

week7-b

September 20, 2024

```
[1]: import pandas as pd
from sklearn.tree import DecisionTreeClassifier, plot_tree
import matplotlib.pyplot as plt
import seaborn as sns

data = {
    'Outlook': ['Sunny', 'Sunny', 'Overcast', 'Rain', 'Rain', 'Rain',
    ↪ 'Overcast', 'Sunny', 'Sunny', 'Rain', 'Sunny', 'Overcast', 'Overcast',
    ↪ 'Rain'],
    'Temperature': ['Hot', 'Hot', 'Hot', 'Mild', 'Cool', 'Cool', 'Cool',
    ↪ 'Mild', 'Cool', 'Mild', 'Mild', 'Mild', 'Hot', 'Mild'],
    'Humidity': ['High', 'High', 'High', 'High', 'Normal', 'Normal', 'Normal',
    ↪ 'High', 'Normal', 'Normal', 'High', 'Normal', 'Normal', 'High'],
    'Wind': ['Weak', 'Strong', 'Weak', 'Weak', 'Weak', 'Strong', 'Strong',
    ↪ 'Weak', 'Weak', 'Weak', 'Strong', 'Strong', 'Weak', 'Strong'],
    'PlayTennis': ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes',
    ↪ 'Yes', 'Yes', 'Yes', 'Yes', 'No']
}

df = pd.DataFrame(data)

features = ['Outlook', 'Temperature', 'Humidity', 'Wind']
target = 'PlayTennis'

X = pd.get_dummies(df[features])
y = df[target]

clf = DecisionTreeClassifier()
clf.fit(X, y)

plt.figure(figsize=(12, 8))
sns.set(style="whitegrid")
plot_tree(clf, feature_names=X.columns, class_names=clf.classes_, filled=True,
    ↪ rounded=True)
plt.title("Decision Tree for PlayTennis Dataset", fontsize=16)
plt.show()
```

```

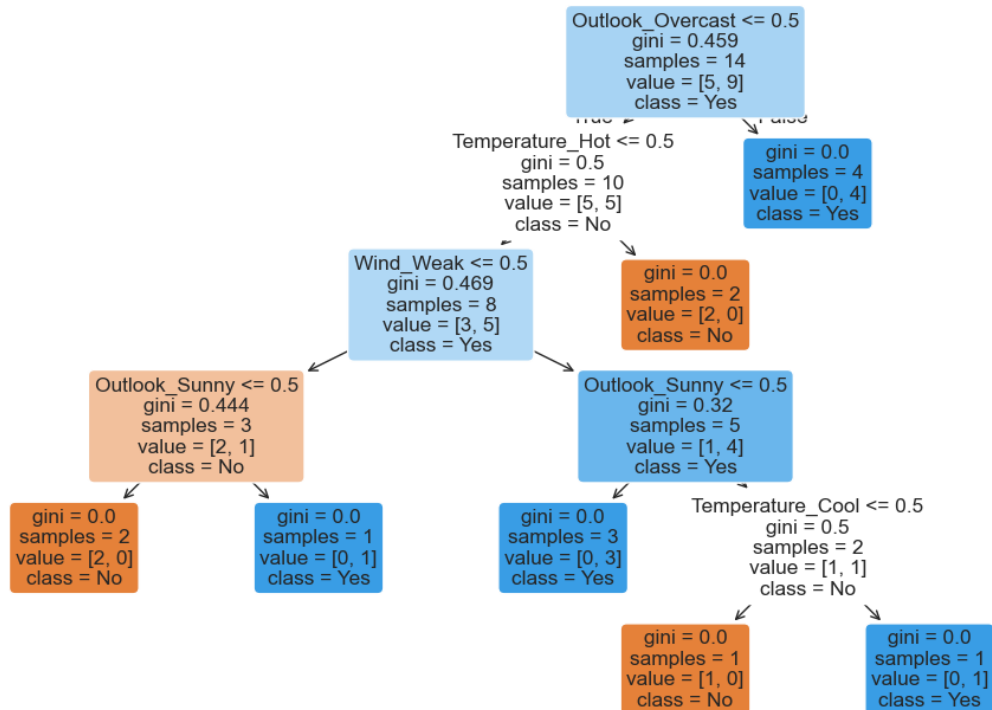
new_sample = {
    'Outlook': input("Enter value for Outlook (Sunny/Overcast/Rain): "),
    'Temperature': input("Enter value for Temperature (Hot/Mild/Cool): "),
    'Humidity': input("Enter value for Humidity (High/Normal): "),
    'Wind': input("Enter value for Wind (Weak/Strong): ")
}

new_sample_df = pd.DataFrame([new_sample])
feature_dummies = pd.get_dummies(new_sample_df)
new_sample_df_aligned = feature_dummies.reindex(columns=X.columns, fill_value=0)

prediction = clf.predict(new_sample_df_aligned)
print(f"Prediction for new sample: {prediction[0]}")

```

Decision Tree for PlayTennis Dataset



Prediction for new sample: No

- 1) Dataset: A table with weather conditions (Outlook, Temperature, Humidity, Wind) and the target variable PlayTennis.
- 2) Data Preprocessing: Categorical weather data is transformed into numerical format using one-hot encoding for compatibility with Scikit-learn's DecisionTreeClassifier.
- 3) Model Training: The decision tree classifier is trained on 14 weather samples to identify patterns in the data.
- 4) New Sample Prediction: Given user input (e.g., Sunny, Cool), the trained model predicts

whether tennis can be played.

Inputs Given: Rain Cool High Strong

Output: No