

```
In [18]: #Name:Chaitali Mahlley
#Roll no: 47
#Sec:A
#Subject:Data Science and Statistics
```

```
In [19]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
```

```
In [20]: import os
```

```
In [21]: os.getcwd()
```

Out[21]: 'C:\\Users\\cmahl\\Desktop'

```
In [22]: os.chdir("C:\\Users\\cmahl\\Desktop")
```

```
In [23]: df=pd.read_csv("framingham.csv")
```

```
In [24]: df.head()
```

Out[24]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.1
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.1
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.1
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.1
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.1

```
In [25]: df.describe()
```

Out[25]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol
count	4240.000000	4240.000000	4135.000000	4240.000000	4211.000000	4187.000000	4240.000000	4240.000000	4240.000000	4190.000000
mean	0.429245	49.580189	1.979444	0.494104	9.005937	0.029615	0.005896	0.310613	0.025708	236.69952
std	0.495027	8.572942	1.019791	0.500024	11.922462	0.169544	0.076569	0.462799	0.158280	44.59128
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	107.00000
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	206.00000
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	234.00000
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	0.000000	1.000000	0.000000	263.00000
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.000000	1.000000	1.000000	696.00000

```
In [26]: 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 [27]: df.isna().sum()
```

```
Out[27]: 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 [28]: df['glucose'].fillna(value = df['glucose'].mean(),inplace=True)

df['education'].fillna(value = df['education'].mean(),inplace=True)

df['heartRate'].fillna(value = df['heartRate'].mean(),inplace=True)

df['BMI'].fillna(value = df['BMI'].mean(),inplace=True)

df['cigsPerDay'].fillna(value = df['cigsPerDay'].mean(),inplace=True)

df['totChol'].fillna(value = df['totChol'].mean(),inplace=True)

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

```
In [29]: df.isna().sum()
```

```
Out[29]: 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 [30]: x=df.drop("TenYearCHD",axis=1)
y=df['TenYearCHD']
```

```
In [31]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

```
In [32]: y_train
```

```
Out[32]: 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 [34]: from sklearn.linear_model import LogisticRegression
model = LogisticRegression().fit(x_train,y_train)
model.score(x_train, y_train)
```

```
Out[34]: 0.8487617924528302
```

## Decision Tree Algorithm

```
In [36]: from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
dtc.fit(x_train, y_train)
model.score(x_train, y_train)
cc = dtc.score(x_test, y_test)*100
print(acc)
```

```
97.99528301886792
```

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js