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

In [143]:
```
data=pd.read_csv(r"C:\Users\user\Desktop\vicky\C2_test.gender_submission (1).csv"
```

In [144]:
```
data.fillna(value=1)
```

Out[144]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emb
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	1	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	1	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	1	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	1	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	1	
...	...	...	...	...	...	...	...	...	...	...	
413	1305	3	Spector, Mr. Woolf	male	1.0	0	0	A.5. 3236	8.0500	1	
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	1	
416	1308	3	Ware, Mr. Frederick	male	1.0	0	0	359309	8.0500	1	
417	1309	3	Peter, Master. Michael J	male	1.0	1	1	2668	22.3583	1	

418 rows × 11 columns

In [145]:
```
data.head()
```

Out[145]:

	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

In [146]:
```
data.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 [147]:
```
data1=data[['PassengerId','Pclass','Sex','SibSp','Parch']]
```

In [148]:
```
data1['Sex'].value_counts()
```

Out[148]:
```
male      266
female    152
Name: Sex, dtype: int64
```

In [130]:
```
x=data1.drop('Sex',axis=1)
y=data1['Sex']
```

In [ ]:
```

```

In [149]:
```
g1={"Sex":{'male':1,'female':0,}}
data1=data1.replace(g1)
print(data1)
```

```
     PassengerId  Pclass  Sex  SibSp  Parch
0            892       3    1      0      0
1            893       3    0      1      0
2            894       2    1      0      0
3            895       3    1      0      0
4            896       3    0      1      1
..           ...     ...  ...    ...    ...
413         1305       3    1      0      0
414         1306       1    0      0      0
415         1307       3    1      0      0
416         1308       3    1      0      0
417         1309       3    1      1      1

[418 rows x 5 columns]
```

In [150]:
```
from sklearn.model_selection import train_test_split
```

In [151]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [152]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [153]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[153]:
```
RandomForestClassifier()
```

In [154]:
```
parameters = {'max_depth':[1,2,3,4,5],
              'min_samples_leaf':[5,10,15,20,25],
              'n_estimators':[10,20,30,40,50]

}
```

In [155]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accura
grid_search.fit(x_train,y_train)
```

Out[155]:
```
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 [156]:
```
grid_search.best_score_
```

Out[156]:
```
0.8465947403910992
```

In [157]:
```
from sklearn.tree import plot_tree
```

In [158]:
```
rfc_best=grid_search.best_estimator_
```

In [159]:
```
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'
```

Out[159]:
```
[Text(2232.0, 1630.8000000000002, 'male <= 0.5\ngini = 0.262\nsamples = 1885\nva
lue = [2506, 460]\nclass = Yes'),
 Text(1116.0, 543.5999999999999, 'gini = 0.23\nsamples = 1065\nvalue = [1449, 22
1]\nclass = Yes'),
 Text(3348.0, 543.5999999999999, 'gini = 0.301\nsamples = 820\nvalue = [1057, 23
9]\nclass = Yes')]
```

In [ ]:

In [ ]: