Exp no: 5 implement program to apply moving average smoothing for data preparation and time series forecasting.

Date: 20/03/25

**Objectives:**

The objective of this experiment is to forecast stock prices using smoothing techniques. This includes transforming time series data using seasonal differencing, and applying forecasting models such as Simple Exponential Smoothing (SES) and Holt-Winters Exponential Smoothing (HWES). Additionally, rolling statistics such as Simple Moving Averages (SMA) are computed and visualized to identify trends.

**Background/Scope:**

Smoothing techniques are widely used in time series forecasting to reduce noise and identify trends. Simple Exponential Smoothing (SES) assumes no trend or seasonality, while Holt-Winters Exponential Smoothing (HWES) incorporates trend and seasonal components. Rolling averages such as SMA provide insights into short-term and long-term trends by averaging data points within a window.

Steps for Time Series Forecasting Using Smoothing Techniques:

\*\*Step 1: Load the Dataset\*\*

import pandas as pd

# Load the dataset

df = pd.read\_csv("market.csv")

# Display first few rows

df.head()

\*\*Step 2: Data Cleaning and Preprocessing\*\*

# Convert Date column to datetime format

df['Date'] = pd.to\_datetime(df['Date'])

# Remove missing and duplicate values

df.dropna(subset=['Date', 'Close', 'Adj Close', 'Open', 'High', 'Low', 'Volume'], inplace=True)

df.drop\_duplicates(subset=['Date', 'Ticker'], inplace=True)

# Remove outliers using IQR

Q1 = df['Close'].quantile(0.25)

Q3 = df['Close'].quantile(0.75)

IQR = Q3 - Q1

df = df[(df['Close'] >= (Q1 - 1.5 \* IQR)) & (df['Close'] <= (Q3 + 1.5 \* IQR))]

\*\*Step 3: Seasonal Differencing\*\*

# Apply seasonal differencing to remove trend

df['seasonal\_first\_difference'] = df['Volume'] - df['Volume'].shift(12)

df['seasonal\_first\_difference'].fillna(0, inplace=True)

\*\*Step 4: Simple Exponential Smoothing (SES)\*\*

from statsmodels.tsa.holtwinters import SimpleExpSmoothing

# Apply SES

df['SES'] = SimpleExpSmoothing(df['seasonal\_first\_difference']).fit(smoothing\_level=0.5, optimized=False).fittedvalues

\*\*Step 5: Holt-Winters Exponential Smoothing (HWES)\*\*

from statsmodels.tsa.holtwinters import ExponentialSmoothing

# Apply HWES

df['HWES'] = ExponentialSmoothing(df['seasonal\_first\_difference'], trend='add', seasonal='add', seasonal\_periods=7).fit().fittedvalues

\*\*Step 6: Visualization of Smoothing Techniques\*\*

import matplotlib.pyplot as plt

# Original volume data

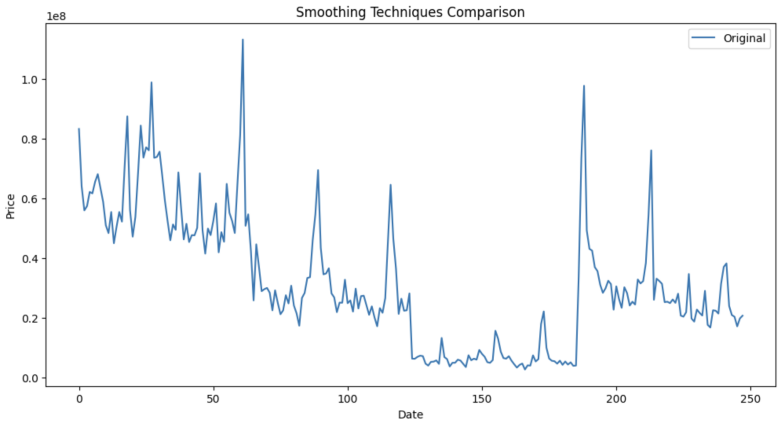
plt.figure(figsize=(12, 6))

plt.plot(df['Volume'], label='Original')

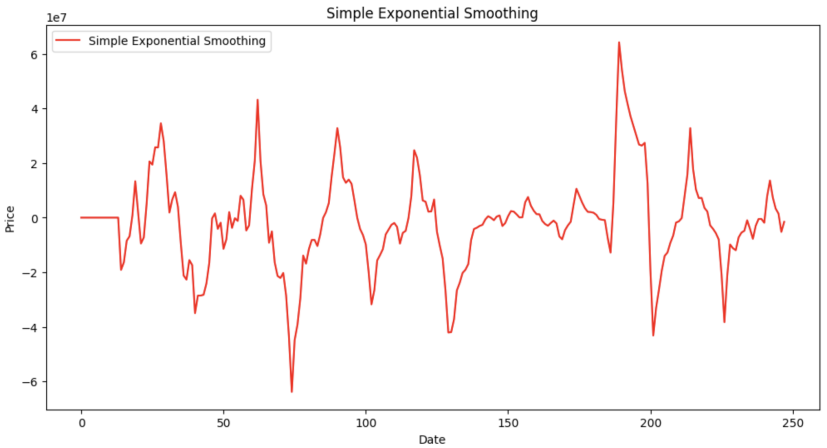
plt.title('Original Stock Volume')

plt.legend()

plt.show()



# SES output

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

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

plt.title('Simple Exponential Smoothing')

plt.legend()

plt.show()

# HWES output

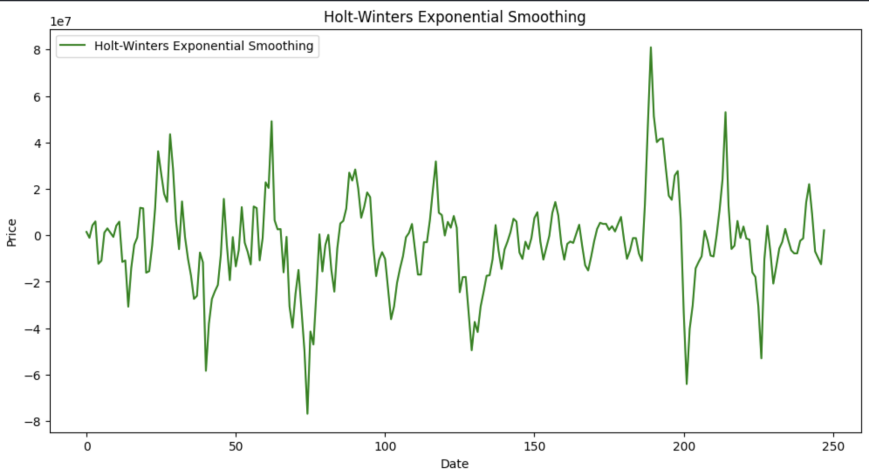
plt.figure(figsize=(12, 6))

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

plt.title('Holt-Winters Exponential Smoothing')

plt.legend()

plt.show()



\*\*Step 7: Weekly Resampling\*\*

weekly\_data = df.resample('W', on='Date').mean(numeric\_only=True)

\*\*Step 8: Moving Averages (SMA 20 and SMA 50)\*\*

# Calculate 20-day and 50-day SMA

sma\_20 = df['High'].rolling(window=20).mean()

sma\_50 = df['High'].rolling(window=50).mean()

# Combine into a DataFrame

priceSma\_df = pd.DataFrame({

'High': df['High'],

'SMA 20': sma\_20,

'SMA 50': sma\_50

})

# Plot moving averages

priceSma\_df.plot(figsize=(12, 6))

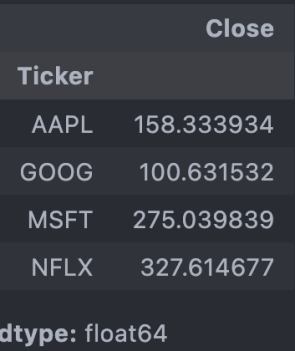
plt.title('SMA 20 vs SMA 50')

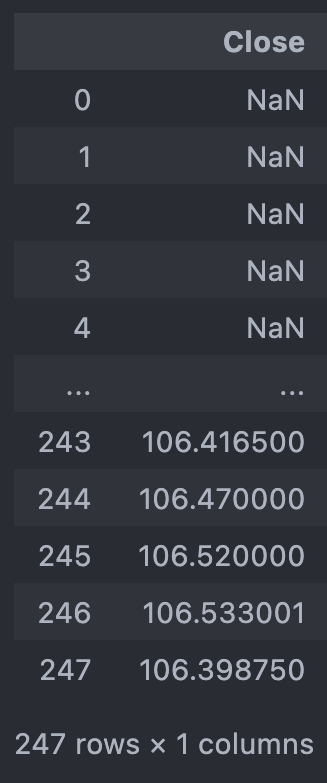
plt.xlabel('Date')

plt.ylabel('Stock Price')

plt.grid(True)

plt.show()





\*\*Result:\*\*

Thus, the stock market dataset was successfully preprocessed and used to forecast trends using SES and HWES smoothing techniques. Rolling averages further enhanced the understanding of short-term and long-term market behavior.