import pandas as pd

import numpy as np

from sklearn.ensemble import RandomForestRegressor

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

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

import matplotlib.pyplot as plt

# 1. Load Zomato stock data (download from Yahoo Finance or Kaggle)

df = pd.read\_csv('zomato-1.csv')  # CSV should contain 'Date' and 'Close' columns

df['Date'] = pd.to\_datetime(df['Date'])

df.set\_index('Date', inplace=True)

# 2. Feature Engineering: Create lag features

df['Close\_1'] = df['Close'].shift(1)

df['Close\_2'] = df['Close'].shift(2)

df['Close\_3'] = df['Close'].shift(3)

df.dropna(inplace=True)

# 3. Define features and target

X = df[['Close\_1', 'Close\_2', 'Close\_3']]

y = df['Close']

# 4. Train-test split

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

# 5. Train Random Forest Regressor

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

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

# 6. Make predictions

predictions = model.predict(X\_test)

# 7. Evaluate performance

mae = mean\_absolute\_error(y\_test, predictions)

rmse = np.sqrt(mean\_squared\_error(y\_test, predictions))

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

print(f"MAE: {mae:.2f}")

print(f"RMSE: {rmse:.2f}")

print(f"R² Score: {r2:.2f}")

# 8. Plot actual vs predicted

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

plt.plot(y\_test.index, y\_test.values, label='Actual Price', color='blue')

plt.plot(y\_test.index, predictions, label='Predicted Price', color='orange')

plt.title('Zomato Stock Price: Actual vs Predicted')

plt.xlabel('Date')

plt.ylabel('Stock Price')

plt.legend()

plt.grid(True)

plt.show()