MACHINE LEARNING LAB

EXERCISE :: 4

NAME:: Saptarshi Datta

REG NO.:: 19BAI1041

Naive Bayes ::

The dataset used is the about possibility of heart attack based on some given details about the patient.

The same dataset was used for Logistic Regression Exercise as well.

Link ::

https://www.kaggle.com/ronitf/heart-disease-uci?select=heart.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
2	0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
3	1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
4	2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
5	3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
6	4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
7	5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
8	6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
9	7	44	1	1	120	263	0	1	173	0	0	2	0	3	1
10	8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
11	9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
12	10	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
13	11	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
14	12	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
15	13	64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
16	14	58	0	3	150	283	1	0	162	0	1	2	0	2	1
17	15	50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
18	16	58	0	2	120	340	0	1	172	0	0	2	0	2	1
19	17	66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
20	18	43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
21	19	69	0	3	140	239	0	1	151	0	1.8	2	2	2	1
22	20	59	1	0	135	234	0	1	161	0	0.5	1	0	3	1

Code ::

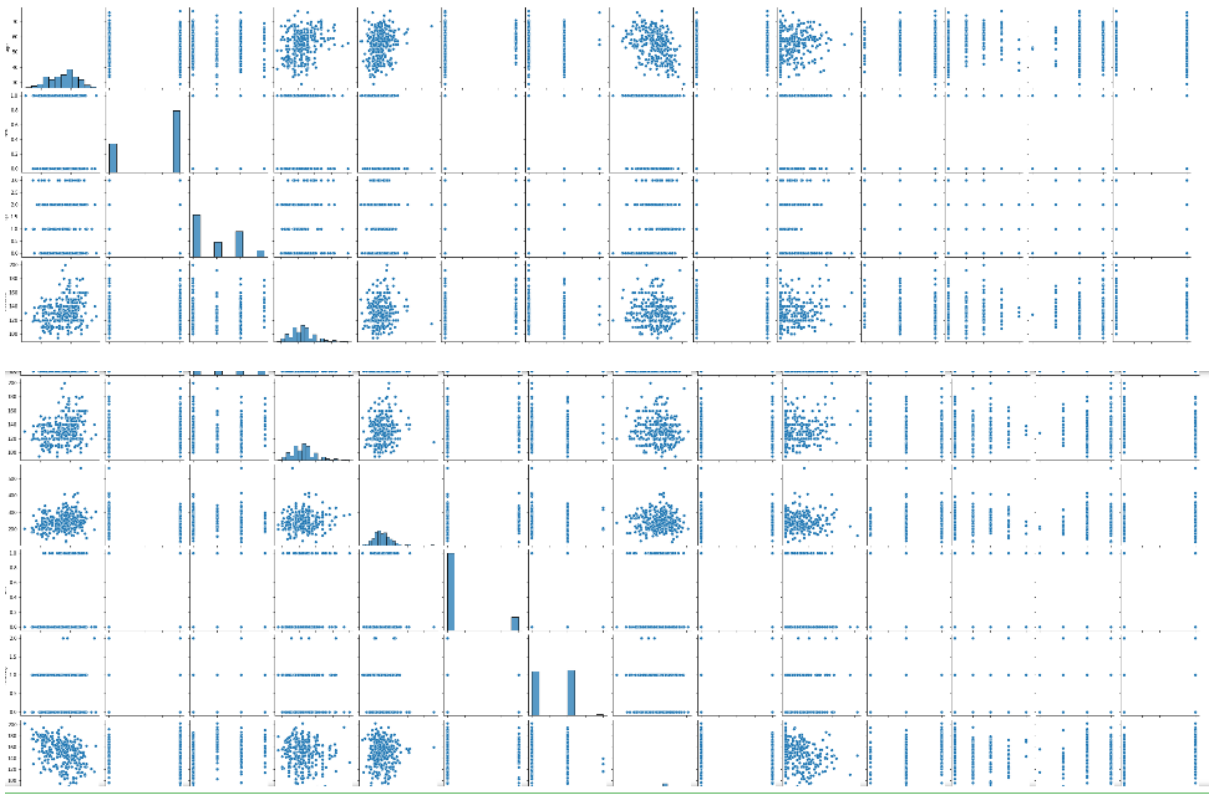
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
```

```
from sklearn.naive_bayes import BernoulliNB
from sklearn.naive_bayes import  MultinomialNB
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("heart.csv")
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

```
sns.pairplot(df)
```



```
df.drop("target", axis = 1, inplace = True)
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2

```
x = df["sex"]
y = df.drop("sex", axis = "columns")
from sklearn.model_selection import train_test_split
from sklearn import metrics
X_train, X_test, y_train, y_test = train_test_split(y,x,test_size=0.30)
from sklearn.naive_bayes import GaussianNB
GausNB=GaussianNB()
GausNB.fit(X_train,y_train)
```

```
GaussianNB()
```

```
GausNB.score(X_test,y_test)
```

```
0.7032967032967034
```

```
X_test[:10]
```

	age	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
24	40	3	140	199	0	1	178	1	1.4	2	0	3
273	58	0	100	234	0	1	156	0	0.1	2	1	3
76	51	2	125	245	1	0	166	0	2.4	1	0	2
276	58	0	146	218	0	1	105	0	2.0	1	1	3
141	43	0	115	303	0	1	181	0	1.2	1	0	2
297	59	0	164	176	1	0	90	0	1.0	1	2	1
63	41	1	135	203	0	1	132	0	0.0	1	0	1
217	63	0	130	330	1	0	132	1	1.8	2	3	3
204	62	0	160	164	0	0	145	0	6.2	0	3	3
296	63	0	124	197	0	1	136	1	0.0	1	0	2

y_test[:10] #1 implies positive chance of heart attack and 0 implies less chance

```
24     1
273    1
76     1
276    1
141    1
297    1
63     1
217    1
204    0
296    0
Name: sex, dtype: int64
```

GausNB.predict(X_test[:10])

```
array([1, 1, 1, 1, 0, 1, 1, 1, 1, 1])
```

GausNB.predict_proba(X_test[:10])

```
array([[7.97756960e-02, 9.20224304e-01],
       [2.78539847e-01, 7.21460153e-01],
       [6.41278714e-02, 9.35872129e-01],
       [4.07379388e-02, 9.59262061e-01],
       [7.73005670e-01, 2.26994330e-01],
       [8.82557018e-03, 9.91174430e-01],
       [3.58185333e-01, 6.41814667e-01],
       [2.55127718e-03, 9.97448723e-01],
       [2.82235714e-07, 9.99999718e-01],
       [3.32377613e-01, 6.67622387e-01]])
```

GausNB.score(X_test,y_test)

Brb = BernoulliNB()

Brb.fit(X_train,y_train)

```
BernoulliNB()
```

Brb.score(X_test,y_test)

```
0.6153846153846154
```

X_test[:10]

	age	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
24	40	3	140	199	0	1	178	1	1.4	2	0	3
273	58	0	100	234	0	1	156	0	0.1	2	1	3
76	51	2	125	245	1	0	166	0	2.4	1	0	2
276	58	0	146	218	0	1	105	0	2.0	1	1	3
141	43	0	115	303	0	1	181	0	1.2	1	0	2
297	59	0	164	176	1	0	90	0	1.0	1	2	1
63	41	1	135	203	0	1	132	0	0.0	1	0	1
217	63	0	130	330	1	0	132	1	1.8	2	3	3
204	62	0	160	164	0	0	145	0	6.2	0	3	3
296	63	0	124	197	0	1	136	1	0.0	1	0	2

y_test[:10]

```
24     1
273    1
76     1
276    1
141    1
297    1
63     1
217    1
204    0
296    0
Name: sex, dtype: int64
```

Brb.score(X_test,y_test)

```
0.6153846153846154
```

Mnb =MultinomialNB()

Mnb.fit(X_train,y_train)

Mnb.score(X_test,y_test)

```
0.6043956043956044
```

X_test[:10]

	age	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
24	40	3	140	199	0	1	178	1	1.4	2	0	3
273	58	0	100	234	0	1	156	0	0.1	2	1	3
76	51	2	125	245	1	0	166	0	2.4	1	0	2
276	58	0	146	218	0	1	105	0	2.0	1	1	3
141	43	0	115	303	0	1	181	0	1.2	1	0	2
297	59	0	164	176	1	0	90	0	1.0	1	2	1
63	41	1	135	203	0	1	132	0	0.0	1	0	1
217	63	0	130	330	1	0	132	1	1.8	2	3	3
204	62	0	160	164	0	0	145	0	6.2	0	3	3
296	63	0	124	197	0	1	136	1	0.0	1	0	2

y_test[:10]

```
24     1
273    1
76     1
276    1
141    1
297    1
63     1
217    1
204    0
296    0
Name: sex, dtype: int64
```

Mnb.predict(X_test[:10])

```
array([1, 1, 1, 1, 0, 1, 1, 0, 1, 1])
```

Mnb.predict_proba(X_test[:10])

```
array([[9.79743067e-03, 9.90202569e-01],
       [3.74506350e-01, 6.25493650e-01],
       [1.81597451e-01, 8.18402549e-01],
       [1.02427524e-01, 8.97572476e-01],
       [9.27870710e-01, 7.21292903e-02],
       [5.73279821e-03, 9.94267202e-01],
```

```
[2.27167024e-01, 7.72832976e-01],
[7.92746293e-01, 2.07253707e-01],
[3.23253940e-05, 9.99967675e-01],
[6.06158995e-02, 9.39384100e-01]])
```

Mnb.score(X_test,y_test)

```
0.6043956043956044
```