

In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv('exp4.csv')
df.head()
```

Out[2]:

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	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	S

In [3]:

```
df.drop(['PassengerId', 'Name', 'SibSp', 'Parch', 'Ticket', 'Cabin', 'Embarked'], axis='columns')
df.head()
```

Out[3]:

	Survived	Pclass	Sex	Age	Fare
0	0	3	male	22.0	7.2500
1	1	1	female	38.0	71.2833
2	1	3	female	26.0	7.9250
3	1	1	female	35.0	53.1000
4	0	3	male	35.0	8.0500

In [17]:

```
inputs = df.drop('Survived', axis='columns')
inputs.head()
```

Out[17]:

	Pclass	Sex	Age	Fare
0	3	male	22.0	7.2500
1	1	female	38.0	71.2833
2	3	female	26.0	7.9250
3	1	female	35.0	53.1000
4	3	male	35.0	8.0500

In [5]:

```
inputs.Sex = inputs.Sex.map({'male':1, 'female':2})
inputs
```

Out[5]:

	Pclass	Sex	Age	Fare
0	3	1	22.0	7.2500
1	1	2	38.0	71.2833
2	3	2	26.0	7.9250
3	1	2	35.0	53.1000
4	3	1	35.0	8.0500
...
886	2	1	27.0	13.0000
887	1	2	19.0	30.0000
888	3	2	NaN	23.4500
889	1	1	26.0	30.0000
890	3	1	32.0	7.7500

891 rows × 4 columns

In [6]:

```
inputs.Age = inputs.Age.fillna(inputs.Age.mean())
inputs
```

Out[6]:

	Pclass	Sex	Age	Fare
count	891.000000	891.000000	891.000000	891.000000
mean	2.308642	1.352413	29.699118	32.204208
std	0.836071	0.477990	13.002015	49.693429
min	1.000000	1.000000	0.420000	0.000000
25%	2.000000	1.000000	22.000000	7.910400
50%	3.000000	1.000000	29.699118	14.454200
75%	3.000000	2.000000	35.000000	31.000000
max	3.000000	2.000000	80.000000	512.329200

In [7]:

```
target = df.Survived
target.head()
```

Out[7]:

```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

In [8]:

```
from sklearn import tree
from sklearn.model_selection import train_test_split
```

In [9]:

```
x_train, x_test, y_train, y_test = train_test_split(inputs, target, test_size=0.3)
```

In [10]:

```
model = tree.DecisionTreeClassifier()

model.fit(x_train, y_train)
model.score(x_test, y_test)
```

Out[10]:

```
0.7947761194029851
```

In [11]:

```
print(x_test)
print(y_test)
```

	Pclass	Sex	Age	Fare
297	1	2	2.000000	151.5500
486	1	2	35.000000	90.0000
825	3	1	29.699118	6.9500
836	3	1	21.000000	8.6625
509	3	1	26.000000	56.4958
..
243	3	1	22.000000	7.1250
121	3	1	29.699118	8.0500
272	2	2	41.000000	19.5000
701	1	1	35.000000	26.2875
455	3	1	29.000000	7.8958

[268 rows x 4 columns]

```
297    0
486    1
825    0
836    0
509    1
..
243    0
121    0
272    1
701    1
455    1
```

Name: Survived, Length: 268, dtype: int64

In [12]:

```
model.predict([[2, 1, 23.000000, 13.0000]])
```

C:\Users\manth\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
 warnings.warn(

Out[12]:

```
array([0], dtype=int64)
```

In [13]:

```
text_tree = tree.export_text(model)
print(text_tree)
```

```
|--- feature_1 <= 1.50
|   |--- feature_0 <= 1.50
|   |   |--- feature_2 <= 57.00
|   |   |   |--- feature_2 <= 17.50
|   |   |   |   |--- class: 1
|   |   |   |--- feature_2 > 17.50
|   |   |   |   |--- feature_3 <= 37.00
|   |   |   |   |   |--- feature_3 <= 26.11
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_3 > 26.11
|   |   |   |   |   |   |--- feature_3 <= 27.14
|   |   |   |   |   |   |   |--- feature_2 <= 31.85
|   |   |   |   |   |   |   |   |--- feature_2 <= 28.85
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_2 > 28.85
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_2 > 31.85
|   |   |   |   |   |   |   |--- feature_2 <= 43.50
|   |   |   |   |   |   |   |   |--- class: 1
```

In [14]:

```
from matplotlib import pyplot as plt
```

In [15]:

```
fig = plt.figure(figsize=(200,200))
tree_ = tree.plot_tree(model, feature_names=["Pclass", "Sex", "Age", "Fare"], class_names='Survived')
```