Logistic Regression

In [4]:

```
EXP : 10
```

In [6]:

```
#Aim : To perform and find the accuracy of Logistic Regression
```

In [8]:

```
#Name: Leena Rajeshwar Kale
#Roll No.:71
#Sec: C
#Subject:ET - 1
```

In [1]:

```
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')
```

In [2]:

```
import os
```

In [3]:

```
os.getcwd()
```

Out[3]:

```
'C:\\Users\\dishi\\Downloads'
```

In [4]:

```
os.chdir("C:\\Users\\dishi\\Downloads")
```

In [19]:

```
df=pd.read_csv("framingham.csv")
```

In [20]:

```
df.head()
```

Out[20]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	dia
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 [21]:

```
df.tail()
```

Out[21]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
4233	1	50	1.0	1	1.0	0.0	0	1
4234	1	51	3.0	1	43.0	0.0	0	0
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

In [22]:

```
df
```

Out[22]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
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
...	...	...	...	...	...	...	...	...
4233	1	50	1.0	1	1.0	0.0	0	1
4234	1	51	3.0	1	43.0	0.0	0	0
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 rows × 16 columns

train test split

In [23]:

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

In [24]:

```
x
```

Out[24]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
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
...	...	...	...	...	...	...	...	...
4233	1	50	1.0	1	1.0	0.0	0	1
4234	1	51	3.0	1	43.0	0.0	0	0
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 rows × 15 columns

In [25]:

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

In [69]:

```
y_train
```

Out[69]:

```
3252    0
3946    0
1261    0
2536    0
4089    0
       ..
3444    0
466     0
3092    0
3772    0
860     0
Name: TenYearCHD, Length: 3390, dtype: int64
```

Logistic Regression Algorithm

In [74]:

```
# Drop rows with missing values
x_train_clean = x_train.dropna()

# Ensure that the target variable (y_train) is also filtered accordingly
y_train_clean = y_train[x_train_clean.index]

# Fit the Logistic Regression model on the cleaned data
model = LogisticRegression().fit(x_train_clean, y_train_clean)

# Check the model's score
model_score = model.score(x_train_clean, y_train_clean)
print(model_score)
```

```
0.8463389480921278
```

In [76]:

```
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression
```

```python
# Create an imputer for replacing missing values with the mean of the column
imputer = SimpleImputer(strategy='mean')

# Fit the imputer on the training data and transform it
x_train_imputed = imputer.fit_transform(x_train)

# Fit the Logistic Regression model on the imputed data
model = LogisticRegression().fit(x_train_imputed, y_train)

# Check the model's score
model_score = model.score(x_train_imputed, y_train)
print(model_score)
```

```
0.8492625368731563
```