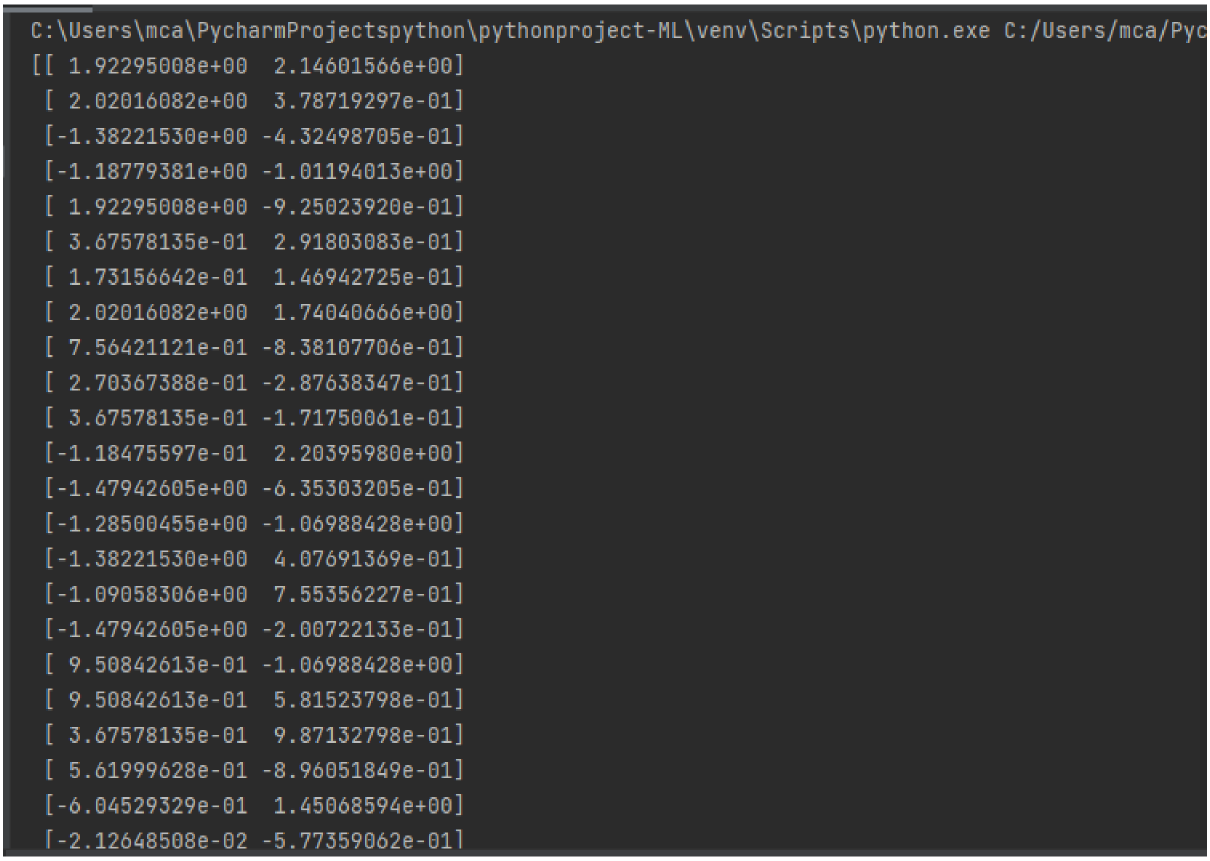
Program no :5

Aim:Program to implement Naïve Bays algorithm using any standard dataset available in the public domain and the accuracy of the algorithm

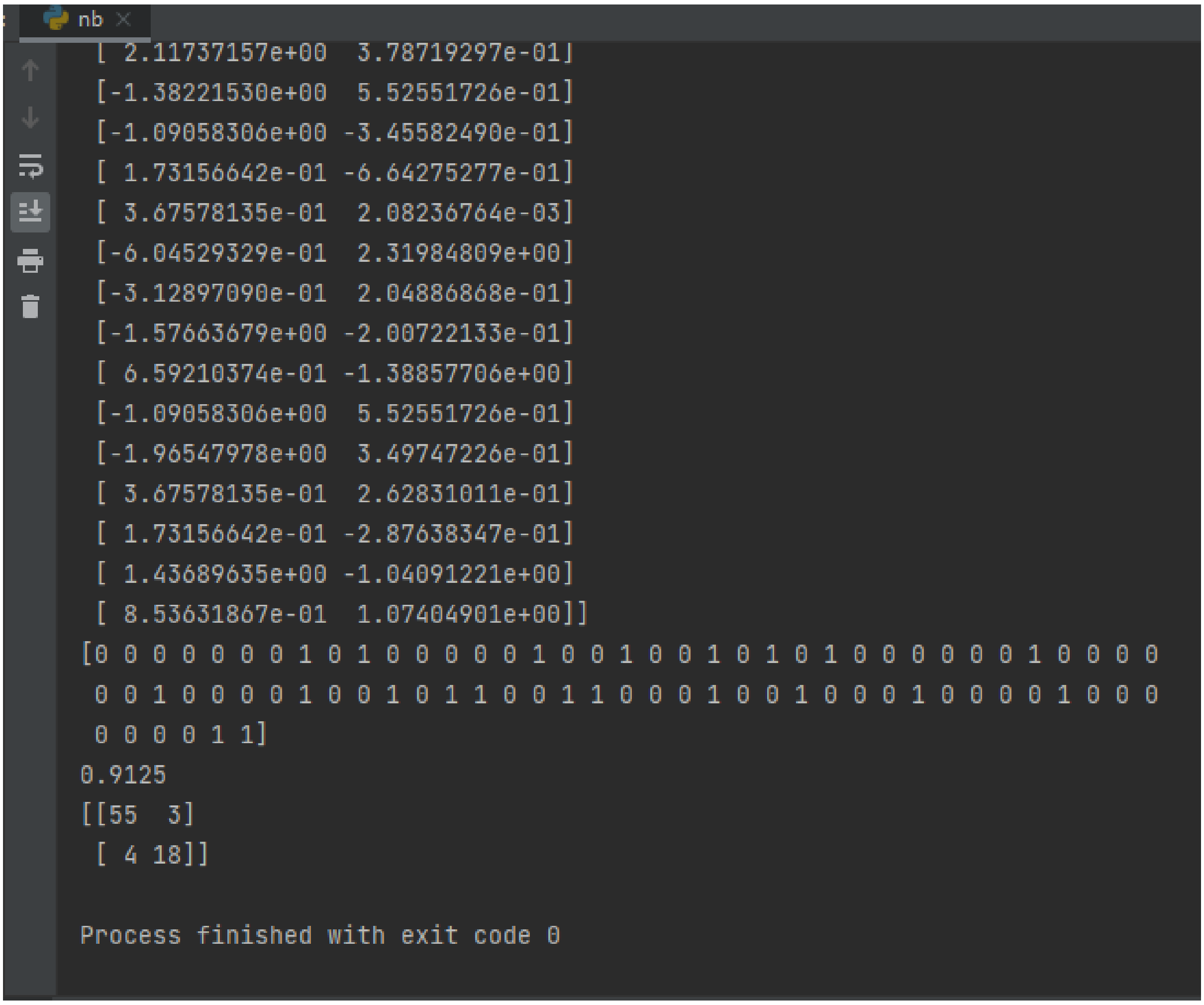
PROGRAM

```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset = pd.read_csv('n.csv')

X = dataset.iloc[:, [2,3]].values

Y = dataset.iloc[:, -1].values

from sklearn.model_selection import train_test_split

X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size = 0.20,random_state = 0)

from sklearn.preprocessing import StandardScaler

sc =StandardScaler()

X_train = sc.fit_transform(X_train)

X_test = sc.transform(X_test)

print(X_train)

print(X_test)

from sklearn.naive_bayes  import GaussianNB

classifier = GaussianNB()

classifier.fit(X_train,Y_train)

Y_pred = classifier.predict(X_test)

print(Y_pred)

from sklearn.metrics import confusion_matrix,accuracy_score

ac = accuracy_score(Y_test,Y_pred)

cm = confusion_matrix(Y_test,Y_pred)

print(ac)

print(cm)
```

OUTPUT



```
C:\Users\mca\PycharmProjectspython\pythonproject-ML\venv\Scripts\python.exe C:/Users/mca/Pyc
[[ 1.92295008e+00  2.14601566e+00]
 [ 2.02016082e+00  3.78719297e-01]
 [-1.38221530e+00 -4.32498705e-01]
 [-1.18779381e+00 -1.01194013e+00]
 [ 1.92295008e+00 -9.25023920e-01]
 [ 3.67578135e-01  2.91803083e-01]
 [ 1.73156642e-01  1.46942725e-01]
 [ 2.02016082e+00  1.74040666e+00]
 [ 7.56421121e-01 -8.38107706e-01]
 [ 2.70367388e-01 -2.87638347e-01]
 [ 3.67578135e-01 -1.71750061e-01]
 [-1.18475597e-01  2.20395980e+00]
 [-1.47942605e+00 -6.35303205e-01]
 [-1.28500455e+00 -1.06988428e+00]
 [-1.38221530e+00  4.07691369e-01]
 [-1.09058306e+00  7.55356227e-01]
 [-1.47942605e+00 -2.00722133e-01]
 [ 9.50842613e-01 -1.06988428e+00]
 [ 9.50842613e-01  5.81523798e-01]
 [ 3.67578135e-01  9.87132798e-01]
 [ 5.61999628e-01 -8.96051849e-01]
 [-6.04529329e-01  1.45068594e+00]
 [-2.12648508e-02 -5.77359062e-01]
```



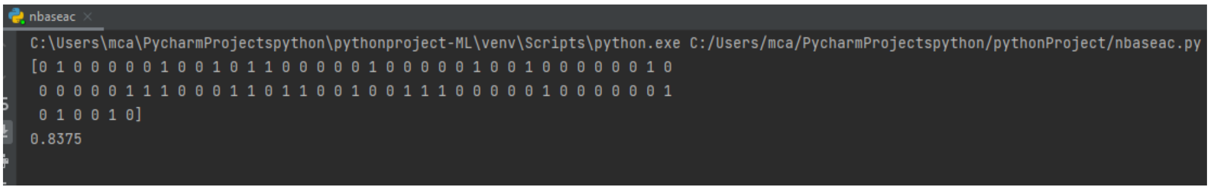
```
 [ 2.11737157e+00  3.78719297e-01]
 [-1.38221530e+00  5.52551726e-01]
 [-1.09058306e+00 -3.45582490e-01]
 [ 1.73156642e-01 -6.64275277e-01]
 [ 3.67578135e-01  2.08236764e-03]
 [-6.04529329e-01  2.31984809e+00]
 [-3.12897090e-01  2.04886868e-01]
 [-1.57663679e+00 -2.00722133e-01]
 [ 6.59210374e-01 -1.38857706e+00]
 [-1.09058306e+00  5.52551726e-01]
 [-1.96547978e+00  3.49747226e-01]
 [ 3.67578135e-01  2.62831011e-01]
 [ 1.73156642e-01 -2.87638347e-01]
 [ 1.43689635e+00 -1.04091221e+00]
 [ 8.53631867e-01  1.07404901e+00]]
[0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0
 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0
 0 0 0 0 1 1]
0.9125
[[55  3]
 [ 4 18]]

Process finished with exit code 0
```

# Plotting graph

```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn import preprocessing

from sklearn.model_selection import train_test_split

from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import accuracy_score


dataset=pd.read_csv('n.csv')

X= dataset.iloc[:,[2,3]].values
```

```
Y= dataset.iloc[:,-1].values


X_train , X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.20,random_state=20)
from sklearn.preprocessing import  StandardScaler


sc = StandardScaler()
X_train =sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
clf = GaussianNB()
clf.fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
print(Y_pred)
print(accuracy_score(Y_test, Y_pred, normalize = True))


plt.plot([i for i in range (0,50)],Y_pred[20:70])
plt.plot([i for i in range (0,80)],Y_test)




plt.legend(["train","test"])
plt.xlabel('X axis')
plt.ylabel('Y axis')
plt.title('Accuracy')
plt.show()
```

OUTPUT



```
C:\Users\mca\PycharmProjectspython\pythonproject-ML\venv\Scripts\python.exe C:/Users/mca/PycharmProjectspython/pythonProject/nbaseac.py
[0 1 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0
 0 0 0 0 0 1 1 1 0 0 0 1 1 0 1 1 0 0 1 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1
 0 1 0 0 1 0]
0.8375
```

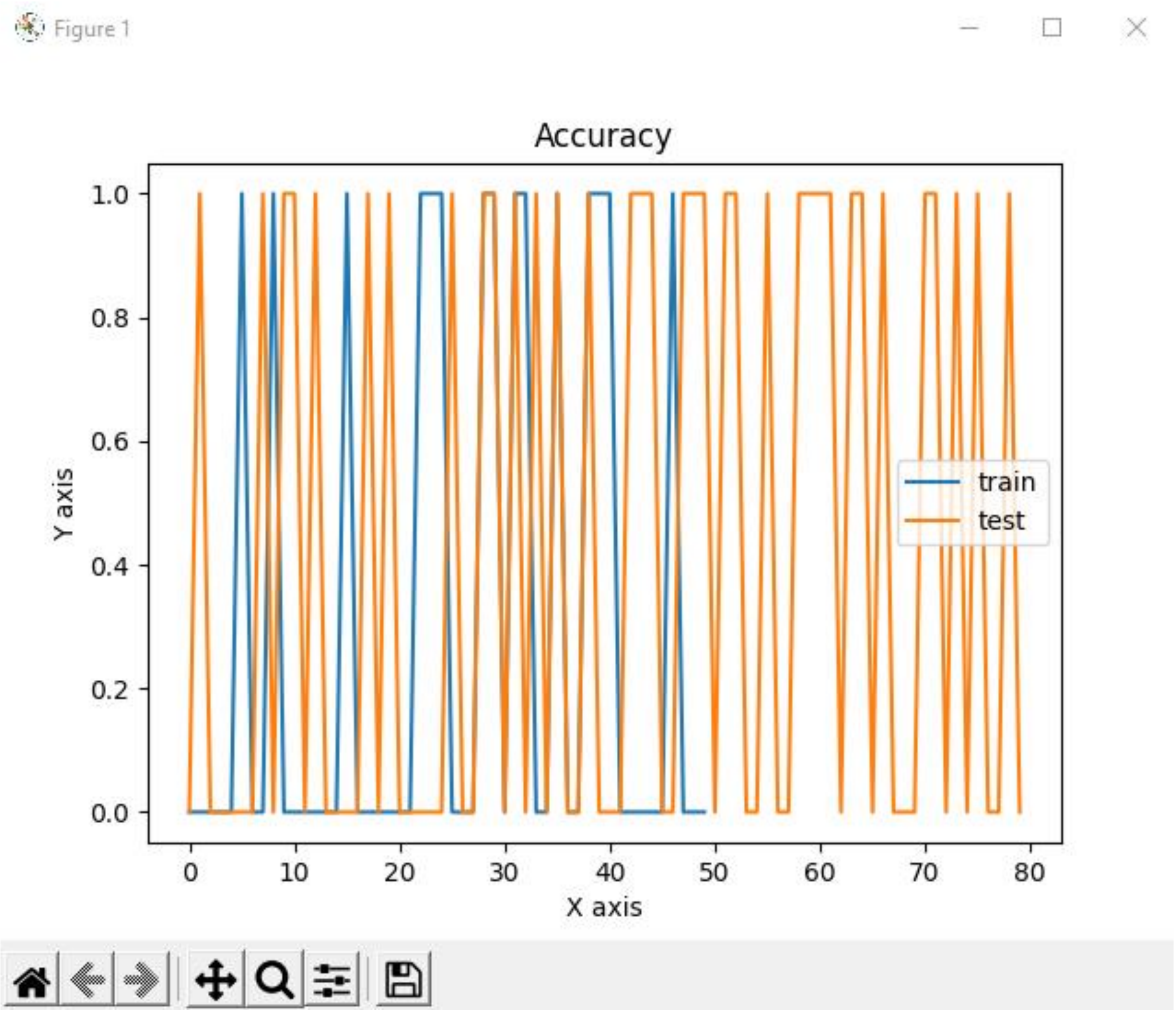


Figure 1
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Y axis
0
10
20
30
40
50
60
70
80
X axis
train
test