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

In [2]:
```
from sklearn.linear_model import LogisticRegression
```

In [3]:
```
df=pd.read_csv(r"bot_detection_data")
```

In [4]:

```
1 df
```

Out[4]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location	Created At	Hashtags
0	132131	flong	Station activity person against natural majori...	85	1	2353	False	1	Adkinston	2020-05-11 15:29:50	NaN
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	True	0	Sanderston	2022-11-26 05:18:10	both live
2	779715	roberttran	Manage whose quickly especially foot none to g...	6	2	4363	True	0	Harrisonfurt	2022-08-08 03:16:54	phone ahead
3	696168	pmason	Just cover eight opportunity strong policy which.	54	5	2242	True	1	Martinezberg	2021-08-14 22:27:05	ever quickly new I
4	704441	noah87	Animal sign six data good or.	26	3	8438	False	1	Camachoville	2020-04-13 21:24:21	foreign mention
...	...	...	...	...	...	...	...	...	...	...	...
49995	491196	uberg	Want but put card direction know miss former h...	64	0	9911	True	1	Lake Kimberlyburgh	2023-04-20 11:06:26	teach quality ten education any
49996	739297	jessicamunoz	Provide whole maybe agree church respond most ...	18	5	9900	False	1	Greenbury	2022-10-18 03:57:35	add walk among believe
49997	674475	lynncunningham	Bring different everyone international capital...	43	3	6313	True	1	Deborahfort	2020-07-08 03:54:08	onto admit artist first
49998	167081	richardthompson	Than about single generation itself seek sell ...	45	1	6343	False	0	Stephenside	2022-03-22 12:13:44	star
49999	311204	daniel29	Here morning class various room human true bec...	91	4	4006	False	0	Novakberg	2022-12-03 06:11:07	home

50000 rows × 11 columns

In [5]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   User ID         50000 non-null  int64
 1   Username        50000 non-null  object
 2   Tweet           50000 non-null  object
 3   Retweet Count   50000 non-null  int64
 4   Mention Count   50000 non-null  int64
 5   Follower Count  50000 non-null  int64
 6   Verified        50000 non-null  bool
 7   Bot Label       50000 non-null  int64
 8   Location        50000 non-null  object
 9   Created At      50000 non-null  object
 10  Hashtags        41659 non-null  object
dtypes: bool(1), int64(5), object(5)
memory usage: 3.9+ MB
```

In [8]:
```
1 df.dropna(inplace=True)
```

In [12]:
```
1 df.columns
```

Out[12]:
```
Index(['User ID', 'Username', 'Tweet', 'Retweet Count', 'Mention Count',
       'Follower Count', 'Verified', 'Bot Label', 'Location', 'Created At',
       'Hashtags'],
      dtype='object')
```

In [9]:

```
1 df
```

Out[9]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location	Created At	Hashtags
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	True	0	Sanderston	2022-11-26 05:18:10	both live
2	779715	roberttran	Manage whose quickly especially foot none to g...	6	2	4363	True	0	Harrisonfurt	2022-08-08 03:16:54	phone ahead
3	696168	pmason	Just cover eight opportunity strong policy which.	54	5	2242	True	1	Martinezberg	2021-08-14 22:27:05	ever quickly new I
4	704441	noah87	Animal sign six data good or.	26	3	8438	False	1	Camachoville	2020-04-13 21:24:21	foreign mention
5	570928	james00	See wonder travel this suffer less yard office...	41	4	3792	True	1	West Cheyenne	2023-05-07 22:24:47	anyone respond perhaps market run
...	...	...	...	...	...	...	...	...	...	...	...
49995	491196	uberg	Want but put card direction know miss former h...	64	0	9911	True	1	Lake Kimberlyburgh	2023-04-20 11:06:26	teach quality ten education any
49996	739297	jessicamunoz	Provide whole maybe agree church respond most ...	18	5	9900	False	1	Greenbury	2022-10-18 03:57:35	add walk among believe
49997	674475	lynncunningham	Bring different everyone international capital...	43	3	6313	True	1	Deborahfort	2020-07-08 03:54:08	onto admit artist first
49998	167081	richardthompson	Than about single generation itself seek sell ...	45	1	6343	False	0	Stephenside	2022-03-22 12:13:44	star
49999	311204	daniel29	Here morning class various room human true bec...	91	4	4006	False	0	Novakberg	2022-12-03 06:11:07	home

41659 rows × 11 columns

In [10]:
```
from sklearn.linear_model import LogisticRegression
```

In [11]:
```
logr =LogisticRegression()
```

In [13]:
```
feature_matrix=df[['User ID','Retweet Count', 'Mention Count',
       'Follower Count','Bot Label']]
target_vector=df['Verified']
```

In [14]:
```
feature_matrix.shape
```

Out[14]: (41659, 5)

In [15]:
```
target_vector.shape
```

Out[15]: (41659,)

In [16]:
```
from sklearn.preprocessing import StandardScaler
```

In [17]:
```
fs=StandardScaler().fit_transform(feature_matrix)
```

In [18]:
```
logr=LogisticRegression()
logr.fit(fs,target_vector)
```

Out[18]: LogisticRegression()

In [27]:
```
observation=[[1,2,3,4,5]]
```

In [28]:
```
prediction = logr.predict(observation)
print(prediction)
```

[False]

In [29]:
```
1 logr.classes_
```

Out[29]:
```
array([False,  True])
```

In [30]:
```
1 logr.predict_proba(observation)[0][1]
```

Out[30]:
```
0.49508486971875193
```

In [31]:
```
1 logr.predict_proba(observation)[0][0]
```

Out[31]:
```
0.504915130281248
```

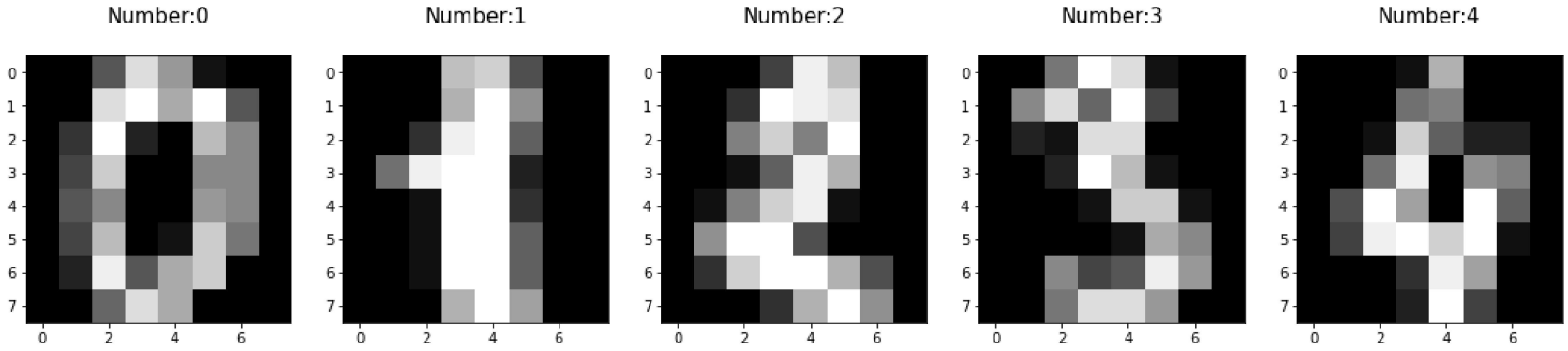
Linear regression 2

In [32]:
```
1 import re
2 from sklearn.datasets import load_digits
3 import numpy as np
4 import pandas as pd
5 import matplotlib.pyplot as plt
6 import seaborn as sns
7 from sklearn.linear_model import LogisticRegression
8 from sklearn.model_selection import train_test_split
```

In [33]:
```
digits =load_digits()
digits
```

Out[33]:
```
{'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 [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 [37]:
```
logre=LogisticRegression(max_iter=10000)
logre.fit(x_train,y_train)
```

Out[37]:
```
LogisticRegression(max_iter=10000)
```

In [38]:
```
print(logre.predict(x_test))
```

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

In [39]:
```
print(logre.score(x_test,y_test))
```

```
0.9777777777777777
```

In [ ]:

```
1
```