|  |  |
| --- | --- |
| **EX.N0 : 6** | **Implement program to apply moving average smoothing for data preparation and time series forecasting** |
| **DATE : 20/03/2025** |

**AIM:**

# Implement program to apply moving average smoothing for data preparation and time series forecasting

# ALGORITHM:

# Step 1: Import Libraries Step 2: Load the Dataset Step 3: Preprocess the Data Step 4: Plot Original Time Series Data Step 5: Apply Moving Average Smoothing Step 6: Forecast Future Values Step 7: Plot Forecasted Data

# PROGRAM:

# Step 1: Import Libraries

import pandas as pd

import matplotlib.pyplot as plt

# Step 2: Load the Dataset

file\_path = '/mnt/data/climate\_change\_data.csv'

df = pd.read\_csv(file\_path)

# Step 3: Preprocess the Data

# Assume the first column is date and the second is the value column

df.columns = [col.strip() for col in df.columns]

date\_col = df.columns[0]

df[date\_col] = pd.to\_datetime(df[date\_col], errors='coerce')

df.set\_index(date\_col, inplace=True)

# Drop non-numeric columns

df\_numeric = df.select\_dtypes(include='number')

# Ensure we have numeric data

if df\_numeric.empty:

raise ValueError("No numeric columns found!")

# Select the first numeric column for analysis

value\_col = df\_numeric.columns[0]

data = df\_numeric[[value\_col]].copy()

# Step 4: Plot Original Data

plt.figure(figsize=(12, 5))

plt.plot(data, label='Original Data')

plt.title('Original Time Series')

plt.grid(True)

plt.legend()

plt.show()

# Step 5: Apply Moving Average Smoothing

window\_size = 12 # Can be changed based on data

data['Smoothed'] = data[value\_col].rolling(window=window\_size).mean()

# Plot Smoothed Data

plt.figure(figsize=(12, 5))

plt.plot(data[value\_col], label='Original')

plt.plot(data['Smoothed'], label=f'{window\_size}-Point Moving Average', color='orange')

plt.title('Moving Average Smoothing')

plt.grid(True)

plt.legend()

plt.show()

# Step 6: Simple Forecasting using Last Smoothed Value

forecast\_periods = 12

last\_date = data.index[-1]

forecast\_index = pd.date\_range(start=last\_date, periods=forecast\_periods + 1, freq='M')[1:]

last\_smoothed\_value = data['Smoothed'].dropna().iloc[-1]

forecast\_values = [last\_smoothed\_value] \* forecast\_periods

forecast\_df = pd.DataFrame({value\_col: forecast\_values}, index=forecast\_index)

# Plot Forecast

plt.figure(figsize=(12, 5))

plt.plot(data[value\_col], label='Original')

plt.plot(data['Smoothed'], label='Smoothed', color='orange')

plt.plot(forecast\_df, label='Forecast (Flat)', linestyle='--', color='green')

plt.title('Forecast Using Moving Average')

plt.grid(True)

plt.legend()

plt.show()

**OUTPUT:**

A blue line graph on a white background

AI-generated content may be incorrect.

A graph showing a blue and orange line

AI-generated content may be incorrect.

A graph showing a wave of blue and orange

AI-generated content may be incorrect.

# RESULT:

Thus, the program for Implement programs for a time series data is executed successfully.