

# Logistic Regression

Exp no.: 10

Aim: Logistic Regression

```
In [ ]: #Name:Nikhil kakar  
#Roll no.: 52  
#Sec: A  
#Aim:SVM Classifier  
#Year:3rd Year
```

```
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\\Downloads'
```

```
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
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	prevalenti
4233	1	50	1.0	1	1.0	0.0	0	
4234	1	51	3.0	1	43.0	0.0	0	
4235	0	48	2.0	1	20.0	NaN	0	
4236	0	44	1.0	1	15.0	0.0	0	
4237	0	52	2.0	0	0.0	0.0	0	



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

```
Out[8]: <bound method DataFrame.info of
cigsPerDay  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
...       ...    ...      ...        ...         ...         ...
4233        1    50      1.0          1          1.0          0.0
4234        1    51      3.0          1         43.0          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

prevalentStroke  prevalentHyp  diabetes  totChol  sysBP  diaBP  BM
I  \
0              0              0          0     195.0   106.0   70.0  26.9
7
1              0              0          0     250.0   121.0   81.0  28.7
3
2              0              0          0     245.0   127.5   80.0  25.3
4
3              0              1          0     225.0   150.0   95.0  28.5
8
4              0              0          0     285.0   130.0   84.0  23.1
0
...           ...           ...        ...         ...         ...
...
4233          0              1          0     313.0   179.0   92.0  25.9
7
4234          0              0          0     207.0   126.5   80.0  19.7
1
4235          0              0          0     248.0   131.0   72.0  22.0
0
4236          0              0          0     210.0   126.5   87.0  19.1
6
4237          0              0          0     269.0   133.5   83.0  21.4
7

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
...       ...     ...        ...
4233      66.0     86.0          1
4234      65.0     68.0          0
4235      84.0     86.0          0
4236      86.0      NaN          0
4237      80.0    107.0          0

[4238 rows x 16 columns]>
```

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

```
Out[9]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	4185.000000
mean	0.429212	49.584946	1.978950	0.494101	9.003089	0.029630	0.029630
std	0.495022	8.572160	1.019791	0.500024	11.920094	0.169584	0.169584
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 [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)
```

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

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

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

```
In [17]: df.isna().sum()
```

```
Out[17]: male                0
         age                  0
         education            0
         currentSmoker        0
         cigsPerDay           29
         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 [18]: df.isna().sum()
```

```
Out[18]: male                0
         age                  0
         education            0
         currentSmoker        0
         cigsPerDay           29
         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]: #Splitting the dependent and independent variables.
         x = df.drop("TenYearCHD",axis=1)
         y = df['TenYearCHD']
```

```
In [20]: x #checking the features
```

```
Out[20]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentH
0	1	39	4.0	0	0.0	0.00000	0	
1	0	46	2.0	0	0.0	0.00000	0	
2	1	48	1.0	1	20.0	0.00000	0	
3	0	61	3.0	1	30.0	0.00000	0	
4	0	46	3.0	1	23.0	0.00000	0	
...	...	...	...	...	...	...	...	...
4233	1	50	1.0	1	1.0	0.00000	0	
4234	1	51	3.0	1	43.0	0.00000	0	
4235	0	48	2.0	1	20.0	0.02963	0	
4236	0	44	1.0	1	15.0	0.00000	0	
4237	0	52	2.0	0	0.0	0.00000	0	

4238 rows × 15 columns



## Train Test Split

```
In [21]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_s
```

```
In [22]: y_train
```

```
Out[22]:
```

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

```
-----
ValueError                                Traceback (most recent call last)
```

```
Cell In[23], line 2
```

```
      1 from sklearn.linear_model import LogisticRegression
----> 2 model = LogisticRegression().fit(x_train,y_train)
      3 model.score(x_train, y_train)
```

```
File ~\anaconda3\Lib\site-packages\sklearn\base.py:1151, in _fit_context
t.<locals>.decorator.<locals>.wrapper(estimator, *args, **kwargs)
```

```
    1144 estimator._validate_params()
    1146 with config_context(
    1147     skip_parameter_validation=(
    1148         prefer_skip_nested_validation or global_skip_validation
    1149     )
    1150 ):
-> 1151     return fit_method(estimator, *args, **kwargs)
```

```
File ~\anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:140
```