

LogisticRegression1

```
In [2]: import numpy as np
import pandas as pd
```

```
In [3]: df=pd.read_csv(r"C:\Users\user\Downloads\C4_framingham - C4_framingham.csv")
df
```

Out[3]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	c
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 [5]: 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 [6]: df1=df.fillna(0)
df1

Out[6]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	c
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	0.0	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 [7]: df1.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              4238 non-null   float64 
3   currentSmoker          4238 non-null   int64  
4   cigsPerDay             4238 non-null   float64 
5   BPMeds                 4238 non-null   float64 
6   prevalentStroke        4238 non-null   int64  
7   prevalentHyp           4238 non-null   int64  
8   diabetes               4238 non-null   int64  
9   totChol                4238 non-null   float64 
10  sysBP                  4238 non-null   float64 
11  diaBP                  4238 non-null   float64 
12  BMI                    4238 non-null   float64 
13  heartRate              4238 non-null   float64 
14  glucose                4238 non-null   float64 
15  TenYearCHD             4238 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

In [9]: `from sklearn.linear_model import LogisticRegression`

In [10]: `feature_matrix = df1.iloc[:,0:5]`
`target_vector = df1.iloc[:,1]`

In [11]: `feature_matrix.shape`

Out[11]: (4238, 5)

In [12]: `target_vector.shape`

Out[12]: (4238,)

In [13]: `from sklearn.preprocessing import StandardScaler`

In [14]: `fs=StandardScaler().fit_transform(feature_matrix)`

```
In [15]: logr=LogisticRegression()  
logr.fit(fs,target_vector)
```

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model_logistic.py:76
3: 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://scikit-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-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
n_iter_i = _check_optimize_result(

```
Out[15]: LogisticRegression()
```

```
In [16]: observation=[[5,7,9,5,6]]
```

```
In [17]: prediction=logr.predict(observation)  
print(prediction)
```

```
[67]
```

```
In [18]: logr.classes_
```

```
Out[18]: array([32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48,  
                49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65,  
                66, 67, 68, 69, 70], dtype=int64)
```

```
In [19]: logr.predict_proba(observation)[0][0]
```

```
Out[19]: 3.5148424147070445e-35
```

```
In [20]: logr.predict_proba(observation)[0][0]
```

```
Out[20]: 3.5148424147070445e-35
```

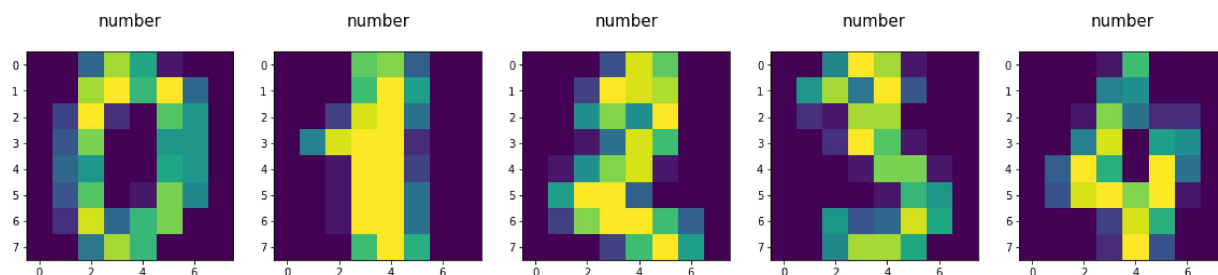
LogisticRegression2

```
In [21]: import re  
from sklearn.datasets import load_digits  
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.model_selection import train_test_split
```

```
In [22]: digits = load_digits()
digits
```

```
Out[22]: {'data': array([[ 0.,  0.,  5., ...,  0.,  0.,  0.],
 [ 0.,  0.,  0., ..., 10.,  0.,  0.],
 [ 0.,  0.,  0., ..., 16.,  9.,  0.],
 ...,
 [ 0.,  0.,  1., ...,  6.,  0.,  0.],
 [ 0.,  0.,  2., ..., 12.,  0.,  0.],
 [ 0.,  0., 10., ..., 12.,  1.,  0.])),
 'target': array([0, 1, 2, ..., 8, 9, 8]),
 'frame': None,
 'feature_names': ['pixel_0_0',
 'pixel_0_1',
 'pixel_0_2',
 'pixel_0_3',
 'pixel_0_4',
 'pixel_0_5',
 'pixel_0_6',
 'pixel_0_7',
 'pixel_1_0',
 'pixel_1_1',
 ...,
 'pixel_1_0',
 'pixel_1_1']
```

```
In [23]: plt.figure(figsize=(20,4))
for index,(image,label) in enumerate(zip(digits.data[0:5],digits.target[0:8])):
    plt.subplot(1,5,index+1)
    plt.imshow(np.reshape(image,(8,8)))
    plt.title("number\n"%label,fontsize=15)
```



```
In [24]: x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0.2)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(1257, 64)
(540, 64)
(1257,)
(540,)
```

```
In [25]: logre=LogisticRegression()
logre.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://scikit-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-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
 n_iter_i = _check_optimize_result(

```
Out[25]: LogisticRegression()
```

```
In [26]: print(logre.predict(x_test))
```

```
[2 6 3 4 1 6 8 9 4 5 1 8 2 5 0 3 9 8 6 8 7 5 4 8 7 5 1 6 2 9 4 3 7 3 2 0 4
 7 7 4 3 4 5 2 5 1 8 1 2 8 1 6 4 1 5 9 0 3 4 0 6 9 3 9 1 4 2 9 7 6 7 7 5 5
 3 5 8 7 0 9 2 9 2 3 3 9 3 3 0 3 3 1 4 5 9 1 7 4 3 5 1 7 0 8 9 2 4 2 9 4 9
 8 8 8 8 0 7 0 7 6 1 2 5 6 1 8 4 5 6 9 4 0 9 2 7 7 6 4 5 9 6 6 4 7 0 7 5 1
 5 1 9 7 8 3 9 0 1 5 1 4 9 7 0 6 7 2 0 6 2 6 1 9 9 9 2 5 1 1 1 1 7 1 8 7 3
 6 0 3 2 8 7 8 3 8 1 6 1 6 0 2 5 8 8 2 5 0 1 7 6 5 5 2 3 7 4 9 7 5 8 5 7 1
 0 0 3 6 6 7 1 9 7 1 6 3 6 8 9 3 3 2 5 6 4 0 4 5 2 8 4 5 6 9 5 1 4 9 6 3 0
 8 1 2 1 4 4 4 2 3 0 2 3 6 2 9 0 8 3 5 7 5 8 5 7 7 8 0 9 4 5 8 5 8 9 1 6 8
 4 3 4 9 8 7 7 8 3 1 4 4 4 3 5 1 5 5 5 9 0 7 1 8 2 9 2 3 2 1 6 1 7 3 5 3 1
 7 9 4 3 0 6 3 2 2 1 6 6 4 1 8 4 8 8 1 4 9 1 0 0 6 1 7 0 4 7 6 5 6 0 4 3 9
 5 0 5 5 8 8 6 7 0 7 1 4 8 7 5 2 3 3 1 8 0 7 5 3 9 3 0 0 7 2 8 4 3 1 6 4 5
 3 3 2 9 7 9 0 8 6 5 8 2 6 1 6 9 1 8 0 2 2 8 0 6 8 9 8 7 3 8 8 0 1 2 9 3 3
 9 3 4 1 8 6 2 5 0 6 4 0 8 3 5 0 4 4 6 7 3 3 4 7 1 1 1 4 6 6 4 3 5 4 3 4 5
 9 2 6 7 7 0 6 6 9 3 4 7 5 9 8 2 0 8 5 0 7 1 8 0 4 8 4 3 8 0 3 0 1 9 5 1 0
 5 8 0 6 9 3 7 2 3 2 3 7 3 9 8 5 7 9 4 2 3 3]
```

```
In [27]: print(logre.score(x_test,y_test))
```

```
0.9666666666666667
```