# Import necessary libraries

import pandas as pd

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

from sklearn.tree import DecisionTreeRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score

import matplotlib.pyplot as plt

# Generate a simple example dataset for illustration purposes

data = {

    'fixed\_acidity': [7.4, 7.8, 7.8, 11.2, 7.4],

    'volatile\_acidity': [0.7, 0.88, 0.76, 0.28, 0.7],

    'citric\_acid': [0.0, 0.0, 0.04, 0.56, 0.0],

    'residual\_sugar': [1.9, 2.6, 2.3, 1.9, 1.9],

    'quality': [5, 5, 5, 6, 5]

}

df = pd.DataFrame(data)

# Assume 'quality' is the target variable, and other columns are features

X = df.drop('quality', axis=1)

y = df['quality']

# Split the data into training and testing sets

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

# Create a Decision Tree Regressor

model = DecisionTreeRegressor(random\_state=42)

# Train the model

model.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = model.predict(X\_test)

# Evaluate the model

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f'Mean Squared Error: {mse}')

print(f'R-squared: {r2}')

# Plotting predicted vs. actual values

plt.scatter(y\_test, y\_pred)

plt.xlabel('Actual Quality')

plt.ylabel('Predicted Quality')

plt.title('Actual vs. Predicted Wine Quality')

plt.show()

output:

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/\_regression.py:918: UndefinedMetricWarning: R^2 score is not well-defined with less than two samples.

warnings.warn(msg, UndefinedMetricWarning)

Mean Squared Error: 0.0

R-squared: nan

