EX:No.4 221501062

**Implement programs for estimating & eliminating trend in time series data- aggregation, smoothing.**

**Aim:**

Write a program to estimating & eliminating trend in time series data- aggregation, smoothing.

**Algorithm:**

Import Libraries

* Load required libraries: numpy, pandas, matplotlib, and statsmodels.

Load the Dataset

* Read the weather dataset (weather\_data.csv) and set the 'Date' column as the index.

Select a Time Series Column

* Choose a column for analysis (e.g., Temperature).

Apply Aggregation (Rolling Mean)

* Compute a 7-day rolling mean to smooth fluctuations in the data.

Apply Smoothing Techniques:

* Moving Average Smoothing: Compute a centered 7-day moving average.
* Exponential Smoothing: Apply Holt-Winters Exponential Smoothing to capture trends.

Extract Trend using Seasonal Decomposition

* Decompose the time series into trend, seasonal, and residual components using additive decomposition.

Visualize the Data

* Plot the original time series.
* Overlay the rolling mean, moving average, and exponential smoothing for comparison.
* Plot the trend component extracted from decomposition.

Output the Results

* Display the smoothed series and extracted trend.
* If needed, detrend the series by subtracting the estimated trend.

**Code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.seasonal import seasonal\_decompose

from statsmodels.tsa.holtwinters import ExponentialSmoothing

# Load the dataset

df = pd.read\_csv(r"C:\Users\kishore\Downloads\rain\_precipitaion.csv", parse\_dates=['date'], index\_col='date')

# Selecting the time series column

time\_series = df['Tdew']

# 1. Aggregation - Rolling Mean

rolling\_mean = time\_series.rolling(window=7).mean()

# 2. Smoothing - Moving Average

moving\_avg = time\_series.rolling(window=7, center=True).mean()

# 3. Exponential Smoothing

exp\_smooth = ExponentialSmoothing(time\_series, trend='add', seasonal=None, damped\_trend=False).fit().fittedvalues

# 4. Seasonal Decomposition (Trend Extraction)

decomposition = seasonal\_decompose(time\_series, model='additive', period=30)

trend = decomposition.trend

# Plot the results

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

plt.plot(time\_series, label='Original', color='blue', alpha=0.5)

plt.plot(rolling\_mean, label='Rolling Mean (7-day)', color='red')

plt.plot(moving\_avg, label='Moving Average (7-day)', color='green')

plt.plot(exp\_smooth, label='Exponential Smoothing', color='purple')

plt.title('Trend Estimation & Smoothing')

plt.legend()

plt.show()

# Plot decomposed trend

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

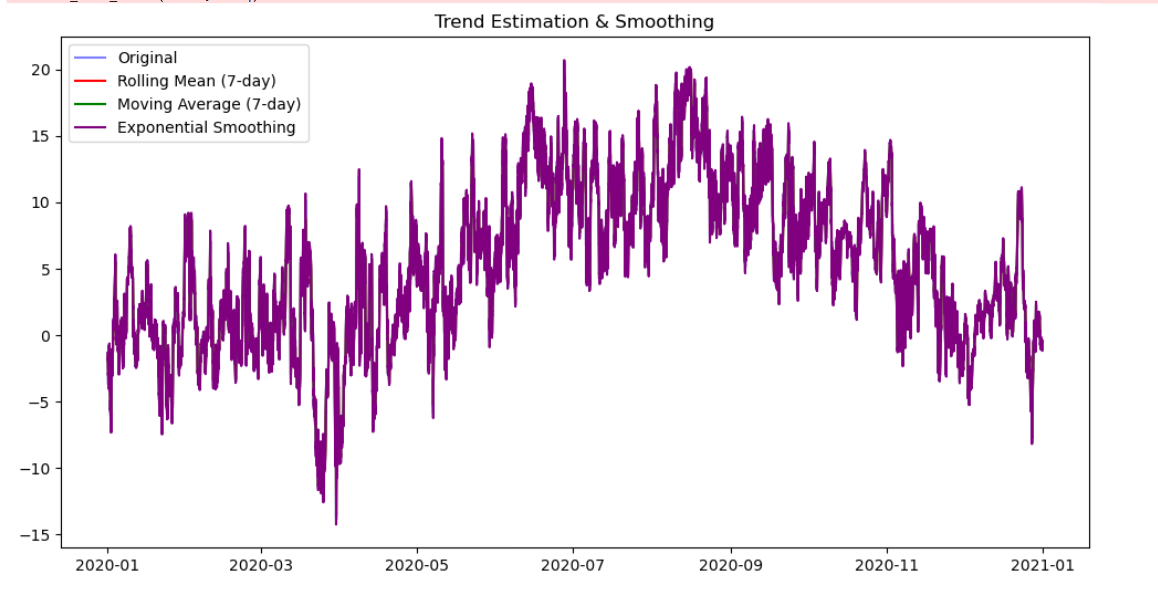
plt.plot(trend, label='Extracted Trend', color='brown')

plt.title('Trend Component from Seasonal Decomposition')

plt.legend()

plt.show()

**Output:**

****

**A graph showing a trend diagram

AI-generated content may be incorrect.**

**Result:**

Thus, the program to estimating & eliminating trend in time series data- aggregation, smoothing was done.