

Moving average smoothing for data preparation and time series forecasting

Aim

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

Procedure

1. Import necessary libraries such as `pandas` for data manipulation and `matplotlib` for visualization.
2. Load the supermarket sales dataset using `read_csv()`.
3. Convert the Date column to datetime format and sort the data in chronological order.
4. Set the Date as the index and aggregate sales data on a daily basis using `resample()`.
5. Apply moving average smoothing using `rolling(window=n).mean()` to reduce noise and highlight trends.
6. Plot original vs smoothed data to visualize the effect of smoothing.
7. Forecast future values using a simple method (e.g., extend the last moving average value) and plot the forecast.

Code

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Step 1: Load the dataset
```

```
df = pd.read_csv(r"C:\Users\Poong\Downloads\supermarket_sales - Sheet1.csv")
```

```
# Step 2: Convert 'Date' column to datetime
```

```
df['Date'] = pd.to_datetime(df['Date'])
```

```
# Step 3: Sort by date and set as index
```

```
df = df.sort_values('Date')
```

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

```
# Step 4: Aggregate daily sales
```

```
daily_sales = df['Total'].resample('D').sum()
```

```
# Step 5: Apply moving average smoothing (7-day window)
```

```
smoothed_sales = daily_sales.rolling(window=7).mean()
```

```
# Step 6: Plot original vs smoothed
```

```
plt.figure(figsize=(14, 5))
```

```
plt.plot(daily_sales, label='Original Sales', alpha=0.5)
```

```
plt.plot(smoothed_sales, label='7-Day Moving Average', color='red')
```

```
plt.title('Original vs Smoothed Sales')
```

```
plt.xlabel('Date')
```

```
plt.ylabel('Sales')
```

```
plt.legend()
```

```
plt.show()
```

```
# Step 7: Forecasting using last known moving average value (naive approach)
```

```
# Let's predict next 7 days using the last rolling average
```

```
last_moving_avg = smoothed_sales[-1]
```

```
future_dates = pd.date_range(start=daily_sales.index[-1] + pd.Timedelta(days=1),  
periods=7)
```

```
forecast = pd.Series([last_moving_avg]*7, index=future_dates)
```

```
# Step 8: Plot forecast
```

```
plt.figure(figsize=(14, 5))
```

```
plt.plot(daily_sales, label='Original Sales')
```

```
plt.plot(forecast, label='Forecast (Naive)', linestyle='--', color='green')
```

```
plt.title('Time Series Forecast using Moving Average')
```

```
plt.xlabel('Date')
```

```
plt.ylabel('Sales')
```

```
plt.legend()
```

```
plt.show()
```

Output



Result:

The above program has been successfully executed.