EX:No.4 221501043

25/01/25

**Program to Develop a linear regression model for forecasting time series data.**

**Aim:**

Write a program to implement time series data for import library, load data, Preprocessing and visualising.

**Algorithm:**

1. **Load the Data**:
   * Read the CSV file containing the weather data.
   * Parse the date column as a datetime index.
2. **Clean the Data**:
   * Handle missing values by performing forward and backward filling.
   * Drop any remaining NaN values.
3. **Normalize the Data**:
   * Apply **Min-Max Scaling** to normalize each column's values between 0 and 1.
4. **Add Time-Based Features**:
   * Extract additional features from the datetime index: day, month and year
5. **Visualize the Data**:
   * Plot the time series for a specific column (e.g., temperature T) over time.
6. **Execute the Program**:
   * Sequentially call the functions to load, clean, normalize, add features, and visualize the data.

**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# 1. Simulate Energy Consumption Data

# Let's create a sample hourly dataset for 60 days

date\_rng = pd.date\_range(start='2024-01-01', end='2024-03-01', freq='H')

np.random.seed(42)

energy\_usage = np.random.normal(loc=1.5, scale=0.5, size=len(date\_rng)) + np.linspace(0, 3, len(date\_rng)) # adding a trend

df = pd.DataFrame({'timestamp': date\_rng, 'energy\_kWh': energy\_usage})

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

# 2. Plot Original Data

plt.figure(figsize=(12, 4))

df['energy\_kWh'].plot(title='Original Energy Consumption Data')

plt.show()

# 3. Aggregation (Hourly to Daily)

daily\_data = df.resample('D').mean()

plt.figure(figsize=(12, 4))

daily\_data.plot(title='Daily Aggregated Energy Consumption')

plt.show()

# 4. Smoothing using Moving Average (7-day window)

daily\_data['smoothed'] = daily\_data['energy\_kWh'].rolling(window=7, center=True).mean()

plt.figure(figsize=(12, 4))

daily\_data[['energy\_kWh', 'smoothed']].plot(title='Smoothed Energy Consumption (7-day MA)')

plt.show()

# 5. Detrending (Eliminating Trend)

# Subtract the smoothed series to remove the trend

daily\_data['detrended'] = daily\_data['energy\_kWh'] - daily\_data['smoothed']

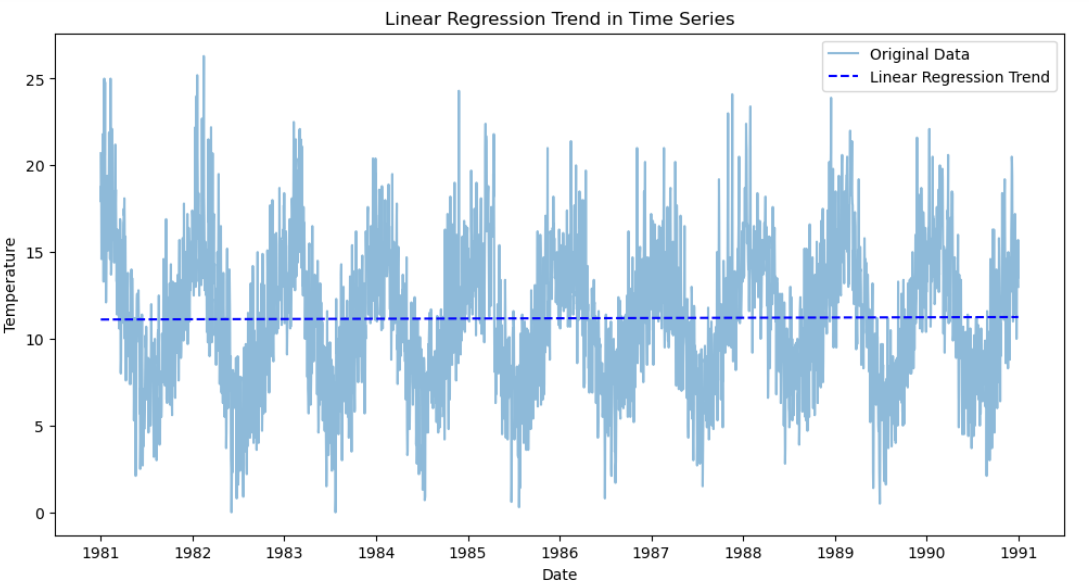
plt.figure(figsize=(12, 4))

daily\_data['detrended'].plot(title='Detrended Energy Consumption')

plt.axhline(0, color='gray', linestyle='--')

plt.show()

**Output:**



**Result:**

Thus, the program using the time series data implementation has been done successfully.