4goj4mdo8

April 8, 2024

[1]:
```
# Aim: To perform and find the accuracy of Naive bayes Classifier
```

[2]:
```
# Name: Mandar K Satpute
# Class: 3rd year
# Sec: B
# Roll No. : 54
```

[3]:
```
import pandas as pd
import os
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
import warnings
warnings.filterwarnings('ignore')
```

[4]:
```
df=pd.read_csv('C:\\Users\\hp\\Desktop\\CHD_preprocessed.csv')
```

[5]:
```
df.head()
```

[5]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
0	1	39	1	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.0	77.0	0
1	0	46	0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.0	76.0	0
2	1	48	0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.0	70.0	0
3	0	61	1	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.0	103.0	1
4	0	46	1	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.0	85.0	0

```
1           0
2           0
3           1
4           0
```

[6]:
```
df.tail()
```

[6]:
```
      male  age  education  currentSmoker  cigsPerDay  BPMeds  \
4128     1   50          0              1         1.0     0.0
4129     1   51          1              1        43.0     0.0
4130     0   48          0              1        20.0     0.0
4131     0   44          0              1        15.0     0.0
4132     0   52          0              0         0.0     0.0

      prevalentStroke  prevalentHyp  diabetes  totChol  sysBP  diaBP    BMI  \
4128                0             1         0    313.0  179.0   92.0  25.97
4129                0             0         0    207.0  126.5   80.0  19.71
4130                0             0         0    248.0  131.0   72.0  22.00
4131                0             0         0    210.0  126.5   87.0  19.16
4132                0             0         0    269.0  133.5   83.0  21.47

      heartRate  glucose  TenYearCHD
4128       66.0     86.0           1
4129       65.0     68.0           0
4130       84.0     86.0           0
4131       86.0     82.0           0
4132       80.0    107.0           0
```

[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4133 entries, 0 to 4132
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   male             4133 non-null   int64
 1   age              4133 non-null   int64
 2   education        4133 non-null   int64
 3   currentSmoker    4133 non-null   int64
 4   cigsPerDay       4133 non-null   float64
 5   BPMeds           4133 non-null   float64
 6   prevalentStroke  4133 non-null   int64
 7   prevalentHyp     4133 non-null   int64
 8   diabetes         4133 non-null   int64
 9   totChol          4133 non-null   float64
 10  sysBP            4133 non-null   float64
 11  diaBP            4133 non-null   float64
```

```
 12  BMI              4133 non-null   float64
 13  heartRate        4133 non-null   float64
 14  glucose          4133 non-null   float64
 15  TenYearCHD       4133 non-null   int64
dtypes: float64(8), int64(8)
memory usage: 516.8 KB
```

[8]:
```
df.describe()
```

[8]:

	male	age	education	currentSmoker	cigsPerDay
count	4133.000000	4133.000000	4133.000000	4133.000000	4133.000000
mean	0.427293	49.557222	0.280668	0.494798	9.101621
std	0.494745	8.561628	0.449380	0.500033	11.918440
min	0.000000	32.000000	0.000000	0.000000	0.000000
25%	0.000000	42.000000	0.000000	0.000000	0.000000
50%	0.000000	49.000000	0.000000	0.000000	0.000000
75%	1.000000	56.000000	1.000000	1.000000	20.000000
max	1.000000	70.000000	1.000000	1.000000	70.000000

	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol
count	4133.000000	4133.000000	4133.000000	4133.000000	4133.000000
mean	0.034358	0.006049	0.311154	0.025647	236.664408
std	0.182168	0.077548	0.463022	0.158100	43.909188
min	0.000000	0.000000	0.000000	0.000000	107.000000
25%	0.000000	0.000000	0.000000	0.000000	206.000000
50%	0.000000	0.000000	0.000000	0.000000	234.000000
75%	0.000000	0.000000	1.000000	0.000000	262.000000
max	1.000000	1.000000	1.000000	1.000000	600.000000

	sysBP	diaBP	BMI	heartRate	glucose
count	4133.000000	4133.000000	4133.000000	4133.000000	4133.000000
mean	132.367046	82.872248	25.778571	75.925236	81.946528
std	22.080332	11.952654	4.074360	12.049188	22.860954
min	83.500000	48.000000	15.540000	44.000000	40.000000
25%	117.000000	75.000000	23.060000	68.000000	72.000000
50%	128.000000	82.000000	25.380000	75.000000	80.000000
75%	144.000000	89.500000	27.990000	83.000000	85.000000
max	295.000000	142.500000	56.800000	143.000000	394.000000

	TenYearCHD
count	4133.000000
mean	0.151948
std	0.359014
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000

```
max       1.000000
```

[9]:
```
df.size
```

[9]:
```
66128
```

[10]:
```
df.shape
```

[10]:
```
(4133, 16)
```

[11]:
```
df.isna().sum()
```

[11]:
```
male               0
age                0
education          0
currentSmoker      0
cigsPerDay         0
BPMeds             0
prevalentStroke    0
prevalentHyp       0
diabetes           0
totChol            0
sysBP              0
diaBP              0
BMI                0
heartRate          0
glucose            0
TenYearCHD         0
dtype: int64
```

[12]:
```
x = df.drop("TenYearCHD",axis=1)
y = df['TenYearCHD']
```

[13]:
```
x
```

[13]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	\
0	1	39	1	0	0.0	0.0	
1	0	46	0	0	0.0	0.0	
2	1	48	0	1	20.0	0.0	
3	0	61	1	1	30.0	0.0	
4	0	46	1	1	23.0	0.0	
…	…	…	…	…	…	…	
4128	1	50	0	1	1.0	0.0	
4129	1	51	1	1	43.0	0.0	
4130	0	48	0	1	20.0	0.0	
4131	0	44	0	1	15.0	0.0	
4132	0	52	0	0	0.0	0.0	

```
      prevalentStroke  prevalentHyp  diabetes  totChol  sysBP  diaBP    BMI  \
0                   0             0         0    195.0  106.0   70.0  26.97
1                   0             0         0    250.0  121.0   81.0  28.73
2                   0             0         0    245.0  127.5   80.0  25.34
3                   0             1         0    225.0  150.0   95.0  28.58
4                   0             0         0    285.0  130.0   84.0  23.10
…                 …             …       …       …    …     …     …
4128                0             1         0    313.0  179.0   92.0  25.97
4129                0             0         0    207.0  126.5   80.0  19.71
4130                0             0         0    248.0  131.0   72.0  22.00
4131                0             0         0    210.0  126.5   87.0  19.16
4132                0             0         0    269.0  133.5   83.0  21.47

      heartRate  glucose
0          80.0     77.0
1          95.0     76.0
2          75.0     70.0
3          65.0    103.0
4          85.0     85.0
…          …       …
4128       66.0     86.0
4129       65.0     68.0
4130       84.0     86.0
4131       86.0     82.0
4132       80.0    107.0

[4133 rows x 15 columns]
```

[14]:
```
y
```

[14]:
```
0       0
1       0
2       0
3       1
4       0
       ..
4128    1
4129    0
4130    0
4131    0
4132    0
Name: TenYearCHD, Length: 4133, dtype: int64
```

[15]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.
↪2,random_state=42)
```

[16]:
```
y_train
```

[16]:
```
173     1
1022    0
3182    0
331     1
2222    0
       ..
3444    0
466     0
3092    0
3772    0
860     0
Name: TenYearCHD, Length: 3306, dtype: int64
```

[17]:
```
y_test
```

[17]:
```
1864    0
1210    0
1924    0
1752    0
1095    0
       ..
881     0
25      1
3256    0
2269    0
1074    0
Name: TenYearCHD, Length: 827, dtype: int64
```

[18]:
```
nb_model = GaussianNB()
nb_model.fit(x_train, y_train)
```

[18]:
```
GaussianNB()
```

[19]:
```
# Evaluate the model
train_accuracy = nb_model.score(x_train, y_train)
test_accuracy = nb_model.score(x_test, y_test)
```

[20]:
```
print("Training Accuracy:", train_accuracy)
print("Testing Accuracy:", test_accuracy)
```

```
Training Accuracy: 0.8236539624924379
Testing Accuracy: 0.8101571946795647
```