In [1]:

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

In [2]:

```
df=pd.read_csv('titanic_test.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	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
			Hirvonen,								

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 [22]:

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

In [23]:

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

```
     PassengerId  Pclass                                          Name  \
0            892       3                              Kelly, Mr. James
1            893       3              Wilkes, Mrs. James (Ellen Needs)
2            894       2                     Myles, Mr. Thomas Francis
3            895       3                             Wirz, Mr. Albert
4            896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)
..           ...     ...                                           ...
413         1305       3                           Spector, Mr. Woolf
414         1306       3                 Oliva y Ocana, Dona. Fermina
415         1307       3                 Saether, Mr. Simon Sivertsen
416         1308       3                          Ware, Mr. Frederick
417         1309       3                     Peter, Master. Michael J

        Sex   Age  SibSp  Parch              Ticket      Fare Cabin Embark
ed
0      male  34.5      0      0              330911    7.8292   NaN
Q
1    female   NaN      1      0              363272   21.6792   NaN
S
2      male  22.5      0      0              240276    9.6875   NaN
Q
3      male  22.5      0      0              315154    8.6625   NaN
S
4    female   NaN      1      0             3101298   12.2875   NaN
S
..      ...   ...    ...    ...                 ...       ...   ...
...
413    male   NaN      0      0           A.5. 3236    8.0500   NaN
S
414  female   NaN      0      0            PC 17758  108.9000  C105
C
415    male  38.5      0      0  SOTON/O.Q. 3101262    7.2500   NaN
S
416    male   NaN      0      0              359309    8.0500   NaN
S
417    male   NaN      1      0                2668   22.3583   NaN
C

[418 rows x 11 columns]
```

In [24]:

```
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 [25]:

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[25]:

```
RandomForestClassifier()
```

Depth of Tree

In [26]:

```
parameters={"max_depth":[1,2,3,4,5],"min_samples_leaf":[5,23,45,76,78],'n_estimators':[10
```

Cross Validate

In [27]:

```
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[27]:

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 23, 45, 76, 78],
                         'n_estimators': [10, 23, 45, 65, 7]},
             scoring='accuracy')
```

Score

In [28]:

```
grid_search.best_score_
```

Out[28]:

```
0.6242959549411162
```

In [29]:

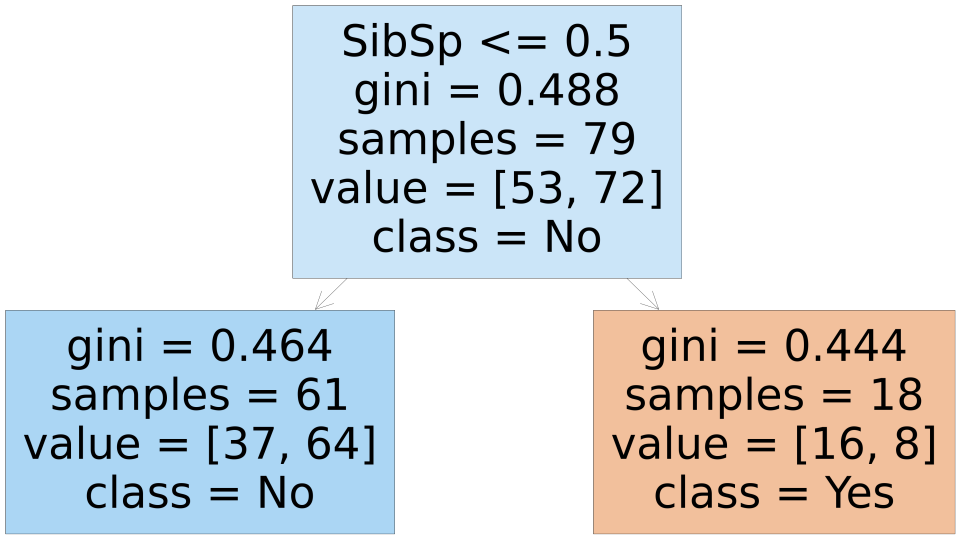
```
rfc_best=grid_search.best_estimator_
```

In [30]:

```
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'],filled
```

Out[30]:

```
[Text(2232.0, 1630.8000000000002, 'SibSp <= 0.5\ngini = 0.488\nsamples = 7
9\nvalue = [53, 72]\nclass = No'),
 Text(1116.0, 543.5999999999999, 'gini = 0.464\nsamples = 61\nvalue = [37,
64]\nclass = No'),
 Text(3348.0, 543.5999999999999, 'gini = 0.444\nsamples = 18\nvalue = [16,
8]\nclass = Yes')]
```



In [ ]: