

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Matplotlib is building the font cache; this may take a moment.

```
In [90]: from sklearn.linear_model import LogisticRegression
```

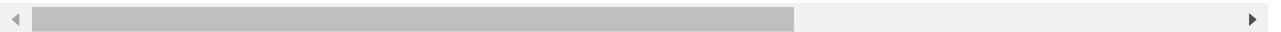
```
In [203]: df=pd.read_csv(r"C:\Users\user\Downloads\C4_framingham.csv")
```

```
In [204]: df
```

Out[204]:

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

4238 rows × 10 columns

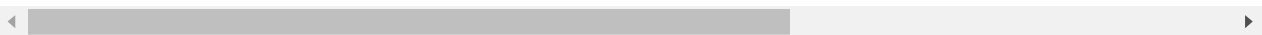


```
In [223]: data=df.dropna()
data
```

Out[223]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes
0	1	39	4.0	0	0.0	0.0	0	0	0
1	0	46	2.0	0	0.0	0.0	0	0	0
2	1	48	1.0	1	20.0	0.0	0	0	0
3	0	61	3.0	1	30.0	0.0	0	1	0
4	0	46	3.0	1	23.0	0.0	0	0	0
...	...	...	...	...	...	...	...	...	...
4231	1	58	3.0	0	0.0	0.0	0	1	0
4232	1	68	1.0	0	0.0	0.0	0	1	0
4233	1	50	1.0	1	1.0	0.0	0	1	0
4234	1	51	3.0	1	43.0	0.0	0	0	0
4237	0	52	2.0	0	0.0	0.0	0	0	0

3656 rows × 16 columns



In [ ]:

In [ ]:

```
In [224]: data.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 [226]: feature_matrix = data.iloc[:,0:16]
          target_vector = data.iloc[:, -1]
```

```
In [227]: feature_matrix.shape
```

```
Out[227]: (3656, 16)
```

```
In [228]: target_vector.shape
```

```
Out[228]: (3656,)
```

```
In [ ]:
```

```
In [229]: from sklearn.preprocessing import StandardScaler
```

```
In [230]: fs = StandardScaler().fit_transform(feature_matrix)
```

```
In [231]: logr=LogisticRegression()
```

```
In [232]: logr.fit(fs,target_vector)
```

```
Out[232]: LogisticRegression()
```

```
In [236]: observation = [[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]]
```

```
In [237]: prediction =logr.predict(observation)
          print(prediction)
```

```
[1]
```

```
In [238]: logr.classes_
```

```
Out[238]: array([0, 1], dtype=int64)
```

```
In [241]: logr.predict_proba(observation)[0][0]
```

```
Out[241]: 0.0
```

```
In [242]: logr.predict_proba(observation)[0][1]
```

```
Out[242]: 1.0
```

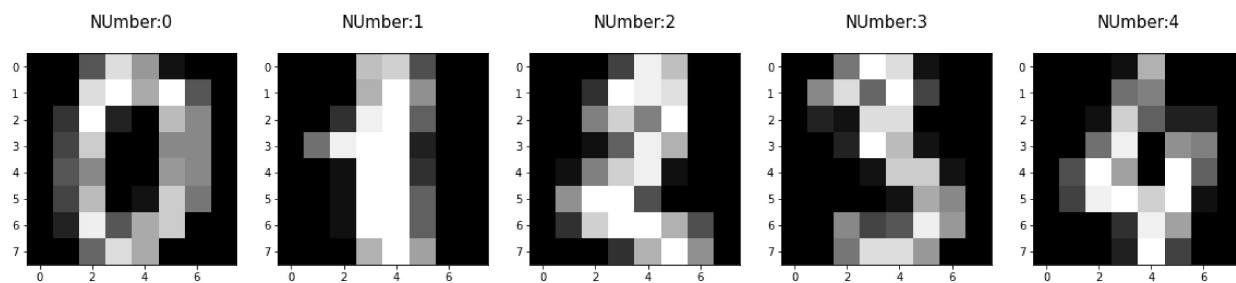
## Logistic REgression 2

```
In [172]: import re
from sklearn.datasets import load_digits
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

```
In [173]: digits=load_digits()
digits
```

```
['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_2',
 'pixel_1_3',
 'pixel_1_4',
 'pixel_1_5',
 'pixel_1_6',
 'pixel_1_7',
 'pixel_2_0',
 'pixel_2_1',
 'pixel_2_2',
 'pixel_2_3',
 'pixel_2_4',
 'pixel_2_5',
```

```
In [174]: plt.figure(figsize=(20,4))
for index,(image,label) in enumerate(zip(digits.data[0:5],digits.target[0:5])):
    plt.subplot(1,5,index+1)
    plt.imshow(np.reshape(image,(8,8)),cmap=plt.cm.gray)
    plt.title("NUmber:%i\n"%label,fontsize=15)
```



```
In [175]: x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0.30)
```

```
In [176]: print(x_train.shape)
          print(x_test.shape)
          print(y_train.shape)
          print(y_test.shape)
```

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

```
In [177]: logr=LogisticRegression(max_iter=100000)
```

```
In [178]: logr.fit(x_train,y_train)
```

```
Out[178]: LogisticRegression(max_iter=100000)
```

```
In [88]: print(logr.predict(x_test))
```

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

```
In [89]: print(logr.score(x_test,y_test))
```

```
0.9648148148148148
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```