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

**EX:No.5**

**DATE:29/03/25**

**AIM:**

To implement a program that applies moving average smoothing for time series data preparation and forecasting.

**ALGORITHM:**

 Import the dataset containing time series values (e.g., stock prices).

 Convert the date column to datetime format and set it as the index.

 Handle missing values using methods like forward fill.

 Choose a window size (e.g., 7, 30, or 50 days) for the moving average.

 Apply the moving average using a rolling window on the 'Close' price column.

 Plot the original data along with the smoothed data for comparison.

 Use the smoothed values for trend analysis or simple forecasting.

**CODE:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Load dataset

file\_path = '/content/AAPL.csv'  # Replace with your file path

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

# Convert 'Date' to datetime and set as index

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

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

# Use the 'Close' price

ts = data['Close']

# Apply Moving Average Smoothing (e.g., 7-day window)

window\_size = 7

smoothed\_ts = ts.rolling(window=window\_size).mean()

# Plot original vs smoothed series

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

plt.plot(ts, label='Original', color='blue', alpha=0.5)

plt.plot(smoothed\_ts, label=f'{window\_size}-Day Moving Average', color='red')

plt.title('AAPL Closing Price with Moving Average Smoothing')

plt.xlabel('Date')

plt.ylabel('Price')

plt.legend()

plt.show()

# Simple Forecasting: Use the last smoothed value to forecast next n days

forecast\_days = 5

last\_smoothed\_value = smoothed\_ts.dropna().iloc[-1]

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

forecast\_dates = pd.date\_range(start=ts.index[-1] + pd.Timedelta(days=1), periods=forecast\_days)

# Create forecast DataFrame

forecast\_df = pd.DataFrame({'Forecast': forecast\_values}, index=forecast\_dates)

# Plot forecast

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

plt.plot(ts, label='Original', color='blue', alpha=0.5)

plt.plot(forecast\_df.index, forecast\_df['Forecast'], label='Forecast', color='green', linestyle='dashed')

plt.title('AAPL Forecast using Moving Average Smoothing')

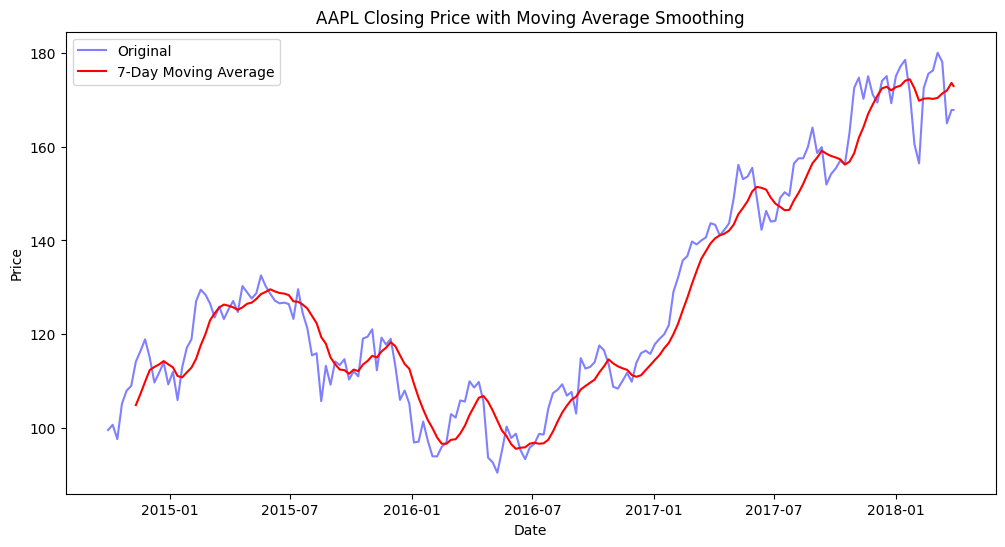
plt.xlabel('Date')

plt.ylabel('Price')

plt.legend()

plt.show()

**OUTPUT:**



**RESULT:**

Thus the program has been completed and verified successfully.