In [1]:
```
#Aim:To perform and find the accuracy of Logistic Regression
```

In [ ]:
```
#Name: Achal Subhash Kharwade
#Roll No: 36
#Sec: B
#Date :09-10-2023
```

In [4]:
```
import pandas as pd
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')
import os
```

In [5]:
```
os.getcwd()
```

Out[5]:
```
'C:\\Users\\Lenovo\\DSS 5th Sem'
```

In [29]:
```
os.chdir("D:\DSS\DSS PRAC PG")
```

In [7]:
```
df=pd.read_csv("framingham.csv")
```

In [8]:
```
df.head()
```

Out[8]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRat
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.(
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.(
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.(
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.(
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.(

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

Out[9]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totCh
count	4240.000000	4240.000000	4135.000000	4240.000000	4211.000000	4187.000000	4240.000000	4240.000000	4240.000000	4190.00000
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 [10]:
```
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 [11]:
```python
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]:
```python
df
```

Out[12]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heart
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4235	0	48	2.0	1	20.0	NaN	0	0	0	248.0	131.0	72.0	22.00	
4236	0	44	1.0	1	15.0	0.0	0	0	0	210.0	126.5	87.0	19.16	
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0	133.5	83.0	21.47	
4238	1	40	3.0	0	0.0	0.0	0	1	0	185.0	141.0	98.0	25.60	
4239	0	39	3.0	1	30.0	0.0	0	0	0	196.0	133.0	86.0	20.91	

4240 rows × 16 columns

In [13]:
```python
# Missing value Treatment
# Since,'glucose' and 'education' columns hadd a significant amount of null values , so we replaced them with the
```

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

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

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

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

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

In [19]:
```python
df['totChol'].fillna(value = df['totChol'].mean(),inplace=True)
```

In [20]:

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

In [21]:

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

Out[21]:

```
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 [22]:

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

In [23]:

```
x #checking the features
```

Out[23]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heart
0	1	39	4.0	0	0.0	0.000000	0	0	0	195.0	106.0	70.0	26.97	
1	0	46	2.0	0	0.0	0.000000	0	0	0	250.0	121.0	81.0	28.73	
2	1	48	1.0	1	20.0	0.000000	0	0	0	245.0	127.5	80.0	25.34	
3	0	61	3.0	1	30.0	0.000000	0	1	0	225.0	150.0	95.0	28.58	
4	0	46	3.0	1	23.0	0.000000	0	0	0	285.0	130.0	84.0	23.10	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4235	0	48	2.0	1	20.0	0.029615	0	0	0	248.0	131.0	72.0	22.00	
4236	0	44	1.0	1	15.0	0.000000	0	0	0	210.0	126.5	87.0	19.16	
4237	0	52	2.0	0	0.0	0.000000	0	0	0	269.0	133.5	83.0	21.47	
4238	1	40	3.0	0	0.0	0.000000	0	1	0	185.0	141.0	98.0	25.60	
4239	0	39	3.0	1	30.0	0.000000	0	0	0	196.0	133.0	86.0	20.91	

4240 rows × 15 columns

In [24]:

```
#Train Test Split
```

In [25]:

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

In [26]:

```
y_train
```

Out[26]:

```
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 [27]:

```
#Logistic Regression Algorithm
```

In [28]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LogisticRegression
model=LogisticRegression().fit(x_train,y_train)
model.score(x_train,y_train)
```

Out[28]:

```
0.8484669811320755
```

In [ ]: