Build a Decision Tree Model to Predict Survival

In this task, we will build a Decision Tree model using the Titanic dataset to predict whether a person would survive or not, based on the following features:

-Pclass (Passenger Class)

- Sex
- Age
- Fare

Evaluation Metric:

Model Score: Accuracy of the model on the test set.

In [40]:
```
import pandas as pd

# Load CSV
df = pd.read_csv("/Users/priyanshi/Downloads/titanic.csv")
print(df.head())
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

In [41]:
```
df.drop(['PassengerId','Name','SibSp','Parch','Ticket','Cabin','Embarked'],axis='columns',inplace=True)
df
```

Out[41]:

	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
...	...	...	...	...	...
886	0	2	male	27.0	13.0000
887	1	1	female	19.0	30.0000
888	0	3	female	NaN	23.4500
889	1	1	male	26.0	30.0000
890	0	3	male	32.0	7.7500

891 rows × 5 columns

In [42]:
```
inputs = df.drop('Survived',axis='columns')
target = df.Survived
```

In [43]:
```
inputs.Sex = inputs.Sex.map({'male': 1, 'female': 2})
inputs.Age[:10]
```

Out[43]:
```
0    22.0
1    38.0
2    26.0
3    35.0
4    35.0
5     NaN
6    54.0
7     2.0
8    27.0
9    14.0
Name: Age, dtype: float64
```

In [44]:
```
inputs.Age = inputs.Age.fillna(inputs.Age.mean())
```

In [45]:
```
from sklearn.model_selection import train_test_split
```

The data split into 80% training and 20% testing sets.

In [47]:
```
X_train, X_test, y_train, y_test = train_test_split(inputs,target,test_size=0.2)
len(X_train)
```

Out[47]: 712

In [48]:
```
len(X_test)
```

Out[48]: 179

In [49]:
```
from sklearn import tree
model = tree.DecisionTreeClassifier()
model.fit(X_train,y_train)
```

Out[49]:

▾ DecisionTreeClassifier

DecisionTreeClassifier()

In [52]:
```
model.score(X_test,y_test)
```

Out[52]:
```
0.7374301675977654
```

In [53]:
```
model.predict([[3,1,22,7.2500]])
```

```
/Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages/sklearn/utils/validation.p
y:2739: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with featur
e names
  warnings.warn(
```

Out[53]:
```
array([0])
```

The person with Pclass 3, Male gender, 22 years old, and a fare of 7.2 was predicted did not survive according to the Decision Tree model.

In [54]:
```
model.predict([[1,2,38.0,71.28]])
```

```
/Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages/sklearn/utils/validation.p
y:2739: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with featur
e names
  warnings.warn(
```

Out[54]:
```
array([1])
```

In [ ]:
```
The person with Pclass 1, Female gender, 38years old, and a fare of 71.28 was predicted to survive accordir
```