mk 02-09-2023

In [729]:

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

In [730]:

```python
from sklearn.linear_model import LogisticRegression
a=pd.read_csv(r"C:\USERS\user\Downloads\C2_train.gender_submission.csv")
a
```

Out[730]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
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	female	35.0	1	0	113803	53.1000	C123	

In [732]:

```
a=a.head(100)
a
```

Out[732]:

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

In [733]:

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

In [734]:

```
a.columns
```

Out[734]:

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

In [735]:

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

Out[735]:

	PassengerId	Survived	Pclass	SibSp	Parch
0	1	0	3	1	0
1	2	1	1	1	0
2	3	1	3	0	0
3	4	1	1	1	0
4	5	0	3	0	0
5	6	0	3	0	0
6	7	0	1	0	0
7	8	0	3	3	1
8	9	1	3	0	2
9	10	1	2	1	0

In [736]:

```
c=b.iloc[:,0:5]
d=b.iloc[:,-1]
```

In [737]:

```
c.shape
```

Out[737]:

```
(10, 5)
```

In [738]:

```
d.shape
```

Out[738]:

```
(10,)
```

In [739]:

```
from sklearn.preprocessing import StandardScaler
fs=StandardScaler().fit_transform(c)
fs
```

Out[739]:

```
array([[-1.5666989 , -1.        ,  0.77777778,  0.33333333, -0.46852129],
       [-1.21854359,  1.        , -1.44444444,  0.33333333, -0.46852129],
       [-0.87038828,  1.        ,  0.77777778, -0.77777778, -0.46852129],
       [-0.52223297,  1.        , -1.44444444,  0.33333333, -0.46852129],
       [-0.17407766, -1.        ,  0.77777778, -0.77777778, -0.46852129],
       [ 0.17407766, -1.        ,  0.77777778, -0.77777778, -0.46852129],
       [ 0.52223297, -1.        , -1.44444444, -0.77777778, -0.46852129],
       [ 0.87038828, -1.        ,  0.77777778,  2.55555556,  1.09321633],
       [ 1.21854359,  1.        ,  0.77777778, -0.77777778,  2.65495395],
       [ 1.5666989 ,  1.        , -0.33333333,  0.33333333, -0.46852129]])
```

In [740]:

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

Out[740]:

```
LogisticRegression()
```

In [741]:

```
e=[[77,9,55,5,76]]
```

In [742]:

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

Out[742]:

```
array([2], dtype=int64)
```

In [743]:

```
logr.classes_
```

Out[743]:

```
array([0, 1, 2], dtype=int64)
```

In [744]:

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

Out[744]:

```
1.3533762495429735e-71
```

In [745]:

```
import re
from sklearn.datasets import load_digits
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [746]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

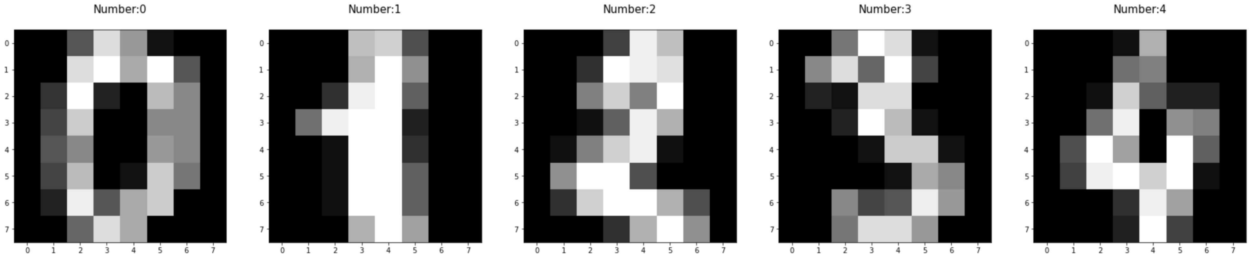
In [747]:

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

```
 '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',
 'pixel_3_0',
 'pixel_3_1',
 'pixel_3_2',
 'pixel_3_3',
 'pixel_3_4',
 'pixel_3_5',
 'pixel_3_6',
```

In [748]:

```
plt.figure(figsize=(50,25))
for index,(image,label) in enumerate(zip(digits.data[0:8],digits.target[0:5])):
    plt.subplot(1,8,index+1)
    plt.imshow(np.reshape(image,(8,8)),cmap=plt.cm.gray)
    plt.title('Number:%i\n'%label,fontsize=15)
```



In [749]:

```
x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0.
```

In [750]:

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

```
(898, 64)
(899, 64)
(898,)
(899,)
```

In [751]:

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

```

Out[751]:

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

In [752]:

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

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

In [753]:

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

In [754]:

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

In [770]:

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

Out[770]:

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

In [771]:

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

Out[771]:

	PassengerId	Survived	Pclass	SibSp	Parch	Embarked
0	1	0	3	1	0	S
1	2	1	1	1	0	C
2	3	1	3	0	0	S
3	4	1	1	1	0	S
4	5	0	3	0	0	S
5	6	0	3	0	0	Q
6	7	0	1	0	0	S
7	8	0	3	3	1	S
8	9	1	3	0	2	S
9	10	1	2	1	0	C

In [785]:

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

Out[785]:

```
0    8
1    1
2    1
Name: Parch, dtype: int64
```

In [786]:

```
x=b[['PassengerId', 'Survived', 'Pclass']]
y=b['Parch']
print(b)
```

```
   PassengerId  Survived  Pclass  SibSp  Parch Embarked
0            1         0       3      1      0        S
1            2         1       1      1      0        C
2            3         1       3      0      0        S
3            4         1       1      1      0        S
4            5         0       3      0      0        S
5            6         0       3      0      0        Q
6            7         0       1      0      0        S
7            8         0       3      3      1        S
8            9         1       3      0      2        S
9           10         1       2      1      0        C
```

In [787]:

```
g1={"Parch":{'g1':1}}
a=a.replace(g1)
print(a)
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3   
1            2         1       1   
2            3         1       3   
3            4         1       1   
4            5         0       3   
5            6         0       3   
6            7         0       1   
7            8         0       3   
8            9         1       3   
9           10         1       2   

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1   
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1   
2                             Heikkinen, Miss. Laina  female  26.0      0   
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1   
4                           Allen, Mr. William Henry    male  35.0      0   
5                                   Moran, Mr. James    male   NaN      0   
6                            McCarthy, Mr. Timothy J    male  54.0      0   
7                     Palsson, Master. Gosta Leonard    male   2.0      3   
8  Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)  female  27.0      0   
9                Nasser, Mrs. Nicholas (Adele Achem)  female  14.0      1   

   Parch            Ticket     Fare Cabin Embarked  
0      0         A/5 21171   7.2500   NaN        S  
1      0          PC 17599  71.2833   C85        C  
2      0  STON/O2. 3101282   7.9250   NaN        S  
3      0            113803  53.1000  C123        S  
4      0            373450   8.0500   NaN        S  
5      0            330877   8.4583   NaN        Q  
6      0             17463  51.8625   E46        S  
7      1            349909  21.0750   NaN        S  
8      2            347742  11.1333   NaN        S  
9      0            237736  30.0708   NaN        C  
```

In [788]:

```
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 [789]:

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

In [790]:

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

Out[790]:

```
RandomForestClassifier()
```

In [791]:

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

In [792]:

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

In [793]:

```
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:666: UserWa
rning: 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[793]:

```
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 [794]:

```
grid_search.best_score_
```

Out[794]:

```
0.875
```

In [795]:

```
rfc_best=grid_search.best_estimator_
```

In [796]:

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

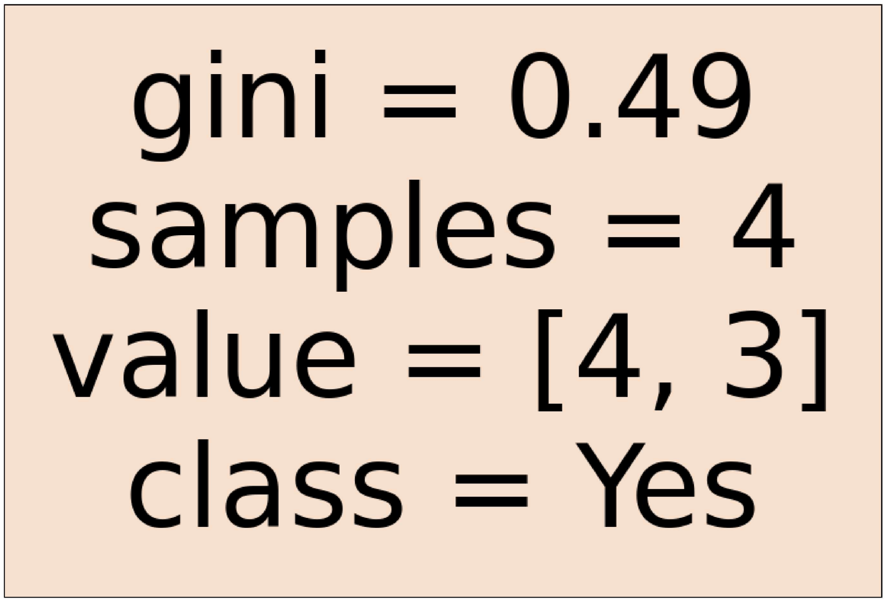
In [797]:

```
plt.figure(figsize=(20,10))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],fi

```

Out[797]:

```
[Text(558.0, 271.8, 'gini = 0.49\nsamples = 4\nvalue = [4, 3]\nclass = Yes')]
```


gini = 0.49
samples = 4
value = [4, 3]
class = Yes