

**EX:No.7**

**DATE:4/04/25**

**Implement program for decomposing time series data into trend and seasonality**

### **AIM:**

Implement program for decomposing time series data into trend and seasonality

### **OBJECTIVE:**

To implement time series decomposition to identify and analyze the trend and seasonal patterns in air pollution data for informed decision-making and improved forecasting.

### **BACKGROUND:**

- Time series data often contains trend, seasonal, and irregular components that influence overall patterns.
- Understanding these components helps in better analyzing long-term behavior and periodic variations.
- Decomposition techniques like additive or multiplicative models break the series into interpretable parts.
- This is especially useful in environmental studies to uncover hidden trends in pollution data.

### **SCOPE OF THE PROGRAM:**

1. Apply time series decomposition to air pollution data (e.g., PM2.5) from 2012 to 2021.
2. Visualize and analyze trend and seasonality to observe pollution behavior over time.
3. Enhance forecasting accuracy by separating components before applying predictive models.
4. The approach can be extended to other pollutants like CO, NO<sub>2</sub>, SO<sub>2</sub>, and O<sub>3</sub>.

### **CODE:**

```
#Import libraries
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
from google.colab import files

#Upload your CSV file
uploaded = files.upload()

#Load the dataset
df = pd.read_csv('gold_data (1).csv') # Make sure the filename matches

#Convert 'Date' to datetime and sort
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)
df = df.sort_values('Date')
```

```

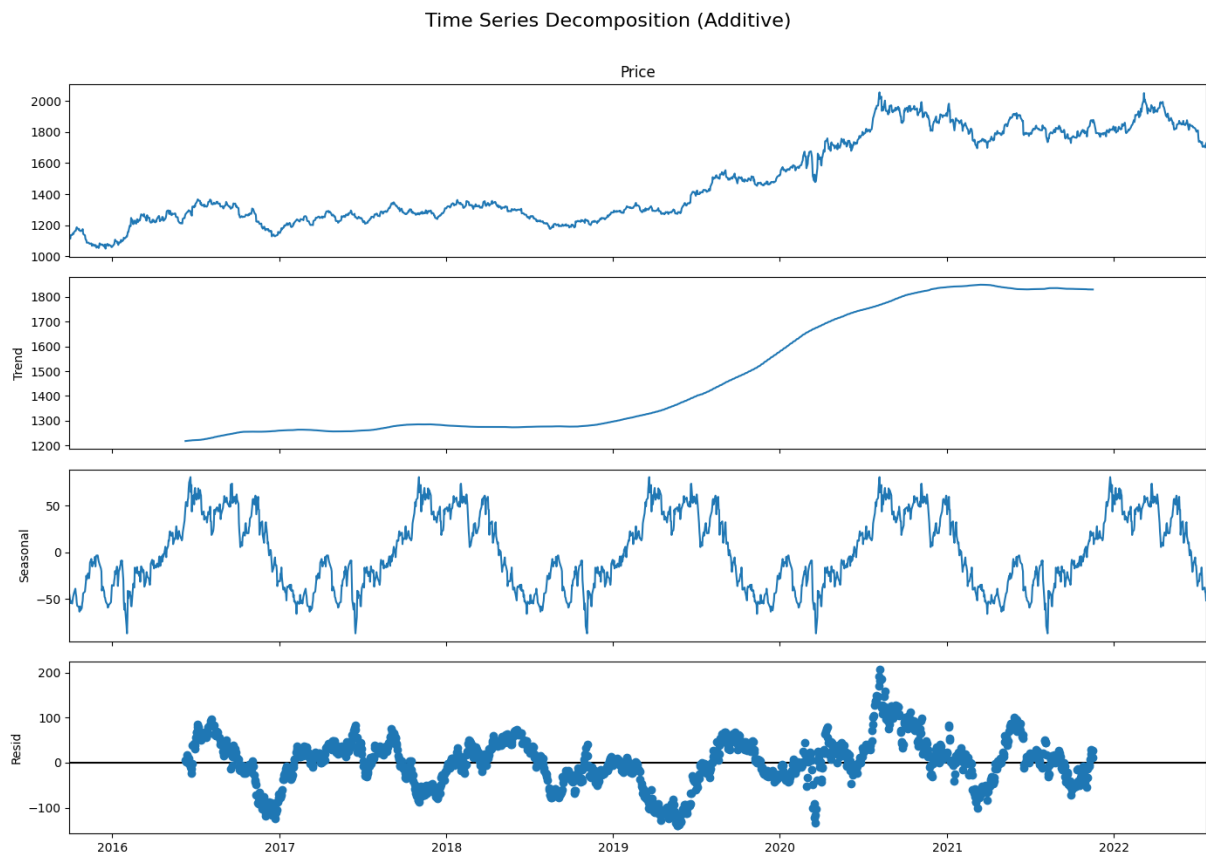
#Set 'Date' as index
df.set_index('Date', inplace=True)

#Perform time series decomposition
decomposition = seasonal_decompose(df['Price'], model='additive', period=365)

#Plot the decomposition
fig = decomposition.plot()
fig.set_size_inches(14, 10)
plt.suptitle("Time Series Decomposition (Additive)", fontsize=16)
plt.tight_layout(rect=[0, 0, 1, 0.97])
plt.show()

```

## OUTPUT:



## RESULT:

Thus, the program using the time series data implementation has been done successful.

