EX:No.7

Implement program for decomposing time series data into trend and seasonality

Aim:

Write a program for decomposing time series data into trend and seasonality

Algorithm:

1.Import Required Libraries

- Use pandas for data handling.
- Use matplotlib.pyplot for plotting.
- Use seasonal_decompose from statsmodels.

2.Load the Dataset

- Read the CSV file (weather_data.csv) using pandas.read_csv.
- Convert the 'Date' column to datetime and set it as the index.

3.Select the Target Time Series Column

- Choose a column like 'Price' for decomposition.

4.Apply Seasonal Decomposition

- Use seasonal_decompose() with model type:
- 'additive' (if variations are roughly constant over time).
- Set period=365 for yearly seasonality (adjust based on data granularity).
- This will return:
- observed: the original data
- trend: the long-term movement
- seasonal: recurring pattern
- resid: random noise

5.Plot the Components

- Create subplots to display:
- Observed data
- Trend component
- Seasonal component
- Residual component

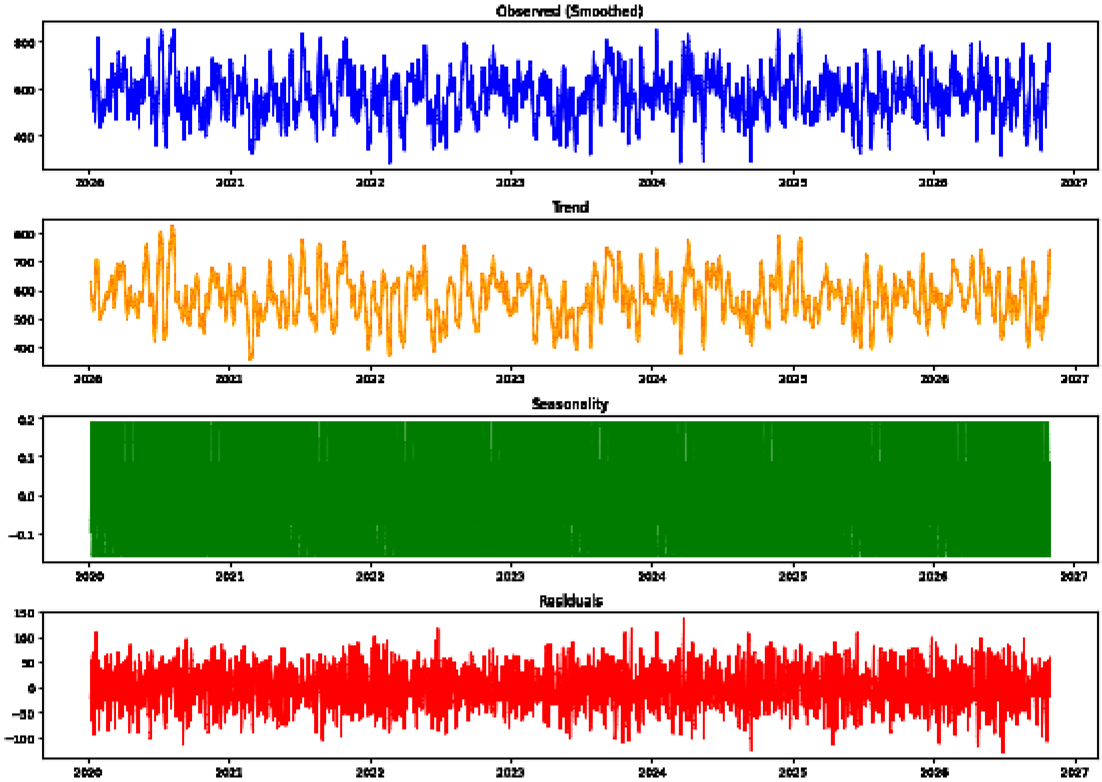
6.Display the Plot

- Use plt.tight_layout() and plt.show() for a clean, readable output.

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
from google.colab import files
# Load the CSV (adjust the filename if needed)
uploaded =  files.upload()
df = pd.read_csv('artmarket_with_dates.csv')

# Convert Date column to datetime
df['Date'] = pd.to_datetime(df['Date'])

# Set date as index and sort it
df.set_index('Date', inplace=True)
df = df.sort_index()

# Resample to daily and interpolate missing values
daily_price = df['Price ($)'].resample('D').mean().interpolate()

# Smooth the data using a 7-day moving average
smoothed_price = daily_price.rolling(window=7, center=True).mean().dropna()

# Decompose the time series
result = seasonal_decompose(smoothed_price, model='additive')

# STEP 3: Plot the components
plt.figure(figsize=(14, 10))

plt.subplot(411)
plt.plot(result.observed, color='blue')
plt.title('Observed (Smoothed)')

plt.subplot(412)
plt.plot(result.trend, color='orange')
plt.title('Trend')

plt.subplot(413)
plt.plot(result.seasonal, color='green')
plt.title('Seasonality')

plt.subplot(414)
plt.plot(result.resid, color='red')
plt.title('Residuals')

plt.tight_layout()
plt.show()
```

Output:



Result:

Thus, the program for decomposing time series data into trend and seasonality was done.