Aim : To perform and find accuracy of Logistic Regression.

In [1]:
```
#Name : Rajshri Kirandas Satpute
#Roll no. : 55
#Section:B
#Year:3rd Year
#Date : 09/10/2023
```

In [2]:
```
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 [3]:
```
os.getcwd()
```

Out[3]:
```
'C:\\Users\\HP'
```

In [4]:
```
os.chdir('C:\\Users\\HP\\Desktop')
```

In [5]:
```
df=pd.read_csv('framingham.csv')
```

In [6]:
```
df.head()
```

Out[6]:

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

In [7]:
```
df.tail()
```

Out[7]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	c
4235	0	48	2.0	1	20.0	NaN	0	0	
4236	0	44	1.0	1	15.0	0.0	0	0	
4237	0	52	2.0	0	0.0	0.0	0	0	
4238	1	40	3.0	0	0.0	0.0	0	1	
4239	0	39	3.0	1	30.0	0.0	0	0	

In [8]:
```
df.info
```

Out[8]:
```
<bound method DataFrame.info of       male  age  education  currentSmoker  cig
sPerDay  BPMeds  \
0        1   39        4.0              0         0.0      0.0
1        0   46        2.0              0         0.0      0.0
2        1   48        1.0              1        20.0      0.0
3        0   61        3.0              1        30.0      0.0
4        0   46        3.0              1        23.0      0.0
...    ...  ...        ...            ...         ...      ...
4235     0   48        2.0              1        20.0      NaN
4236     0   44        1.0              1        15.0      0.0
4237     0   52        2.0              0         0.0      0.0
4238     1   40        3.0              0         0.0      0.0
4239     0   39        3.0              1        30.0      0.0

      prevalentStroke  prevalentHyp  diabetes  totChol  sysBP  diaBP    BMI  \
0                   0             0         0    195.0  106.0   70.0  26.97
1                   0             0         0    250.0  121.0   81.0  28.73
2                   0             0         0    245.0  127.5   80.0  25.34
3                   0             1         0    225.0  150.0   95.0  28.58
4                   0             0         0    285.0  130.0   84.0  23.10
...               ...           ...       ...      ...    ...    ...    ...
4235                0             0         0    248.0  131.0   72.0  22.00
4236                0             0         0    210.0  126.5   87.0  19.16
4237                0             0         0    269.0  133.5   83.0  21.47
4238                0             1         0    185.0  141.0   98.0  25.60
4239                0             0         0    196.0  133.0   86.0  20.91

      heartRate  glucose  TenYearCHD
0          80.0     77.0           0
1          95.0     76.0           0
2          75.0     70.0           0
3          65.0    103.0           1
4          85.0     85.0           0
...         ...      ...         ...
4235       84.0     86.0           0
4236       86.0      NaN           0
4237       80.0    107.0           0
4238       67.0     72.0           0
4239       85.0     80.0           0

[4240 rows x 16 columns]>
```

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

Out[9]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalent
count	4240.000000	4240.000000	4135.000000	4240.000000	4211.000000	4187.000000	4240.0
mean	0.429245	49.580189	1.979444	0.494104	9.005937	0.029615	0.0
std	0.495027	8.572942	1.019791	0.500024	11.922462	0.169544	0.0
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	0.0
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	0.0
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	0.0
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	0.0
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.0

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

In [12]:
```
df['education'].fillna(value = df['education'].mean(),inplace=True)
```

In [13]:
```
df['heartRate'].fillna(value = df['heartRate'].mean(),inplace=True)
```

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

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

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

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

In [18]:
```
df.isna().sum()
```

Out[18]:
```
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 [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	d
0	1	39	4.0	0	0.0	0.000000	0	0	
1	0	46	2.0	0	0.0	0.000000	0	0	
2	1	48	1.0	1	20.0	0.000000	0	0	
3	0	61	3.0	1	30.0	0.000000	0	1	
4	0	46	3.0	1	23.0	0.000000	0	0	
...	...	...	...	...	...	...	...	...	...
4235	0	48	2.0	1	20.0	0.029615	0	0	
4236	0	44	1.0	1	15.0	0.000000	0	0	
4237	0	52	2.0	0	0.0	0.000000	0	0	
4238	1	40	3.0	0	0.0	0.000000	0	1	
4239	0	39	3.0	1	30.0	0.000000	0	0	

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_sta
```

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
```

Logistic Regression Algorithm

In [24]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression().fit(x_train,y_train)
model.score(x_train, y_train)
```

Out[24]:
```
0.8484669811320755
```

In [ ]:
```

```