

Name - Gayatri Bhakare
Roll No - 10
Class - 3rd year
Section - CSE

```
In [1]: 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 [2]: os.getcwd()
```

```
Out[2]: 'C:\\\\Users\\sarth\\Data Analytics projects\\College Proj'
```

```
In [3]: os.chdir('D:\\\\Users\\SARTHAK\\Picture\\Desktop')
```

```
In [4]: df=pd.read_csv('framingham.csv')
```

```
In [5]: df.head()
```

```
Out[5]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
0	1	39	4.0	0	0.0	0.0	0	(
1	0	46	2.0	0	0.0	0.0	0	(
2	1	48	1.0	1	20.0	0.0	0	(
3	0	61	3.0	1	30.0	0.0	0	.
4	0	46	3.0	1	23.0	0.0	0	(



```
In [6]: 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 [8]: df.size
```

```
Out[8]: 67840
```

```
In [9]: df.shape
```

```
Out[9]: (4240, 16)
```

```
In [10]: df.isna().sum()
```

```
Out[10]: 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 [11]: df.describe()
```

```
Out[11]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	pre
count	4240.000000	4240.000000	4135.000000	4240.000000	4211.000000	4187.000000	
mean	0.429245	49.580189	1.979444	0.494104	9.005937	0.029615	
std	0.495027	8.572942	1.019791	0.500024	11.922462	0.169544	
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	

```
In [12]: x = df.drop("TenYearCHD",axis=1)
y = df['TenYearCHD']
```

```
In [13]: x
```

```
Out[13]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalent
0	1	39	4.0	0	0.0	0.0	0	
1	0	46	2.0	0	0.0	0.0	0	
2	1	48	1.0	1	20.0	0.0	0	
3	0	61	3.0	1	30.0	0.0	0	
4	0	46	3.0	1	23.0	0.0	0	
...	
4235	0	48	2.0	1	20.0	NaN	0	
4236	0	44	1.0	1	15.0	0.0	0	
4237	0	52	2.0	0	0.0	0.0	0	
4238	1	40	3.0	0	0.0	0.0	0	
4239	0	39	3.0	1	30.0	0.0	0	

4240 rows × 15 columns

```
In [14]: y
```

```
Out[14]: 0      0
          1      0
          2      0
          3      1
          4      0
          ..
        4235     0
        4236     0
        4237     0
        4238     0
        4239     0
        Name: TenYearCHD, Length: 4240, dtype: int64
```

Train - Test Splitting

```
In [15]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_s
```

```
In [16]: y_train
```

```
Out[16]: 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 [17]: y_test
```

```
Out[17]: 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 [19]: from sklearn.linear_model import LogisticRegression
from sklearn.impute import SimpleImputer

# Assuming x_train is your input data with missing values
# Replace strategy with 'mean', 'median', or 'most_frequent' as needed
imputer = SimpleImputer(strategy='mean')
x_train_imputed = imputer.fit_transform(x_train)

# Now you can fit your model
model = LogisticRegression().fit(x_train_imputed, y_train)
accuracy = model.score(x_train_imputed, y_train)
print("Training Accuracy:", accuracy)
```

Training Accuracy: 0.8484669811320755

```
In [20]: 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 [21]: print(type(H))
```

<class 'list'>

```
In [ ]:
```