In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

In [3]:
```
dftrain=pd.read_csv(r"C:\USERS\user\Downloads\C4_framingham - C4_framingham.cs
```

Out[3]:

	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
...	...	...	...	...	...	...	...	...
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 [4]:

Out[4]:
```
Index(['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
       'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
       'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD'],
      dtype='object')
```

In [6]:
```
a=dftrain[['male','age','education','currentSmoker','cigsPerDay','BPMeds','pre
```

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
...	...	...	...	...	...	...	...	...
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 × 9 columns

In [7]:
```
b=dftrain.head(10)
```

Out[7]:

	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	
5	0	43	2.0	0	0.0	0.0	0	1	
6	0	63	1.0	0	0.0	0.0	0	0	
7	0	45	2.0	1	20.0	0.0	0	0	
8	1	52	1.0	0	0.0	0.0	0	1	
9	1	43	1.0	1	30.0	0.0	0	1	

In [8]:
```
a=b[['male','age','education','currentSmoker','cigsPerDay','BPMeds','prevalent
```

Out[8]:

	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	
5	0	43	2.0	0	0.0	0.0	0	1	
6	0	63	1.0	0	0.0	0.0	0	0	
7	0	45	2.0	1	20.0	0.0	0	0	
8	1	52	1.0	0	0.0	0.0	0	1	
9	1	43	1.0	1	30.0	0.0	0	1	

In [9]:
```
c=a.iloc[:,0:9]
```

In [10]:

Out[10]:
```
(10, 9)
```

In [11]:

Out[11]:
```
(10,)
```

In [12]:

In [13]:

In [14]:
```
logr=LogisticRegression()
```

Out[14]:
```
LogisticRegression()
```

In [15]:

In [16]:
```
prediction=logr.predict(observation)
```

Out[16]:
```
array([1], dtype=int64)
```

In [17]:

Out[17]:
```
array([0, 1], dtype=int64)
```

In [18]:

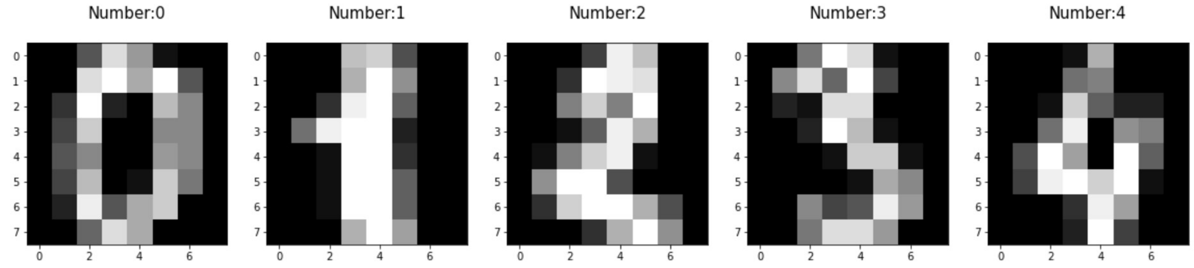
Out[18]:
```
0.04613373500012041
```

In [19]:
```
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
```

In [20]:
```
digits=load_digits()
```

Out[20]:
```
{'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',
```

In [21]:
```
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)
```



In [22]:

In [23]:
```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
```

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

In [24]:
```
logre=LogisticRegression(max_iter=10000)
```

Out[24]:
```
LogisticRegression(max_iter=10000)
```

In [25]:

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

In [26]:

```
0.9629629629629629
```

In [ ]: