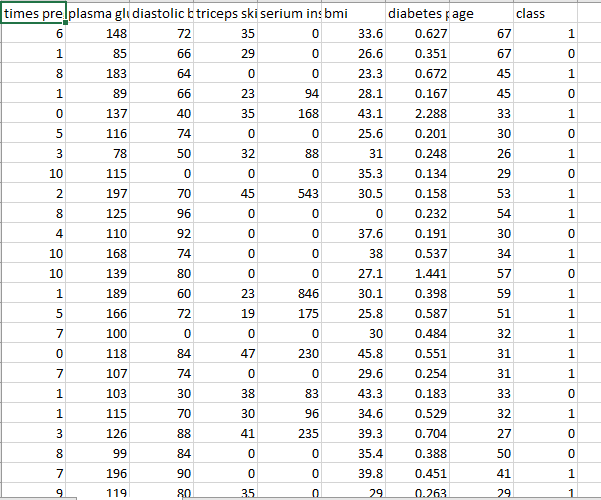
Dataset:-



Code:--

import pandas as pd

import numpy as np

from sklearn.naive\_bayes import BernoulliNB

#import seaborn as sns

#import matplotlib.pyplot as plt

#% matplotlib inline

data=pd.read\_csv("DiabetesData.csv")

print(data.head(10))

print("# no of passenger in the data set:", +(len(data)))

##Analysis data

#sns.countplot(x="diastolic bp",data=data)

#sns.countplot(x="diastolic bp",hue="age", data=data)

#plt.show()

# sns.countplot(x="loans",hue="homeowner",data=bank\_data)

# plt.show()

#data["age"].plot.hist()

#plt.show()

#

## TRAIN MY DATASET

x=data.drop("class",axis=1)

y=data["class"]

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

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=1)

from sklearn.linear\_model import LogisticRegression

logmodel=LogisticRegression()

print(logmodel.fit(x\_train,y\_train))

predictions=logmodel.predict(x\_test)

from sklearn.metrics import classification\_report

print(classification\_report(y\_test,predictions)) #generate classification report

##generate accrucy

from sklearn.metrics import confusion\_matrix

print(confusion\_matrix(y\_test,predictions))

from sklearn.metrics import accuracy\_score

print( "The Accuracy of the Logistic Regression prediction is: ",accuracy\_score(y\_test,predictions))

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=1)

from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier()

classifier.fit(x\_train, y\_train)

y\_pred = classifier.predict(x\_test)

from sklearn.metrics import classification\_report, confusion\_matrix

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

from sklearn.metrics import accuracy\_score

print("\n======================================================================================")

print( "The Accuracy of the prediction using Decision Tree is: ", accuracy\_score(y\_test, y\_pred))

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.35,random\_state=1)

BernNB = BernoulliNB(binarize= 0.1)

BernNB.fit(x\_train,y\_train)

print(BernNB)

y\_expt = y\_test

y\_pred = BernNB.predict(x\_test)

print("\n======================================================================================")

print("The Accuracy using Naive Bayes is: ",accuracy\_score(y\_expt, y\_pred))

OutPut:-

