EXP:6

DATE:03.04.2025

**IMPLEMENT PROGRAM TO APPLY MOVING AVERAGE SMOOTHING FOR DATA PREPARATION AND TIME SERIES FORECASTING**

**AIM:** Implement program to apply moving average smoothing for data preparation and time series forecasting.

**PROCEDURE:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

def simple\_moving\_average(data, window\_size):

return data.rolling(window=window\_size).mean()

def weighted\_moving\_average(data, window\_size):

weights = np.arange(1, window\_size + 1)

return data.rolling(window=window\_size).apply(lambda x: np.dot(x, weights) / weights.sum(), raw=True)

def exponential\_moving\_average(data, span):

return data.ewm(span=span, adjust=False).mean()

# Example usage

def main():

# Generate sample daily website visitors data

np.random.seed(42)

days = pd.date\_range(start='2024-01-01', periods=100, freq='D')

visitors = np.random.randint(100, 500, size=100) # Simulated visitor counts

visitor\_series = pd.Series(visitors, index=days)

window\_size = 7

span = 7

sma = simple\_moving\_average(visitor\_series, window\_size)

wma = weighted\_moving\_average(visitor\_series, window\_size)

ema = exponential\_moving\_average(visitor\_series, span)

# Plot the results

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

plt.plot(visitor\_series, label='Daily Visitors', color='gray', alpha=0.5)

plt.plot(sma, label=f'SMA (window={window\_size})', linestyle='dashed')

plt.plot(wma, label=f'WMA (window={window\_size})', linestyle='dotted')

plt.plot(ema, label=f'EMA (span={span})', linestyle='solid')

plt.legend()

plt.title('Moving Average Smoothing for Daily Website Visitors')

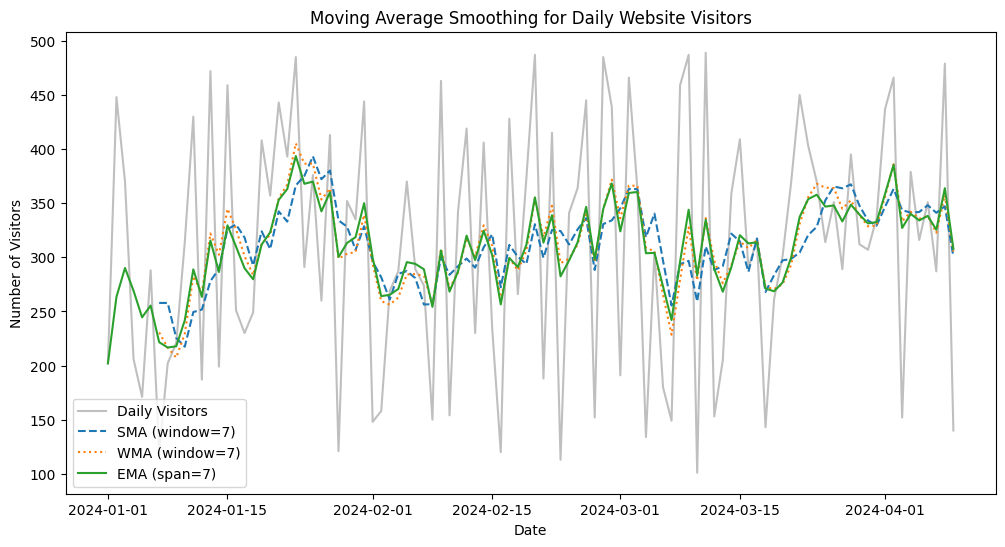
plt.xlabel('Date')

plt.ylabel('Number of Visitors')

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

main()

**OUTPUT:**

**RESULT:**

Thus the expected output has been excecuted successfully.