**Exp No 7**

**Date:31/3/25**

**Objective**

The objective of this implementation is to decompose a time series dataset into its key components: trend, seasonality, and residuals. This helps in understanding the underlying patterns in the data, which is crucial for forecasting and analysis.

**Background & Scope**

Time series decomposition is a fundamental technique used in time series analysis to separate data into:

* Trend: The long-term progression of the data.
* Seasonality: The repeated pattern at fixed intervals.
* Residuals: The random noise or irregular fluctuations in the data.

This technique is essential for fields such as economics, finance, weather forecasting, and energy consumption analysis.

**Implementation Steps**

**Step 1: Import Required Libraries**

We use Python libraries such as pandas for data handling, matplotlib for visualization, and statsmodels for decomposition.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from statsmodels.tsa.seasonal import seasonal\_decompose

**Step 2: Load and Preprocess Data**

We load the dataset, parse the date column, and set it as the index.

file\_path = "Electric\_Production.csv" # Update path if needed

df = pd.read\_csv(file\_path)

# Convert DATE column to datetime and set as index

df['DATE'] = pd.to\_datetime(df['DATE'], format='%d-%m-%Y') # Adjusted format

df.set\_index('DATE', inplace=True)

# Handle missing values

df.dropna(inplace=True)

**Step 3: Perform Seasonal Decomposition**

We apply the seasonal decomposition method to break the data into its components.

decomposed = seasonal\_decompose(df['Value'], model='additive', period=12) # Adjust period as needed

# Extract Components

df['Trend'] = decomposed.trend

df['Seasonality'] = decomposed.seasonal

df['Residuals'] = decomposed.resid

**Step 4: Visualizing the Decomposed Components**

We plot the original data along with its trend, seasonality, and residual components.

plt.figure(figsize=(12, 8))

plt.subplot(4, 1, 1)

plt.plot(df['Value'], label="Original Data", color='blue')

plt.legend(loc='best')

plt.title("Original Time Series Data")

plt.subplot(4, 1, 2)

plt.plot(df['Trend'], label="Trend", color='red')

plt.legend(loc='best')

plt.title("Trend Component")

plt.subplot(4, 1, 3)

plt.plot(df['Seasonality'], label="Seasonality", color='green')

plt.legend(loc='best')

plt.title("Seasonal Component")

plt.subplot(4, 1, 4)

plt.plot(df['Residuals'], label="Residuals", color='purple')

plt.legend(loc='best')

plt.title("Residual Component")

plt.tight\_layout()

plt.show()

**Conclusion**

This implementation successfully decomposes a time series dataset into its core components. The results help in identifying patterns such as long-term trends and seasonal variations, which are critical for making data-driven decisions in various domains. By analyzing the residuals, we can also assess how well the model captures the data's structure and identify any anomalies or outliers.