Source code

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestRegressor

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

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

# 1. Create a sample air quality dataset

np.random.seed(42)

date\_range = pd.date\_range(start="2023-01-01", periods=200, freq='H')

data = {

'Temperature': np.random.normal(20, 5, 200),

'Humidity': np.random.normal(60, 10, 200),

'WindSpeed': np.random.normal(3, 1, 200),

'PM2.5': np.random.normal(25, 8, 200),

'NO2': np.random.normal(40, 12, 200) # Target variable

}

df = pd.DataFrame(data, index=date\_range)

print("Sample Data:\n", df.head())

# 2. Define features and target

X = df[['Temperature', 'Humidity', 'WindSpeed', 'PM2.5']]

y = df['NO2']

# 3. Split data

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

# 4. Train model

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

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

# 5. Make predictions

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

# 6. Evaluate

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

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

print("\nModel Evaluation:")

print(f"Mean Squared Error: {mse:.2f}")

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

# 7. Plot actual vs predicted

plt.figure(figsize=(10, 5))

plt.plot(y\_test.values[:50], label='Actual', marker='o')

plt.plot(y\_pred[:50], label='Predicted', marker='x')

plt.title("Actual vs Predicted NO₂ Levels")

plt.xlabel("Sample Index")

plt.ylabel("NO₂")

plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()