In [110]:

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

In [111]:

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

Out[111]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Eml
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	
...	...	...	...	...	...	...	...	...	...	...	
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	

418 rows × 11 columns

In [112]:

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

In [113]:

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

Out[113]:

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

In [114]:

```
a.columns
```

Out[114]:

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

In [116]:

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

Out[116]:

	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 [117]:

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

In [118]:

```
c.shape
```

Out[118]:

```
(10, 6)
```

In [119]:

```
d.shape
```

Out[119]:

```
(10,)
```

In [120]:

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

In [121]:

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

In [122]:

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

Out[122]:

```
LogisticRegression()
```

In [123]:

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

In [124]:

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

Out[124]:

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

In [125]:

```
logr.classes_
```

Out[125]:

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

In [126]:

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

Out[126]:

```
5.258911934097103e-27
```

In [127]:

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

Out[127]:

```
1.0
```

In [128]:

```
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
```

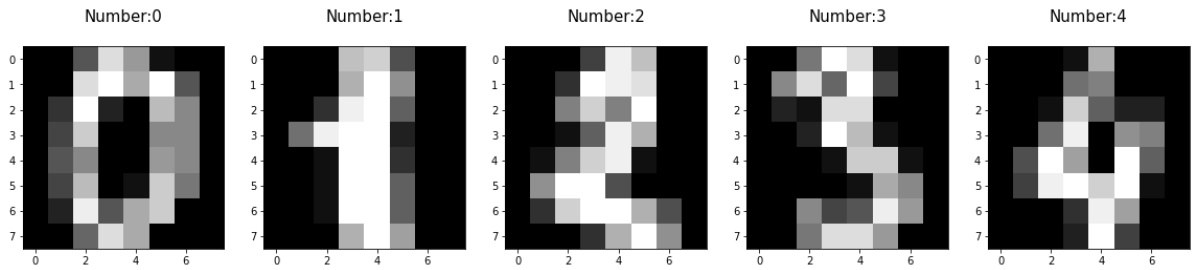
In [129]:

```
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',
```

In [130]:

```
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 [131]:

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

In [132]:

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

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

In [133]:

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

Out[133]:

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

In [134]:

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

Out[134]:

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

In [135]:

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

Out[135]:

```
0.9592592592592593
```

In [136]:

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

In [137]:

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

Out[137]:

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

In [144]:

```
b['Embarked'].value_counts()
```

Out[144]:

```
S    6
Q    3
C    1
Name: Embarked, dtype: int64
```

In [145]:

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

In [146]:

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

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

In [147]:

```
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 [148]:

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

In [149]:

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

Out[149]:

```
RandomForestClassifier()
```

In [150]:

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

In [151]:

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

In [152]:

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

Out[152]:

```
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 [153]:

```
grid_search.best_score_
```

Out[153]:

```
0.7083333333333333
```

In [154]:

```
rfc_best=grid_search.best_estimator_
```

In [155]:

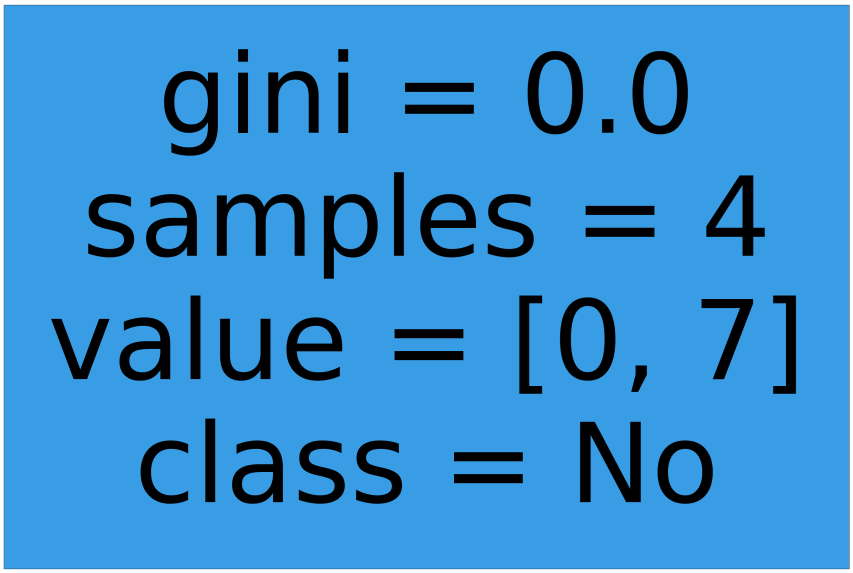
```
from sklearn.tree import plot_tree
```

In [156]:

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

Out[156]:

```
[Text(2232.0, 1087.2, 'gini = 0.0\nsamples = 4\nvalue = [0, 7]\nclass = No')]
```



In [ ]:

```

```