**Practical 9**

**Aim:**

**Write a program to apply decision tree classifier on pima Indian Diabetes dataset.**

**Code:**

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

from sklearn.preprocessing import StandardScaler

from google.colab import files

uploaded = files.upload()

data = pd.read\_csv('diabetes.csv',header = 0)

data.head()

data.describe()

data.isnull().sum()

X\_features = pd.DataFrame(data = data, columns = ["Glucose","BMI","Age"])

X\_features.head(2)

#Considering the 3 features showing the max correlation.

Y = data.iloc[:,8]

Y.head(3)

scaler = StandardScaler()

X\_features = scaler.fit\_transform(X\_features)

X\_features

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X\_features, Y, test\_size=0.25, random\_state=10)

from sklearn.model\_selection import KFold

from sklearn.model\_selection import cross\_val\_score

from sklearn.tree import DecisionTreeClassifier

models = []

models.append(("Decision Tree:",DecisionTreeClassifier()))

print('Model appended...')

results = []

names = []

for name,model in models:

kfold = KFold(n\_splits=5, random\_state=3)

cv\_result = cross\_val\_score(model,X\_train,Y\_train, cv = kfold,scoring = "accuracy")

names.append(name)

results.append(cv\_result)

for i in range(len(names)):

print(names[i],results[i].mean()\*100)

**Output:**







