EX:No.6 221501057

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

Aim:

Write a program to apply moving average smoothing for data preparation and time series forecasting.

Algorithm:

- 1. Import Required Libraries
- Load pandas, numpy, matplotlib.pyplot, and sklearn.metrics.
- 2.Load Dataset
- Read the CSV file and parse 'Date' as a datetime index.
- 3. Select Target Column
- Choose a time series column for smoothing (e.g., Price).
- 4. Apply Moving Average Smoothing
- Calculate the 7-day rolling average of the target column.
- 5. Forecast Future Values
- Use the last value of the moving average as the forecast for the next 10 days (simple static forecast).
- Create future dates and build a forecast

DataFrame. 6. Visualization

- Plot:
- Original time series.
- Smoothed series.
- Forecast for the next 10 days.
- Add labels, title, and legend for clarity.
- 7. Evaluate the Smoothing (Optional)
- Compute RMSE between original and smoothed values (excluding initial NaN values).

Code:

import numpy as np

import pandas as pd

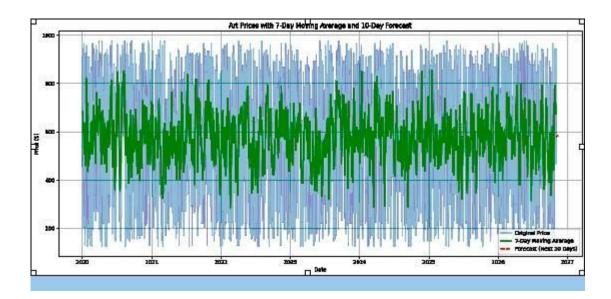
import matplotlib.pyplot as plt

from sklearn.metrics import mean_squared_error

uploaded = files.upload()

```
df = pd.read_csv("artmarket_with_dates.csv")
df['Date'] = pd.to_datetime(df['Date'])
df = df.sort_values('Date')
df['Moving Average'] = df['Price ($)'].rolling(window=7, center=True).mean()
ts = df.set_index('Date')['Price ($)']
model = ExponentialSmoothing(ts, trend="add", seasonal=None)
fitted_model = model.fit()
last_date = df['Date'].max()
future_dates = [last_date + timedelta(days=i) for i in range(1, 11)]
forecast = fitted model.forecast(10)
forecast_df = pd.DataFrame({'Date': future_dates, 'Forecast': forecast.values})
plt.figure(figsize=(15, 7))
plt.plot(df['Date'], df['Price ($)'], label='Original Price', alpha=0.5)
plt.plot(df['Date'], df['Moving Average'], label='7-Day Moving Average', linewidth=2,
color='green')
plt.plot(forecast df['Date'], forecast df['Forecast'], label='Forecast (Next 10
Days)', linestyle='--', color='red', linewidth=2)
plt.title('Art Prices with 7-Day Moving Average and 10-Day Forecast')
plt.xlabel('Date')
plt.ylabel('Price
($)') plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Output:



Result:

Thus, the program to apply moving average smoothing for data preparation and time series forecasting was done.