EXERCISE NO 06

MOVING AVERAGE SMOOTHING FOR DATA PREPARATION AND TIME SERIES FORECASTING

AIM:

To prepare data by moving average smoothing and time series forecasting.

PROCEDURE:

1. Import the necessary libraries.

   ```
   import numpy as np
   import pandas as pd
   import matplotlib.pyplot as plt
   from sklearn.linear_model import LinearRegression
   ```

2. Load the time series data.

   ```
   df = pd.read_csv(“.../amazon.csv”, encoding = “latin1”)
   ```

3. Pre-process the data.

   ```
   month_map = {
     'Janeiro': 'January', 'Fevereiro': 'February', 'Março': 'March',
     'Abril': 'April', 'Maio': 'May', 'Junho': 'June',
     'Julho': 'July', 'Agosto': 'August', 'Setembro': 'September',
     'Outubro': 'October', 'Novembro': 'November', 'Dezembro': 'December'
   }
   ```

```
df['month'] = df['month'].map(month_map)

df['date'] = pd.to_datetime(df['month'] + ' ' + df['year'].astype(str),
format='%B %Y')

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

4. Aggregate the preprocessed data.

```
df_monthly = df.resample('ME')['number'].sum()

df_yearly = df.resample('YE')['number'].sum()
```

5. Apply moving average smoothing.

```
window_size = 5

df_monthly_smooth = df_monthly.rolling(window=window_size,
center=True).mean()
```

6. Forecast the future values using the moving average.

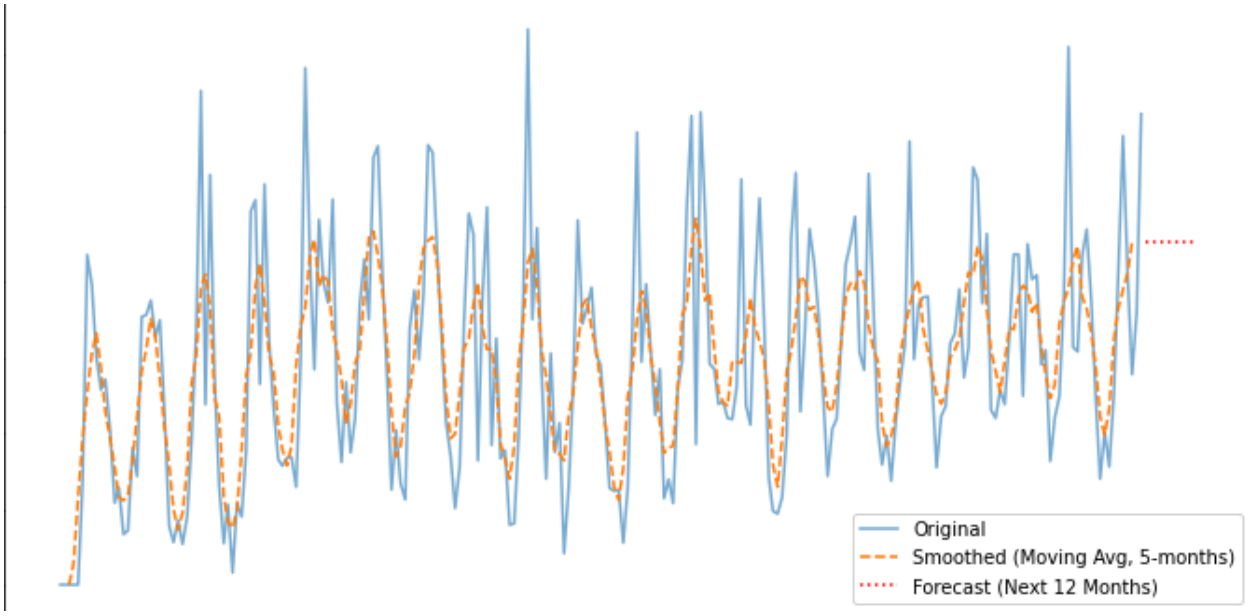
```
forecast_period = 12

last_values = df_monthly.tail(window_size)

moving_avg_forecast = np.mean(last_values)

forecast_dates = pd.date_range(start=df_monthly.index[-1],
periods=forecast_period+1, freq='M')[1:]

forecast_values = np.full(forecast_period, moving_avg_forecast)
```

7. Visualise the forecasting.

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

plt.plot(df_monthly.index, df_monthly, label='Original', alpha=0.6)
```

```
plt.plot(df_monthly.index, df_monthly_smooth, label=f'Smoothed (Moving Avg, {window_size}-months)', linestyle='dashed')

plt.plot(forecast_dates, forecast_values, label=f'Forecast (Next {forecast_period} Months)', linestyle='dotted', color='red')

plt.legend()

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

plt.show()
```

OUTPUT:



RESULT:

Thus the program has been successfully implemented and verified.