**Exp no: 2 Implement programs for visualizing time series data**

**Date: 31/1/25**

**Objectives:**

The goal of this preprocessing pipeline is to prepare time series sales data for accurate predictive modeling. This involves handling missing values, removing anomalies, ensuring correct time order, and potentially transforming features. The aim is to create a clean and consistent dataset that improves the performance of time series forecasting models, enabling better predictions of future sales trends and the identification of seasonality patterns.

**Background/Scope:**

Time series sales data is a sequence of observations indexed by time. Preprocessing is crucial because missing values, outliers, incorrect time order, and other data quality issues can negatively impact forecasting models. This process addresses these challenges by filling gaps, removing outliers, ordering data chronologically, and potentially creating new features. Proper preprocessing is essential for time series models to accurately capture trends, predict future sales, and support informed business decisions.

### ****Steps for Time Series Sales Data Preprocessing****:

**Step 1: Load the Dataset and libraries**

1. **Load** the dataset from a CSV file into a Pandas DataFrame.

# Step 1: Load the Dataset

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from statsmodels.graphics.tsaplots import plot\_acf

from statsmodels.tsa.stattools import adfuller

df=pd.read\_csv("/content/NFLX.csv")

**Step 2: Visualizing techniques**

**Autocorrelation Plot:** Create an autocorrelation plot to identify potential seasonality using the acf\_plot

if 'Date' not in df.columns:

    print("'Date' is already the index or not present in the DataFrame.")

else:

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

# Plot the ACF

plt.figure(figsize=(12, 6))

plot\_acf(df['Volume'], lags=40) # You can adjust the number of lags as needed

plt.xlabel('Lag')

plt.ylabel('Autocorrelation')

plt.title('Autocorrelation Function (ACF) Plot')

plt.show()

**Lag Plot:** Create a lag plot (scatter plot of the time series against lagged values of itself) to visualize autocorrelation.

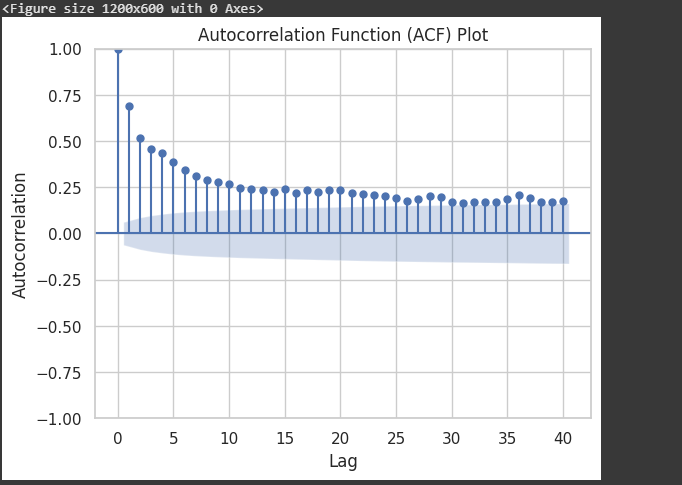
from pandas.plotting import lag\_plot

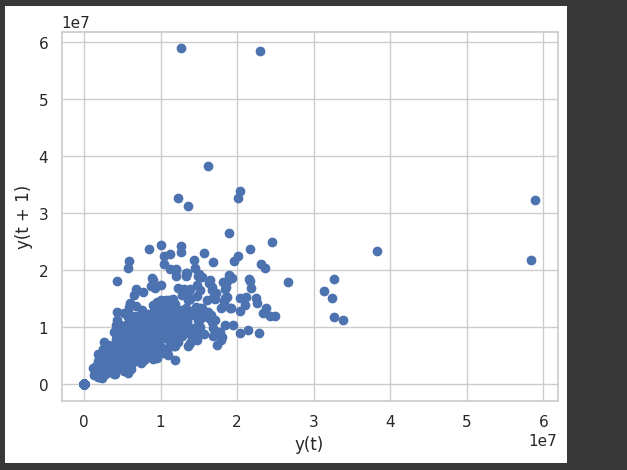
# lag scatter plot

lag\_plot(df)

plt.show()

output :





Result:

Thus the time series dataset is visualized successfully.