

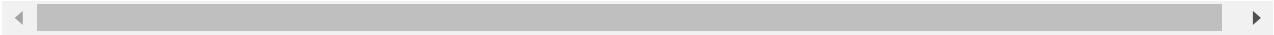
```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: df=pd.read_csv(r"C:\Users\user\Downloads\test gender.csv")
df
```

Out[2]:

	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	C
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	C
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	C
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	C

418 rows × 11 columns



In [3]:

```
df.columns
```

Out[3]:

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

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 418 entries, 0 to 417  
Data columns (total 11 columns):  
#   Column      Non-Null Count  Dtype  
---  ---  
0   PassengerId  418 non-null    int64  
1   Pclass       418 non-null    int64  
2   Name         418 non-null    object  
3   Sex          418 non-null    object  
4   Age          332 non-null    float64  
5   SibSp        418 non-null    int64  
6   Parch        418 non-null    int64  
7   Ticket       418 non-null    object  
8   Fare         417 non-null    float64  
9   Cabin        91 non-null     object  
10  Embarked     418 non-null    object  
dtypes: float64(2), int64(4), object(5)  
memory usage: 36.0+ KB
```

In [5]:

```
df['Sex'].value_counts()
```

Out[5]:

```
male      266  
female    152  
Name: Sex, dtype: int64
```

In [6]:

```
df['Sex'].value_counts()
```

Out[6]:

```
male      266  
female    152  
Name: Sex, dtype: int64
```

In [7]:

```
x=df[['PassengerId', 'Pclass', 'SibSp',  
      'Parch',]]  
y=df['Sex']
```

In [8]:

```
g1={"Sex":{"male":1,'female':2}}  
df=df.replace(g1)  
print(df)
```

	PassengerId	Pclass	Name	Sex	\
0	892	3	Kelly, Mr. James	1	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	2	
2	894	2	Myles, Mr. Thomas Francis	1	
3	895	3	Wirz, Mr. Albert	1	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	2	
..	
413	1305	3	Spector, Mr. Woolf	1	
414	1306	1	Oliva y Ocana, Dona. Fermina	2	

415	1307	3	Saether, Mr. Simon Sivertsen	1
416	1308	3	Ware, Mr. Frederick	1
417	1309	3	Peter, Master. Michael J	1

	Age	SibSp	Parch		Ticket	Fare	Cabin	Embarked
0	34.5	0	0		330911	7.8292	NaN	Q
1	47.0	1	0		363272	7.0000	NaN	S
2	62.0	0	0		240276	9.6875	NaN	Q
3	27.0	0	0		315154	8.6625	NaN	S
4	22.0	1	1		3101298	12.2875	NaN	S
...
413	NaN	0	0	A.5.	3236	8.0500	NaN	S
414	39.0	0	0	PC	17758	108.9000	C105	C
415	38.5	0	0	SOTON/O.Q.	3101262	7.2500	NaN	S
416	NaN	0	0		359309	8.0500	NaN	S
417	NaN	1	1		2668	22.3583	NaN	C

[418 rows x 11 columns]

```
In [9]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.70)
```

```
In [10]: from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[10]: RandomForestClassifier()

```
In [11]: parameters= {
    "max_depth":[1,2,3,4,5],
    "min_samples_leaf":[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

```
In [12]: from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

```
Out[12]: 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')
```

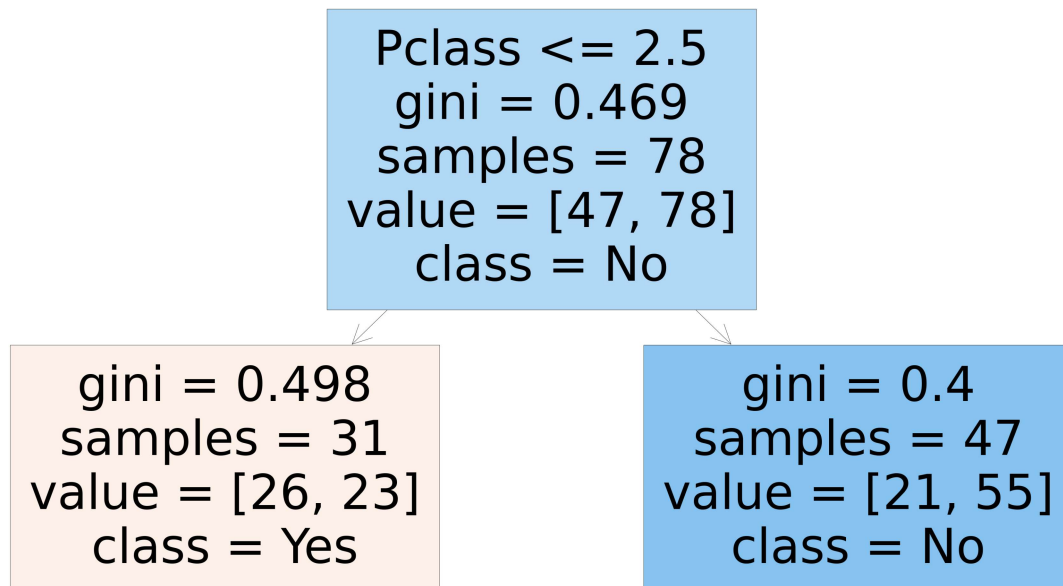
```
In [13]: grid_search.best_score_
```

Out[13]: 0.6159754224270353

```
In [14]: rfc_best=grid_search.best_estimator_
```

```
In [15]: from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],fill
```

```
Out[15]: [Text(2232.0, 1630.8000000000002, 'Pclass <= 2.5\ngini = 0.469\nsamples = 78\nvalue = [47, 78]\nnclass = No'),  
Text(1116.0, 543.5999999999999, 'gini = 0.498\nsamples = 31\nvalue = [26, 23]\nnclass = Yes'),  
Text(3348.0, 543.5999999999999, 'gini = 0.4\nsamples = 47\nvalue = [21, 55]\nnclass = No')]
```



In []: