In [17]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [18]:
```
from sklearn.linear_model import LogisticRegression
```

In [19]:
```
df=pd.read_csv("framingham.csv").dropna()
df
```

0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.0	7
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.0	7
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.0	7
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.0	10
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.0	8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4231	1	58	3.0	0	0.0	0.0	0	1	0	187.0	141.0	81.0	24.96	80.0	8
4232	1	68	1.0	0	0.0	0.0	0	1	0	176.0	168.0	97.0	23.14	60.0	7
4233	1	50	1.0	1	1.0	0.0	0	1	0	313.0	179.0	92.0	25.97	66.0	8
4234	1	51	3.0	1	43.0	0.0	0	0	0	207.0	126.5	80.0	19.71	65.0	6
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0	133.5	83.0	21.47	80.0	10

3656 rows × 16 columns

In [20]:
```
df.dropna(inplace=True)
```

In [21]:
```
df.info()
```

```
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             3656 non-null   int64  
 1   age              3656 non-null   int64  
 2   education        3656 non-null   float64
 3   currentSmoker    3656 non-null   int64  
 4   cigsPerDay       3656 non-null   float64
 5   BPMeds           3656 non-null   float64
 6   prevalentStroke  3656 non-null   int64  
 7   prevalentHyp     3656 non-null   int64  
 8   diabetes         3656 non-null   int64  
 9   totChol          3656 non-null   float64
 10  sysBP            3656 non-null   float64
 11  diaBP            3656 non-null   float64
 12  BMI              3656 non-null   float64
 13  heartRate        3656 non-null   float64
 14  glucose          3656 non-null   float64
 15  TenYearCHD       3656 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 485.6 KB
```

In [22]:
```
Smoker','cigsPerDay','BPMeds','prevalentStroke','prevalentHyp','diabetes','totChol','sysBP','diaBP','BMI','heartRate','glucose'
```

In [23]:
```
feature_matrix.shape
```

Out[23]:
```
(3656, 15)
```

In [24]:
```
target_vector.shape
```

Out[24]:
```
(3656,)
```

In [25]:
```
from sklearn.preprocessing import StandardScaler
```

In [26]:
```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [27]:
```
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

Out[27]:
```
LogisticRegression()
```

In [28]:
```
feature_matrix.shape
```

Out[28]:
```
(3656, 15)
```

In [29]:
```
target_vector.shape
```

Out[29]:
```
(3656,)
```

In [30]:
```
from sklearn.preprocessing import StandardScaler
```

In [31]:
```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [32]:
```
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

Out[32]:
```
LogisticRegression()
```

In [33]:
```
tSmoker','cigsPerDay','BPMeds','prevalentStroke','prevalentHyp','diabetes','totChol','sysBP','diaBP','BMI','heartRate','glucose
```

In [34]:
```
prediction = logr.predict(observation)
prediction
```

Out[34]:
```
array([1, 1, 1, ..., 1, 1, 1], dtype=int64)
```

In [35]:
```
logr.classes_
```

Out[35]:
```
array([0, 1], dtype=int64)
```

In [36]:
```
logr.predict_proba(observation)[0][1]
```

Out[36]:
```
1.0
```