In [56]:

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

In [57]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C2_test.gender_submission.csv")
a
```

Out[57]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	Na
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	Na
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	Na
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	Na
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	Na
...	...	...	...	...	...	...	...	...	...	
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	Na
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C1(
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	Na
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	Na
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	Na

418 rows × 11 columns

In [58]:

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

In [59]:

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

Out[59]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emb
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	
5	897	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	7538	9.2250	NaN	
6	898	3	Connolly, Miss. Kate	female	30.0	0	0	330972	7.6292	NaN	
7	899	2	Caldwell, Mr. Albert Francis	male	26.0	1	1	248738	29.0000	NaN	
8	900	3	Abrahim, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	0	0	2657	7.2292	NaN	
9	901	3	Davies, Mr. John Samuel	male	21.0	2	0	A/4 48871	24.1500	NaN	

In [60]:

```
a.columns
```

Out[60]:

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

In [61]:

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

Out[61]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
0	892	3	34.5	0	0	7.8292
1	893	3	47.0	1	0	7.0000
2	894	2	62.0	0	0	9.6875
3	895	3	27.0	0	0	8.6625
4	896	3	22.0	1	1	12.2875
5	897	3	14.0	0	0	9.2250
6	898	3	30.0	0	0	7.6292
7	899	2	26.0	1	1	29.0000
8	900	3	18.0	0	0	7.2292
9	901	3	21.0	2	0	24.1500

In [62]:

```
c=b.iloc[:,0:6]
d=a.iloc[:,-1]
```

In [63]:

```
c.shape
```

Out[63]:

```
(10, 6)
```

In [64]:

```
d.shape
```

Out[64]:

```
(10,)
```

In [65]:

```
from sklearn.preprocessing import StandardScaler
```

In [66]:

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

In [67]:

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

In [68]:

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

Out[68]:

```
LogisticRegression()
```

In [69]:

```
e=[[2,5,77,8,65,4]]
```

In [70]:

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

Out[70]:

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

In [71]:

```
logr.classes_
```

Out[71]:

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

In [72]:

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

Out[72]:

```
5.638398564699393e-26
```

In [73]:

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

Out[73]:

```
0.9999999999903915
```

In [74]:

```
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
```

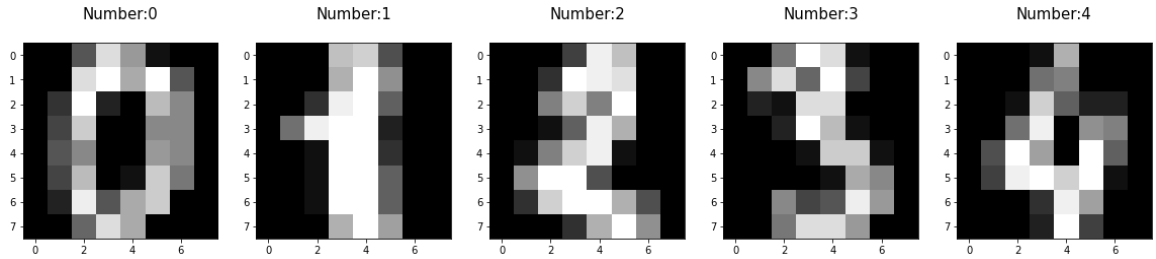
In [75]:

```
digits=load_digits()
digits
```

```
  'pixel_2_0',
  'pixel_2_1',
  'pixel_2_2',
  'pixel_2_3',
  'pixel_2_4',
  'pixel_2_5',
  'pixel_2_6',
  'pixel_2_7',
  'pixel_3_0',
  'pixel_3_1',
  'pixel_3_2',
  'pixel_3_3',
  'pixel_3_4',
  'pixel_3_5',
  'pixel_3_6',
  'pixel_3_7',
  'pixel_4_0',
  'pixel_4_1',
  'pixel_4_2',
```

In [76]:

```
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 [77]:

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

In [78]:

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

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

In [79]:

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

Out[79]:

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

In [80]:

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

Out[80]:

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

In [81]:

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

Out[81]:

```
0.9629629629629629
```

In [ ]: