DECISION TREE IMPLEMENTATION

May 15, 2025

```python
[2]: import csv

data = [
    ['Outlook', 'Temperature', 'Humidity', 'Windy', 'PlayTennis'],
    ['Sunny', 'Hot', 'High', False, 'No'],
    ['Sunny', 'Hot', 'High', True, 'Yes'],
    ['Overcast', 'Hot', 'High', False, 'Yes'],
    ['Rainy', 'Mild', 'High', False, 'Yes'],
    ['Rainy', 'Cool', 'Normal', False, 'Yes'],
    ['Rainy', 'Cool', 'Normal', True, 'No'],
    ['Overcast', 'Cool', 'Normal', True, 'Yes'],
    ['Sunny', 'Mild', 'High', False, 'No'],
    ['Sunny', 'Cool', 'Normal', True, 'Yes'],
    ['Rainy', 'Mild', 'Normal', False, 'Yes'],
    ['Sunny', 'Mild', 'Normal', True, 'Yes'],
    ['Overcast', 'Mild', 'High', True, 'Yes'],
    ['Overcast', 'Hot', 'Normal', False, 'Yes'],
    ['Rainy', 'Mild', 'High', True, 'No']
]

file_name = 'tennisdata.csv'
with open(file_name, mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerows(data)

print(f"CSV file '{file_name}' created successfully!")
```

```
CSV file 'tennisdata.csv' created successfully!
```

```python
[9]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix
```

```python
file_path = 'tennisdata.csv'
df = pd.read_csv(file_path)

df_encoded = df.copy()
categorical_columns = ['Outlook', 'Temperature', 'Humidity', 'Windy',
 'PlayTennis']
for col in categorical_columns:
    df_encoded[col] = df_encoded[col].astype('category').cat.codes

X = df_encoded.drop('PlayTennis', axis=1)
y = df_encoded['PlayTennis']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
 random_state=42)

clf = DecisionTreeClassifier(random_state=1)
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print("\nClassification Report:\n", classification_report(y_test, y_pred))

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['No', 'Yes'],
 yticklabels=['No', 'Yes'])
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion Matrix')
plt.show()

plt.figure(figsize=(18, 10))
plot_tree(clf, filled=True, feature_names=X.columns, class_names=['No', 'Yes'],
 rounded=True, fontsize=10)
plt.title('Decision Tree Visualization')
plt.show()
```
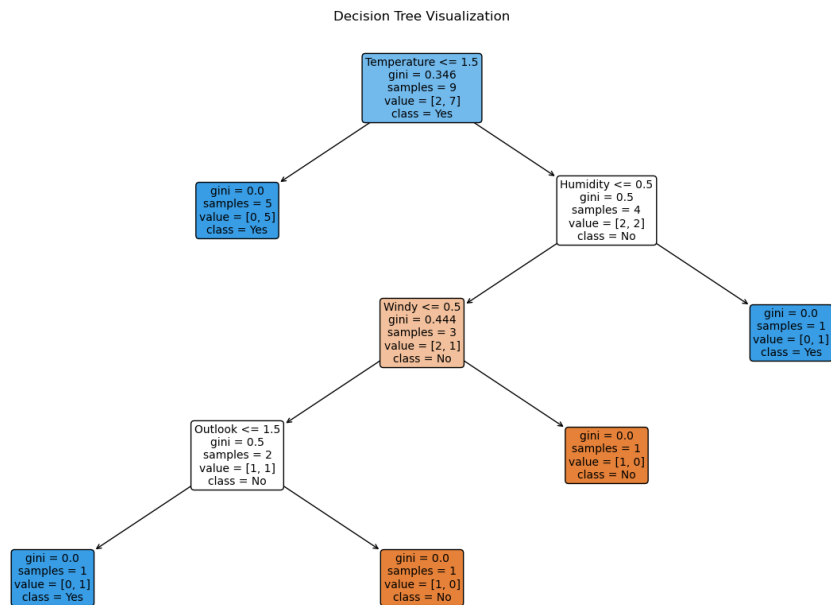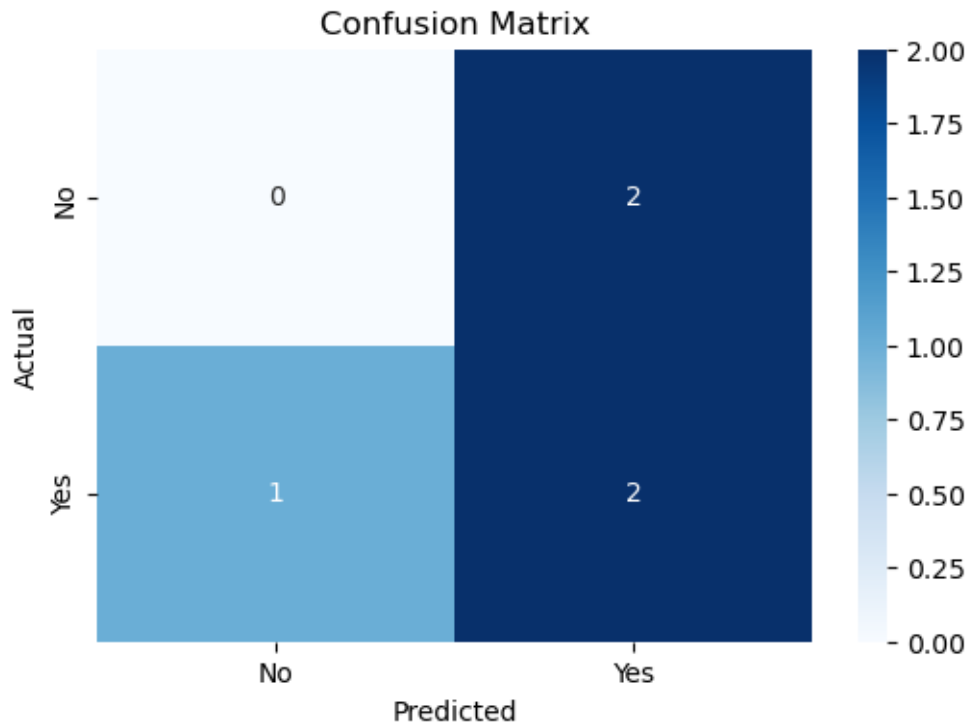
```
Accuracy: 0.40

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         2
           1       0.50      0.67      0.57         3
```

accuracy			0.40	5
macro avg	0.25	0.33	0.29	5
weighted avg	0.30	0.40	0.34	5

Confusion Matrix

Decision Tree Visualization

[ ]: