**EX:No.6 221501028**

**01/04/25**

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

**AIM:**

To develop a Python program to apply moving average smoothing for data preparation and time series forecasting using the "Autism Screening for Toddlers" dataset, simulating a time series, and visualizing the results.

.

**ALGORITHM:**

1. Load the dataset and simulate a time series by assigning synthetic timestamps.
2. Calculate the total score from screening variables (A1 to A10) as the time series variable.
3. Apply moving average smoothing to the time series data to reduce noise.
4. Use the smoothed data to forecast future values with a simple moving average model.
5. Visualize the original, smoothed, and forecasted data for comparison.

**PROCESS:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

**# Load the Autism Screening for Toddlers dataset**

df = pd.read\_csv('autism.csv') # Replace with actual file path

**# Simulate a time series by adding a timestamp (since the dataset lacks explicit time data)**

date\_rng = pd.date\_range(start='2018-01-01', periods=len(df), freq='D') # Daily frequency for demo

df['timestamp'] = date\_rng

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

**# Calculate Total\_Score as the sum of A1 to A10 scores**

df['Total\_Score'] = df[['A1\_Score', 'A2\_Score', 'A3\_Score', 'A4\_Score', 'A5\_Score',

'A6\_Score', 'A7\_Score', 'A8\_Score', 'A9\_Score', 'A10\_Score']].sum(axis=1)

**# Clean data (handle any potential missing values)**

df['Total\_Score'] = df['Total\_Score'].fillna(method='ffill').fillna(method='bfill').fillna(0)

**# Apply moving average smoothing with a 7-day window**

df['smoothed'] = df['Total\_Score'].rolling(window=7, center=True).mean()

**# Prepare data for forecasting (simple moving average forecast for the next 7 days)**

forecast\_periods = 7

last\_smoothed\_value = df['smoothed'].iloc[-1]

forecast = np.full(forecast\_periods, last\_smoothed\_value) # Simple forecast using last smoothed value

forecast\_index = pd.date\_range(start=df.index[-1] + pd.Timedelta(days=1), periods=forecast\_periods, freq='D')

**# Combine original, smoothed, and forecast data for plotting**

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

plt.plot(df['Total\_Score'], label='Original Data', color='blue')

plt.plot(df['smoothed'], label='Smoothed Data (7-day MA)', color='red')

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

plt.title('Moving Average Smoothing and Forecasting of Total Autism Screening Scores')

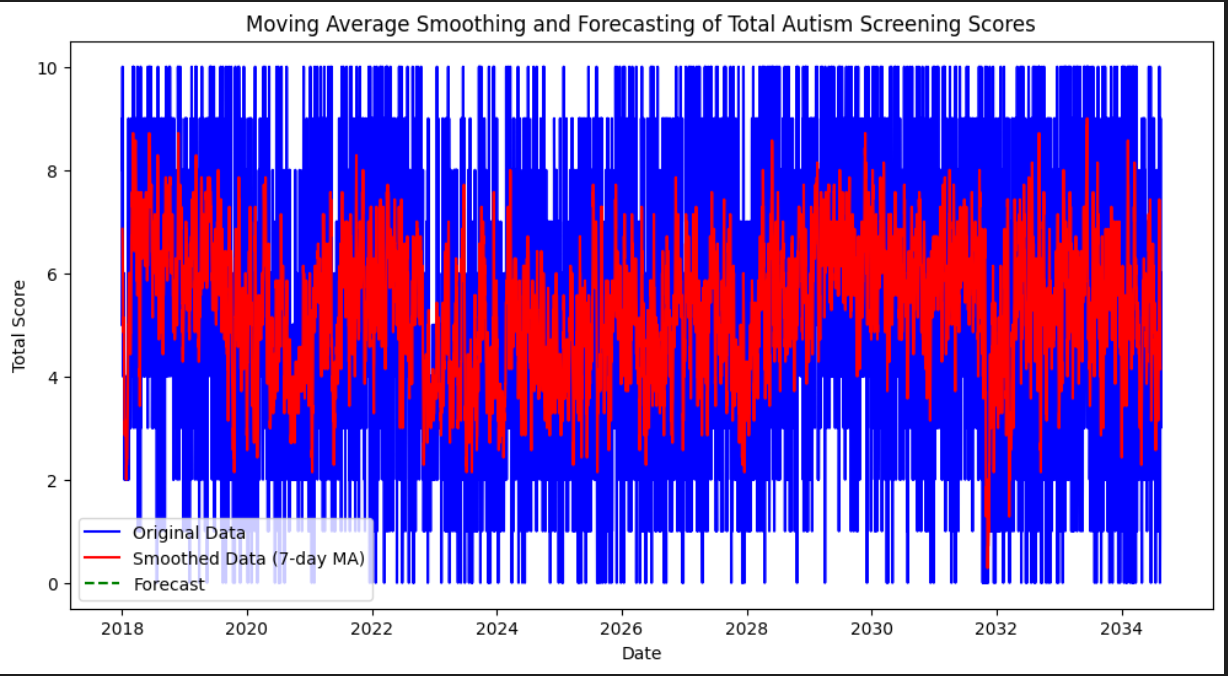
plt.xlabel('Date')

plt.ylabel('Total Score')

plt.legend()

plt.show()

**OUTPUT:**

****

**RESULT:**

The program successfully applies moving average smoothing to the simulated time series from the "Autism Screening for Toddlers" dataset, prepares the data, and performs a simple forecast, with visualizations comparing the original, smoothed, and forecasted values.