02/08/2023

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

In [52]:
```
a=pd.read_csv(r"C:\Users\user\Downloads\C2_test.gender_submission.csv")
a
```

Out[52]:

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

418 rows × 11 columns

In [11]:
```
from sklearn.linear_model import LogisticRegression
```

In [53]:
```
a=a.head(10)
a
```

Out[53]:

	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 [54]:
```
a.columns
```

Out[54]:
```
Index(['PassengerId', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch',
       'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [56]:
```
b=a[['PassengerId', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']]
b
```

Out[56]:

	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 [57]:
```
c=b.iloc[:,0:11]
d=a.iloc[:,-1]
```

In [58]:
```
c.shape
```

Out[58]: (10, 6)

In [59]:
```
d.shape
```

Out[59]: (10,)

In [60]:
```
from sklearn.preprocessing import StandardScaler
```

In [61]:
```
fs=StandardScaler().fit_transform(c)
```

In [62]:
```
logr=LogisticRegression()
logr.fit(fs,d)
```

Out[62]:
```
LogisticRegression()
```

In [67]:
```
e=[[2,5,77,8,6,5]]
```

In [68]:
```
prediction=logr.predict(e)
prediction
```

Out[68]:
```
array(['Q'], dtype=object)
```

In [69]:
```
logr.classes_
```

Out[69]:
```
array(['C', 'Q', 'S'], dtype=object)
```

In [70]:
```
logr.predict_proba(e)[0][0]
```

Out[70]:
```
5.258911934097103e-27
```

In [71]:
```
logr.predict_proba(e)[0][1]
```

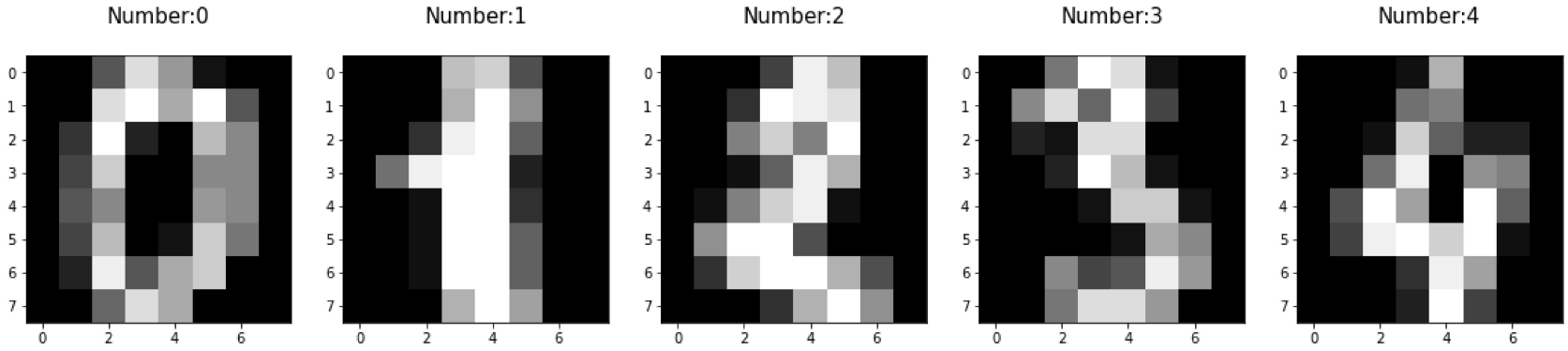
Out[71]:
```
1.0
```

In [72]:
```
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 [27]:
```
digits=load_digits()
digits
```

```
 'target': array([0, 1, 2, ..., 8, 9, 8]),
 'frame': None,
 'feature_names': ['pixel_0_0',
  'pixel_0_1',
  'pixel_0_2',
  'pixel_0_3',
  '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',
```

In [73]:
```
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 [74]:
```
x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0.30)
```

In [75]:
```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

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

In [76]:
```
logre=LogisticRegression(max_iter=10000)
logre.fit(x_train,y_train)
```

Out[76]:
```
LogisticRegression(max_iter=10000)
```

In [32]:
```
logre.predict(x_test)
```

Out[32]:
```
array([8, 4, 3, 1, 7, 4, 0, 9, 4, 7, 4, 9, 5, 4, 2, 6, 2, 5, 9, 7, 7, 5,
       7, 0, 3, 4, 9, 8, 6, 7, 1, 4, 1, 9, 1, 2, 2, 9, 9, 3, 4, 8, 9, 5,
       3, 0, 3, 0, 0, 4, 4, 9, 1, 2, 2, 7, 5, 9, 8, 2, 0, 4, 0, 2, 1, 4,
       0, 5, 8, 1, 8, 3, 5, 9, 3, 6, 7, 7, 0, 0, 5, 3, 5, 5, 6, 8, 8, 4,
       8, 8, 2, 5, 6, 5, 5, 8, 1, 6, 5, 6, 1, 3, 6, 6, 7, 6, 0, 7, 1, 6,
       9, 3, 0, 0, 4, 3, 9, 5, 3, 7, 2, 4, 7, 5, 6, 6, 0, 6, 6, 7, 1, 1,
       1, 0, 4, 8, 4, 9, 9, 1, 2, 1, 5, 0, 2, 6, 7, 5, 3, 4, 9, 2, 7, 5,
       1, 8, 5, 6, 0, 4, 1, 6, 2, 8, 9, 2, 1, 3, 7, 6, 5, 3, 4, 1, 3, 3,
       8, 7, 9, 6, 7, 9, 5, 4, 0, 2, 0, 5, 1, 1, 8, 9, 1, 7, 2, 4, 6, 7,
       4, 4, 2, 7, 9, 4, 7, 1, 3, 2, 7, 9, 7, 1, 9, 8, 2, 0, 1, 8, 6, 8,
       5, 1, 7, 8, 6, 0, 5, 0, 4, 9, 2, 6, 2, 8, 1, 3, 3, 6, 9, 4, 2, 4,
       5, 7, 4, 6, 5, 4, 6, 4, 0, 2, 4, 1, 9, 3, 7, 3, 0, 3, 9, 9, 6, 6,
       0, 1, 3, 2, 6, 2, 3, 3, 7, 8, 8, 4, 5, 5, 9, 6, 9, 0, 9, 2, 4, 9,
       4, 6, 2, 2, 1, 7, 0, 3, 5, 5, 2, 2, 4, 1, 6, 1, 8, 6, 5, 9, 7, 7,
       1, 8, 6, 7, 8, 8, 5, 9, 1, 5, 4, 4, 2, 5, 8, 4, 0, 7, 6, 2, 5, 6,
       9, 0, 6, 7, 7, 5, 0, 9, 6, 3, 1, 2, 8, 3, 3, 6, 5, 7, 7, 0, 8, 6,
       6, 3, 7, 8, 1, 0, 5, 9, 8, 4, 4, 6, 8, 3, 4, 6, 5, 7, 8, 4, 7, 3,
       2, 5, 1, 3, 0, 6, 7, 9, 8, 4, 9, 8, 1, 2, 8, 8, 6, 2, 0, 1, 5, 1,
       3, 3, 0, 5, 1, 3, 7, 1, 8, 9, 6, 0, 2, 3, 4, 3, 1, 8, 7, 1, 2, 8,
       1, 2, 0, 5, 5, 9, 2, 2, 9, 4, 8, 4, 4, 4, 2, 2, 9, 3, 7, 4, 5, 5,
       7, 8, 5, 8, 9, 0, 2, 6, 0, 5, 1, 8, 9, 6, 3, 4, 4, 3, 7, 8, 3, 6,
       5, 9, 5, 3, 6, 3, 0, 3, 8, 3, 2, 1, 6, 6, 8, 7, 2, 1, 6, 6, 5, 8,
       3, 6, 6, 3, 0, 3, 1, 3, 1, 9, 5, 1, 4, 0, 0, 0, 8, 7, 1, 0, 5, 2,
       5, 2, 8, 4, 6, 7, 0, 9, 3, 0, 0, 3, 9, 9, 0, 3, 0, 1, 0, 7, 7, 3,
       4, 7, 2, 6, 8, 1, 9, 1, 3, 8, 3, 8])
```

In [77]:
```
logre.score(x_test,y_test)
```

Out[77]:
```
0.9629629629629629
```

In [78]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [35]:
```
b=a.head(10)
b
```

Out[35]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	Make
0	0	T-Roc	2019	25000	Automatic	13904	Diesel	145	49.6	2.0	VW
1	1	T-Roc	2019	26883	Automatic	4562	Diesel	145	49.6	2.0	VW
2	2	T-Roc	2019	20000	Manual	7414	Diesel	145	50.4	2.0	VW
3	3	T-Roc	2019	33492	Automatic	4825	Petrol	145	32.5	2.0	VW
4	4	T-Roc	2019	22900	Semi-Auto	6500	Petrol	150	39.8	1.5	VW
5	5	T-Roc	2020	31895	Manual	10	Petrol	145	42.2	1.5	VW
6	6	T-Roc	2020	27895	Manual	10	Petrol	145	42.2	1.5	VW
7	7	T-Roc	2020	39495	Semi-Auto	10	Petrol	145	32.5	2.0	VW
8	8	T-Roc	2019	21995	Manual	10	Petrol	145	44.1	1.0	VW
9	9	T-Roc	2019	23285	Manual	10	Petrol	145	42.2	1.5	VW

In [81]:
```
b=a[['PassengerId', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare','Embarked']]
b
```

Out[81]:

	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 [83]:
```
b['Embarked'].value_counts()
```

Out[83]:
```
S    6
Q    3
C    1
Name: Embarked, dtype: int64
```

In [84]:
```
x=b.drop('Embarked',axis=1)
y=b['Embarked']
```

In [39]:
```
g1={"Make":{'Make':1,'b':2}}
b=b.replace(g1)
print(b)
```

```
   Unnamed: 0  mileage  tax   mpg  engineSize Make
0           0    13904  145  49.6         2.0   VW
1           1     4562  145  49.6         2.0   VW
2           2     7414  145  50.4         2.0   VW
3           3     4825  145  32.5         2.0   VW
4           4     6500  150  39.8         1.5   VW
5           5       10  145  42.2         1.5   VW
6           6       10  145  42.2         1.5   VW
7           7       10  145  32.5         2.0   VW
8           8       10  145  44.1         1.0   VW
9           9       10  145  42.2         1.5   VW
```

In [85]:
```
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 [86]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [87]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[87]:
```
RandomForestClassifier()
```

In [88]:
```
parameters={'max_depth':[1,2,3,4,5],
 'min_samples_leaf':[5,10,15,20,25],
 'n_estimators':[10,20,30,40,50]}
```

In [89]:
```
from sklearn.model_selection import GridSearchCV
```

In [90]:
```
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: UserWarning: The least pop
ulated 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[90]:
```
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 [91]:
```
grid_search.best_score_
```

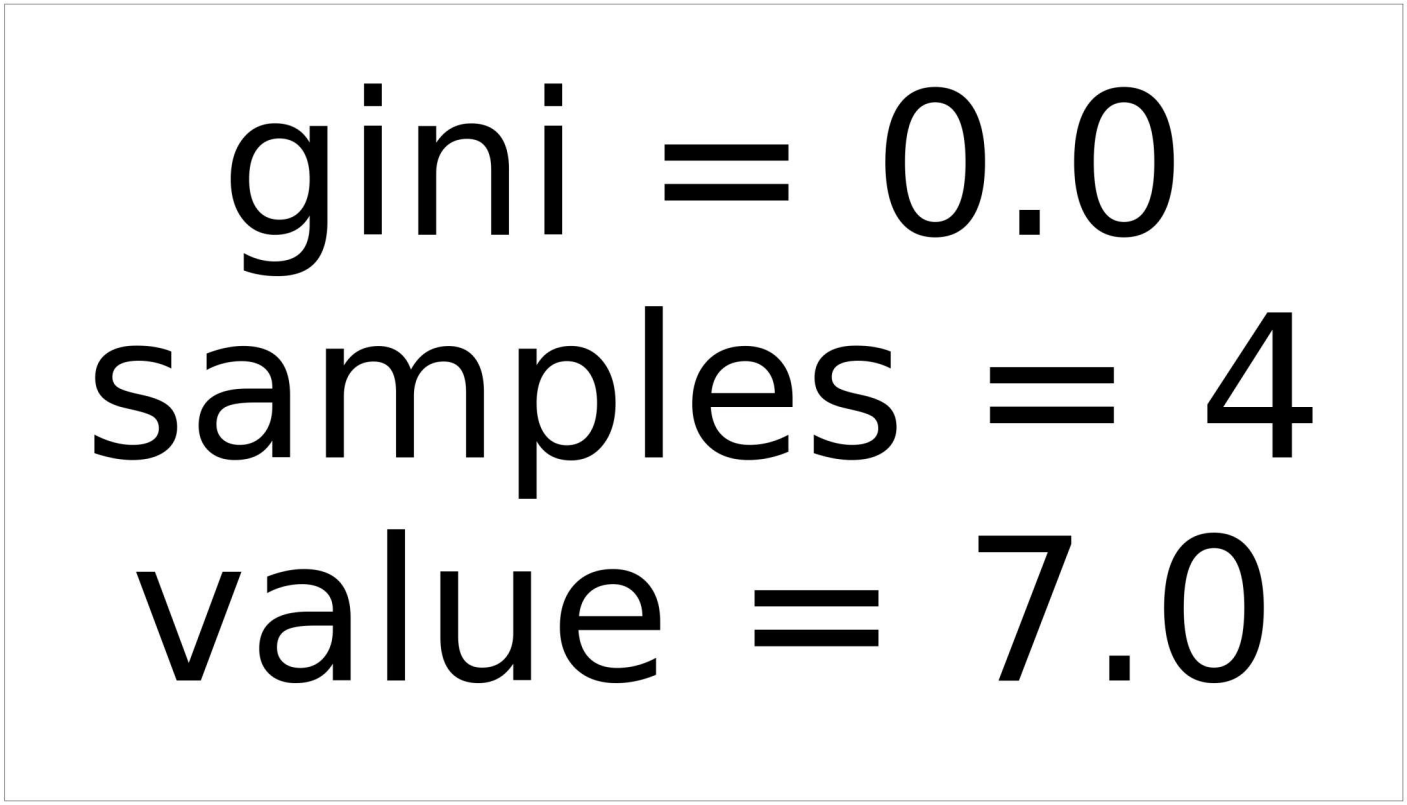
Out[91]:
```
0.5833333333333333
```

In [92]:
```
rfc_best=grid_search.best_estimator_
```

In [93]:
```
from sklearn.tree import plot_tree
```

In [50]:
```
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```

Out[50]:
```
[Text(2232.0, 1087.2, 'gini = 0.0\nsamples = 4\nvalue = 7.0')]
```



In [ ]: