# 6**.** 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 & CODE:**

*1.Import all necessary lib.*

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.holtwinters import ExponentialSmoothing

*2.Load and preprocess the data*

file\_path = '/mnt/data/dataset.csv'

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

data['Date'] = pd.to\_datetime(data['Date'])

data.set\_index('Date', inplace=True)

*3.Apply Moving Average Smoothing*

window\_size = 7 # You can adjust this for longer/shorter smoothing

data['MA\_Smooth'] = data['Close'].rolling(window=window\_size).mean()

*4.Time Series Forecasting using Holt-Winters Exponential Smoothing*

# Split data into training and testing sets

train\_size = int(len(data) \* 0.8)

train, test = data.iloc[:train\_size], data.iloc[train\_size:]

*5. Fit the model on training data*

model = ExponentialSmoothing(train['Close'], trend='add', seasonal=None, seasonal\_periods=None)

fitted\_model = model.fit()

*5. Forecasting and plotting*

forecast\_steps = len(test)

forecast = fitted\_model.forecast(steps=forecast\_steps)

plt.figure(figsize=(14, 8))

*6.Original & Smoothed Data , displaying forecasting values*

plt.plot(data.index, data['Close'], label='Original Close Price', color='blue')

plt.plot(data.index, data['MA\_Smooth'], label=f'{window\_size}-Day Moving Average', color='orange')

# Forecasting

plt.plot(test.index, forecast, label='Forecast', color='green', linestyle='dashed')

plt.title('Moving Average Smoothing & Forecasting')

plt.xlabel('Date')

plt.ylabel('Close Price')

plt.legend()

plt.grid(True)

plt.show()

forecast\_df = pd.DataFrame({'Date': test.index, 'Forecast': forecast.values})

print("\nForecasted Values:")

print(forecast\_df)