

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

```
In [2]: # Name : Shriya Mechineni  
# class : 3rd year  
# Section : A  
# Roll No. : 49
```

```
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\\admin'
```

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

```
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
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
4238	1	40	3.0	0	0.0	0.0	0	1	0	18
4239	0	39	3.0	1	30.0	0.0	0	0	0	19

```
In [9]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4240 entries, 0 to 4239
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   male                  4240 non-null   int64
1   age                   4240 non-null   int64
2   education              4135 non-null   float64
3   currentSmoker         4240 non-null   int64
4   cigsPerDay             4211 non-null   float64
5   BPMeds                4187 non-null   float64
6   prevalentStroke       4240 non-null   int64
7   prevalentHyp          4240 non-null   int64
8   diabetes              4240 non-null   int64
9   totChol               4190 non-null   float64
10  sysBP                 4240 non-null   float64
11  diaBP                 4240 non-null   float64
12  BMI                   4221 non-null   float64
13  heartRate             4239 non-null   float64
14  glucose               3852 non-null   float64
15  TenYearCHD            4240 non-null   int64
dtypes: float64(9), int64(7)
memory usage: 530.1 KB
```

In [10]:

df.describe()

Out[10]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	preva
count	4240.000000	4240.000000	4135.000000	4240.000000	4211.000000	4187.000000	4240.000000	4240
mean	0.429245	49.580189	1.979444	0.494104	9.005937	0.029615	0.005896	0
std	0.495027	8.572942	1.019791	0.500024	11.922462	0.169544	0.076569	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)

```
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	totChol
0	1	39	4.0	0	0.0	0.000000	0	0	0	191
1	0	46	2.0	0	0.0	0.000000	0	0	0	253
2	1	48	1.0	1	20.0	0.000000	0	0	0	246
3	0	61	3.0	1	30.0	0.000000	0	1	0	234
4	0	46	3.0	1	23.0	0.000000	0	0	0	246
...	...	...	...	...	...	...	...	...	...	...
4235	0	48	2.0	1	20.0	0.029615	0	0	0	246
4236	0	44	1.0	1	15.0	0.000000	0	0	0	234
4237	0	52	2.0	0	0.0	0.000000	0	0	0	246
4238	1	40	3.0	0	0.0	0.000000	0	1	0	191
4239	0	39	3.0	1	30.0	0.000000	0	0	0	191

4240 rows × 15 columns

# Train Test Split

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

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

```
Out[23]: 1427      0
          3257      0
          3822      0
          1263      0
          3575      0
          ..
          3444      0
          466       0
          3092      0
          3772      0
          860       0
          Name: TenYearCHD, Length: 3392, dtype: int64
```

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

```
Out[24]: 1350      1
          1434      0
          2500      0
          1128      0
          4144      1
          ..
          1844      0
          4178      0
          4193      1
          2897      0
          910       0
          Name: TenYearCHD, Length: 848, dtype: int64
```

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

```
Out[25]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totC
1427	0	53	3.0	1	20.0	0.0	0	0	0	22
3257	0	64	4.0	1	6.0	0.0	0	1	0	23
3822	0	38	3.0	0	0.0	0.0	0	0	0	18
1263	0	49	1.0	0	0.0	0.0	0	0	0	27
3575	1	56	2.0	1	20.0	0.0	0	0	0	18
...	...	...	...	...	...	...	...	...	...	...
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

3392 rows × 15 columns

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

Out [26]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol
1350	0	49	3.0	1	10.0	0.0	0	0	0	260
1434	1	43	1.0	1	25.0	0.0	0	0	0	200
2500	1	45	1.0	1	1.0	0.0	0	1	0	270
1128	0	63	3.0	1	10.0	0.0	0	1	0	230
4144	1	59	2.0	0	0.0	0.0	0	0	0	230
...	...	...	...	...	...	...	...	...	...	...
1844	1	35	3.0	1	15.0	0.0	0	0	0	190
4178	1	41	3.0	1	30.0	0.0	0	0	0	210
4193	0	63	1.0	0	0.0	0.0	0	1	0	300
2897	0	45	1.0	0	0.0	0.0	0	0	0	290
910	1	39	1.0	0	0.0	0.0	0	0	0	220

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)
```

84.19811320754717

In [ ]: