import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.tree import DecisionTreeClassifier, plot\_tree

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

# Loading the dataset

df = pd.read\_csv("play\_tennis.csv")

# Encoding categorical variables

le = LabelEncoder()

for col in df.columns:

df[col] = le.fit\_transform(df[col])

# Splitting dataset

X = df.drop(columns=['PlayTennis'])

y = df['PlayTennis']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Training the Decision Tree Model

dt\_model = DecisionTreeClassifier(criterion='entropy', random\_state=42)

dt\_model.fit(X\_train, y\_train)

# Visualizing the Decision Tree

plt.figure(figsize=(12, 6))

plot\_tree(dt\_model, feature\_names=X.columns, class\_names=['No', 'Yes'], filled=True)

plt.show()

# Making predictions

y\_pred = dt\_model.predict(X\_test)

# Evaluating the model

accuracy = accuracy\_score(y\_test, y\_pred)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

class\_report = classification\_report(y\_test, y\_pred)

print(f'Accuracy: {accuracy}\n')

print('Confusion Matrix:\n', conf\_matrix)

print('\nClassification Report:\n', class\_report)

# Display confusion matrix

plt.figure(figsize=(6, 4))

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['No', 'Yes'], yticklabels=['No', 'Yes'])

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix')

plt.show()