In [1]:
```
# Aim: To perform and find the accuracy of K-Nearest Neighbors Algorithm i.e. KNN Classi
```

In [2]:
```
# Name : Kaushal A. Bharade
# class : 3rd year
# Section : A
# Roll No. : 11
```

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\\HP'
```

In [5]:
```
os.chdir ("C:\\Users\\HP\\Desktop\\BDA")
```

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()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        4133 non-null   float64
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64  
 7   prevalentHyp     4238 non-null   int64  
 8   diabetes         4238 non-null   int64  
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

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()
```

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)
```

Loading [MathJax]/extensions/Safe.js

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

In [21]:
```
x #Checking the features
```

Out[21]:

	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

Loading [MathJax]/extensions/Safe.js

Train Test Split

In [22]:
```python
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

In [23]:
```python
y_train
```

Out[23]:
```
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
```

In [24]:
```python
y_test
```

Out[24]:
```
3188    0
764     0
3264    0
1967    0
2185    0
       ..
3303    1
4056    0
4210    0
3971    0
2540    0
Name: TenYearCHD, Length: 848, dtype: int64
```

In [25]:
```python
x_train
```

Out[25]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totC
3252	1	40	4.0	1	30.0	0.0	0	0	0	20
3946	0	57	2.0	0	0.0	0.0	0	1	0	25
1261	0	47	1.0	0	0.0	0.0	0	0	0	23
2536	1	41	2.0	1	30.0	0.0	0	0	0	22
4089	0	64	1.0	0	0.0	0.0	0	1	0	23
...	...	...	...	...	...	...	...	...	...	
3444	0	36	1.0	1	5.0	0.0	0	1	0	22
466	0	57	3.0	1	15.0	0.0	0	0	0	25
3092	0	60	2.0	0	0.0	0.0	0	1	0	29
3772	1	39	2.0	1	10.0	0.0	0	0	0	21
860	0	35	2.0	0	0.0	0.0	0	0	0	24

3390 rows × 15 columns

In [26]:
```python
x_test
```

Out[26]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totC
3188	1	63	1.0	0	0.0	0.0	0	1	0	19
764	1	45	3.0	0	0.0	0.0	0	0	0	16
3264	0	51	1.0	1	2.0	0.0	0	0	0	26
1967	1	45	3.0	1	30.0	0.0	0	0	0	25
2185	0	45	2.0	1	3.0	0.0	0	0	0	25
...	...	...	...	...	...	...	...	...	...	
3303	1	47	1.0	0	0.0	0.0	0	0	0	25
4056	1	44	2.0	0	0.0	0.0	0	0	0	25
4210	1	50	1.0	0	0.0	0.0	0	0	0	28
3971	1	64	3.0	0	0.0	0.0	0	1	1	19
2540	1	55	3.0	1	20.0	0.0	0	0	0	21

848 rows × 15 columns

In [27]:
```
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
```

In [ ]: