

6. 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:

◇ Step 1: Import Required Libraries

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

This imports the necessary libraries for data manipulation (pandas, numpy) and visualization (matplotlib).

◇ Step 2: Load and Inspect Your Data

```
# Load your CSV file
```

```
df = pd.read_csv("cleaned_weather.csv")
```

```
# Strip any extra whitespace from column names
```

```
df.columns = df.columns.str.strip()
```

```
# Check the first few rows
```

```
print(df.head())
```

```
# Check column names
```

```
print(df.columns)
```

This step helps confirm the structure of your dataset and identify the exact column names ('date', 'Tpot' etc.).

◇ Step 3: Convert date Column to DateTime Format & Set Index

```
# Convert 'date' column to datetime
```

```
df['date'] = pd.to_datetime(df['date'])
```

```
# Set 'date' as the index
```

```
df.set_index('date', inplace=True)
```

This ensures that we can perform time-based operations like rolling averages.

◇ Step 4: Verify 'Tpot' Column Exists

```
if "Tpot" not in df.columns:
```

```
    raise ValueError("The dataset must contain a 'Tpot' column.")
```

This is a safety check to avoid errors if the column is missing.

◇ Step 5: Apply Moving Average Smoothing

```
# Define the moving average function
```

```
def moving_average(series, window):
```

```
    return series.rolling(window=window).mean()
```

```
# Apply the smoothing
```

```
window_size = 7 # 7-day rolling average
```

```
df['Smoothed'] = moving_average(df['Tpot'], window_size)
```

This adds a new column Smoothed that contains the 7-day moving average of Tpot.

◇ Step 6: Forecast Next 10 Days (Simple Average Forecast)

```
# Forecasting next 10 days
```

```
future_steps = 10
```

```
future_dates = pd.date_range(start=df.index[-1] + pd.Timedelta(days=1),  
periods=future_steps, freq="D")
```

```
# Use the average of the last 7 smoothed values as forecast
```

```
future_temperatures = [df['Smoothed'].iloc[-window_size:].mean()] *  
future_steps
```

This creates 10 future dates and predicts the same value (average of last 7 smoothed days).

◇ Step 7: Plot the Results

```
# Plot original, smoothed, and forecasted Tpot
```

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

```
plt.plot(df.index, df["Tpot"], label="Original Tpot", linestyle="dashed",  
alpha=0.6)
```

```
plt.plot(df.index, df["Smoothed"], label="Smoothed Tpot", linewidth=2)
```

```
plt.plot(future_dates, future_temperatures, label="Forecasted Tpot",  
linestyle="dotted", marker="o")
```

```
plt.xlabel("Date")
```

```
plt.ylabel("Tpot")
```

```
plt.legend()
```

```
plt.title("Moving Average Smoothing & Forecasting for Tpot")
```

```
plt.grid(True)
```

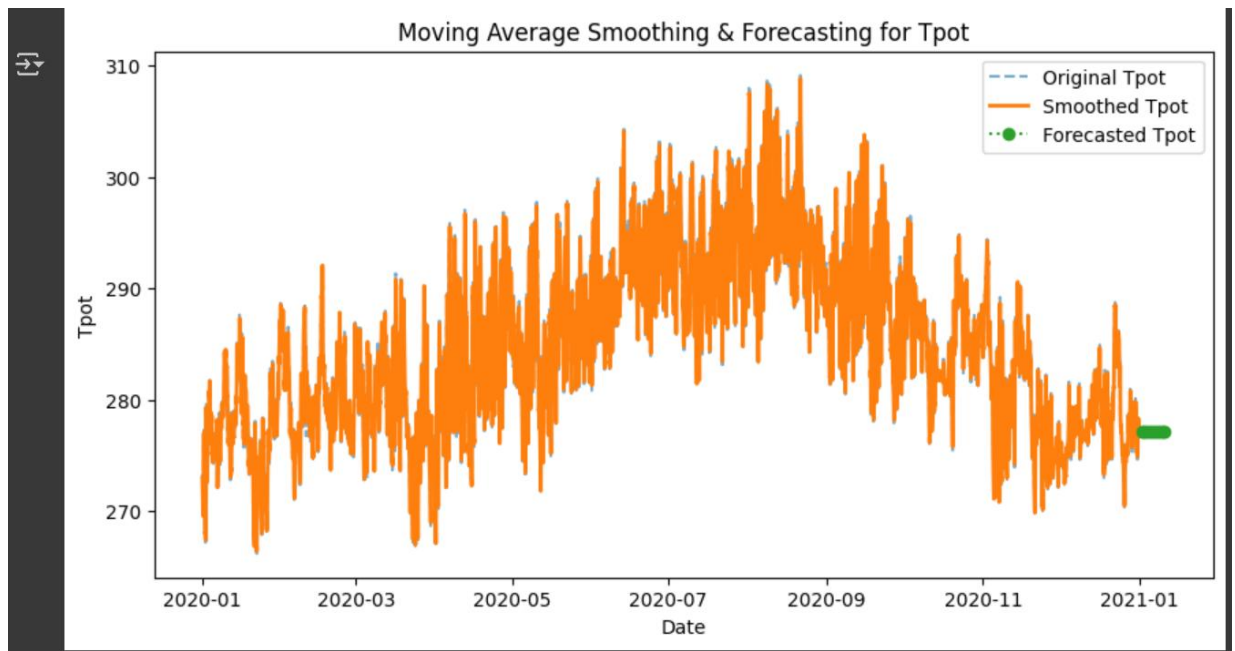
```
plt.tight_layout()
```

```
plt.show()
```

This gives you a full visual of:

- **Raw data (Tpot)**
- **Smoothed data (rolling average)**
- **Forecast for 10 days ahead**

OUTPUT :



RESULT:

The program to implement to apply moving average smoothing on the autism screening dataset has been implemented successfully.