In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

In [2]:
```
df=pd.read_csv(r"C4_framingham.csv")
df
```

Out[2]:

| | 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

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        4133 non-null   float64
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64  
 7   prevalentHyp     4238 non-null   int64  
 8   diabetes         4238 non-null   int64  
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

In [4]:
```
df=df.dropna()
```

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

Out[5]:

| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalen |
|---|---|---|---|---|---|---|---|
| **count** | 3656.000000 | 3656.000000 | 3656.000000 | 3656.000000 | 3656.000000 | 3656.000000 | 3656 |
| **mean** | 0.443654 | 49.557440 | 1.979759 | 0.489059 | 9.022155 | 0.030361 | 0 |
| **std** | 0.496883 | 8.561133 | 1.022657 | 0.499949 | 11.918869 | 0.171602 | 0 |
| **min** | 0.000000 | 32.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0 |
| **25%** | 0.000000 | 42.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0 |
| **50%** | 0.000000 | 49.000000 | 2.000000 | 0.000000 | 0.000000 | 0.000000 | 0 |
| **75%** | 1.000000 | 56.000000 | 3.000000 | 1.000000 | 20.000000 | 0.000000 | 0 |
| **max** | 1.000000 | 70.000000 | 4.000000 | 1.000000 | 70.000000 | 1.000000 | 1 |

In [7]:
```
df.columns
```

Out[7]:
```
Index(['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
       'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
       'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD'],
      dtype='object')
```

In [8]:
```
df1=df[['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
       'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
       'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD']]
```

In [20]:
```
x=df1[['age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
       'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
       'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD']]
y=df1['male']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
lr=LogisticRegression()
lr.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:
763: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
```

Out[20]:
```
LogisticRegression()
```

In [21]:
```
lr.predict(x_test)
```

Out[21]:
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [22]:
```
lr.score(x_test,y_test)
```

Out[22]:
```
0.6918869644484958
```

In [23]:
```
from sklearn.preprocessing import StandardScaler
fs=StandardScaler().fit_transform(x)
logr=LogisticRegression()
logr.fit(fs,y)
```

Out[23]:
```
LogisticRegression()
```

In [24]:
```
o=[[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]]
prediction=logr.predict(o)
print(prediction)
```

```
[1]
```

In [25]:
```
logr.classes_
```

Out[25]:
```
array([0, 1], dtype=int64)
```

In [26]:
```
logr.predict_proba(o)[0][0]
```

Out[26]:
```
0.023655703523128735
```

In [27]:
```
logr.predict_proba(o)[0][1]
```

Out[27]:
```
0.9763442964768713
```

In [ ]:
```

```