In [278...

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [279...

```
a1 = pd.read_csv(r'D:\DATA_SCIENCE_COURSE\PROJECTS\PROJECT-3(GOLD)\Gold_data.csv',

# Set the 'date' column as the index
a1.set_index('date', inplace=True)
a1
```

Out[279]:

date	price
2016-01-01	2252.60
2016-01-02	2454.50
2016-01-03	2708.10
2016-01-04	2577.80
2016-01-05	2597.75
...	...
2021-12-17	4394.40
2021-12-18	4389.50
2021-12-19	4389.50
2021-12-20	4354.10
2021-12-21	4346.50

2182 rows × 1 columns

In [280...

```
a1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2182 entries, 2016-01-01 to 2021-12-21
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   price   2182 non-null   float64
dtypes: float64(1)
memory usage: 34.1 KB
```

In [281...

```
a1.isnull().sum()
```

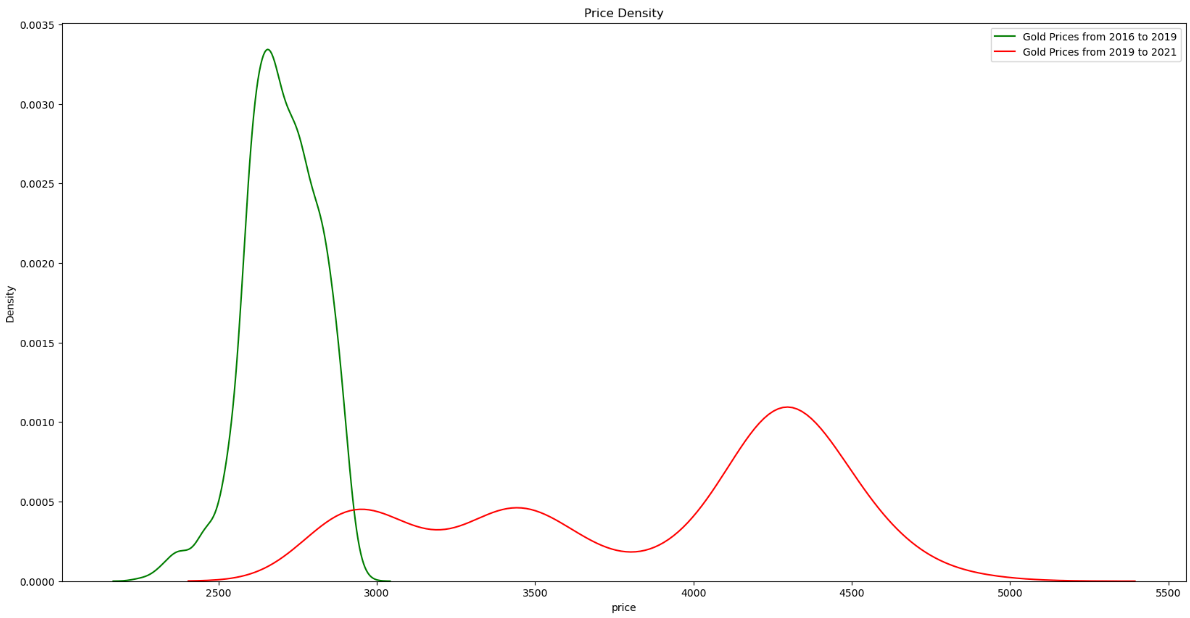
Out[281]:

```
price    0
dtype: int64
```

In [282...

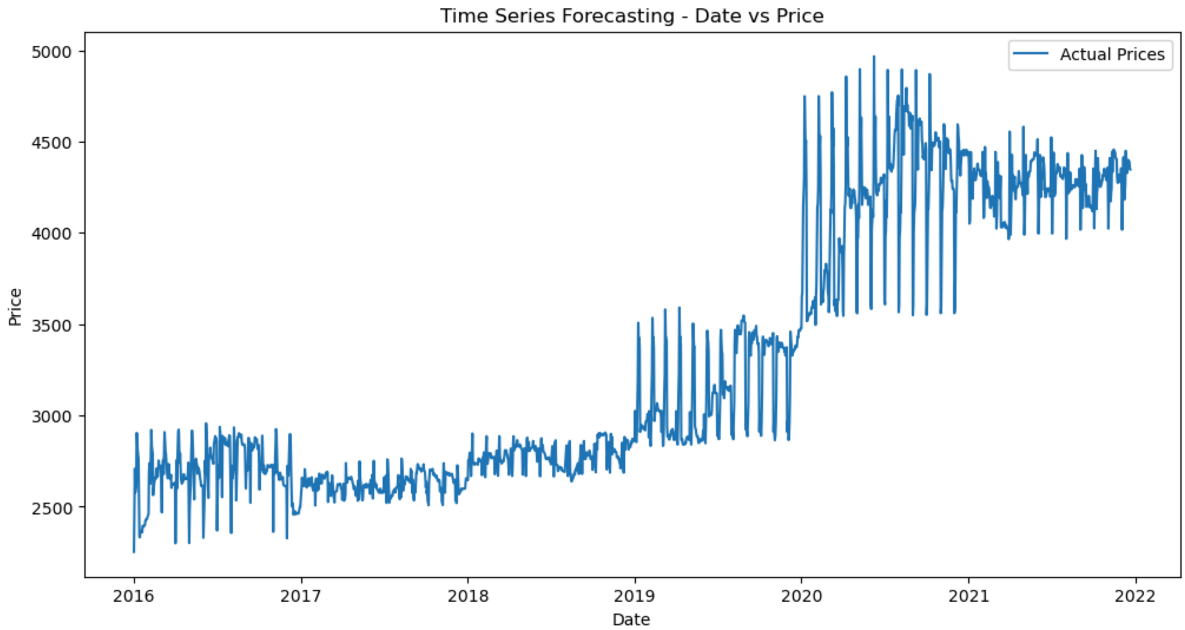
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(20, 10))

# Plot 1
sns.kdeplot(a1['price'].iloc[:1096], color='green', label='Gold Prices from 2016 to
plt.title('Price Density')

# Plot 2
```

```
sns.kdeplot(a1['price'].iloc[1096:2183], color='red', label='Gold Prices from 2019
plt.title('Price Density')

plt.legend()
plt.show()
```



In [283…

```
# Plot the original data
plt.figure(figsize=(12, 6))
plt.plot(a1.index, a1['price'], label='Actual Prices')

plt.title('Time Series Forecasting - Date vs Price')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```



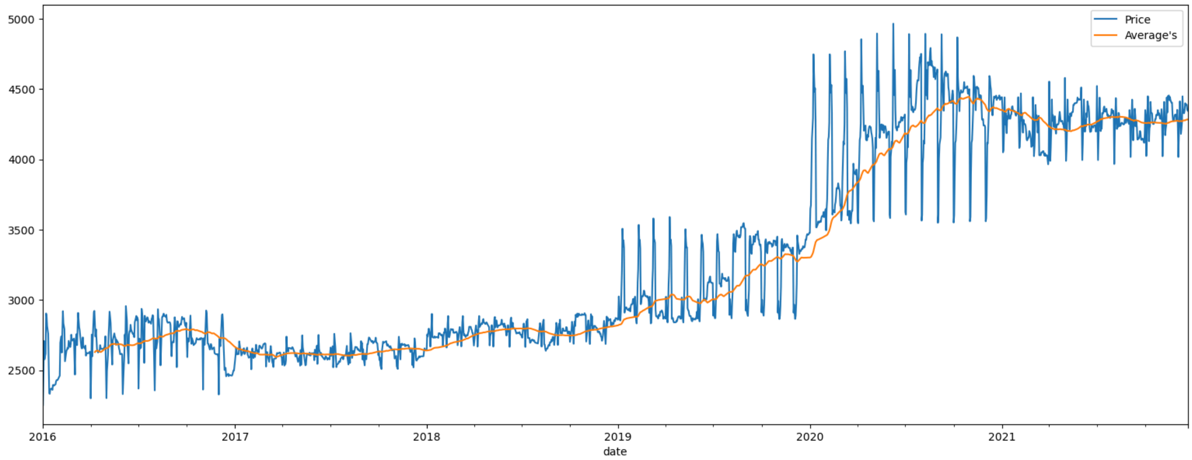
Moving Average

In [284…

```
plt.figure(figsize=(19,7))
a1.price.plot(label="Price")
pred = pd.DataFrame()
pred["Moving_Avg"] = a1["price"].rolling(100).mean()
```

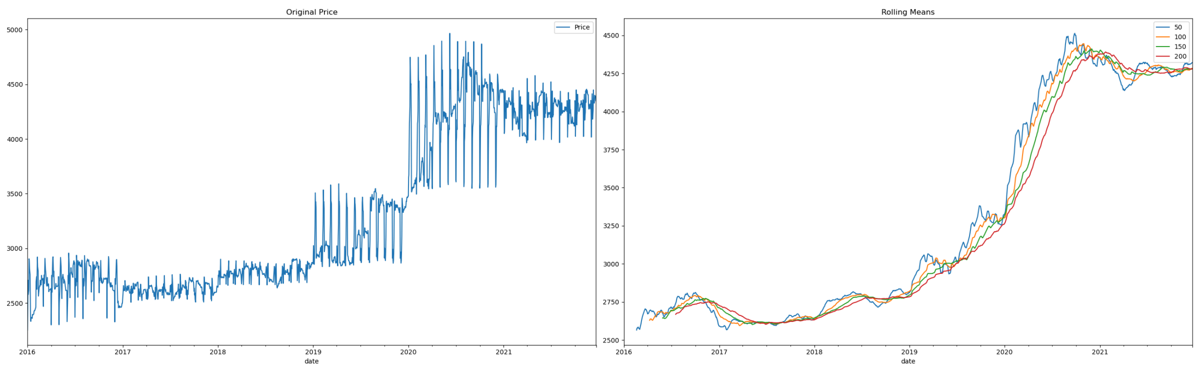
```
pred["Moving_Avg"].plot(label="Average's")
plt.legend(loc='best')
```

Out[284]: <matplotlib.legend.Legend at 0x1943264c490>



In [285…

```
import matplotlib.pyplot as plt

plt.figure(figsize=(26, 8))

# Plot original price
plt.subplot(1, 2, 1)
a1['price'].plot(label="Price")
plt.title('Original Price')
plt.legend()

# Plot rolling means
plt.subplot(1, 2, 2)
for i in range(50, 250, 50):
    a1['price'].rolling(i).mean().plot(label=str(i))

plt.title('Rolling Means')
plt.legend(loc='best')

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



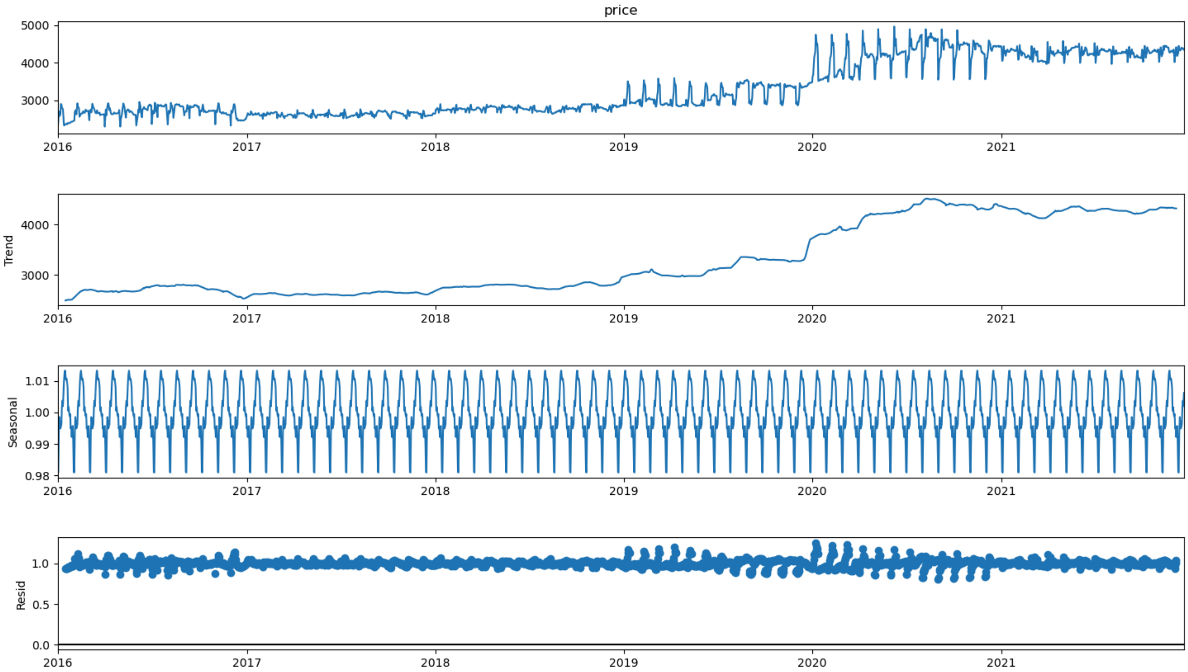
Time series decomposition plot

In [286…

```
from statsmodels.tsa.seasonal import seasonal_decompose
#To separate the trend and the seasonality from a time series,
# we can decompose the series using the following code.
result = seasonal_decompose(a1['price'], model='multiplicative', period = 31)
# period = 31  ,For monthly data with a consistent pattern every month,
fig = plt.figure()
```

```
fig = result.plot()
fig.set_size_inches(16, 9)
```

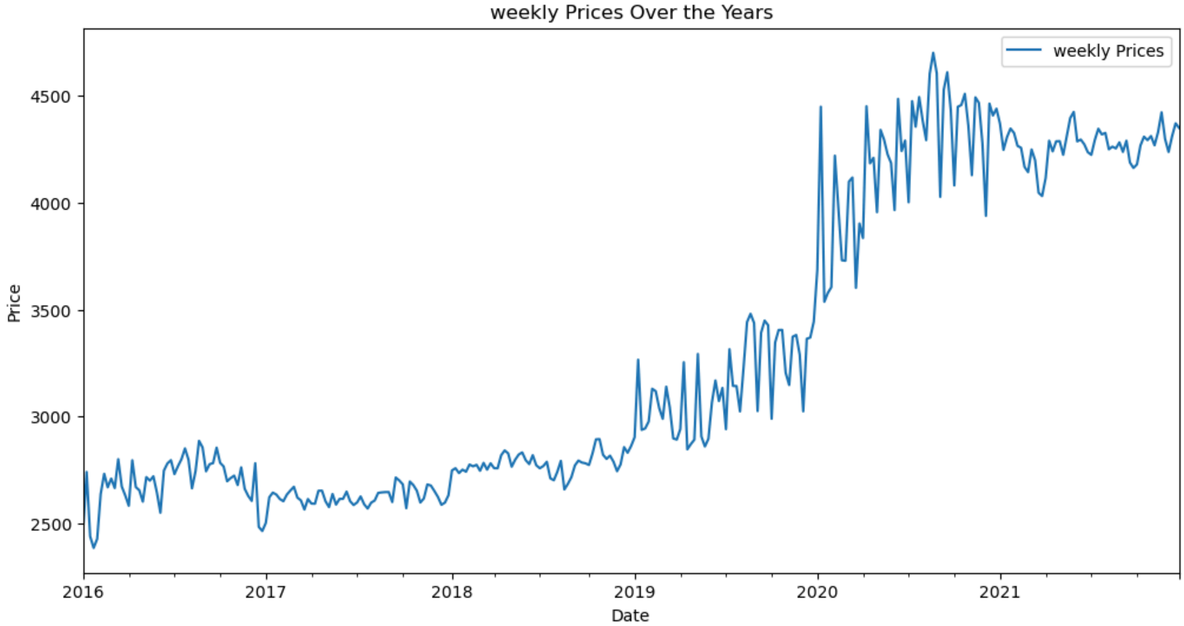
<Figure size 640x480 with 0 Axes>



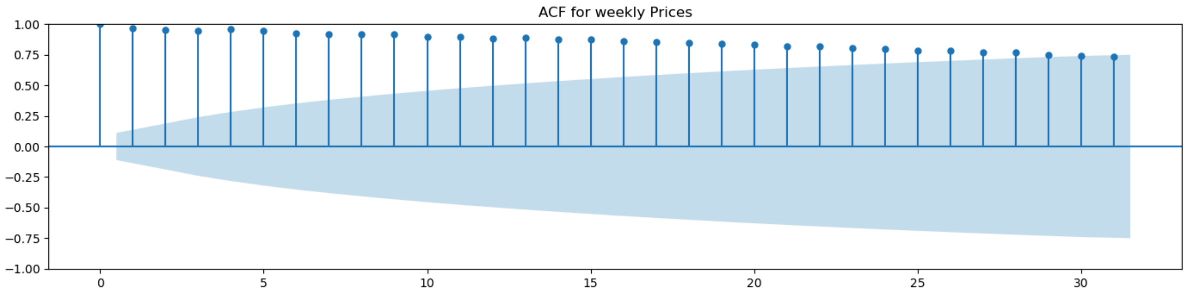
ACF plots and PACF plots

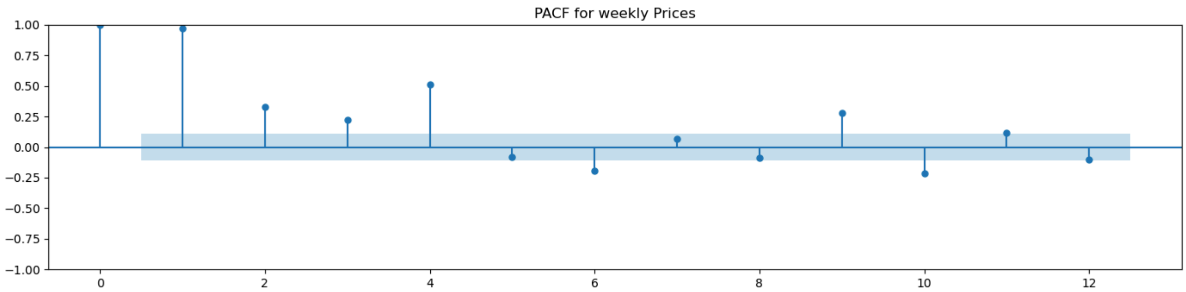
In [287…

```
import statsmodels.graphics.tsaplots as tsa_plots

# Resample the data to weekly frequency
weekly_prices = a1['price'].resample('W').mean()

# Plot the original weekly prices
plt.figure(figsize=(12, 6))
weekly_prices.plot(label='weekly Prices')
plt.title('weekly Prices Over the Years')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

# Plot ACF and PACF
plt.figure(figsize=(14, 7))

# ACF plot
plt.subplot(2, 1, 1)
tsa_plots.plot_acf(weekly_prices, lags=31, ax=plt.gca())
plt.title('ACF for weekly Prices')

# PACF plot
plt.subplot(2, 1, 2)
tsa_plots.plot_pacf(weekly_prices, lags=12, ax=plt.gca())
plt.title('PACF for weekly Prices')

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




```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\graphics\tsaplots.py:348: F
utureWarning: The default method 'yw' can produce PACF values outside of the [-1,
1] interval. After 0.13, the default will change tounadjusted Yule-Walker ('ywm').
You can use this method now by setting method='ywm'.
  warnings.warn(
```





In [288…

```
print(weekly_prices)
```

```
date
2016-01-03    2471.733333
2016-01-10    2740.778571
2016-01-17    2439.657143
2016-01-24    2385.528571
2016-01-31    2427.692857
                  ...
2021-11-28    4298.785714
2021-12-05    4237.885714
2021-12-12    4311.571429
2021-12-19    4371.642857
2021-12-26    4350.300000
Freq: W-SUN, Name: price, Length: 313, dtype: float64
```

ACF (AutoCorrelation Function): ACF measures the correlation between a time series and its lagged values. It helps identify patterns such as seasonality and trends in the data. Used to determine the order of the autoregressive (AR) component in time series forecasting models.

PACF (Partial AutoCorrelation Function): PACF measures the correlation between a time series and its lagged values while removing the effects of intermediate lags. It helps identify the direct relationship between an observation and its past observations. Used to determine the order of the moving average (MA) component in time series forecasting models.

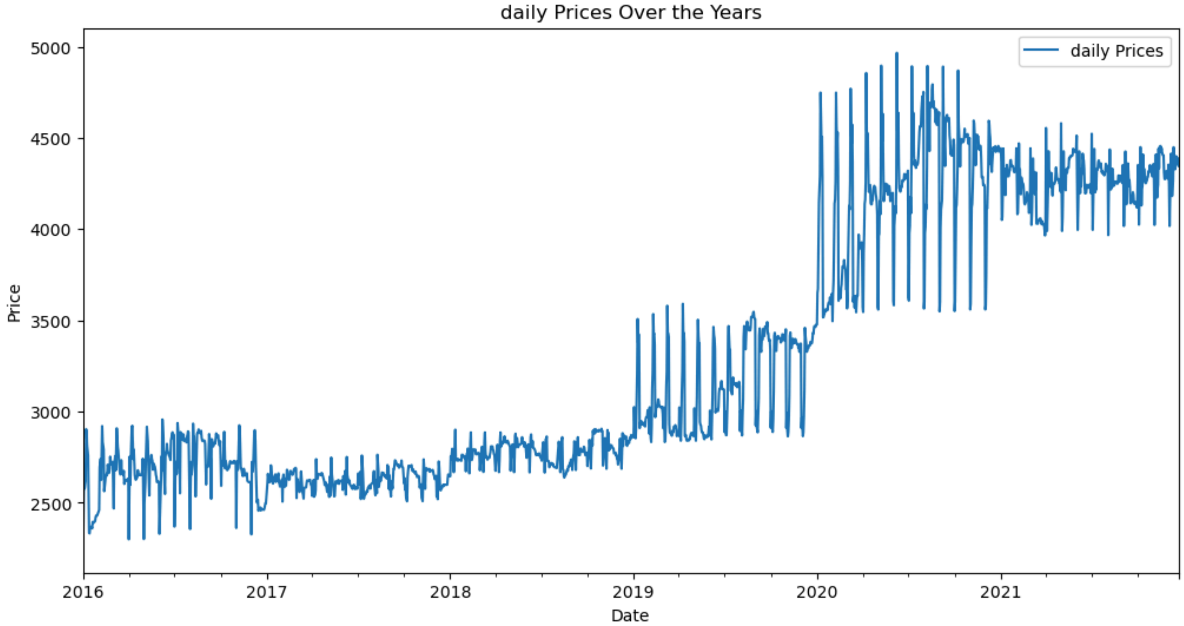
Uses:

-Identifying seasonality and trends.

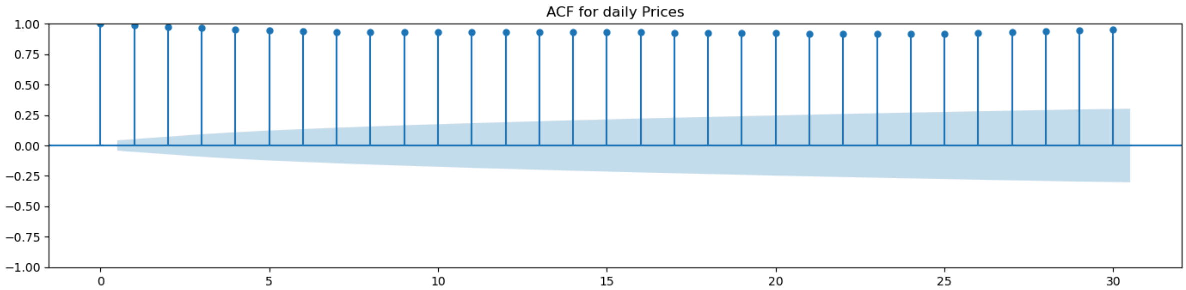
-Determining the order of autoregressive (AR) and moving average (MA) components in time series models such as ARIMA (AutoRegressive Integrated Moving Average).

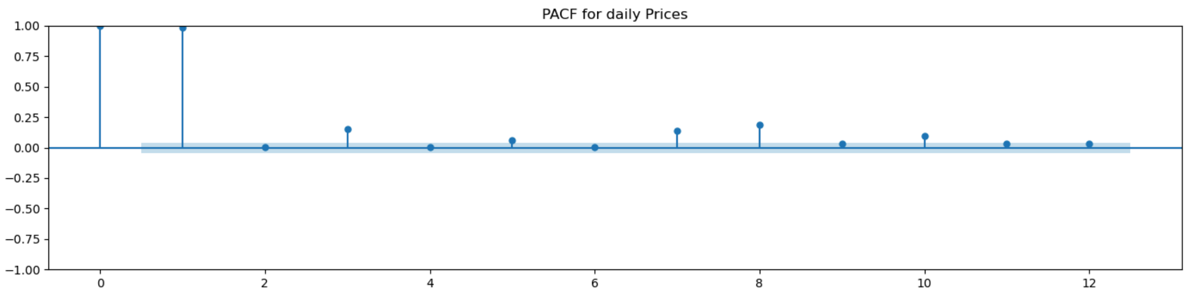
In [289…

```python
# Resample the data to daily frequency
daily_prices = a1['price'].resample('D').mean()

# Plot the original daily prices
plt.figure(figsize=(12, 6))
daily_prices.plot(label='daily Prices')
plt.title('daily Prices Over the Years')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

# Plot ACF and PACF
plt.figure(figsize=(14, 7))

# ACF plot
plt.subplot(2, 1, 1)
tsa_plots.plot_acf(daily_prices, lags=30, ax=plt.gca())
plt.title('ACF for daily Prices')

# PACF plot
plt.subplot(2, 1, 2)
tsa_plots.plot_pacf(daily_prices, lags=12, ax=plt.gca())
plt.title('PACF for daily Prices')

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




```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\graphics\tsaplots.py:348: F
utureWarning: The default method 'yw' can produce PACF values outside of the [-1,
1] interval. After 0.13, the default will change tounadjusted Yule-Walker ('ywm').
You can use this method now by setting method='ywm'.
  warnings.warn(
```





In [197…

```
import itertools
import statsmodels.api as sm

# Define ranges for p, d, q
p_range = d_range = q_range = range(0, 3)

# Generate all possible combinations of p, d, q
order_combinations = list(itertools.product(p_range, d_range, q_range))

# Perform grid search and store results
best_aic = float('inf')
best_order = None

for order in order_combinations:
    try:
        model = sm.tsa.ARIMA(daily_prices, order=order)
        results = model.fit()
        aic = results.aic
        if aic < best_aic:
```

```
            best_aic = aic
            best_order = order
    except:
        continue

print(f"Best AIC: {best_aic}")
print(f"Best Order: {best_order}")
```

```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sarimax.py:9
66: UserWarning: Non-stationary starting autoregressive parameters found. Using ze
ros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sarimax.py:9
78: UserWarning: Non-invertible starting MA parameters found. Using zeros as start
ing parameters.
  warn('Non-invertible starting MA parameters found.'
Best AIC: 26628.204946313333
Best Order: (2, 1, 2)
```

In [290…

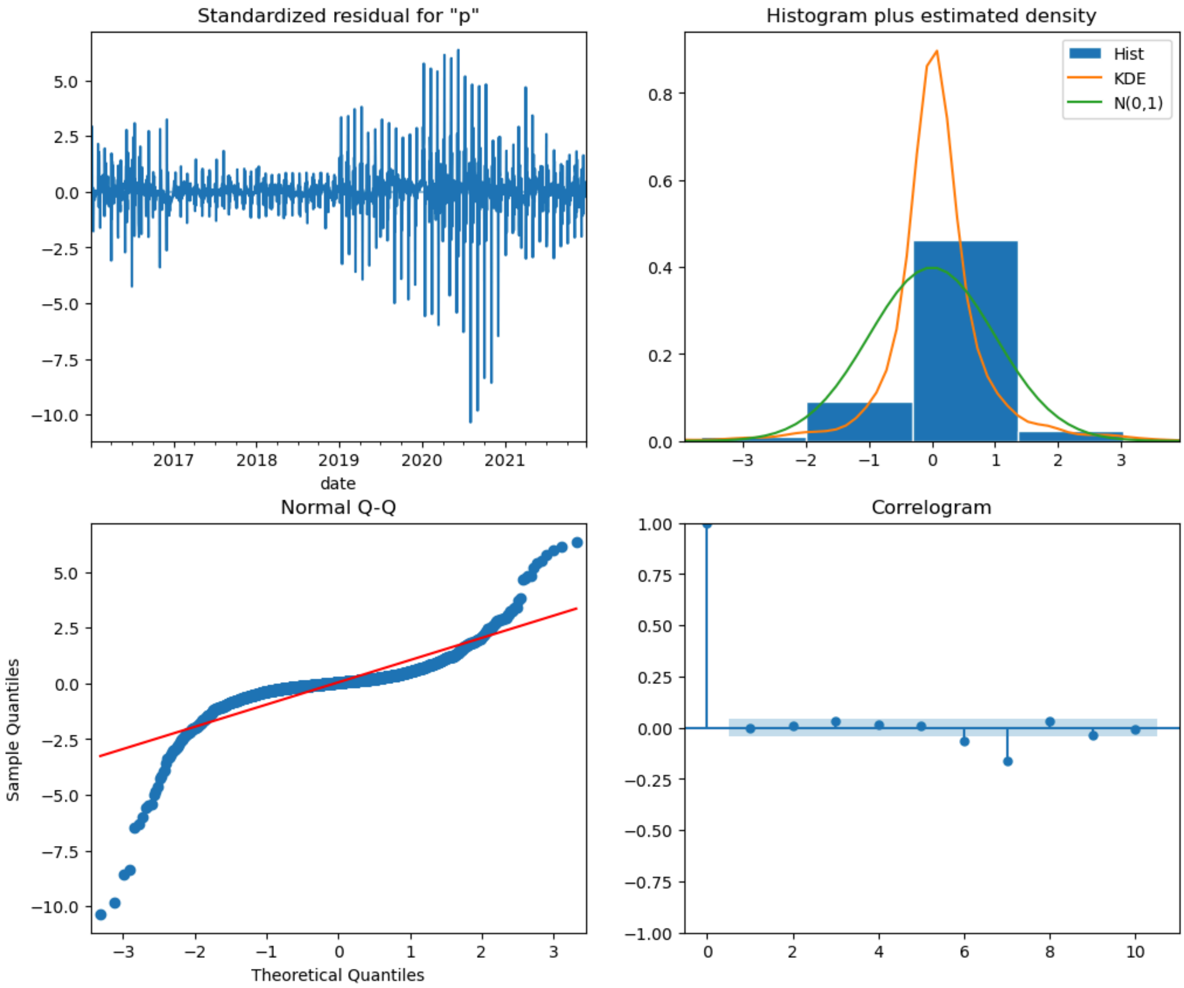
```
from statsmodels.tsa.arima.model import ARIMA

# Example ARIMA model with identified parameters (replace with your values)
arima_model = ARIMA(daily_prices, order=(2, 1, 2))
arima_results = arima_model.fit()

# Print model summary
print(arima_results.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  price   No. Observations:                 2182
Model:                 ARIMA(2, 1, 2)   Log Likelihood              -13309.102
Date:                Mon, 11 Mar 2024   AIC                          26628.205
Time:                        17:42:54   BIC                          26656.643
Sample:                    01-01-2016   HQIC                         26638.601
                         - 12-21-2021
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.3001      0.092      3.276      0.001       0.121       0.480
ar.L2          0.3240      0.079      4.093      0.000       0.169       0.479
ma.L1         -0.3886      0.088     -4.432      0.000      -0.560      -0.217
ma.L2         -0.5555      0.088     -6.282      0.000      -0.729      -0.382
sigma2      1.169e+04    115.090    101.535      0.000    1.15e+04    1.19e+04
==================================================================================
=
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):              50533.5
0
Prob(Q):                              0.91   Prob(JB):                          0.0
0
Heteroskedasticity (H):               4.69   Skew:                             -1.4
7
Prob(H) (two-sided):                  0.00   Kurtosis:                         26.4
0
==================================================================================
=

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-ste
p).
```

Diagnostic Checks

In [291…

```
arima_results.plot_diagnostics(figsize=(12, 10))
plt.show()
```



Model Validation

In [292…

```
train_size = int(len(daily_prices) * 0.8)
train, test = daily_prices[:train_size], daily_prices[train_size:]

arima_model = ARIMA(train, order=(2, 1, 2))
arima_results = arima_model.fit()

predictions = arima_results.predict(start=len(train), end=len(train) + len(test) -
```

In [293…

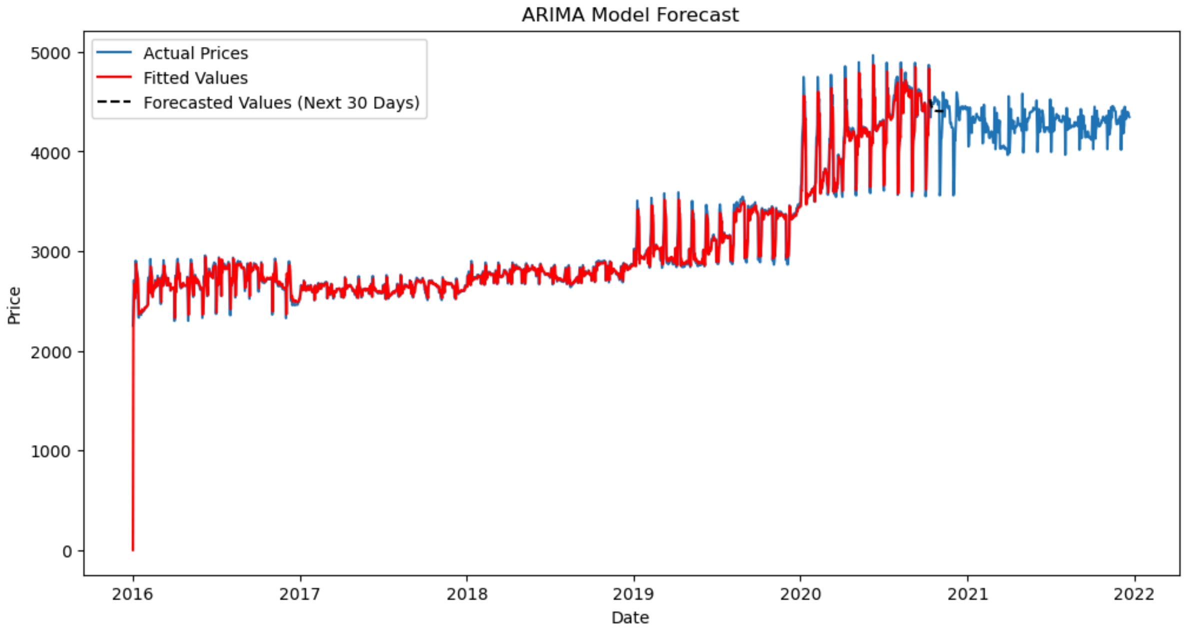
```
from sklearn.metrics import mean_squared_error
import numpy as np
mse = mean_squared_error(test, predictions)
rmse = np.sqrt(mse)
print(f"Root Mean Squared Error: {rmse}")
```

```
Root Mean Squared Error: 189.95471015214216
```

In [294…

```
# Forecast future values for the next 30 days
future_steps = 30
forecast = arima_results.get_forecast(steps=future_steps)
forecast_values = forecast.predicted_mean
```

```
# Plotting the original series, fitted values, and forecasted values
plt.figure(figsize=(12, 6))
plt.plot(daily_prices, label='Actual Prices')
plt.plot(arima_results.fittedvalues, color='red', label='Fitted Values')
plt.plot(forecast_values, color='black', linestyle='dashed', label='Forecasted Valu
plt.legend()
plt.title('ARIMA Model Forecast')
plt.xlabel('Date')
plt.ylabel('Price')
plt.show()
```



In [295…

```
daily_prices
```

Out[295]:

```
date
2016-01-01    2252.60
2016-01-02    2454.50
2016-01-03    2708.10
2016-01-04    2577.80
2016-01-05    2597.75
                ...
2021-12-17    4394.40
2021-12-18    4389.50
2021-12-19    4389.50
2021-12-20    4354.10
2021-12-21    4346.50
Freq: D, Name: price, Length: 2182, dtype: float64
```

In [306…

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import LSTM, Dense


# Convert the time series to a numpy array
price_series = daily_prices.values.reshape(-1, 1)

# Normalize the data
scaler = MinMaxScaler(feature_range=(0, 1))
price_series_scaled = scaler.fit_transform(price_series)

# Function to prepare the LSTM input data
```

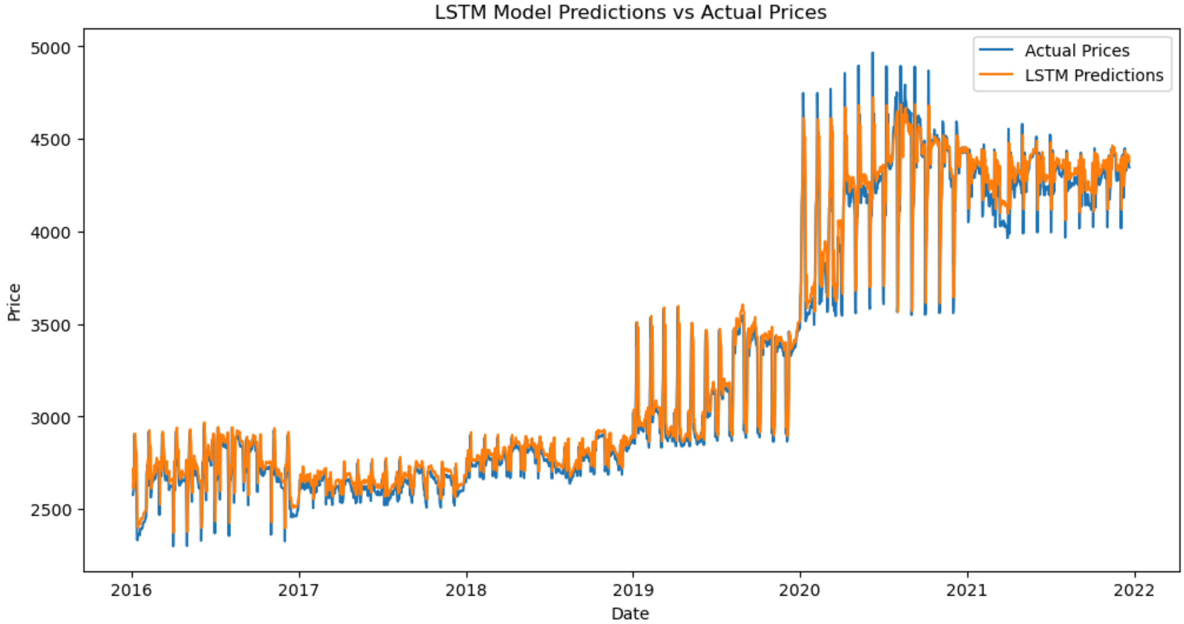
```python
def create_lstm_dataset(dataset, time_steps=1):
    X, y = [], []
    for i in range(len(dataset) - time_steps):
        a = dataset[i:(i + time_steps), 0]
        X.append(a)
        y.append(dataset[i + time_steps, 0])
    return np.array(X), np.array(y)

# Set the number of time steps
time_steps = 3  # You can adjust this parameter based on your data and problem

# Create LSTM input data
X, y = create_lstm_dataset(price_series_scaled, time_steps)

# Reshape the input data for LSTM (samples, time steps, features)
X = np.reshape(X, (X.shape[0], X.shape[1], 1))

# Build the LSTM model
model = Sequential()
model.add(LSTM(units=50, activation='relu', input_shape=(time_steps, 1)))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model.fit(X, y, epochs=100, batch_size=32)

# Predictions on the training data for illustration (replace with future data for f
train_predictions = model.predict(X)

# Invert the predictions to original scale
train_predictions_original = scaler.inverse_transform(train_predictions)
y_original = scaler.inverse_transform(y.reshape(-1, 1))

# Calculate RMSE on the training data for illustration
rmse = np.sqrt(mean_squared_error(y_original, train_predictions_original))
print(f"Root Mean Squared Error (LSTM): {rmse}")

# Visualize the results
plt.figure(figsize=(12, 6))
plt.plot(daily_prices.index[time_steps:], y_original, label='Actual Prices')
plt.plot(daily_prices.index[time_steps:], train_predictions_original, label='LSTM F
plt.title('LSTM Model Predictions vs Actual Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```

```
Epoch 1/100
69/69 [==============================] - 2s 3ms/step - loss: 0.0682
Epoch 2/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0050
Epoch 3/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0027
Epoch 4/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0026
Epoch 5/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0026
Epoch 6/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0026
Epoch 7/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0026
Epoch 8/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0026
Epoch 9/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0025
Epoch 10/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0024
Epoch 11/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0024
Epoch 12/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0023
Epoch 13/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0023
Epoch 14/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0023
Epoch 15/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0022
Epoch 16/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0022
Epoch 17/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0022
Epoch 18/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0021
Epoch 19/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0021
Epoch 20/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0021
Epoch 21/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0021
Epoch 22/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0020
Epoch 23/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0020
Epoch 24/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0019
Epoch 25/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0019
Epoch 26/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0019
Epoch 27/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0019
Epoch 28/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 29/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0020
Epoch 30/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0018
Epoch 31/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 32/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
```

```
Epoch 33/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 34/100
69/69 [==============================] - 0s 4ms/step - loss: 0.0018
Epoch 35/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 36/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 37/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 38/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 39/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 40/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0018
Epoch 41/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 42/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0017
Epoch 43/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 44/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0018
Epoch 45/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0018
Epoch 46/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0018
Epoch 47/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 48/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 49/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 50/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 51/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 52/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 53/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 54/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 55/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 56/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 57/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 58/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0017
Epoch 59/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0017
Epoch 60/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0018
Epoch 61/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0017
Epoch 62/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 63/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 64/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0017
```

```
Epoch 65/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 66/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 67/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0017
Epoch 68/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 69/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 70/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0017
Epoch 71/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0017
Epoch 72/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0017
Epoch 73/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 74/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0016
Epoch 75/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 76/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 77/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 78/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 79/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 80/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 81/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 82/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 83/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 84/100
69/69 [==============================] - 0s 4ms/step - loss: 0.0017
Epoch 85/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 86/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 87/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 88/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 89/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 90/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 91/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 92/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0016
Epoch 93/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 94/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0016
Epoch 95/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0017
Epoch 96/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
```

```
Epoch 97/100
69/69 [==============================] - 0s 2ms/step - loss: 0.0016
Epoch 98/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 99/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
Epoch 100/100
69/69 [==============================] - 0s 3ms/step - loss: 0.0016
69/69 [==============================] - 0s 2ms/step
Root Mean Squared Error (LSTM): 112.48984471129229
```

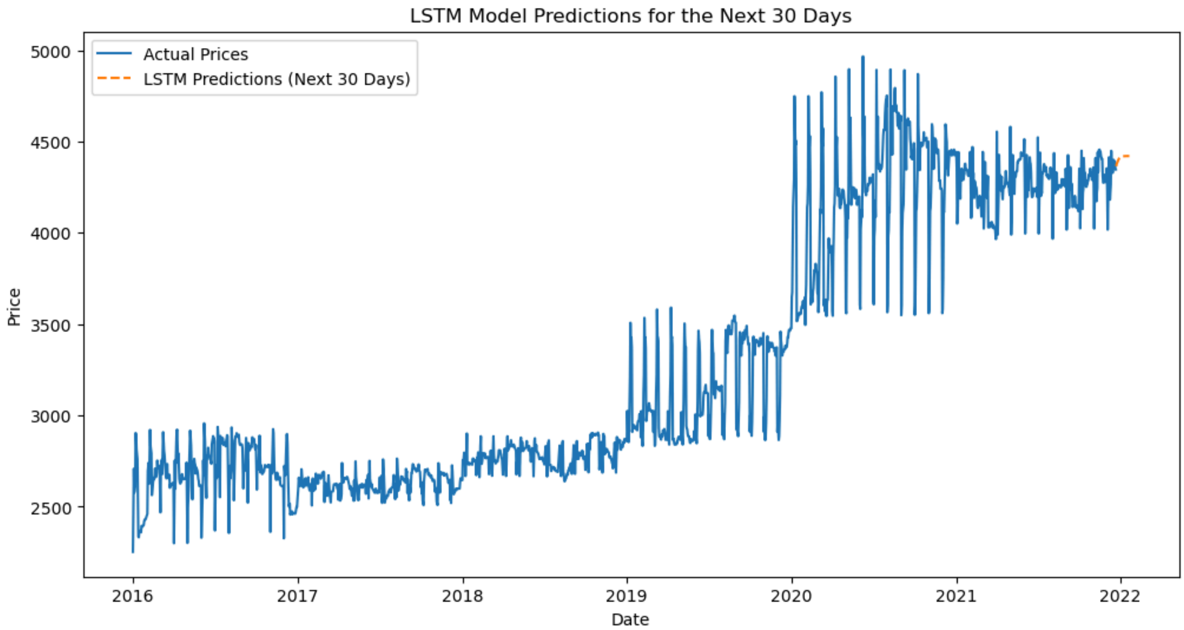


In [305…

```
# Function to generate LSTM predictions for a given input sequence
def generate_lstm_predictions(model, input_sequence, num_predictions):
    predictions = []
    for _ in range(num_predictions):
        # Predict the next value
        next_prediction = model.predict(input_sequence.reshape(1, input_sequence.sh
        predictions.append(next_prediction[0, 0])

        # Update the input sequence for the next prediction
        input_sequence = np.roll(input_sequence, shift=-1)
        input_sequence[-1] = next_prediction[0, 0]

    return predictions

# Number of days to predict into the future
num_days_to_predict = 30

# Create an initial input sequence for prediction (use the last 'time_steps' days f
initial_sequence = price_series_scaled[-time_steps:]

# Generate predictions for the next 30 days
future_predictions_scaled = generate_lstm_predictions(model, initial_sequence, num_

# Invert the predictions to the original scale
future_predictions_original = scaler.inverse_transform(np.array(future_predictions_

# Generate dates for the next 30 days
last_date = daily_prices.index[-1]
next_dates = pd.date_range(start=last_date + pd.DateOffset(days=1), periods=num_day

# Create a DataFrame with the predicted values and dates
future_predictions_df = pd.DataFrame({'Date': next_dates, 'Predicted Price': future
future_predictions_df.set_index('Date', inplace=True)
```

```
# Visualize the predictions for the next 30 days
plt.figure(figsize=(12, 6))
plt.plot(daily_prices.index, daily_prices, label='Actual Prices')
plt.plot(future_predictions_df.index, future_predictions_df['Predicted Price'], lab
plt.title('LSTM Model Predictions for the Next 30 Days')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

# Display the predicted values for the next 30 days
print("Predicted Prices for the Next 30 Days:")
print(future_predictions_df)
```

```
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 12ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 17ms/step
```




```
Predicted Prices for the Next 30 Days:
            Predicted Price
Date
2021-12-22      4364.439941
2021-12-23      4376.914062
2021-12-24      4383.214844
2021-12-25      4389.250000
2021-12-26      4395.519043
2021-12-27      4401.036133
2021-12-28      4405.470215
2021-12-29      4408.776367
2021-12-30      4411.329590
2021-12-31      4413.346191
2022-01-01      4414.917480
2022-01-02      4416.134277
2022-01-03      4417.081543
2022-01-04      4417.820312
2022-01-05      4418.395508
2022-01-06      4418.842285
2022-01-07      4419.190430
2022-01-08      4419.461914
2022-01-09      4419.673340
2022-01-10      4419.837891
2022-01-11      4419.965820
2022-01-12      4420.065430
2022-01-13      4420.143555
2022-01-14      4420.204102
2022-01-15      4420.250977
2022-01-16      4420.287598
2022-01-17      4420.315918
2022-01-18      4420.338379
2022-01-19      4420.355957
2022-01-20      4420.369629
```

In [298…

```
# Calculate the index for splitting (80% for training, 20% for testing)
split_index = int(0.8 * len(a1))

# Split the dataset into training and testing sets
train = a1.iloc[:split_index]
test = a1.iloc[split_index:]

# Display the shapes of the training and testing sets
print("Training Set Shape:", train.shape)
print("Testing Set Shape:", test.shape)
```

```
Training Set Shape: (1745, 1)
Testing Set Shape: (437, 1)
```

Evaluation Metric MAPE

In [299…

```python
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.holtwinters import SimpleExpSmoothing # SES
from statsmodels.tsa.holtwinters import Holt # Holts Exponential Smoothing
from statsmodels.tsa.holtwinters import ExponentialSmoothing
```

In [300…

```python
def MAPE(pred,org):
    temp = np.abs((pred-org)/org)*100
    return np.mean(temp)
```

Simple Exponential Smoothing

In [301…

```python
ses_model = SimpleExpSmoothing(train["price"]).fit(smoothing_level=0.7)
pred_ses = pd.DataFrame()
pred_ses["Exp_Smoothing"] = ses_model.predict(start = test.index[0],end = test.inde
MAPE(pred_ses["Exp_Smoothing"],test.price)
```

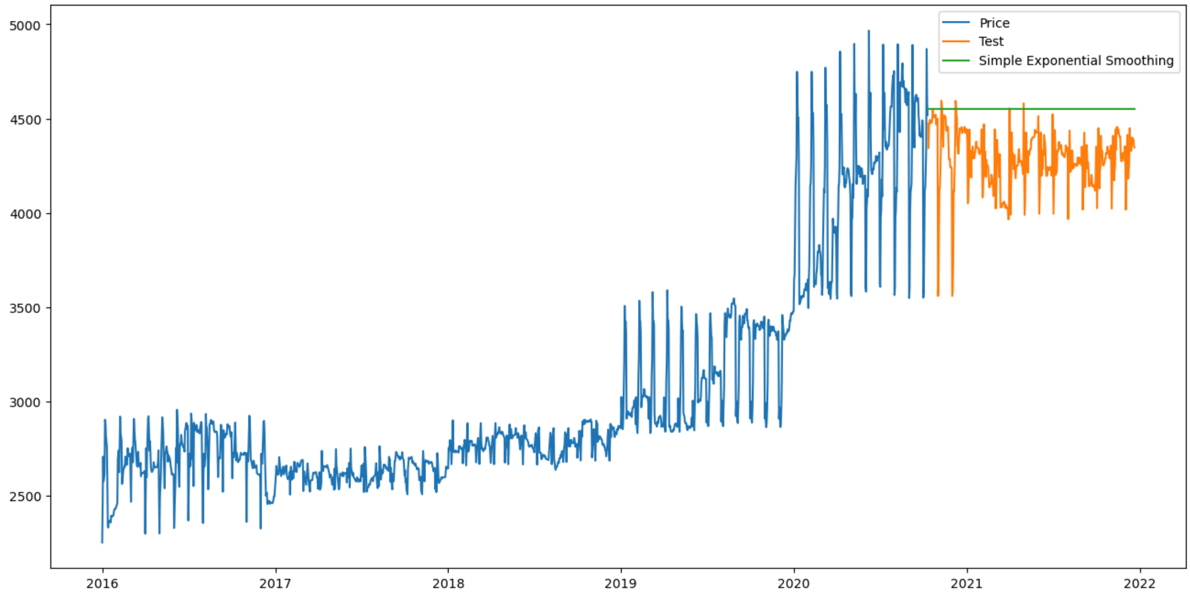
```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
```

Out[301]:

```
6.3174505837272354
```

In [302…

```python
plt.figure(figsize=(16,8))
plt.plot(train["price"], label='Price')
plt.plot(test["price"], label='Test')
plt.plot(pred_ses["Exp_Smoothing"], label='Simple Exponential Smoothing')
plt.legend(loc='best')
plt.show()
```



Holt method

In [250…

```python
# Grid search for Holt-Winters parameters
best_mape = float('inf')
```

```
best_params = None

for smoothing_level in np.arange(0.1, 1.0, 0.1):
    for smoothing_trend in np.arange(0.1, 1.0, 0.1):
        hw_model = ExponentialSmoothing(train["price"], trend="add", seasonal="add'
            smoothing_level=smoothing_level, smoothing_trend=smoothing_trend
        )
        pred_hw = hw_model.predict(start=test.index[0], end=test.index[-1])
        mape = MAPE(pred_hw, test["price"])

        if mape < best_mape:
            best_mape = mape
            best_params = {"smoothing_level": smoothing_level, "smoothing_trend": s

# Train the final Holt-Winters model with the best parameters on the entire dataset
final_hw_model = ExponentialSmoothing(a1["price"], trend="add", seasonal="add", sea
    smoothing_level=best_params["smoothing_level"], smoothing_trend=best_params["sm
)

# Make predictions for the test set
final_pred_hw["holt_method"] = final_hw_model.predict(start=test.index[0], end=test

# Display the best parameters and MAPE
print("Best Holt-Winters Parameters:")
print(best_params)
print("MAPE on Test Set:", best_mape)
```

```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
```

```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
```

```
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
```

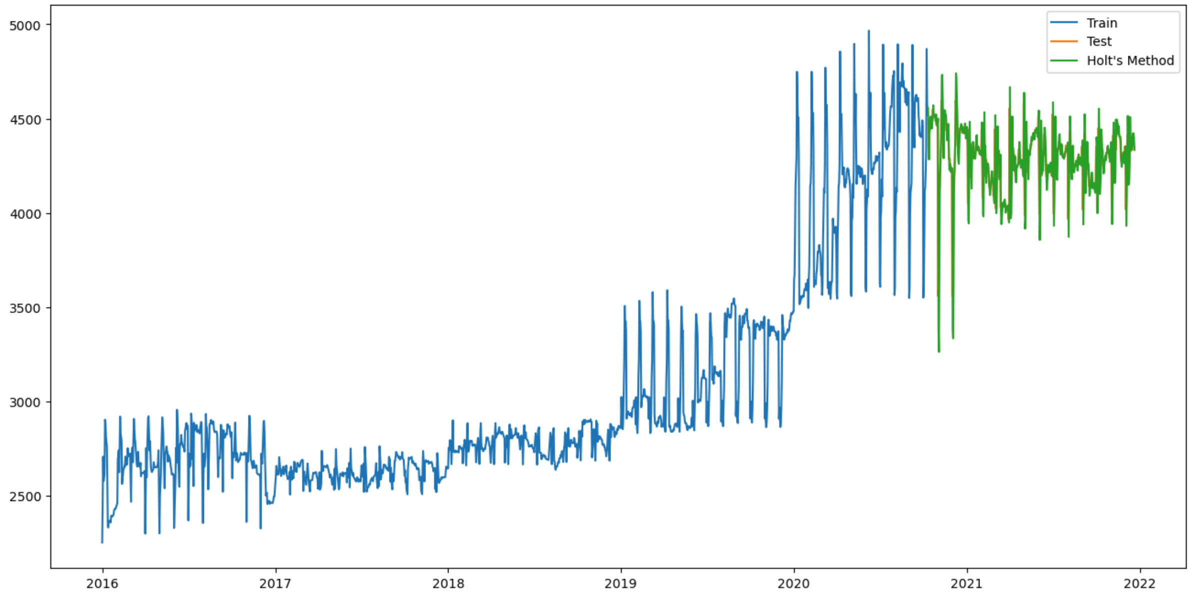
```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
```

```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
```

```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
Best Holt-Winters Parameters:
{'smoothing_level': 0.8, 'smoothing_trend': 0.5}
MAPE on Test Set: 9.708829569157277
```

In [303…

```
plt.figure(figsize=(16,8))
plt.plot(train["price"], label='Train')
plt.plot(test["price"], label='Test')
plt.plot(final_pred_hw["holt_method"], label="Holt's Method")
plt.legend(loc='best')
plt.show()
```



Holts winter exponential smoothing with mul seasonality and additive trend

In [304…

```
hwe_model_add_add = ExponentialSmoothing(train["price"],seasonal="multiplicative",t
pred_hwe_add_add = pd.DataFrame()
pred_hwe_add_add["holt_winter_method"] = hwe_model_add_add.predict(start = test.inc
MAPE(pred_hwe_add_add["holt_winter_method"],test.price)
```

```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
```

Out[304]: 8.84007545907398

In [274…

```
from statsmodels.tsa.holtwinters import ExponentialSmoothing
import numpy as np

# Grid search for Holt-Winters parameters
best_mape = float('inf')
best_params = None

for trend_type in ["add", "additive"]:
    for seasonal_type in ["mul", "multiplicative"]:
        for seasonal_periods in range(20, 50):
            hwe_model = ExponentialSmoothing(train["price"], seasonal=seasonal_type
            pred_hwe = hwe_model.predict(start=test.index[0], end=test.index[-1])
            mape = MAPE(pred_hwe, test["price"])

            if mape < best_mape:
                best_mape = mape
                best_params = {"seasonal": seasonal_type, "trend": trend_type, "sea

# Train the final Holt-Winters model with the best parameters on the entire dataset
final_hwe_model = ExponentialSmoothing(a1["price"], seasonal=best_params["seasonal"

# Make predictions for the test set
final_pred_hwe = final_hwe_model.predict(start=test.index[0], end=test.index[-1])

# Display the best parameters and MAPE
print("Best Holt-Winters Parameters:")
```

```
print(best_params)
print("MAPE on Test Set:", best_mape)
```

```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
```

```
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
```

```
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
```

```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
```

```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
```

```
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:915: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will
```

```
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
```

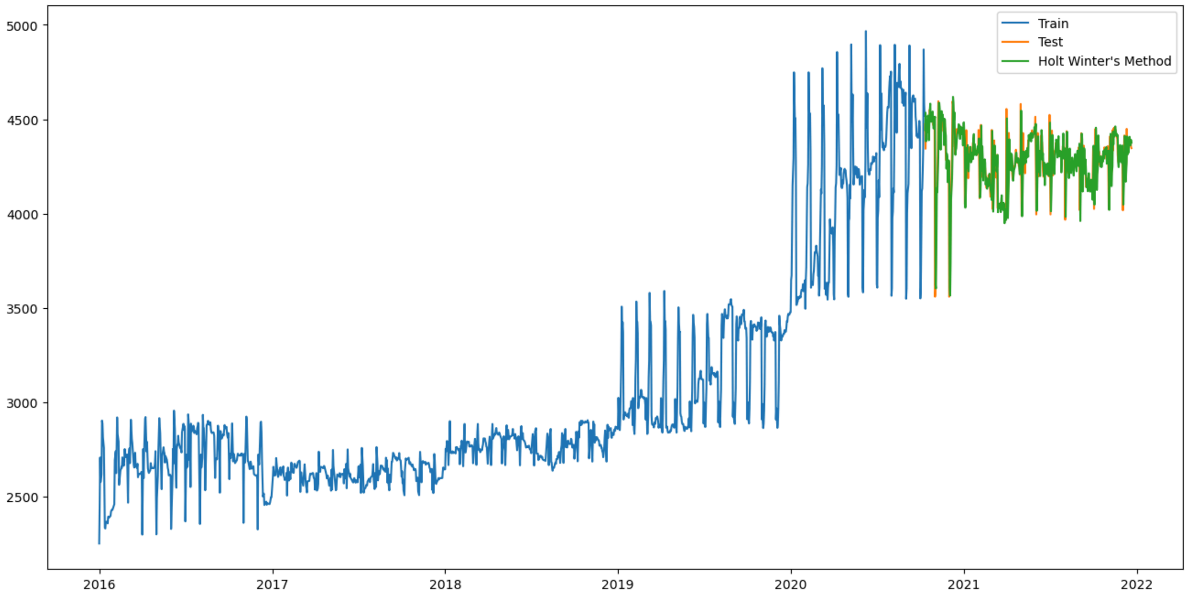
```
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
```

```
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:91
5: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.
  warnings.warn(
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471:
ValueWarning: No frequency information was provided, so inferred frequency D will
be used.
  self._init_dates(dates, freq)
```

```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
Best Holt-Winters Parameters:
{'seasonal': 'mul', 'trend': 'add', 'seasonal_periods': 40}
MAPE on Test Set: 3.033993393050866
```

In [275…

```python
plt.figure(figsize=(16,8))
plt.plot(train["price"], label='Train')
plt.plot(test["price"], label='Test')
plt.plot(final_pred_hwe, label="Holt Winter's Method")
plt.legend(loc='best')
plt.show()
```



Final Model by combining train and test

In [276…

```python
hwe_model_mul_add = ExponentialSmoothing(a1.price,seasonal="mul",trend="add",seasor
```

```
C:\Users\Tirum\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
  self._init_dates(dates, freq)
```

In [318…

```
#Forecasting for next 10 time periods
future_data = hwe_model_mul_add.forecast(30)
future_data
```

Out[318]:

```
2021-12-22    4339.968626
2021-12-23    4374.236842
2021-12-24    4359.347494
2021-12-25    4321.332185
2021-12-26    4318.717686
2021-12-27    4356.627527
2021-12-28    4350.802032
2021-12-29    4340.687340
2021-12-30    4351.899993
2021-12-31    4356.176977
2022-01-01    4361.604455
2022-01-02    4314.495772
2022-01-03    4286.813550
2022-01-04    4302.149821
2022-01-05    4301.363913
2022-01-06    4346.871053
2022-01-07    4333.570138
2022-01-08    4341.496298
2022-01-09    4336.990710
2022-01-10    4332.381512
2022-01-11    4334.956932
2022-01-12    4306.408130
2022-01-13    4340.659525
2022-01-14    4331.853704
2022-01-15    4327.675460
2022-01-16    4322.722809
2022-01-17    4348.699085
2022-01-18    4334.029648
2022-01-19    4341.674099
2022-01-20    4363.689064
Freq: D, dtype: float64
```

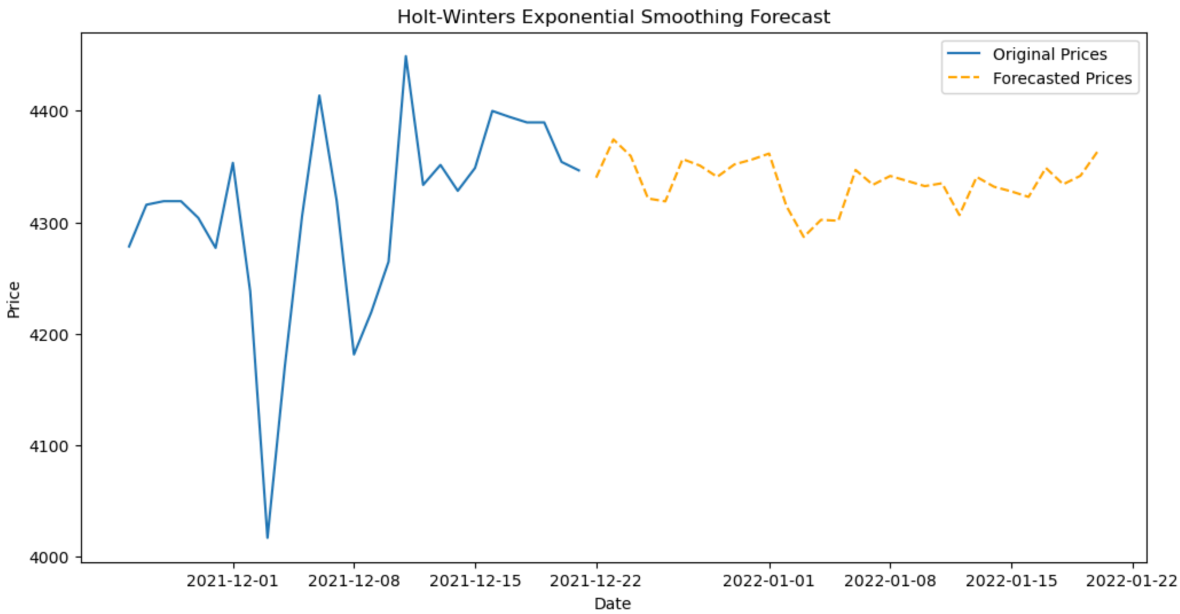
In [320…

```
a1['price'].iloc[2155:]
```

```
Out[320]: date
2021-11-25    4278.3
2021-11-26    4315.7
2021-11-27    4318.9
2021-11-28    4318.9
2021-11-29    4303.9
2021-11-30    4277.0
2021-12-01    4353.2
2021-12-02    4238.1
2021-12-03    4017.1
2021-12-04    4170.4
2021-12-05    4305.5
2021-12-06    4413.6
2021-12-07    4319.4
2021-12-08    4181.5
2021-12-09    4219.4
2021-12-10    4264.7
2021-12-11    4448.9
2021-12-12    4333.5
2021-12-13    4351.4
2021-12-14    4328.2
2021-12-15    4348.7
2021-12-16    4399.8
2021-12-17    4394.4
2021-12-18    4389.5
2021-12-19    4389.5
2021-12-20    4354.1
2021-12-21    4346.5
Name: price, dtype: float64
```

In [328…

```python
import matplotlib.pyplot as plt

# Forecast the next 30 time periods
future_data = hwe_model_mul_add.forecast(steps=30)

# Plotting the original data and the forecasted values
plt.figure(figsize=(12, 6))
plt.plot(a1.index[2155:], a1['price'].iloc[2155:], label='Original Prices')
plt.plot(future_data.index, future_data, label='Forecasted Prices', linestyle='dash
plt.title('Holt-Winters Exponential Smoothing Forecast')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```



In [ ]: