In [10]:
```
#Experiment no 10 To perform and find the accuracy of Naive bayes Classifier
```

In [11]:
```
#Name : :Shravani M Karne
#Roll no : 39
#Sub : Big Data Analysis(ET 2 lab)
```

In [12]:
```
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
import warnings
warnings.filterwarnings('ignore')
```

In [13]:
```
os.getcwd()
```

Out[13]:
```
'C:\\Users\\rautp'
```

In [15]:
```
os.chdir('C:\\Users\\rautp')
```

In [16]:
```
df=pd.read_csv('CHD_preprocessed.csv')
```

In [17]:
```
df.head()
```

Out[17]:

| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp | diabetes | totChol |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 39 | 1 | 0 | 0.0 | 0.0 | 0 | 0 | 0 | 195.0 |
| **1** | 0 | 46 | 0 | 0 | 0.0 | 0.0 | 0 | 0 | 0 | 250.0 |
| **2** | 1 | 48 | 0 | 1 | 20.0 | 0.0 | 0 | 0 | 0 | 245.0 |
| **3** | 0 | 61 | 1 | 1 | 30.0 | 0.0 | 0 | 1 | 0 | 225.0 |
| **4** | 0 | 46 | 1 | 1 | 23.0 | 0.0 | 0 | 0 | 0 | 285.0 |

In [18]:
```
df.tail()
```

Out[18]:

| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp | diabetes | totC |
|---|---|---|---|---|---|---|---|---|---|---|
| **4128** | 1 | 50 | 0 | 1 | 1.0 | 0.0 | 0 | 1 | 0 | 31 |
| **4129** | 1 | 51 | 1 | 1 | 43.0 | 0.0 | 0 | 0 | 0 | 20 |
| **4130** | 0 | 48 | 0 | 1 | 20.0 | 0.0 | 0 | 0 | 0 | 24 |
| **4131** | 0 | 44 | 0 | 1 | 15.0 | 0.0 | 0 | 0 | 0 | 21 |
| **4132** | 0 | 52 | 0 | 0 | 0.0 | 0.0 | 0 | 0 | 0 | 26 |

In [19]:
```
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
```

```
In [20]: df.size
```

```
Out[20]: 66128
```

```
In [21]: df.shape
```

```
Out[21]: (4133, 16)
```

```
In [22]: df.isna().sum()
```

```
Out[22]: 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
```

```
In [23]: df.describe()
```

Loading [MathJax]/extensions/Safe.js

Out[23]:

| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | preva |
|---|---|---|---|---|---|---|---|---|
| **count** | 4133.000000 | 4133.000000 | 4133.000000 | 4133.000000 | 4133.000000 | 4133.000000 | 4133.000000 | 4133 |
| **mean** | 0.427293 | 49.557222 | 0.280668 | 0.494798 | 9.101621 | 0.034358 | 0.006049 | 0 |
| **std** | 0.494745 | 8.561628 | 0.449380 | 0.500033 | 11.918440 | 0.182168 | 0.077548 | 0 |
| **min** | 0.000000 | 32.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0 |
| **25%** | 0.000000 | 42.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0 |
| **50%** | 0.000000 | 49.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0 |
| **75%** | 1.000000 | 56.000000 | 1.000000 | 1.000000 | 20.000000 | 0.000000 | 0.000000 | 1 |
| **max** | 1.000000 | 70.000000 | 1.000000 | 1.000000 | 70.000000 | 1.000000 | 1.000000 | 1 |

In [24]:

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

In [25]:

```
x
```

Out[25]:

| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp | diabetes | totC |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 39 | 1 | 0 | 0.0 | 0.0 | 0 | 0 | 0 | 19 |
| **1** | 0 | 46 | 0 | 0 | 0.0 | 0.0 | 0 | 0 | 0 | 25 |
| **2** | 1 | 48 | 0 | 1 | 20.0 | 0.0 | 0 | 0 | 0 | 24 |
| **3** | 0 | 61 | 1 | 1 | 30.0 | 0.0 | 0 | 1 | 0 | 22 |
| **4** | 0 | 46 | 1 | 1 | 23.0 | 0.0 | 0 | 0 | 0 | 28 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **4128** | 1 | 50 | 0 | 1 | 1.0 | 0.0 | 0 | 1 | 0 | 31 |
| **4129** | 1 | 51 | 1 | 1 | 43.0 | 0.0 | 0 | 0 | 0 | 20 |
| **4130** | 0 | 48 | 0 | 1 | 20.0 | 0.0 | 0 | 0 | 0 | 24 |
| **4131** | 0 | 44 | 0 | 1 | 15.0 | 0.0 | 0 | 0 | 0 | 21 |
| **4132** | 0 | 52 | 0 | 0 | 0.0 | 0.0 | 0 | 0 | 0 | 26 |

4133 rows × 15 columns

In [26]:

```
y
```

Out[26]:

```
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
```

# Train-Test Split

```
In [27]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

```
In [28]: y_train
```

```
Out[28]: 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
```

```
In [29]: y_test
```

```
Out[29]: 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
```

```
In [30]: from sklearn.linear_model import LogisticRegression
         model = LogisticRegression().fit(x_train,y_train)
         model.score(x_train,y_train)
```

```
Out[30]: 0.8557168784029038
```

```
In [31]: H = [1,1,1,2,3,3,4,5,6,4,4,4,5,6,6,6,7,7,8,8,9,9,9,10,10,10,10]
```

```
In [32]: print(type(H))

         <class 'list'>
```

```
In [ ]:
```