

CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.naive_bayes import GaussianNB
```

```
data=pd.read_csv('Pima.csv')
data.head(5)
data.shape
data.info()
```

```
data['x1'].describe()
```

```
data.dtypes
```

```
train=np.array(data.iloc[0:600])
test=np.array(data.iloc[600:768])
```

```
train.shape
test.shape
```

```
model = GaussianNB()
model.fit(train[:,0:8], train[:,8])
predicted= model.predict(test[:,0:8])
print(test[:,8])
print(predicted)
```

```
count=0
for l in range(168):
    if(predicted[l]==test[l,8]):
        count=count+1
```

```
print("Matched samples:",count)
```

```
print("Accuracy:",(count/168))
```

Accuracy: 0.7619047619047619

```
IPython console
Console 1/A x
max 17.000000
Name: x1, dtype: float64

In [39]: data.dtypes
Out[39]:
x1      int64
x2      int64
x3      int64
x4      int64
x5      int64
x6      float64
x7      float64
x8      int64
class   int64
dtype: object

In [40]: train=np.array(data.iloc[0:600])
....: test=np.array(data.iloc[600:768])

In [41]: train.shape
Out[41]: (600, 9)

In [42]: test.shape
Out[42]: (168, 9)

In [43]: model = GaussianNB()
....: model.fit(train[:,0:8], train[:,8])
....: predicted= model.predict(test[:,0:8])

In [44]: print(test[:,8])
[0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1. 1. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1.
 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 1. 1. 1. 0. 1. 1. 1. 0. 0. 0. 0.
 0. 0. 0. 1. 1. 0. 1. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0. 1.
 1. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 0. 1. 1. 0. 0. 1. 0. 0. 1. 1. 0. 0. 1.
 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 1.
 0. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 1. 0.]

In [45]: print(predicted)
[0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1. 0. 0. 1. 1. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1.
 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 1. 0. 1. 1. 1. 0. 0. 0. 0. 0. 1. 1. 0.
 0. 1. 0. 1. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0. 1.
 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.
 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 1. 1. 1. 1. 0. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0.]
```

```
File Edit Search Source Run Debug Consoles Projects Tools View Help
/home/srushti/BE Sem1/my/LP1/DA/2/untitled1.py
untitled1.py* x
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from sklearn.naive_bayes import GaussianNB
6
7 data=pd.read_csv('Pima.csv')
8 data.head(5)
9 data.shape
10 data.info()
11
12 data['x1'].describe()
13
14 data.dtypes
15
16 train=np.array(data.iloc[0:600])
17 test=np.array(data.iloc[600:768])
18
19 train.shape
20 test.shape
21
22 model = GaussianNB()
23 model.fit(train[:,0:8], train[:,8])
24 predicted= model.predict(test[:,0:8])
25 print(test[:,8])
26 print(predicted)
27
28 count=0
29 for l in range(168):
30     if(predicted[l]==test[l,8]):
31         count=count+1
32
33 print("Matched samples:",count)
34
35 print("Accuracy:",(count/168))
36
37
38
IPython console
Console 1/A x
In [44]: print(test[:,8])
[0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1. 1. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1.
 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0.
 0. 0. 0. 1. 1. 0. 1. 0. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0. 1.
 1. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 0. 1. 1. 0. 0. 1. 0. 0. 1. 1. 0. 0. 1.
 0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 1.
 0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 1.
 0. 0. 1. 1. 1. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0.]

In [45]: print(predicted)
[0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1. 1. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1.
 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 1. 1. 0.
 0. 1. 0. 1. 1. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1.
 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 0. 1. 0. 1. 0. 0. 1. 1. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 1.
 1. 1. 1. 0. 1. 0. 0. 0. 0. 1. 1. 1. 1. 0. 0. 1. 0. 1. 0. 0. 0. 0.]

In [46]: count=0
...: for l in range(168):
...:     if(predicted[l]==test[l,8]):
...:         count=count+1

In [47]: print("Matched samples:",count)
...:
...: print("Accuracy:",(count/168))
Matched samples: 128
Accuracy: 0.7619047619047619

In [48]:
Permissions: RW End-of-lines: LF Encod
```