**Implement program to apply moving average**

**smoothing for data preparation and time series forecasting**

**AIM:**

To implement program to apply moving average smoothing for data preparation andforecasting time series data using jupyter notebook.

**ALGORITHM:**

1. Load the taxi dataset then clean and load the values

2. Smooth the time series using a moving average and forecast future values by extending the last smoothed value.

3. Plot the graph and visualize the values

**CODE:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Load and prepare dataset

df = pd.read\_csv("taxi dataset.csv")

df["datetime"] = pd.to\_datetime(df[["year", "month", "day"]]) + pd.to\_timedelta(df["hour\_of\_day"], unit='h')

df.set\_index("datetime", inplace=True)

# Daily average of total\_amount

daily\_avg = df["total\_amount"].resample("D").mean().dropna()

# Apply moving average smoothing (7-day)

smoothed = daily\_avg.rolling(window=7, center=False).mean().dropna()

# Forecast next 7 days using last smoothed value

future\_index = pd.date\_range(start=smoothed.index[-1] + pd.Timedelta(days=1), periods=7)

forecast\_values = np.full(shape=7, fill\_value=smoothed.iloc[-1]) # naive forecast

forecast\_series = pd.Series(forecast\_values, index=future\_index)

# Plot the result

plt.figure(figsize=(14, 6))

plt.plot(daily\_avg, label="Original Daily Avg", alpha=0.4)

plt.plot(smoothed, label="Smoothed (7-day MA)", color='blue')

plt.plot(forecast\_series, label="Forecast (Next 7 Days)", color='red', linestyle='--')

plt.title("Moving Average Smoothing and Forecasting")

plt.xlabel("Date")

plt.ylabel("Total Amount ($)")

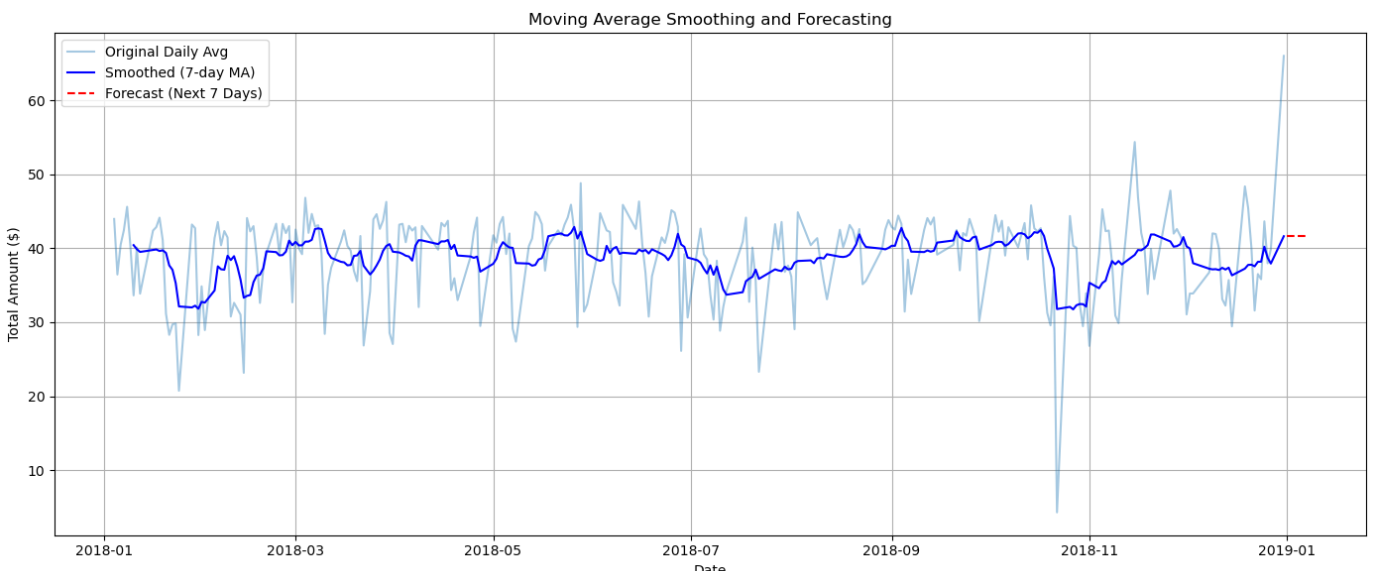
plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()

**OUTPUT:**

****

**RESULT:**

The program to execute applying moving average smoothing for data preparation andforecasting time series data completed successfully and the output is verified.