In [1]:

```
#Experiment no 8 To Perform and find the accuracy of KNN Classifier.
```

In [2]:

```
#Name: Shravani M Karne
#Roll no.: 39
#Sec:A
#Year:3rd Year
#Sub: Big Data Analysis(ET 2 Lab)
```

In [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
import warnings
warnings.filterwarnings('ignore')
```

In [4]:

```
os.getcwd()
```

Out[4]:

```
'C:\\Users\\rautp'
```

In [5]:

```
os.chdir('C:\\Users\\rautp')
```

In [6]:

```
df=pd.read_csv('framingham.csv')
```

In [7]:

```
df.head()
```

Out[7]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0

In [8]:

```
df.tail()
```

Out[8]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totC
4233	1	50	1.0	1	1.0	0.0	0	1	0	31
4234	1	51	3.0	1	43.0	0.0	0	0	0	20
4235	0	48	2.0	1	20.0	NaN	0	0	0	24
4236	0	44	1.0	1	15.0	0.0	0	0	0	21
4237	0	52	2.0	0	0.0	0.0	0	0	0	26

In [9]:

```
df.info
```

Out[9]:

```
<bound method DataFrame.info of       male  age  education  currentSmoker  cigsPerDay  B
PMeds  \
0        1   39        4.0              0         0.0     0.0
1        0   46        2.0              0         0.0     0.0
2        1   48        1.0              1        20.0     0.0
3        0   61        3.0              1        30.0     0.0
4        0   46        3.0              1        23.0     0.0
...    ...  ...        ...            ...         ...     ...
4233     1   50        1.0              1         1.0     0.0
4234     1   51        3.0              1        43.0     0.0
4235     0   48        2.0              1        20.0     NaN
4236     0   44        1.0              1        15.0     0.0
4237     0   52        2.0              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
...               ...           ...       ...      ...    ...    ...    ...
4233                0             1         0    313.0  179.0   92.0  25.97
4234                0             0         0    207.0  126.5   80.0  19.71
4235                0             0         0    248.0  131.0   72.0  22.00
4236                0             0         0    210.0  126.5   87.0  19.16
4237                0             0         0    269.0  133.5   83.0  21.47

      heartRate  glucose  TenYearCHD
0          80.0     77.0           0
1          95.0     76.0           0
2          75.0     70.0           0
3          65.0    103.0           1
4          85.0     85.0           0
...         ...      ...         ...
4233       66.0     86.0           1
4234       65.0     68.0           0
4235       84.0     86.0           0
4236       86.0      NaN           0
4237       80.0    107.0           0

[4238 rows x 16 columns]>
```

In [10]:

```
df.describe()
```

Out[10]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	preva
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	4238.000000	4238
mean	0.429212	49.584946	1.978950	0.494101	9.003089	0.029630	0.005899	0
std	0.495022	8.572160	1.019791	0.500024	11.920094	0.169584	0.076587	0
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	0.000000	1
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.000000	1

In [11]:

```
df.isna().sum()
```

Loading [MathJax]/extensions/Safe.js

```
Out[11]: male                 0
         age                  0
         education          105
         currentSmoker        0
         cigsPerDay          29
         BPMeds              53
         prevalentStroke      0
         prevalentHyp         0
         diabetes             0
         totChol             50
         sysBP                0
         diaBP                0
         BMI                 19
         heartRate            1
         glucose            388
         TenYearCHD           0
         dtype: int64
```

```
In [12]: df['glucose'].fillna(value = df['glucose'].mean(),inplace=True)
```

```
In [13]: df['education'].fillna(value = df['education'].mean(),inplace=True)
```

```
In [14]: df['heartRate'].fillna(value = df['heartRate'].mean(),inplace=True)
```

```
In [15]: df['BMI'].fillna(value = df['BMI'].mean(),inplace=True)
```

```
In [16]: df['cigsPerDay'].fillna(value = df['cigsPerDay'].mean(),inplace=True)
```

```
In [17]: df['totChol'].fillna(value = df['totChol'].mean(),inplace=True)
```

```
In [18]: df['BPMeds'].fillna(value = df['BPMeds'].mean(),inplace=True)
```

```
In [19]: df.isna().sum()
```

```
Out[19]: 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 [20]: df.isna().sum()
```

Loading [MathJax]/extensions/Safe.js

Out[20]:

```
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 [21]:

```
#Splitting the dependent and independent variables.
x = df.drop("TenYearCHD",axis=1)
y = df['TenYearCHD']
```

In [22]:

```
x #checking the features
```

Out[22]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totC
0	1	39	4.0	0	0.0	0.00000	0	0	0	19
1	0	46	2.0	0	0.0	0.00000	0	0	0	25
2	1	48	1.0	1	20.0	0.00000	0	0	0	24
3	0	61	3.0	1	30.0	0.00000	0	1	0	22
4	0	46	3.0	1	23.0	0.00000	0	0	0	28
...	...	...	...	...	...	...	...	...	...	
4233	1	50	1.0	1	1.0	0.00000	0	1	0	31
4234	1	51	3.0	1	43.0	0.00000	0	0	0	20
4235	0	48	2.0	1	20.0	0.02963	0	0	0	24
4236	0	44	1.0	1	15.0	0.00000	0	0	0	21
4237	0	52	2.0	0	0.0	0.00000	0	0	0	26

4238 rows × 15 columns

Train test Split

In [23]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

In [24]:

```
y_train
```

Out[24]:

```
3252    0
3946    0
1261    0
2536    0
4089    0
       ..
3444    0
466     0
3092    0
3772    0
860     0
Name: TenYearCHD, Length: 3390, dtype: int64
```

KNN Classifier

In [25]:

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5, p=2, metric='minkowski')
knn.fit(x_train, y_train)
acc = knn.score(x_test,y_test)*100
print(acc)
```

```
83.13679245283019
```