**Time Series Trend Estimation and Elimination**

**Aim:** To estimate and eliminate trends in time series data using aggregation and smoothing techniques.

**Algorithm:**

1. Import necessary libraries.
2. Load and preprocess the time series data from Microsoft\_Stock.csv.
3. Apply aggregation techniques (Weekly and Monthly mean).
4. Apply smoothing techniques (Moving Average and Savitzky-Golay filter).
5. Visualize original vs transformed series.

**Prerequisites:** Ensure the following Python libraries are installed:

pip install pandas matplotlib scipy

**Code Explanation:**

**Importing Required Libraries**

****import pandas as pd

import matplotlib.pyplot as plt

from scipy.signal import savgol\_filter

These libraries are used for data manipulation, visualization, and smoothing.

**Reading and Preprocessing Data**

****df = pd.read\_csv("Microsoft\_Stock.csv")

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

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

df.sort\_index(inplace=True)

ts = df['Close']

This code loads the stock data, sets the Date as the index, and uses the Close prices as the time series.

**Aggregation**

****weekly\_avg = ts.resample('W').mean()

monthly\_avg = ts.resample('M').mean()

*   
  Weekly and Monthly resampling are performed to observe longer-term trends.

**Smoothing**

****moving\_avg = ts.rolling(window=10, min\_periods=1).mean()

savgol\_smooth = savgol\_filter(ts, window\_length=11, polyorder=2)

*   
  A simple moving average smooths fluctuations over a 10-day window.
* Savitzky-Golay filter fits successive sub-sets of adjacent data points with a polynomial.

**Visualization**

****plt.figure(figsize=(14, 10))

# Smoothing

plt.subplot(3, 1, 1)

plt.plot(ts, label='Original', alpha=0.6)

plt.plot(moving\_avg, label='Moving Average (10)', linestyle='--')

plt.plot(ts.index, savgol\_smooth, label='Savitzky-Golay', linestyle='-.')

plt.title("Smoothing Techniques")

plt.legend()

# Weekly Aggregation

plt.subplot(3, 1, 2)

plt.plot(ts, label='Original', alpha=0.6)

plt.plot(weekly\_avg, label='Weekly Average', color='orange')

plt.title("Aggregation - Weekly")

plt.legend()

# Monthly Aggregation

plt.subplot(3, 1, 3)

plt.plot(ts, label='Original', alpha=0.6)

plt.plot(monthly\_avg, label='Monthly Average', color='green')

plt.title("Aggregation - Monthly")

plt.legend()

plt.tight\_layout()

plt.show()

The plots compare the original time series against smoothed and aggregated versions.

**Result:** This program demonstrates trend estimation and elimination in time series data. Aggregation provides higher-level overviews (weekly/monthly), while smoothing helps reveal underlying patterns by reducing short-term noise.

**OUTPUT:**

