import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler

from scipy import stats

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

# Step 1: Generate Synthetic Time Series Data

np.random.seed(42)

date\_range = pd.date\_range(start='2024-01-01', periods=60, freq='D')

prices = np.random.normal(loc=100, scale=5, size=60)

# Introduce missing values

missing\_indices = np.random.choice(60, 6, replace=False)

prices[missing\_indices] = np.nan

# Introduce outliers

outlier\_indices = np.random.choice(60, 3, replace=False)

prices[outlier\_indices] = [200, 250, 180]

# Create DataFrame

df = pd.DataFrame({'Date': date\_range, 'Price': prices})

# Save to CSV

csv\_path = 'petrol\_prices\_timeseries.csv'

df.to\_csv(csv\_path, index=False)

# Step 2: Load Data

loaded\_df = pd.read\_csv(csv\_path, parse\_dates=['Date'])

# Step 3: Handle Missing Data

loaded\_df['Price'].fillna(method='ffill', inplace=True) # Forward fill

# Step 4: Detect & Remove Outliers

z\_scores = np.abs(stats.zscore(loaded\_df['Price']))

loaded\_df = loaded\_df[z\_scores < 2] # Keep values within 2 standard deviations

# Step 5: Normalize Data

scaler = MinMaxScaler()

loaded\_df['Normalized\_Price'] = scaler.fit\_transform(loaded\_df[['Price']])

# Step 6: Prepare Data for Linear Regression Model

loaded\_df['Days'] = (loaded\_df['Date'] - loaded\_df['Date'].min()).dt.days

X = loaded\_df[['Days']]

y = loaded\_df['Price']

# Split 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)

# Train Linear Regression Model

model = LinearRegression()

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

# Make Predictions

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

# Evaluate Model

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

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

# Step 7: Visualize Predictions

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

plt.scatter(X\_test, y\_test, color='blue', label='Actual Prices')

plt.plot(X\_test, y\_pred, color='red', linewidth=2, label='Predicted Prices')

plt.xlabel('Days')

plt.ylabel('Price')

plt.title('Linear Regression Forecasting of Petrol Prices')

plt.legend()

plt.grid(True)

plt.show()