Vijay 02-09-2023

In [729]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [730]:
```
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	Emb
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 [798]:
```
a=a.head(50)
a
```

Out[798]:

	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 [799]:
```
from sklearn.linear_model import LogisticRegression
```

In [800]:
```
a.columns
```

Out[800]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [801]:
```
b=a[['PassengerId', 'Survived', 'Pclass', 'SibSp','Parch']]
b
```

Out[801]:

	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 [802]:
```
c=b.iloc[:,0:5]
d=b.iloc[:,-1]
```

In [803]:
```
c.shape
```

Out[803]: (10, 5)

In [804]:
```
d.shape
```

Out[804]: (10,)

In [805]:
```
from sklearn.preprocessing import StandardScaler
fs=StandardScaler().fit_transform(c)
fs
```

Out[805]:
```
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 [806]:
```
logr=LogisticRegression()
logr.fit(fs,d)
```

Out[806]: LogisticRegression()

In [807]:
```
e=[[77,9,55,5,76]]
```

In [808]:
```
prediction=logr.predict(e)
prediction
```

Out[808]: array([2], dtype=int64)

In [809]:
```
logr.classes_
```

Out[809]:
```
array([0, 1, 2], dtype=int64)
```

In [810]:
```
logr.predict_proba(e)[0][0]
```

Out[810]:
```
1.3533762495429735e-71
```

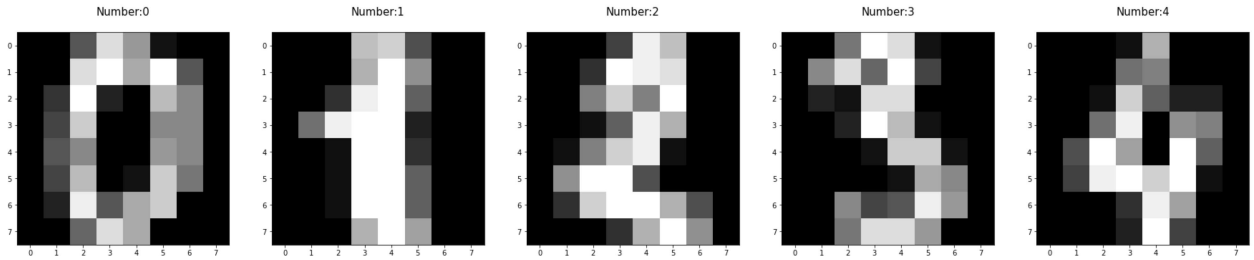
In [811]:
```
import re
from sklearn.datasets import load_digits
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [812]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

In [813]:
```
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 [814]:
```
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 [817]:
```
x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0.
```

In [818]:
```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(1347, 64)
(450, 64)
(1347,)
(450,)
```

In [819]:
```
logre=LogisticRegression(max_iter=10000)
logre.fit(x_train,y_train)

```

Out[819]:
```
LogisticRegression(max_iter=10000)
```

In [820]:
```
print(logre.predict(x_test))
```

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

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 [821]:
```
a=a.head(50)
a
```

Out[821]:

	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 [822]:
```
b=a[['PassengerId', 'Survived', 'Pclass', 'SibSp','Parch']]
b
```

Out[822]:

	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 [823]:
```
b['Pclass'].value_counts()
```

Out[823]:
```
3    6
1    3
2    1
Name: Pclass, dtype: int64
```

In [824]:
```
x=b[['PassengerId', 'Survived', 'SibSp']]
y=b['Pclass']
print(b)
```

```
   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 [825]:
```
g1={"Pclass":{'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 [826]:
```
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 [827]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [828]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[828]:
```
RandomForestClassifier()
```

In [829]:
```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf':[5,10,15,20,25],
            'n_estimators':[10,20,30,40,50]}
```

In [830]:
```
from sklearn.model_selection import GridSearchCV
```

In [831]:
```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[831]:
```
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 [832]:
```
grid_search.best_score_
```

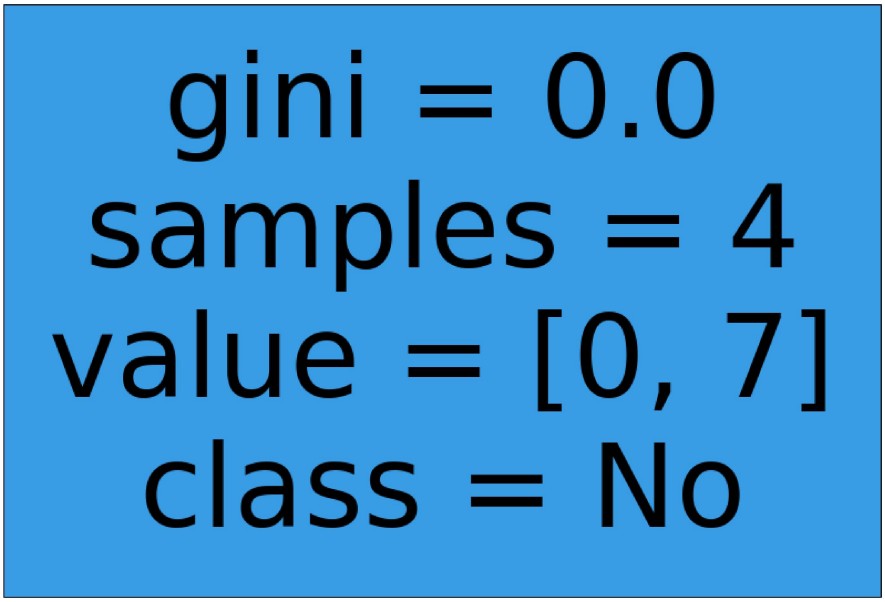
Out[832]:
```
0.7083333333333333
```

In [833]:
```
rfc_best=grid_search.best_estimator_
```

In [834]:
```
from sklearn.tree import plot_tree
```

In [835]:
```
plt.figure(figsize=(20,10))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],fi
```

Out[835]:
```
[Text(558.0, 271.8, 'gini = 0.0\nsamples = 4\nvalue = [0, 7]\nclass = No')]
```



In [ ]:
```
```