

In [110]:

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

In [208]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C3_bot_detection_data.csv")  
a
```

Out[208]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location
0	132131	flong	Station activity person against natural majori...	85	1	2353	False	1	Adkinstr
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	True	0	Sanderstr
2	779715	roberttran	Manage whose quickly especially foot none to g...	6	2	4363	True	0	Harrisonfr
3	696168	pmason	Just cover eight opportunity strong policy which.	54	5	2242	True	1	Martinezbe
4	704441	noah87	Animal sign six data good or.	26	3	8438	False	1	Camachovi
...	...	...	...	...	...	...	...	...	...
49995	491196	uberg	Want but put card direction know miss former h...	64	0	9911	True	1	La Kimberlybur
49996	739297	jessicamunoz	Provide whole maybe agree church respond most ...	18	5	9900	False	1	Greenbu
49997	674475	lynncunningham	Bring different everyone international capital...	43	3	6313	True	1	Deborahfr
49998	167081	richardthompson	Than about single generation itself seek sell ...	45	1	6343	False	0	Stephensi
49999	311204	daniel29	Here morning class various room human true bec...	91	4	4006	False	0	Novakbe

50000 rows × 11 columns



In [209]:

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

In [210]:

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

Out[210]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location
0	132131	flong	Station activity person against natural majori...	85	1	2353	False	1	Adkinston
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	True	0	Sanderston
2	779715	roberttran	Manage whose quickly especially foot none to g...	6	2	4363	True	0	Harrisonfurt
3	696168	pmason	Just cover eight opportunity strong policy which.	54	5	2242	True	1	Martinezberg
4	704441	noah87	Animal sign six data good or.	26	3	8438	False	1	Camachoville
5	570928	james00	See wonder travel this suffer less yard office...	41	4	3792	True	1	West Cheyenne
6	734182	leonard00	Door final sound my guess building rich.	54	0	10	True	0	South Donald
7	107312	lesterdaniel	Job phone price magazine worry stage check view.	64	0	1442	False	1	Smithhaven
8	549888	kimberlymorris	Eye rest prove mission show floor.	25	2	836	False	0	Lake Brittanyville
9	117640	schmittjill	Add letter year performance western what cultu...	67	3	6523	False	1	West Hannahborough

In [211]:

```
a.columns
```

Out[211]:

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

In [213]:

```
b=a[['User ID', 'Retweet Count', 'Mention Count',  
     'Follower Count', 'Bot Label']]  
b
```

Out[213]:

	User ID	Retweet Count	Mention Count	Follower Count	Bot Label
0	132131	85	1	2353	1
1	289683	55	5	9617	0
2	779715	6	2	4363	0
3	696168	54	5	2242	1
4	704441	26	3	8438	1
5	570928	41	4	3792	1
6	734182	54	0	10	0
7	107312	64	0	1442	1
8	549888	25	2	836	0
9	117640	67	3	6523	1

In [223]:

```
c=b.iloc[:,0:11]  
d=b.iloc[:, -1]
```

In [224]:

```
c.shape
```

Out[224]:

```
(10, 5)
```

In [225]:

```
d.shape
```

Out[225]:

```
(10,)
```

In [226]:

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

In [227]:

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

In [228]:

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

Out[228]:

```
LogisticRegression()
```

In [229]:

```
e=[[2,5,77,8,6]]
```

In [230]:

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

Out[230]:

```
array([1], dtype=int64)
```

In [231]:

```
logr.classes_
```

Out[231]:

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

In [232]:

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

Out[232]:

```
8.027930653575766e-09
```

In [233]:

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

Out[233]:

```
0.9999999919720693
```

In [234]:

```
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
```

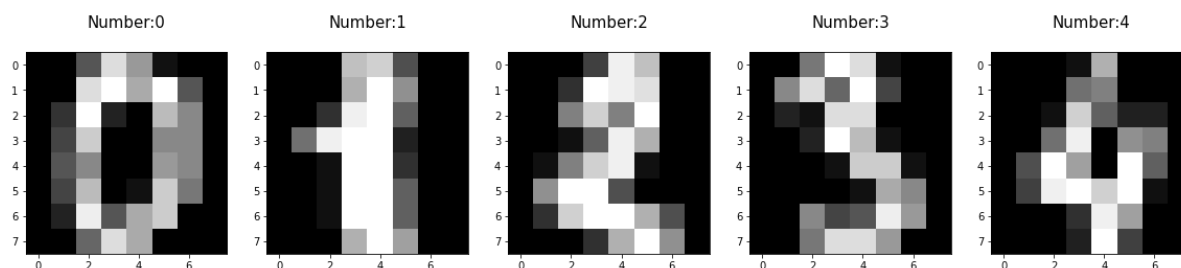
In [235]:

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

```
'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',
'pixel_2_6',
'pixel_2_7'
```

In [236]:

```
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 [237]:

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

In [238]:

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

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

In [239]:

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

Out[239]:

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

In [240]:

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

Out[240]:

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

In [241]:

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

Out[241]:

```
0.9703703703703703
```



In [242]:

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

In [261]:

```
b=a[['User ID', 'Retweet Count', 'Mention Count',
      'Follower Count', 'Bot Label'],]
b
```

Out[261]:

	User ID	Retweet Count	Mention Count	Follower Count	Bot Label
0	132131	85	1	2353	1
1	289683	55	5	9617	0
2	779715	6	2	4363	0
3	696168	54	5	2242	1
4	704441	26	3	8438	1
5	570928	41	4	3792	1
6	734182	54	0	10	0
7	107312	64	0	1442	1
8	549888	25	2	836	0
9	117640	67	3	6523	1

In [262]:

```
b['Bot Label'].value_counts()
```

Out[262]:

```
1    6
0    4
Name: Bot Label, dtype: int64
```

In [263]:

```
x=b.drop('Bot Label',axis=1)
y=b['Bot Label']
```

In [264]:

```
g1={"Bot Label":{"Bot Label":1,'b':2}}
b=b.replace(g1)
print(b)
```

	User ID	Retweet Count	Mention Count	Follower Count	Bot Label
0	132131	85	1	2353	1
1	289683	55	5	9617	0
2	779715	6	2	4363	0
3	696168	54	5	2242	1
4	704441	26	3	8438	1
5	570928	41	4	3792	1
6	734182	54	0	10	0
7	107312	64	0	1442	1
8	549888	25	2	836	0
9	117640	67	3	6523	1

In [265]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [266]:

```
from sklearn.ensemble import RandomForestClassifier
```

In [267]:

```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[267]:

```
RandomForestClassifier()
```

In [268]:

```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf':[5,10,15,20,25],
            'n_estimators':[10,20,30,40,50]}
```

In [269]:

```
from sklearn.model_selection import GridSearchCV
```

In [270]:

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[270]:

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                          'min_samples_leaf': [5, 10, 15, 20, 25],
                          'n_estimators': [10, 20, 30, 40, 50]},
             scoring='accuracy')
```

In [271]:

```
grid_search.best_score_
```

Out[271]:

0.7083333333333333

In [272]:

```
rfc_best=grid_search.best_estimator_
```

In [273]:

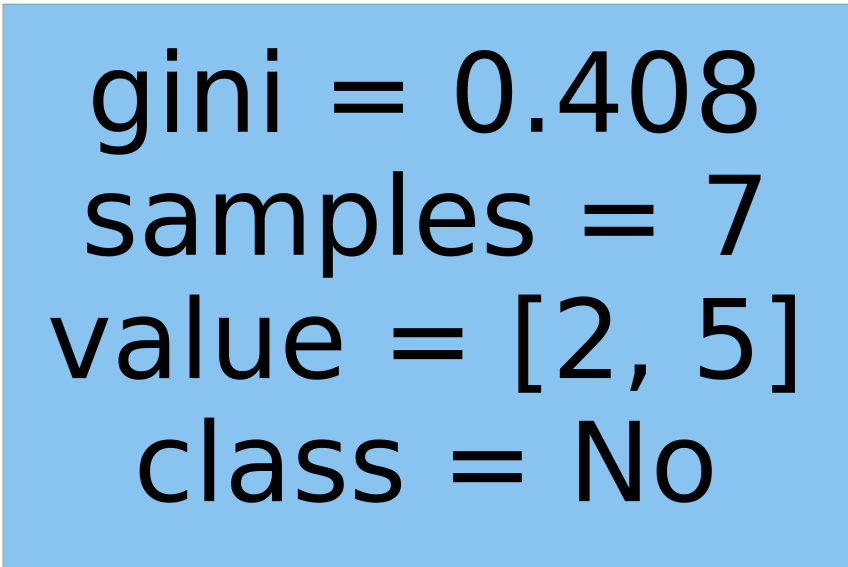
```
from sklearn.tree import plot_tree
```

In [274]:

```
plt.figure(figsize=(80,40))  
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```

Out[274]:

[Text(2232.0, 1087.2, 'gini = 0.408\nsamples = 7\nvalue = [2, 5]\nclass = No')]



gini = 0.408  
samples = 7  
value = [2, 5]  
class = No

In [ ]: