```
# Step 1: Import libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, confusion_matrix


# Step 2: Load data
titanic = sns.load_dataset('titanic')
import seaborn as sns




# Step 3: Look at data
print(titanic.head())
print(titanic.info())
```

```
   survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
0         0       3    male  22.0      1      0   7.2500        S  Third
1         1       1  female  38.0      1      0  71.2833        C  First
2         1       3  female  26.0      0      0   7.9250        S  Third
3         1       1  female  35.0      1      0  53.1000        S  First
4         0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
None
```

```
# Step 4: Choose target value (what we want to predict)
# Target = survived


# Step 5: Choose input features (what helps us predict)
features = ['pclass', 'sex', 'age', 'sibsp', 'parch', 'fare']


# Step 6: Clean data (simple way)
titanic = titanic[features + ['survived']]
titanic.dropna(inplace=True)  # Remove rows with missing values
titanic['sex'] = titanic['sex'].map({'male': 0, 'female': 1})  # Convert sex to numeric
```

```
/tmp/ipython-input-15-4013291390.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  titanic.dropna(inplace=True)  # Remove rows with missing values
/tmp/ipython-input-15-4013291390.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the doc                                    ixing.html#returning-a-view-versus
  titanic['sex'] = titanic['sex'].map({'male': 0, 'female': 1})  # Convert sex to numeric
```

```
# Step 7: Split data (train & test)
X = titanic[features]
y = titanic['survived']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Step 8: Build the Decision Tree
model = DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)
```

```
DecisionTreeClassifier(random_state=42)
```

```
# Step 9: Make prediction
y_pred = model.predict(X_test)
```

```
# Step 10: Check how good the model is (calculate accuracy)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.7132867132867133
```

```
# Step 11: Visualize the Decision Tree
plt.figure(figsize=(15, 10))
plot_tree(model, feature_names=features, class_names=['Died', 'Survived'], filled=True)
plt.title("Decision Tree - Titanic")
plt.show()
```