EX:No.7 221501060

07/04/25

**Implement program for decomposing time series data into trend and seasonality**

**Aim:**

**To Implement program for decomposing time series data into trend and seasonality**

**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

from statsmodels.tsa.seasonal import seasonal\_decompose

np.random.seed(42)

date\_range = pd.date\_range(start='2025-01-01', periods=24\*30, freq='H'

daily\_pattern = 20 + 10 \* np.sin(2 \* np.pi \* date\_range.hour / 24)

trend\_component = np.linspace(100, 130, len(date\_range))

noise = np.random.normal(scale=3, size=len(date\_range))

energy\_consumption = trend\_component + daily\_pattern + noise

df = pd.DataFrame({

'Timestamp': date\_range,

'Energy\_Consumption': energy\_consumption

})

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

decomposition = seasonal\_decompose(df['Energy\_Consumption'], model='additive', period=24)

plt.figure(figsize=(12, 10))

plt.subplot(411)

plt.plot(df['Energy\_Consumption'], label='Original', color='blue')

plt.title('Original Energy Consumption')

plt.legend()

plt.subplot(412)

plt.plot(decomposition.trend, label='Trend', color='orange')

plt.title('Trend Component')

plt.legend()

plt.subplot(413)

plt.plot(decomposition.seasonal, label='Seasonality', color='green')

plt.title('Seasonal Component (Daily Cycle)')

plt.legend()

plt.subplot(414)

plt.plot(decomposition.resid, label='Residuals', color='red')

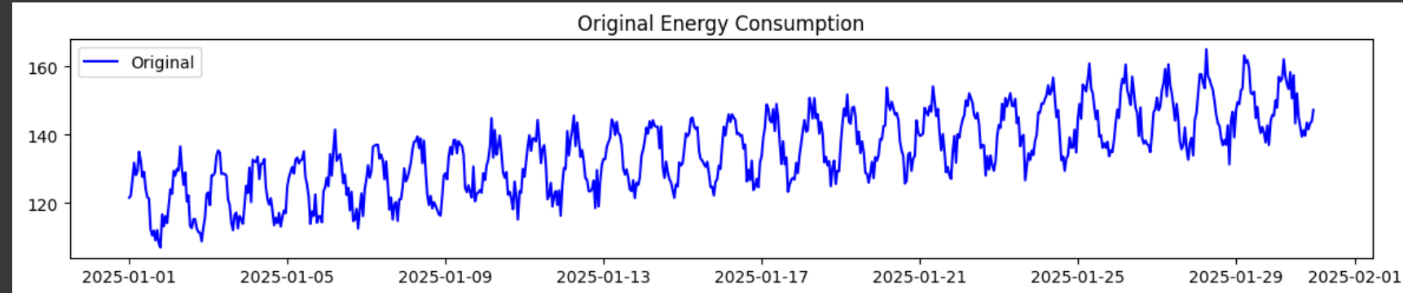
plt.title('Residual Component')

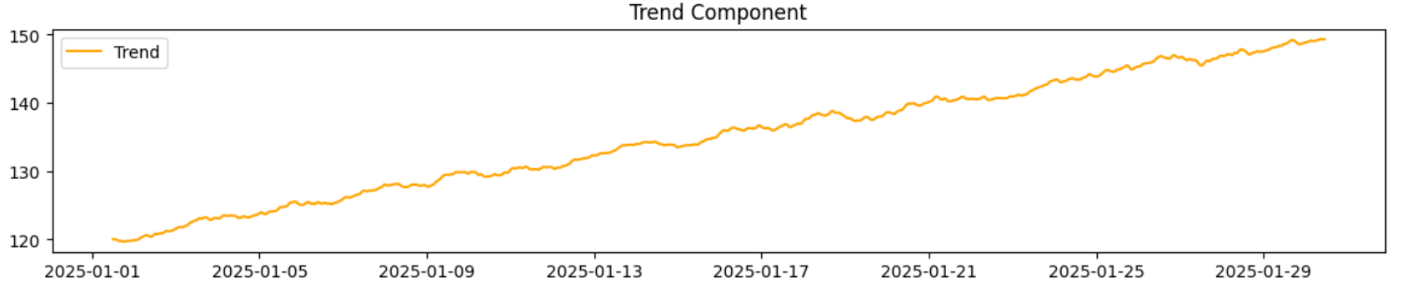
plt.legend()

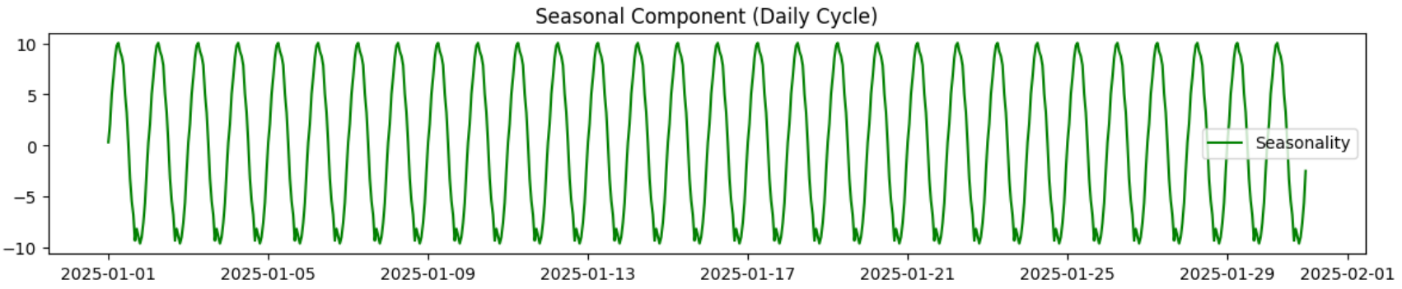
plt.tight\_layout()

plt.show()

**Output:**







**Result:**

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