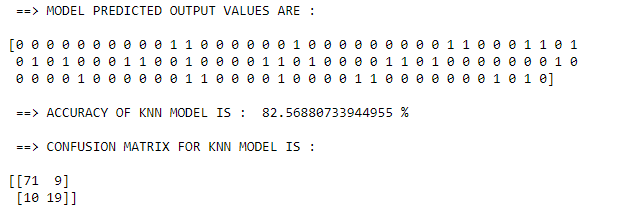


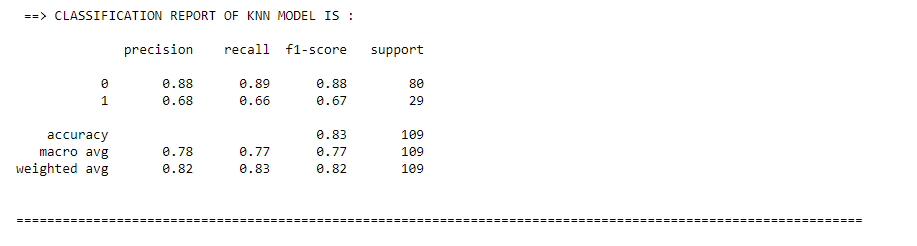














1. **Decision Tree Classification Model :-**

* Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
* In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
* The decisions or the test are performed on the basis of features of the given dataset.
* It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
* It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
* In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
* A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

**How it works :-**

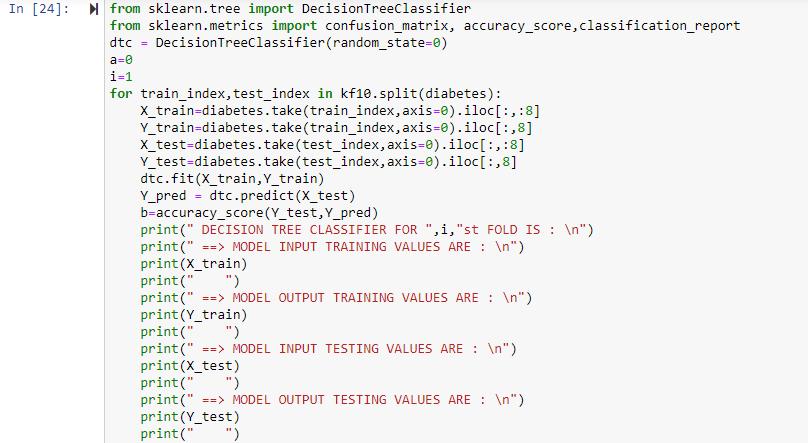
Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

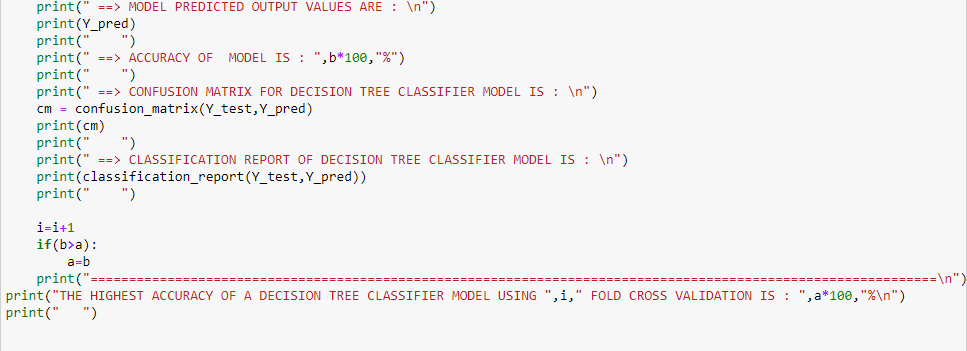
Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).

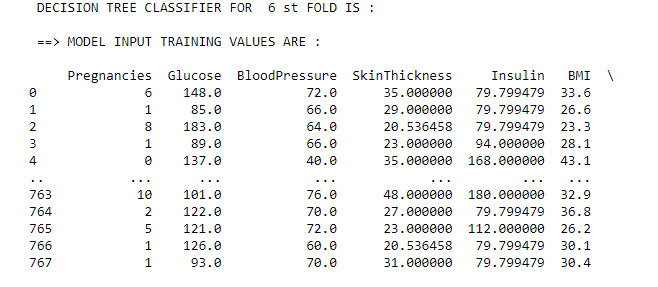
Step-3: Divide the S into subsets that contains possible values for the best attributes.

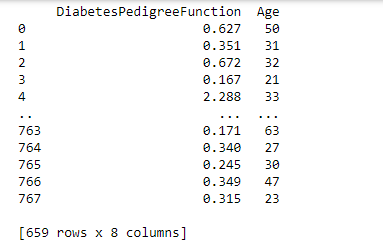
Step-4: Generate the decision tree node, which contains the best attribute

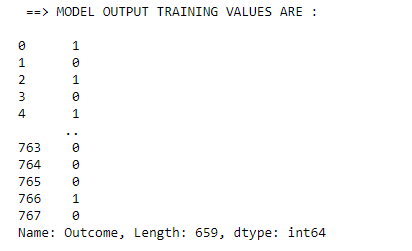
Step-5: Recursively make new decision trees using the subsets of the dataset created in step-3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

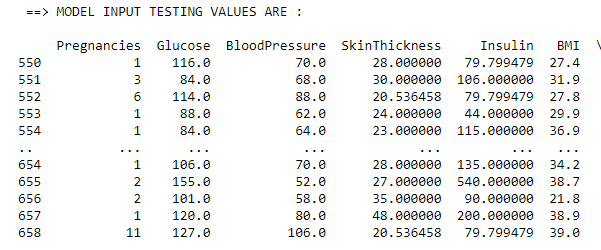


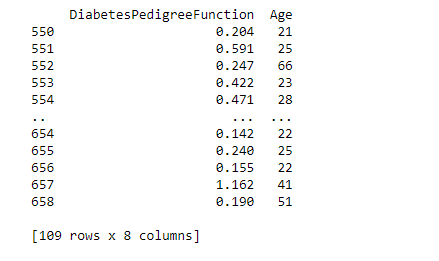


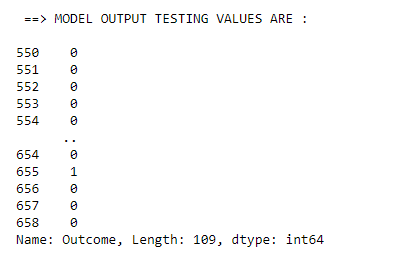
****

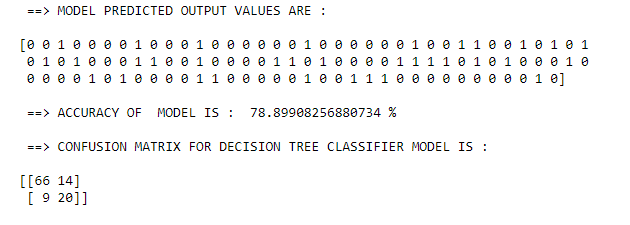
****

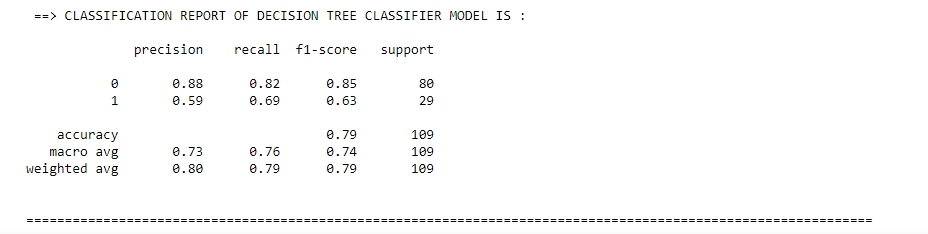
****

****

****

****

****

****

****

**PERFORMANCE COMPARISION BETWEEN MODELS :-**

In training of model the diabetes data is divided using cross validation method of Kfold function. Here, I have used the 8 fold cross validation.

So, that accuracy of KNN algorithm is 83 percent and the accuracy of Decision Tree algorithm is 79 percent.

Hence we can conclude that for given Diabetes dataset KNN model perform better than the Decision Tree model for 8 KFold cross validation of training and test data.