**Implement program to apply moving average smoothing for data preparation and time series forecasting run in python code**

**EX.No:7**

**DATE:**

**AIM:**

To apply Moving Average Smoothing for trend analysis and forecasting using Python.

**ALGORITHM:**

1. Import pandas and matplotlib libraries.
2. Load the time series dataset and convert the date column to datetime format.
3. Set the date column as index and resample the data.
4. Apply Moving Average using rolling().mean() function.
5. Plot original data and smoothed data for comparison.

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.seasonal import seasonal\_decompose

# Step 1: Load the dataset

file\_path = 'Dataset.csv'

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

# Step 2: Fix the month column by adding a year

data['month'] = data['month'] + '-2020'  # Adding dummy year

# Step 3: Convert to datetime

data['month'] = pd.to\_datetime(data['month'], format='%d-%b-%Y')

# Step 4: Set date as index

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

# Step 5: Handle missing values

data['price'] = data['price'].interpolate()  # You can also use .fillna(method='ffill')

# Step 6: Plot raw data

plt.figure(figsize=(10, 4))

plt.plot(data['price'])

plt.title('Raw Time Series Data')

plt.xlabel('Date')

plt.ylabel('Price')

plt.grid(True)

plt.show()

# Step 7: Decompose the time series

result = seasonal\_decompose(data['price'], model='additive', period=12)

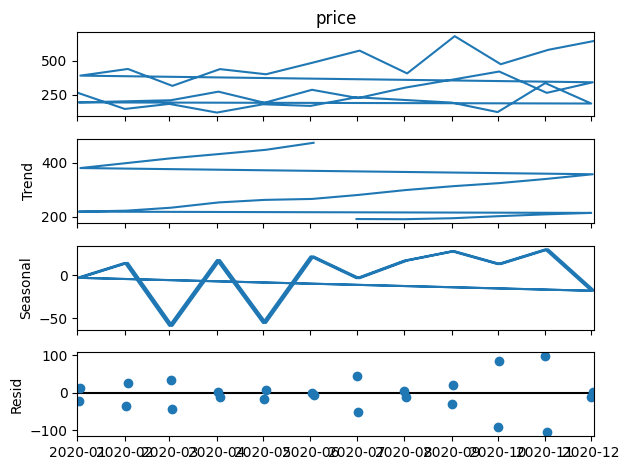
# Step 8: Plot decomposition

result.plot()

plt.tight\_layout()

plt.show()

**OUTPUT:**



**RESULT:**

Thus the program has been completed and verified successfully.