

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

```
In [2]: from sklearn.linear_model import LogisticRegression
```

```
In [17]: df=pd.read_csv(r"E:\154\C4_framingham - C4_framingham.csv").dropna()
df
```

Out[17]:

	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
...
4231	1	58	3.0	0	0.0	0.0	0	1
4232	1	68	1.0	0	0.0	0.0	0	1
4233	1	50	1.0	1	1.0	0.0	0	1
4234	1	51	3.0	1	43.0	0.0	0	0
4237	0	52	2.0	0	0.0	0.0	0	0

3656 rows × 16 columns



```
In [18]: df.head()
```

Out[18]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	di
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 [19]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3656 entries, 0 to 4237
Data columns (total 16 columns):
#   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 [20]: df.describe()

Out[20]:

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

In [21]: df.columns

Out[21]: Index(['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds', 'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP', 'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD'], dtype='object')

```
In [22]: feature_matrix = df.iloc[:,0:15]
        target_vector = df.iloc[:, -1]
```

```
In [23]: fs=StandardScaler().fit_transform(feature_matrix)
        logr=LogisticRegression()
        logr.fit(fs,target_vector)
```

Out[23]: LogisticRegression()

```
In [24]: observation=[[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]]
```

```
In [25]: prediction=logr.predict(observation)
        print(prediction)
```

[1]

```
In [26]: logr.classes_
```

Out[26]: array([0, 1], dtype=int64)

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

Out[27]: 0.0002214783507201723

```
In [28]: logr.predict_proba(observation)[0][1]
```

Out[28]: 0.9997785216492798

```
In [ ]:
```