

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C2_train.gender_submission.csv")
a
```

Out[3]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.250
1	2	1	1Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.283
2	3	1	3Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925
3	4	1	1Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.100
4	5	0	3Allen, Mr. William Henry	male	35.0	0	0	373450	8.050
...
886	887	0	2Montvila, Rev. Juozas	male	27.0	0	0	211536	13.000
887	888	1	1Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.000
888	889	0	3Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.450
889	890	1	1Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.000
890	891	0	3Dooley, Mr. Patrick	male	32.0	0	0	370376	7.750

891 rows × 12 columns

In [4]:

```
from sklearn.linear_model import LogisticRegression
```

In [10]:

```
a=a.head(10)
a
```

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708

In [11]:

```
a.columns
```

Out[11]:

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',  
      'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],  
      dtype='object')
```

In [12]:

```
b=a[['PassengerId', 'Survived', 'Pclass', 'SibSp', 'Parch']]  
b
```

Out[12]:

	PassengerId	Survived	Pclass	SibSp	Parch
0	1	0	3	1	0
1	2	1	1	1	0
2	3	1	3	0	0
3	4	1	1	1	0
4	5	0	3	0	0
5	6	0	3	0	0
6	7	0	1	0	0
7	8	0	3	3	1
8	9	1	3	0	2
9	10	1	2	1	0

In [13]:

```
c=b.iloc[:,0:5]  
d=a.iloc[:, -1]
```

In [14]:

```
c.shape
```

Out[14]:

```
(10, 5)
```

In [15]:

```
d.shape
```

Out[15]:

```
(10,)
```

In [41]:

```
fs=StandardScaler().fit_transform(c)
```

In [17]:

```
logr=LogisticRegression()  
logr.fit(fs,d)
```

Out[17]:

```
LogisticRegression()
```

In [19]:

```
e=[[2,5,77,8,65]]
```

In [20]:

```
prediction=logr.predict(e)  
prediction
```

Out[20]:

```
array(['S'], dtype=object)
```

In [21]:

```
logr.classes_
```

Out[21]:

```
array(['C', 'Q', 'S'], dtype=object)
```

In [22]:

```
logr.predict_proba(e)[0][0]
```

Out[22]:

```
1.3926038933604862e-33
```

In [23]:

```
logr.predict_proba(e)[0][1]
```

Out[23]:

```
8.694120921868689e-08
```

In [29]:

```
import re  
from sklearn.datasets import load_digits  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import sklearn as sns  
from sklearn.model_selection import train_test_split  
from sklearn.linear_model import LogisticRegression
```

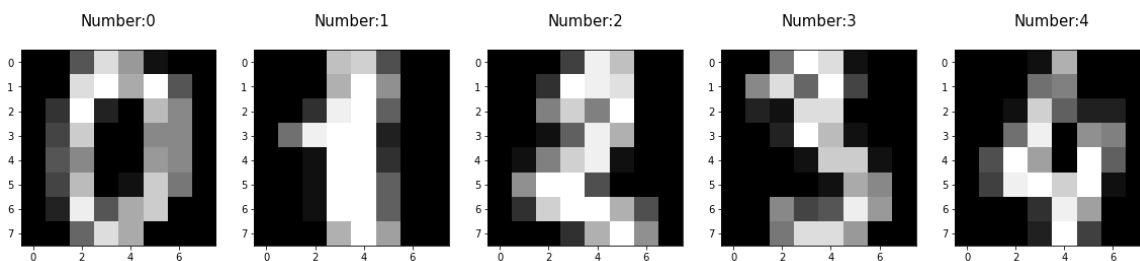
In [30]:

```
digits=load_digits()
digits
pixel_7_0 ,
'pixel_7_1',
'pixel_7_2',
'pixel_7_3',
'pixel_7_4',
'pixel_7_5',
'pixel_7_6',
'pixel_7_7'],
'target_names': array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
'images': array([[ 0.,  0.,  5., ...,  1.,  0.,  0.],
 [ 0.,  0., 13., ..., 15.,  5.,  0.],
 [ 0.,  3., 15., ..., 11.,  8.,  0.],
 ...,
 [ 0.,  4., 11., ..., 12.,  7.,  0.],
 [ 0.,  2., 14., ..., 12.,  0.,  0.],
 [ 0.,  0.,  6., ...,  0.,  0.,  0.]],

 [[ 0.,  0.,  0., ...,  5.,  0.,  0.],
 [ 0.,  0.,  0., ...,  9.,  0.,  0.],
 [ 0.,  0.,  3., ...,  6.,  0.,  0.]],
```

In [34]:

```
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 [35]:

```
x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0.30)
```

In [36]:

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

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

In [38]:

```
logre=LogisticRegression(max_iter=10000)
logre.fit(x_train,y_train)
```

Out[38]:

```
LogisticRegression(max_iter=10000)
```

In [39]:

```
1 logre.predict(x_test)
```

Out[39]:

```
array([6, 9, 4, 9, 6, 1, 1, 2, 9, 5, 8, 5, 3, 0, 3, 2, 4, 6, 3, 8, 2, 0,
        6, 7, 5, 1, 6, 2, 6, 2, 3, 6, 4, 0, 6, 9, 7, 8, 5, 5, 0, 9, 4, 4,
        7, 5, 1, 8, 2, 0, 6, 5, 6, 0, 1, 9, 8, 1, 3, 1, 6, 7, 9, 1, 2, 8,
        1, 2, 2, 5, 7, 6, 8, 4, 5, 4, 4, 3, 9, 0, 7, 8, 6, 4, 2, 2, 8, 8,
        5, 0, 7, 6, 6, 1, 1, 4, 1, 9, 6, 5, 7, 2, 6, 3, 0, 4, 2, 8, 1, 1,
        3, 2, 8, 7, 6, 5, 7, 7, 3, 0, 8, 5, 5, 2, 4, 3, 3, 7, 2, 0, 0, 0,
        9, 9, 6, 5, 6, 8, 8, 4, 6, 9, 6, 2, 6, 8, 9, 7, 1, 9, 8, 4, 3, 8,
        8, 1, 1, 8, 4, 9, 3, 1, 0, 2, 1, 4, 7, 2, 5, 4, 2, 0, 8, 5, 3, 8,
        8, 0, 0, 4, 3, 2, 6, 4, 6, 3, 8, 3, 1, 2, 3, 5, 9, 0, 0, 1, 7, 5,
        6, 5, 8, 1, 3, 0, 9, 2, 0, 7, 4, 8, 7, 2, 6, 7, 8, 5, 5, 9, 5, 4,
        4, 5, 7, 6, 9, 8, 3, 9, 3, 5, 3, 8, 4, 1, 6, 6, 7, 3, 8, 7, 1, 7,
        0, 7, 2, 8, 5, 7, 2, 1, 9, 4, 0, 1, 9, 4, 9, 1, 1, 6, 6, 9, 0, 1,
        6, 4, 4, 2, 8, 6, 5, 6, 4, 3, 8, 7, 2, 0, 8, 7, 3, 6, 2, 8, 2, 9,
        0, 5, 9, 9, 4, 8, 1, 3, 2, 2, 4, 5, 1, 3, 5, 0, 8, 4, 1, 1, 2, 6,
        4, 6, 0, 1, 2, 6, 2, 0, 4, 2, 5, 0, 2, 6, 9, 0, 6, 3, 9, 4, 4, 9,
        6, 6, 7, 8, 9, 8, 4, 9, 2, 8, 9, 0, 3, 7, 0, 1, 0, 2, 6, 0, 7, 2,
        9, 4, 1, 4, 2, 6, 4, 0, 9, 6, 0, 9, 5, 1, 6, 7, 4, 5, 7, 9, 8, 7,
        1, 0, 4, 3, 0, 7, 4, 1, 3, 6, 1, 3, 7, 4, 9, 9, 6, 3, 9, 6, 4, 7,
        9, 4, 0, 2, 9, 9, 0, 8, 2, 9, 9, 9, 0, 1, 4, 6, 5, 2, 1, 0, 8, 8,
        6, 7, 9, 3, 2, 7, 5, 5, 1, 7, 0, 8, 1, 0, 7, 1, 4, 7, 3, 6, 4, 8,
        4, 7, 7, 1, 4, 1, 1, 2, 7, 6, 8, 3, 2, 1, 8, 5, 7, 1, 5, 8, 7, 0,
        7, 1, 0, 1, 0, 4, 2, 5, 2, 6, 9, 9, 2, 7, 4, 6, 3, 7, 3, 1, 7, 9,
        6, 6, 4, 3, 4, 5, 1, 3, 9, 9, 2, 4, 9, 7, 1, 1, 8, 0, 5, 3, 4, 3,
        8, 5, 1, 2, 6, 8, 5, 8, 1, 5, 4, 1, 2, 5, 9, 8, 3, 4, 9, 5, 4, 1,
        2, 1, 0, 9, 2, 7, 6, 1, 5, 0, 8, 8])
```

In [40]:

```
1 logre.score(x_test,y_test)
```

Out[40]:

```
0.9611111111111111
```

In []: