import pandas as pd

import numpy as np

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

from sklearn.ensemble import RandomForestRegressor

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

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset

data = pd.read\_csv('synthetic\_soil\_moisture\_data.csv')

# Display the first few rows of the dataset

print("Dataset Preview:")

print(data.head())

# Display summary statistics

print("\nSummary Statistics:")

print(data.describe())

# Check for missing values

print("\nMissing Values:")

print(data.isnull().sum())

# Fill missing values (if any) with mean of the column

data.fillna(data.mean(), inplace=True)

def predict\_and\_evaluate(target\_variable):

# Define features and target variable

features = data.drop(target\_variable, axis=1)

target = data[target\_variable]

# Split the data into training and testing sets

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

# Initialize the model

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

# Train the model

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

# Make predictions

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

# Calculate evaluation metrics

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

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

print(f'\n{target\_variable.capitalize()} Prediction:')

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

print(f'R^2 Score: {r2}')

# Plotting the results

plt.figure(figsize=(10, 6))

plt.scatter(y\_test, y\_pred, alpha=0.5)

plt.xlabel(f'Actual {target\_variable.capitalize()}')

plt.ylabel(f'Predicted {target\_variable.capitalize()}')

plt.title(f'Actual vs Predicted {target\_variable.capitalize()}')

plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'k--', lw=2)

plt.show()

# Predict and evaluate soil moisture

predict\_and\_evaluate('soil\_moisture')

# Predict and evaluate temperature

predict\_and\_evaluate('temperature')

# Predict and evaluate humidity

predict\_and\_evaluate('humidity')