In [110]:

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

In [157]:

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

Out[157]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Na
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C8
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	Na
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C12
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Na
...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	Na
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B4
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	Na
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C14
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	Na

891 rows × 12 columns

In [158]:

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

In [159]:

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

Out[159]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN

In [160]:

```
a.columns
```

Out[160]:

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

In [169]:

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

Out[169]:

	PassengerId	Pclass	SibSp	Parch	Fare
0	1	3	1	0	7.2500
1	2	1	1	0	71.2833
2	3	3	0	0	7.9250
3	4	1	1	0	53.1000
4	5	3	0	0	8.0500
5	6	3	0	0	8.4583
6	7	1	0	0	51.8625
7	8	3	3	1	21.0750
8	9	3	0	2	11.1333
9	10	2	1	0	30.0708

In [170]:

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

In [171]:

```
c.shape
```

Out[171]:

```
(10, 5)
```

In [172]:

```
d.shape
```

Out[172]:

```
(10,)
```

In [173]:

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

In [174]:

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

In [175]:

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

Out[175]:

```
LogisticRegression()
```

In [179]:

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

In [180]:

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

Out[180]:

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

In [181]:

```
logr.classes_
```

Out[181]:

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

In [182]:

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

Out[182]:

```
0.9999999973932001
```

In [183]:

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

Out[183]:

```
3.0804479291079108e-25
```

In [184]:

```
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
```

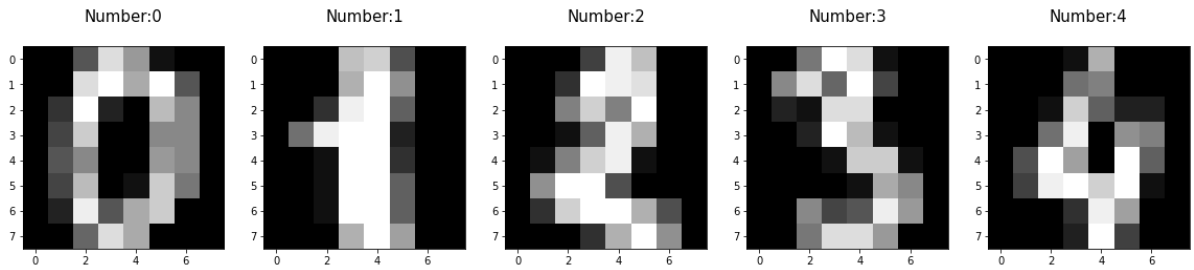
In [185]:

```
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 [186]:

```
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 [187]:

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

In [188]:

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

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

In [189]:

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

Out[189]:

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

In [190]:

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

Out[190]:

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

In [191]:

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

Out[191]:

```
0.975925925925926
```

In [192]:

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

In [194]:

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

Out[194]:

	PassengerId	Pclass	SibSp	Parch	Fare	Embarked
0	1	3	1	0	7.2500	S
1	2	1	1	0	71.2833	C
2	3	3	0	0	7.9250	S
3	4	1	1	0	53.1000	S
4	5	3	0	0	8.0500	S
5	6	3	0	0	8.4583	Q
6	7	1	0	0	51.8625	S
7	8	3	3	1	21.0750	S
8	9	3	0	2	11.1333	S
9	10	2	1	0	30.0708	C

In [195]:

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

Out[195]:

```
S    7
C    2
Q    1
Name: Embarked, dtype: int64
```

In [196]:

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

In [197]:

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

```
   PassengerId  Pclass  SibSp  Parch     Fare Embarked
0            1       3      1      0   7.2500        S
1            2       1      1      0  71.2833        C
2            3       3      0      0   7.9250        S
3            4       1      1      0  53.1000        S
4            5       3      0      0   8.0500        S
5            6       3      0      0   8.4583        Q
6            7       1      0      0  51.8625        S
7            8       3      3      1  21.0750        S
8            9       3      0      2  11.1333        S
9           10       2      1      0  30.0708        C
```

In [198]:

```
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 [199]:

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

In [200]:

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

Out[200]:

```
RandomForestClassifier()
```

In [201]:

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

In [202]:

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

In [203]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:66
6: UserWarning: The least populated class in y has only 1 members, which is less
than n_splits=2.
  warnings.warn(("The least populated class in y has only %d"
```

Out[203]:

```
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 [204]:

```
grid_search.best_score_
```

Out[204]:

```
0.875
```

In [205]:

```
rfc_best=grid_search.best_estimator_
```

In [206]:

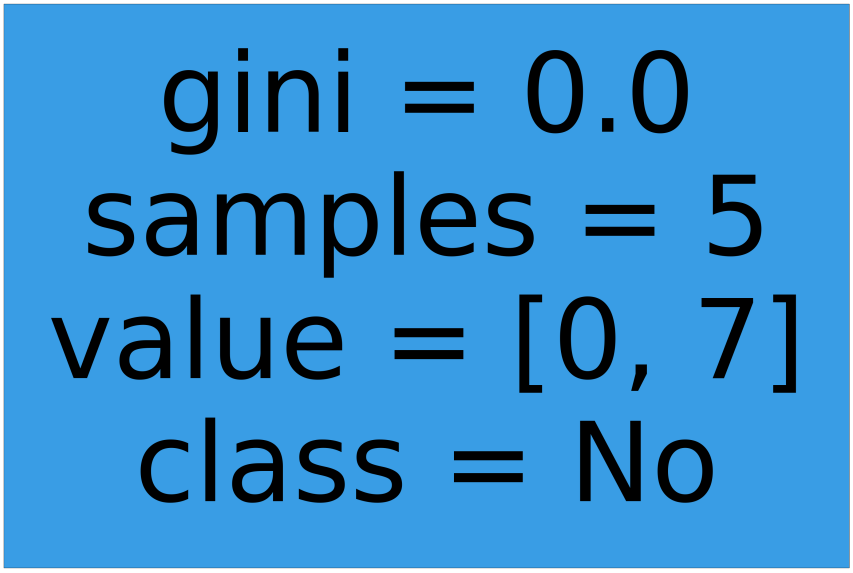
```
from sklearn.tree import plot_tree
```

In [207]:

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

Out[207]:

```
[Text(2232.0, 1087.2, 'gini = 0.0\nsamples = 5\nvalue = [0, 7]\nclass = No')]
```



In [ ]:

```

```