

Logistic Regression

```
In [1]: #Name:janhavi Nitin warghade  
#Roll no.:69  
#Sec :3c  
#Sub:DSS
```

```
In [2]: #Aim :To perform Logistic regression
```

```
In [3]: 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 [4]: import os
```

```
In [5]: os.getcwd()
```

```
Out[5]: 'C:\\Users\\DELL'
```

```
In [19]: df=pd.read_csv("C:\\Users\\DELL\\OneDrive\\Documents\\framingham.csv")
```

```
In [20]: df.head()
```

```
Out[20]:
```

	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 [21]: df.tail()
```

```
Out[21]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	di
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.shape
```

```
Out[22]: (4238, 16)
```

```
In [23]: df.size
```

```
Out[23]: 67808
```

```
In [24]: df.ndim
```

```
Out[24]: 2
```

```
In [25]: 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 [26]: df.describe()
```

```
Out[26]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevale
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	423
mean	0.429212	49.584946	1.978950	0.494101	9.003089	0.029630	
std	0.495022	8.572160	1.019791	0.500024	11.920094	0.169584	
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	

```
In [27]: df.isnull().sum()
```

```
Out[27]: 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
```

Missing value treatment

```
In [32]: df['glucose'].fillna(value =df['glucose'].mean(),inplace=True)
```

```
In [33]: df['education'].fillna(value =df['education'].mean(),inplace=True)
```

```
In [34]: df['heartRate'].fillna(value =df['heartRate'].mean(),inplace=True)
```

```
In [35]: df['BMI'].fillna(value =df['BMI'].mean(),inplace=True)
```

```
In [37]: df['cigsPerDay'].fillna(value =df['cigsPerDay'].mean(),inplace=True)
```

```
In [38]: df['totChol'].fillna(value =df['totChol'].mean(),inplace=True)
```

```
In [39]: df['BPMeds'].fillna(value =df['BPMeds'].mean(),inplace=True)
```

```
In [40]: df.isna().sum()
```

```
Out[40]: male          0
age          0
education     0
currentSmoker 0
cigsPerDay    0
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 [41]: x=df.drop("TenYearCHD",axis=1)
y=df['TenYearCHD']
```

```
In [42]: x
```

```
Out[42]:
```

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

4238 rows × 15 columns



In [43]:

y

Out[43]:

0	0
1	0
2	0
3	1
4	0

..

4233	1
4234	0
4235	0
4236	0
4237	0

Name: TenYearCHD, Length: 4238, dtype: int64

Train test split

In [44]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=2,random_state=
```

In [45]:

y_train

Out[45]:

3264	0
1967	0
2185	0
393	0
2333	1

..

3444	0
466	0
3092	0
3772	0
860	0

Name: TenYearCHD, Length: 4236, dtype: int64

Logistic Regression algorithm

In [46]:

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

Out[46]: 0.8508026440037771

In []: