

BAN 210: Workshop 3

In [2]:

```
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
import seaborn as sns
```

Part I: Simple Linear Regression

In [4]:

```
# Import the data
Advertising =pd.read_csv("Advertising.csv")
Advertising.head()
```

Out[4]:

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

In [5]:

```
# Explore the data: data type, null count
Advertising.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Unnamed: 0    200 non-null    int64  
 1   TV           200 non-null    float64 
 2   radio         200 non-null    float64 
 3   newspaper     200 non-null    float64 
 4   sales         200 non-null    float64 
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
```

```
In [6]: # Explore the data: average, min, max, statistics number
Advertising.describe()
```

```
Out[6]:
```

	Unnamed: 0	TV	radio	newspaper	sales
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500
std	57.879185	85.854236	14.846809	21.778621	5.217457
min	1.000000	0.700000	0.000000	0.300000	1.600000
25%	50.750000	74.375000	9.975000	12.750000	10.375000
50%	100.500000	149.750000	22.900000	25.750000	12.900000
75%	150.250000	218.825000	36.525000	45.100000	17.400000
max	200.000000	296.400000	49.600000	114.000000	27.000000

```
In [7]: Advertising.drop('Unnamed: 0', axis=1, inplace=True)
Advertising.head()
```

```
Out[7]:
```

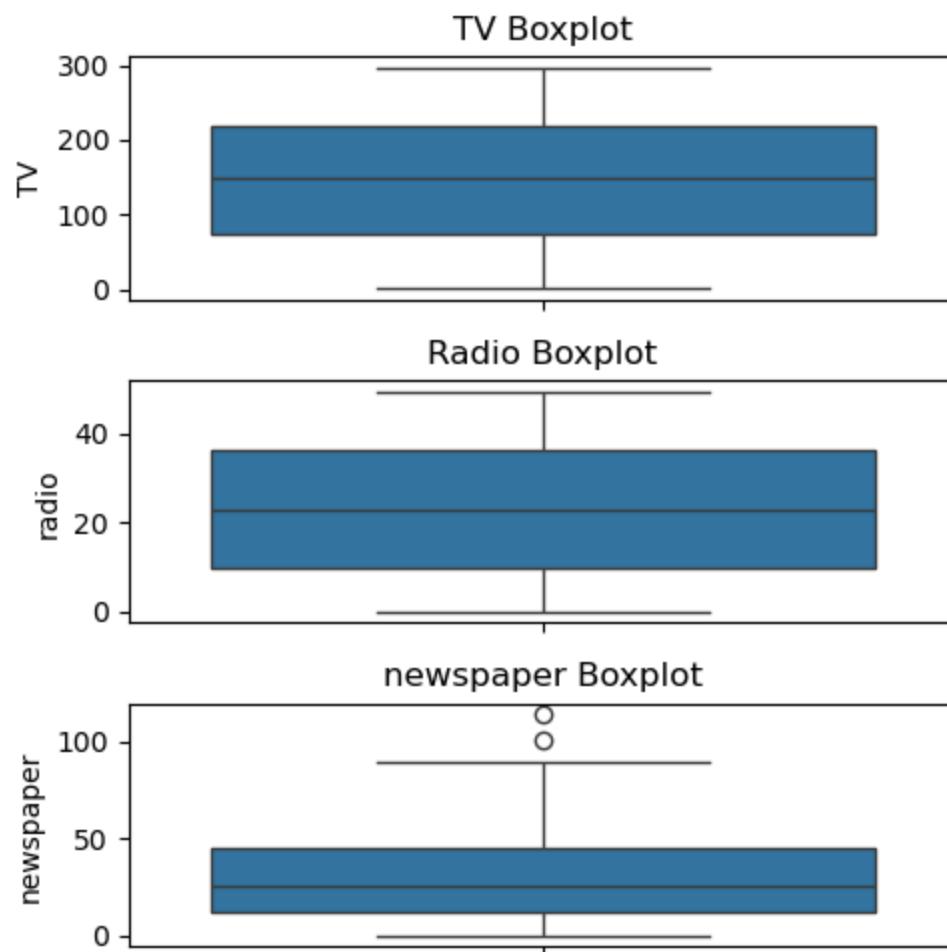
	TV	radio	newspaper	sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

Data Cleaning

```
In [9]: # Checking for Null Values: There is no null here in dataset  
Advertising.isnull().sum()*100/Advertising.shape[0]
```

```
Out[9]: TV      0.0  
radio    0.0  
newspaper 0.0  
sales    0.0  
dtype: float64
```

```
In [10]: # Outlier Analysis  
fig, axs = plt.subplots(3, figsize = (5,5))  
plt1 = sns.boxplot(Advertising['TV'], ax = axs[0])  
plt1.set_title('TV Boxplot')  
plt2 = sns.boxplot(Advertising['radio'], ax = axs[1])  
plt2.set_title('Radio Boxplot')  
plt3 = sns.boxplot(Advertising['newspaper'], ax = axs[2])  
plt3.set_title('newspaper Boxplot')  
plt.tight_layout()  
  
plt.show()
```



There are no considerable outliers present in the data

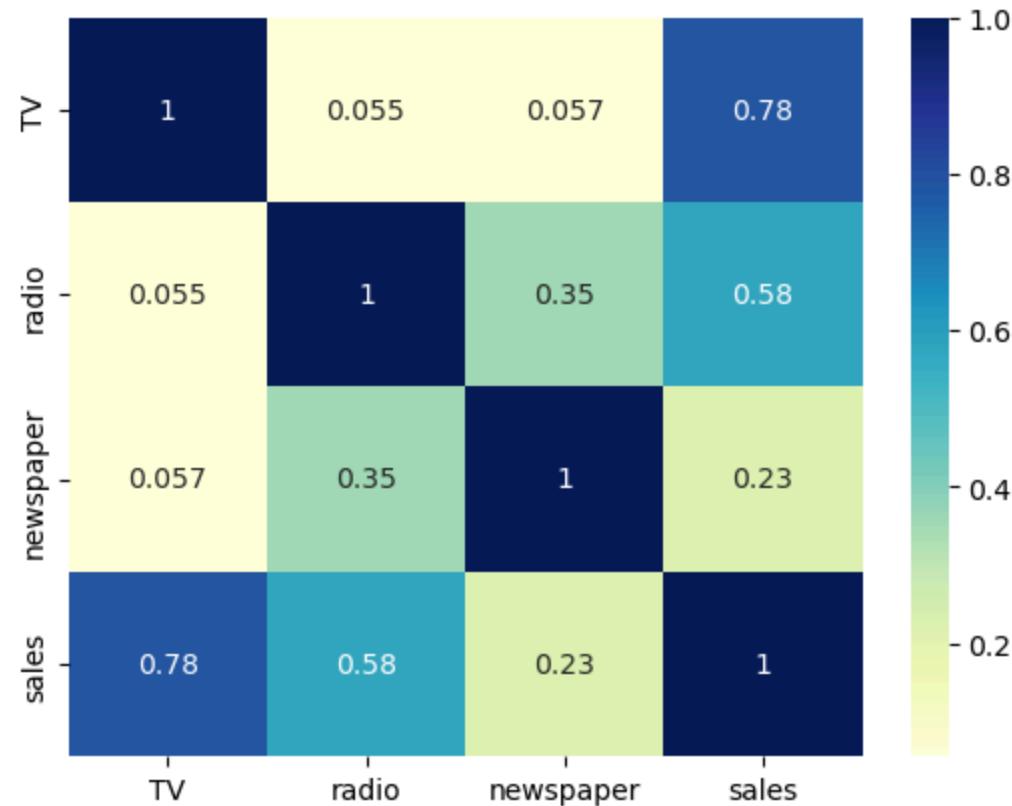
```
In [12]: #Checking the correlation between variables

correlation_tv = Advertising['TV'].corr(Advertising['sales'])
correlation_radio = Advertising['radio'].corr(Advertising['sales'])
correlation_newspaper = Advertising['newspaper'].corr(Advertising['sales'])

print(f'Correlation between TV and sales: {correlation_tv}')
print(f'Correlation between radio and sales: {correlation_radio}')
print(f'Correlation between newspaper and sales: {correlation_newspaper}')

sns.heatmap(Advertising.corr(), cmap="YlGnBu", annot = True)
plt.show()
```

Correlation between TV and sales: 0.7822244248616061
Correlation between radio and sales: 0.5762225745710551
Correlation between newspaper and sales: 0.22829902637616528



TV has the strongest correlation to sales and it is the best choice as feature variable for the simple linear regression.

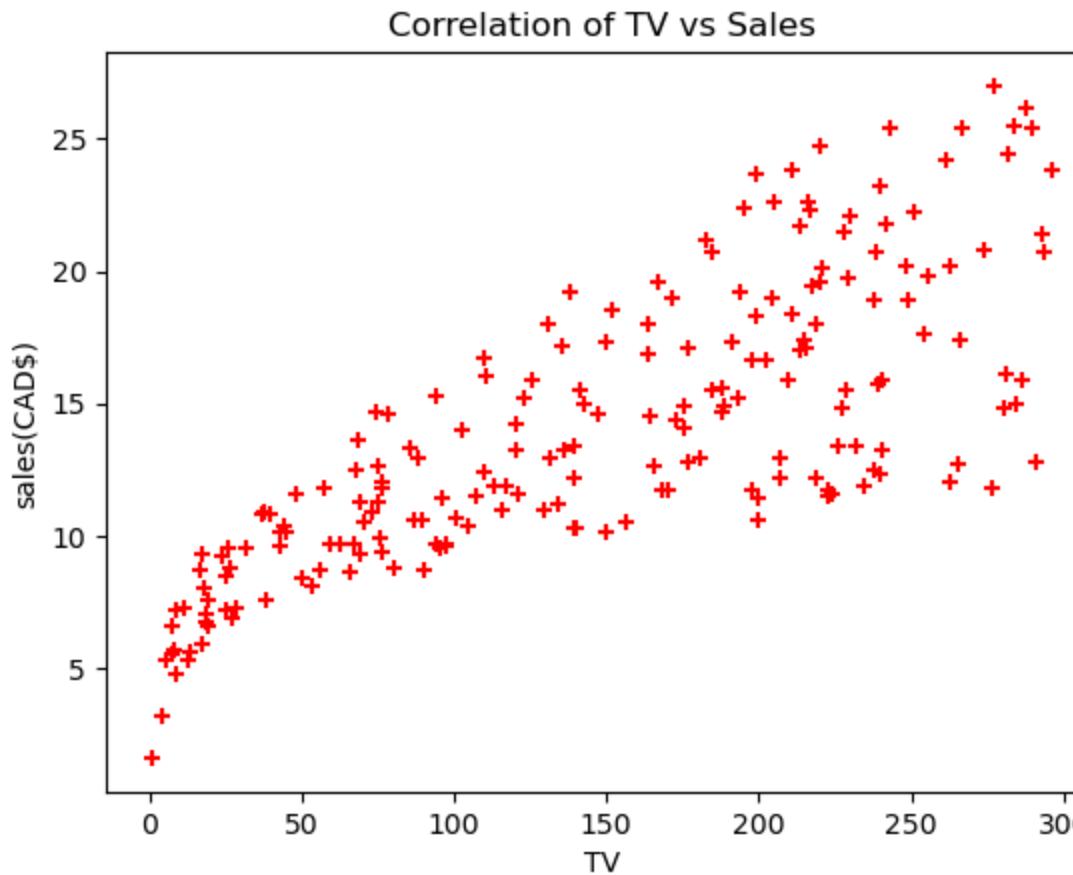
**Q1 : Is the distribution of X suitable for a linear regression model?
(Here: Variable X is "TV")**

In [15]: # Checking the correlation using scatterplot

```
%matplotlib inline
plt.title('Correlation of TV vs Sales')
plt.xlabel('TV')
```

```
plt.ylabel('sales(CAD$)')
plt.scatter(Advertising.TV, Advertising.sales, color='red', marker='+')

# Display the plot
plt.show()
```



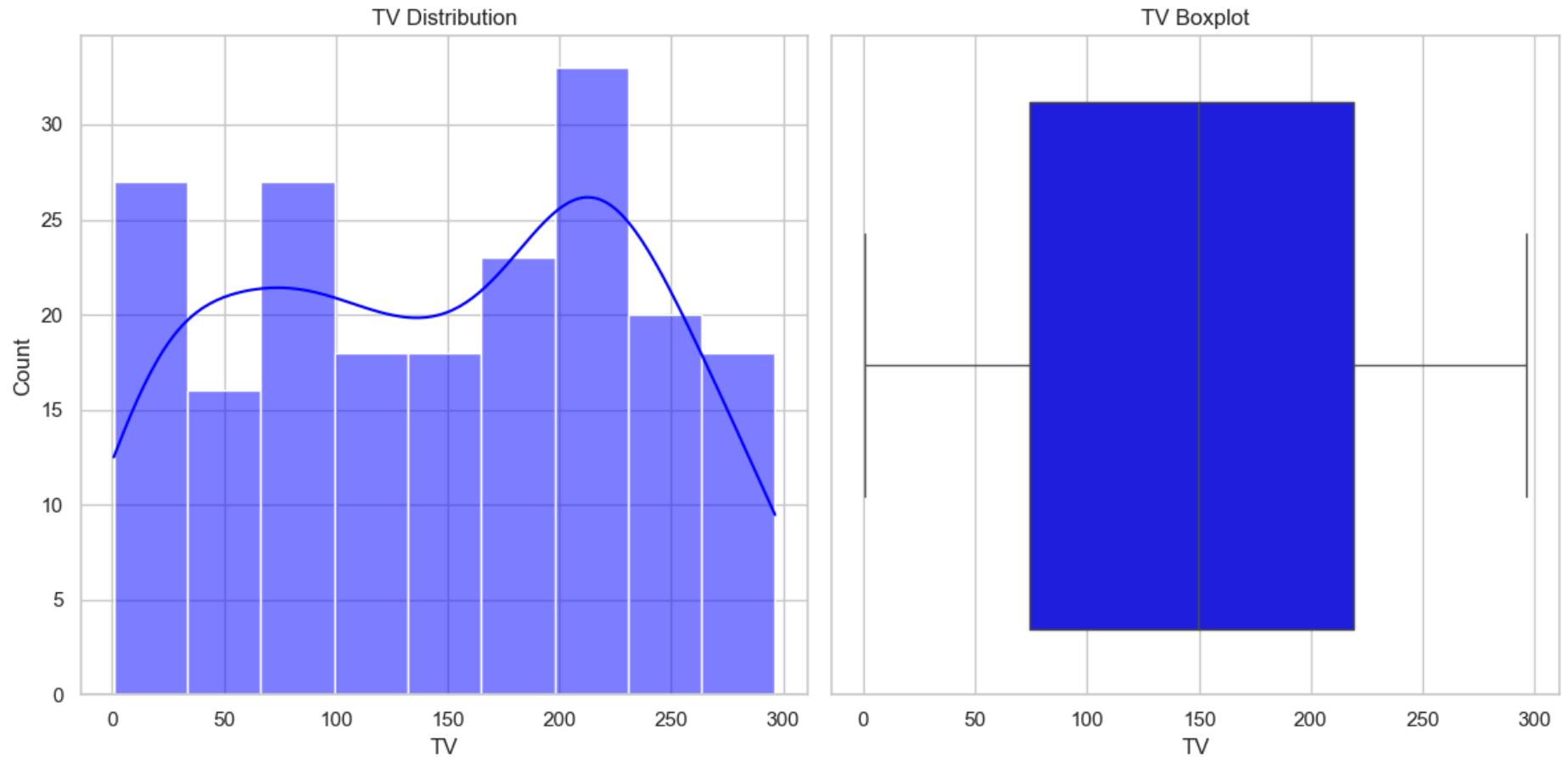
```
In [16]: # create frequency plot for X
# Set the style for seaborn
sns.set(style="whitegrid")

# Create a subplot for the histogram and boxplot of TV
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# TV Histogram with KDE
sns.histplot(Advertising['TV'], kde=True, ax=axes[0], color='blue')
axes[0].set_title('TV Distribution')

# Boxplot for TV
sns.boxplot(x=Advertising['TV'], ax=axes[1], color='blue')
axes[1].set_title('TV Boxplot')
```

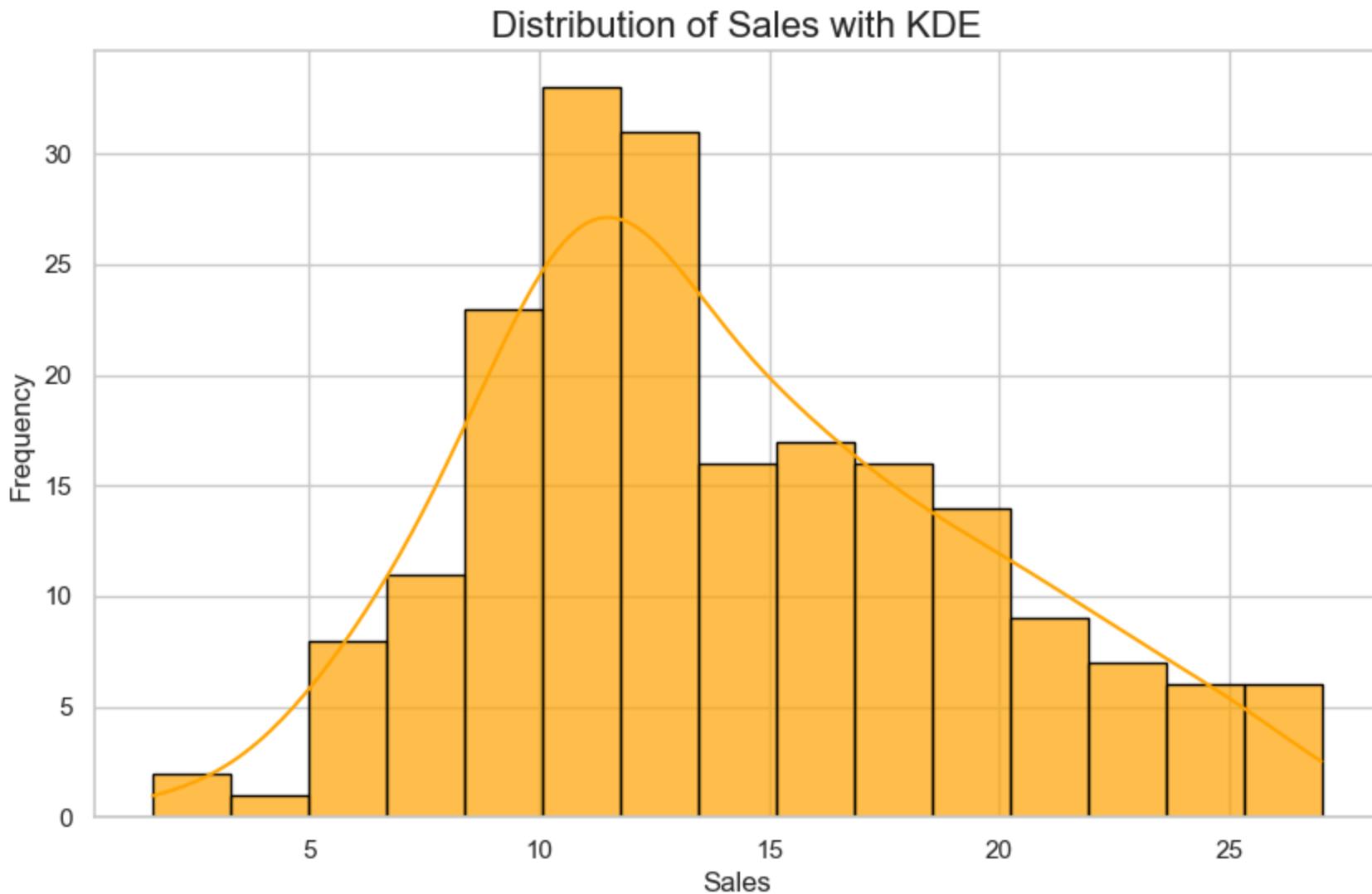
```
# Adjust layout to prevent overlap
plt.tight_layout()
plt.show()
```



```
In [17]: # create frequency plot for Y (sales)
# TV Histogram with KDE

# Set style
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
# Create histogram with KDE
sns.histplot(Advertising['sales'], bins=15, kde=True, color='orange', edgecolor='black', alpha=0.7)
# Add titles and labels
plt.title('Distribution of Sales with KDE', fontsize=16)
plt.xlabel('Sales', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
```

```
# Show the plot  
plt.show() #right skewed distribution
```



We chose TV as the variable for linear regression since it shows the strongest correlation to sales, which makes it the most important predictor in our dataset.

The distribution of TV is right-skewed which may not meet the ideal normality assumption for linear regression. Hence, we're going to do a log transformation on TV to help stabilize the variance and make the connection between TV and sales more linear. The log transformation is really

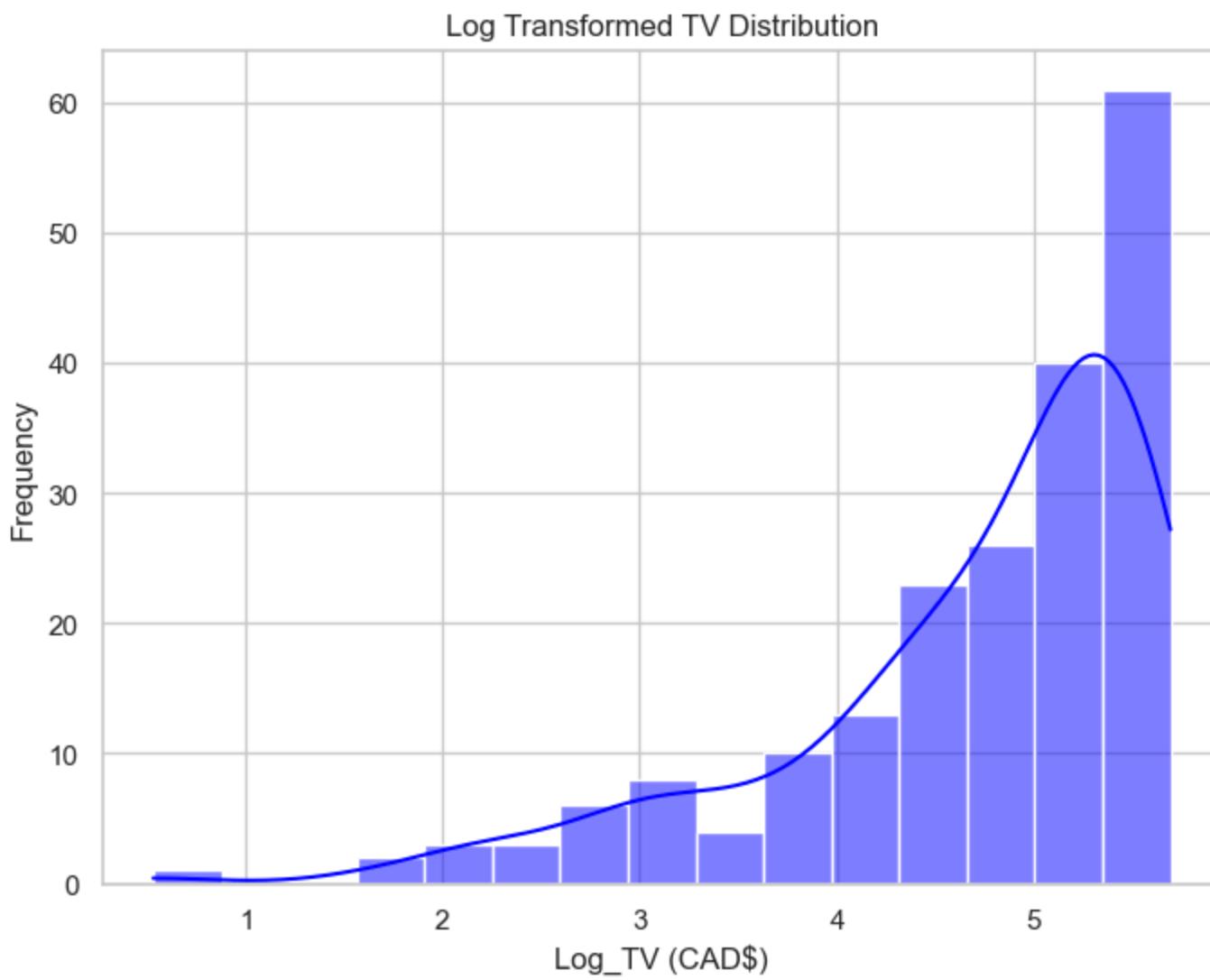
useful because it helps the model satisfy the assumptions of linear regression, which in turn makes the results more accurate and easier to understand.

```
In [19]: # Apply log transformation to the 'TV' column  
Advertising['log_TV'] = np.log(Advertising['TV'] + 1) # Adding 1 to handle zeros in the data  
  
# Checking the first few rows to confirm the transformation  
Advertising[['TV', 'log_TV']].head()
```

```
Out[19]:
```

	TV	log_TV
0	230.1	5.442851
1	44.5	3.817712
2	17.2	2.901422
3	151.5	5.027165
4	180.8	5.202907

```
In [20]: # Creating the histogram of the Log-transformed TV data  
plt.figure(figsize=(8, 6))  
sns.histplot(Advertising['log_TV'], kde=True, color='blue')  
plt.title('Log Transformed TV Distribution')  
plt.xlabel('Log_TV (CAD$)')  
plt.ylabel('Frequency')  
plt.show()
```



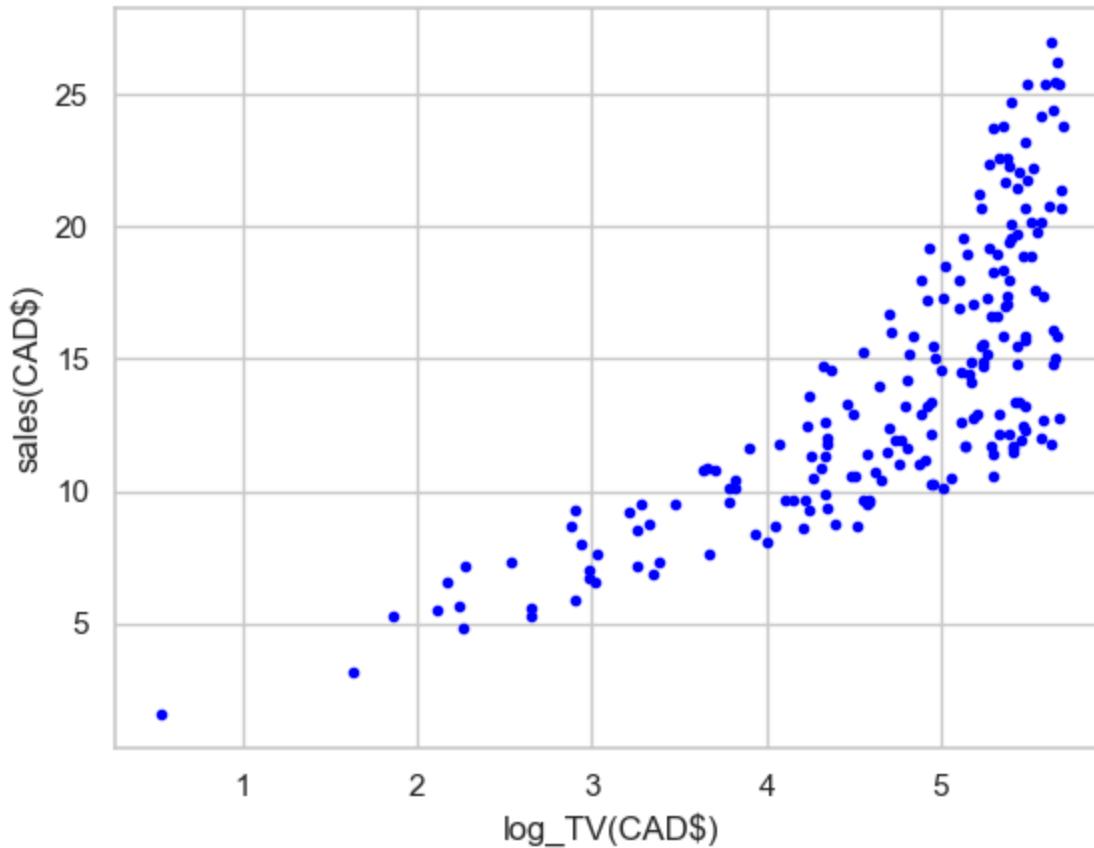
Q2: What is the coefficient for LOG_X? Q3: What is the intercept?

Model Building

Performing Simple Linear Regression

```
In [23]: # Performing the Scatterplot for Log_TV vs. Sales
```

```
%matplotlib inline
plt.xlabel('log_TV(CAD$)')
plt.ylabel('sales(CAD$)')
plt.scatter(Advertising.log_TV, Advertising.sales, color='blue', marker='.')
# Display the scatterplot
plt.show()
```



```
In [24]: # X as the independent variable (Log_TV) and Y (Sales) as dependent variable
```

```
X = Advertising[['log_TV']]
Y = Advertising['sales']

# Initialize the linear regression model
model = LinearRegression()

# Fit the model to the data
model.fit(X, Y)
```

```
# Get the intercept and coefficient
intercept = model.intercept_
coefficient = model.coef_[0]

print(f"Intercept: {intercept:.4f}")
print(f"Coefficient for Log_X (log_TV): {coefficient:.4f}")
```

Intercept: -5.2734
 Coefficient for Log_X (log_TV): 4.1117

Equation of linear regression $y=c+m_1x_1+m_2x_2+\dots+m_nx_n$

y is the response c is the intercept m_1 is the coefficient for the first feature m_n is the coefficient for the n th feature In our case:

$y=c+4.11\times\text{Log_TV}$

The m values are called the model coefficients or model parameters.

Linear Equation Model:

Sales = -5.273 + 4.11*Log_TV

Part II: Multivariate Regression

Step 2:

```
In [29]: # Applying Data Transformation to all predictors (Log(X) and Norm of X)

# Apply Log transformation to the predictors (TV, Radio, Newspaper)
Advertising['log_TV'] = np.log(Advertising['TV'] + 1)
Advertising['log_radio'] = np.log(Advertising['radio'] + 1)
Advertising['log_newspaper'] = np.log(Advertising['newspaper'] + 1)

# Normalizing the transformed predictors
scaler = StandardScaler()
Advertising[['log_TV', 'log_radio', 'log_newspaper']] = scaler.fit_transform(Advertising[['log_TV', 'log_radio', 'log_newspaper']])
```

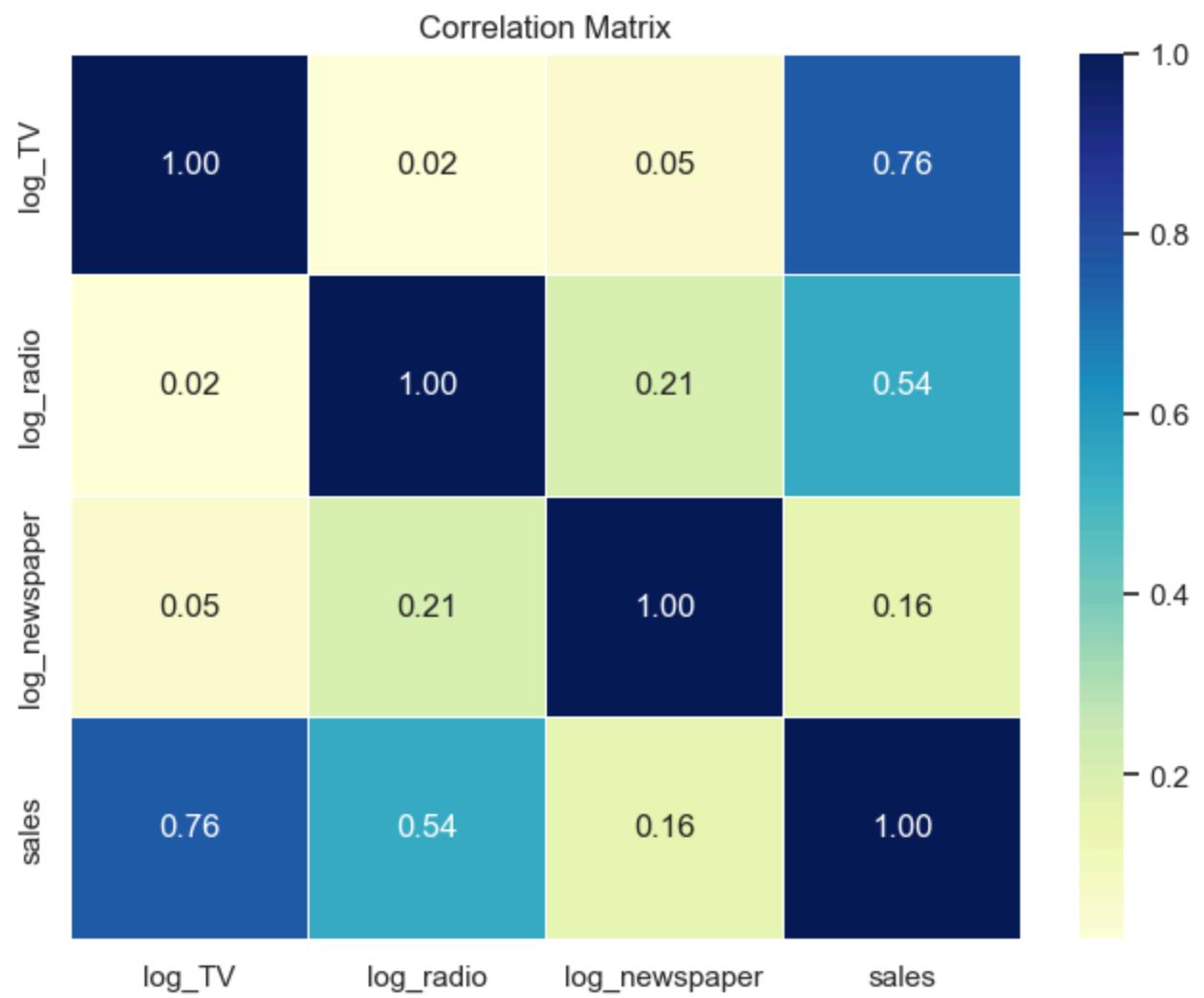
```
# Check the first few rows to confirm the transformation  
Advertising[['log_TV', 'log_radio', 'log_newspaper']].head()
```

Out[29]:

	log_TV	log_radio	log_newspaper
0	0.781107	0.850789	1.270548
1	-0.911701	0.892779	0.784008
2	-1.866146	1.060674	1.272195
3	0.348112	0.946397	1.079221
4	0.531172	-0.466897	1.077274

Step 3:

```
In [31]: # Correlation Matrix  
corr_matrix = Advertising[['log_TV', 'log_radio', 'log_newspaper', 'sales']].corr()  
  
# Heatmap Correlation  
plt.figure(figsize=(8, 6))  
sns.heatmap(corr_matrix, annot=True, cmap='YlGnBu', fmt=".2f", linewidths=0.5)  
plt.title("Correlation Matrix")  
plt.show()
```



The correlation heatmap indicates that log_TV exhibits the most positive association with sales. This indicates that television advertisements exert the most significant influence on sales. A fairly significant correlation exists between log_radio and sales, indicating that radio advertising influences sales, albeit to a lesser extent than television advertising. The weakest positive correlation is shown in log_newspaper, indicating that newspaper advertisements exert minimal influence on sales. Overall, these findings indicate that television advertisements exert the greatest influence on sales. Radio and newspaper advertisements yield modest yet favorable impacts.

In [33]:

```
# Scatter plots for all predictors vs Sales as Target Variable
fig, axes = plt.subplots(1, 3, figsize=(18, 6))

# Scatter plot; Log_TV vs sales
sns.scatterplot(x=Advertising['log_TV'], y=Advertising['sales'], ax=axes[0], color='red', marker='+')
axes[0].set_title('log_TV vs Sales')

# Scatter plot for log_radio vs sales
```

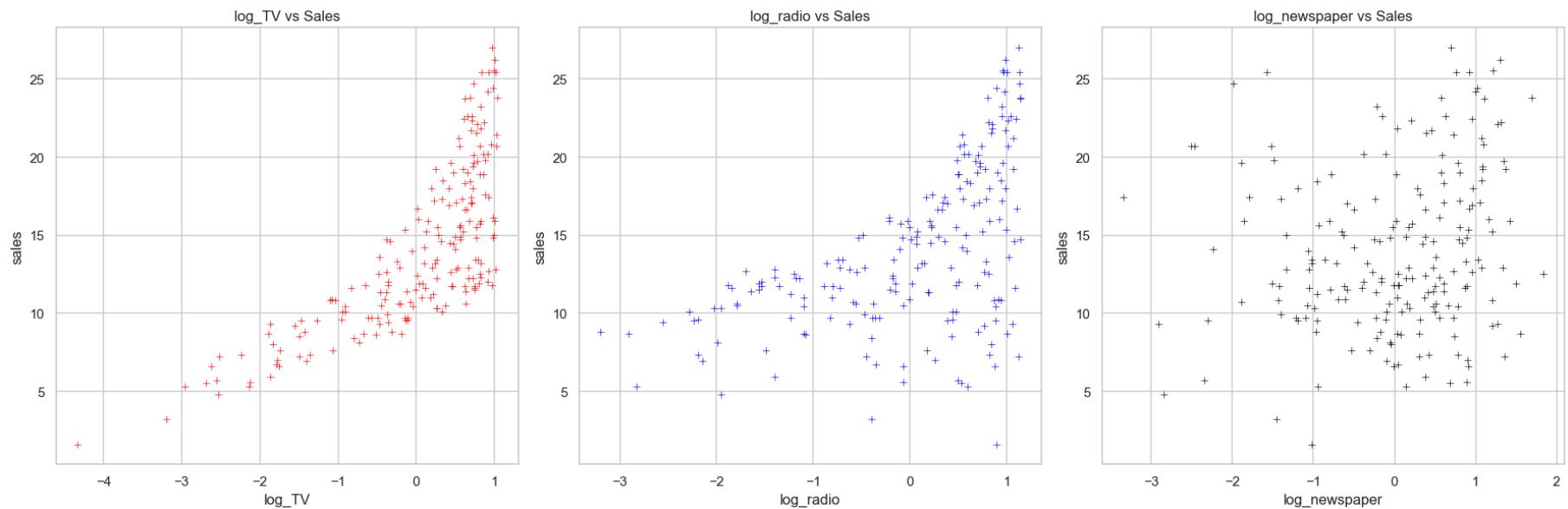
```

sns.scatterplot(x=Advertising['log_radio'], y=Advertising['sales'], ax=axes[1], color='blue', marker='+')
axes[1].set_title('log_radio vs Sales')

# Scatter plot for log_newspaper vs sales
sns.scatterplot(x=Advertising['log_newspaper'], y=Advertising['sales'], ax=axes[2], color='black', marker='+')
axes[2].set_title('log_newspaper vs Sales')

plt.tight_layout()
plt.show()

```



Model Building

Multivariate Regression Model

```

In [35]: # X as the independent variable (log_TV) and Y (Sales) as dependent variable
X1 = Advertising[['log_TV', 'log_radio', 'log_newspaper']]
Y1 = Advertising['sales']

# Initialize the Linear regression model
model_mrm = LinearRegression()

# Fit the model to the data
model_mrm.fit(X1, Y1)

# Get the intercept and coefficient
intercept = model_mrm.intercept_
coefficient = model_mrm.coef_

```

```
print(f"Intercept: {intercept:.4f}")
print(f"Coefficient for Log_X1 (log_TV): {coefficient[0]:.4f}")
print(f"Coefficient for Log_X2 (log_radio): {coefficient[1]:.4f}")
print(f"Coefficient for Log_X3 (log_newspaper): {coefficient[2]:.4f}")
```

```
Intercept: 14.0225
Coefficient for Log_X1 (log_TV): 3.8915
Coefficient for Log_X2 (log_radio): 2.6953
Coefficient for Log_X3 (log_newspaper): 0.1000
```

Regression Equation Model:

$$\text{Sales} = 14.0225 + 3.8915 \text{Log_TV} + 2.6953 \text{Log_radio} + 0.1000 * \text{Log_newspaper}$$

Part III: Regression assessment

```
In [38]: # Predicting using the simple linear regression model
simple_model = LinearRegression()
simple_model.fit(Advertising[['log_TV']], Advertising['sales'])
simple_y_pred = simple_model.predict(Advertising[['log_TV']])

# Predicting using the multivariate regression model
mrm_y_pred = model_mrm.predict(X1)

# Computing the R-squared and MSE for both models
simple_r2 = r2_score(Y, simple_y_pred)
simple_mse = mean_squared_error(Y1, simple_y_pred)

mrm_r2 = r2_score(Y, mrm_y_pred)
mrm_mse = mean_squared_error(Y1, mrm_y_pred)

# Displaying the results

Table = pd.DataFrame({
    'Model': ['Simple Linear Regression', 'Multivariate Linear Regression'],
    'R-squared': [simple_r2, mrm_r2],
    'Mean Squared Error': [simple_mse, mrm_mse]
})
```

```
# Print the table
```

```
Table
```

```
Out[38]:
```

	Model	R-squared	Mean Squared Error
0	Simple Linear Regression	0.575251	11.504629
1	Multivariate Linear Regression	0.847862	4.120757

Q4: What do the above numbers mean? Is it fair to compare them if they are using different data?

The Multivariate Linear Regression model exhibits greater performance, evidenced by a higher R-squared and a lower Mean Squared Error (MSE) relative to the Simple Linear Regression model. This signifies that the multivariate model accounts for greater volatility in the data and yields more precise predictions, indicating a superior fit.

It is essential to recognize that a comparison of the two models is fair only if they are assessed using the identical dataset. To facilitate a valid comparison, both models must be evaluated under identical settings. Assessing models across different datasets may result in skewed comparisons and erroneous results.

Q5: Does a better fit on training data result in a better fit on validation or test data?

An improved fit on the training data does not always guarantee a superior fit on validation or test data. A model may overfit the training data, resulting in inadequate generalization to unseen data. Evaluating models using test data is crucial for determining how applicable they are.

Part IV: Time Series

```
In [44]: # Import the data
```

```
retail = pd.read_csv("retail_sales.csv")
retail.head()
```

```
Out[44]:
```

	month	naics_code	kind_of_business	value
0	2024-12-01	no code	Retail sales, total	702255.0
1	2024-06-01	no code	GAFO(1)	129215.0
2	2024-01-01	no code	GAFO(1)	111547.0
3	2023-12-01	451211	Book stores	1073.0
4	2023-07-01	no code	GAFO(1)	128320.0

```
In [45]: #Summarize and visualize the data
```

```
retail.info()          # Column types & non-null counts
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25740 entries, 0 to 25739
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   month            25740 non-null   object 
 1   naics_code        25739 non-null   object 
 2   kind_of_business  25740 non-null   object 
 3   value             24819 non-null   float64
dtypes: float64(1), object(3)
memory usage: 804.5+ KB
```

```
In [46]: retail.describe()      # Summary stats for numeric columns
```

```
Out[46]:
```

	value
count	24819.000000
mean	47282.821024
std	99366.590580
min	12.000000
25%	2668.000000
50%	9566.000000
75%	38949.500000
max	799769.000000

```
In [47]: retail.isnull().sum() # Missing values
```

```
Out[47]: month          0  
naics_code       1  
kind_of_business 0  
value           921  
dtype: int64
```

Q6: How many products are in this dataset? Which product has the highest frequency?

```
In [49]: # Count the frequency of each product in the 'kind_of_business' column  
product_counts = retail['kind_of_business'].value_counts()
```

```
# Total number of unique products  
product_totals = product_counts.shape[0]
```

```
# Product with the highest frequency and its count  
product_most_frequent = product_counts.idxmax()  
count_most_frequent = product_counts.max()
```

```
# Print the results  
print(f"Amount of Products in the Dataset: {product_totals}")  
print(f"Product with the highest frequency: {product_most_frequent} ({count_most_frequent} occurrences)")
```

Amount of Products in the Dataset: 65

Product with the highest frequency: Automobile dealers (408 occurrences)

Q7:

Looking at Time Series Summary graphs for each Business, how are the sales distributed among each Business value? What is the range of values assigned to Time Series ID? Based on the possible values for the three Business ID variables, is this what you expected?

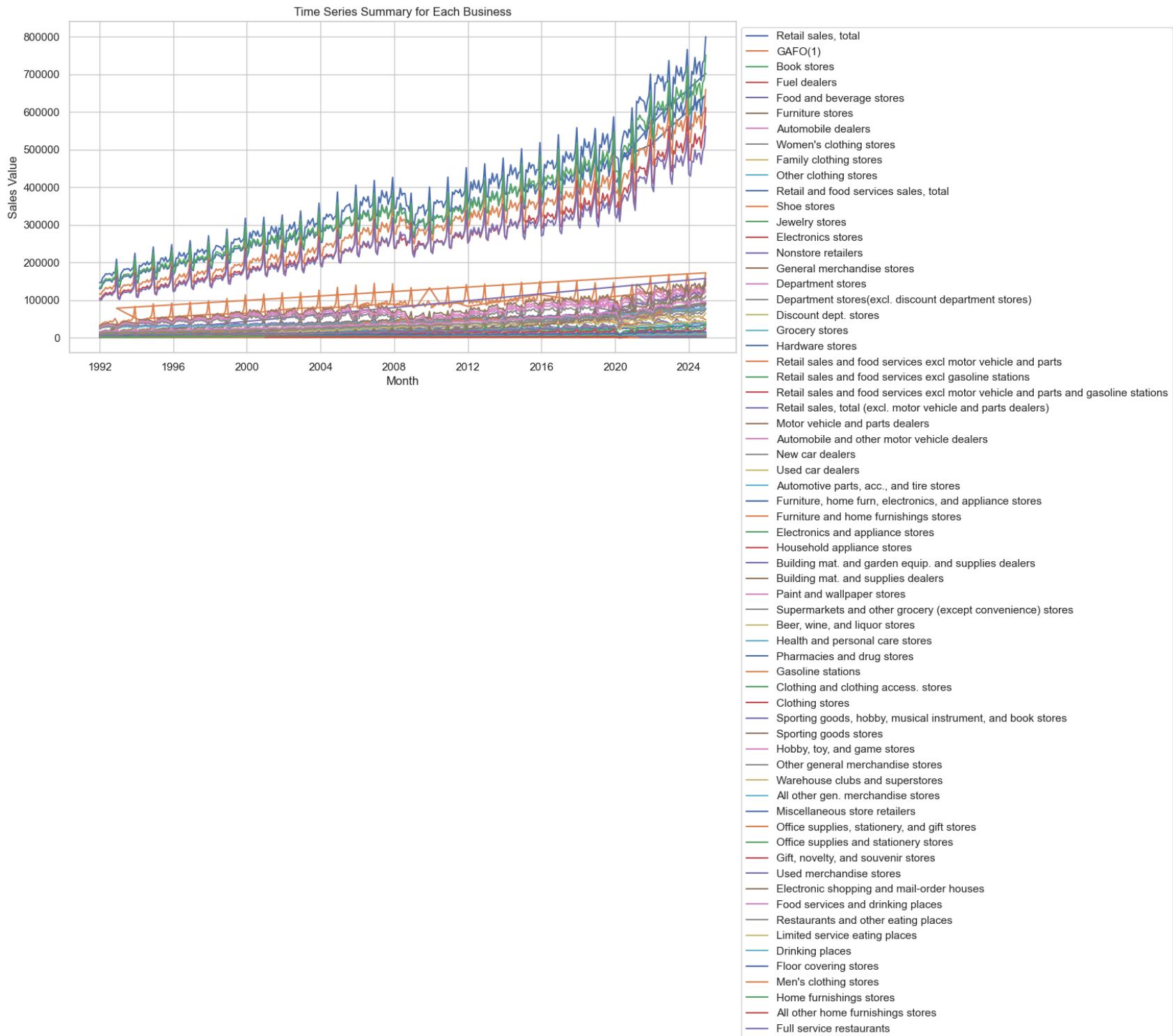
```
In [51]: # Generate Time Series Summary graphs for each Business
# Convert 'month' column to datetime format
retail['month'] = pd.to_datetime(retail['month'])

# Get unique businesses
unique_businesses = retail['kind_of_business'].unique()

# Plot for each business
plt.figure(figsize=(12, 6))
for business in unique_businesses:
    business_data = retail[retail['kind_of_business'] == business]
    plt.plot(business_data['month'], business_data['value'], label=business)

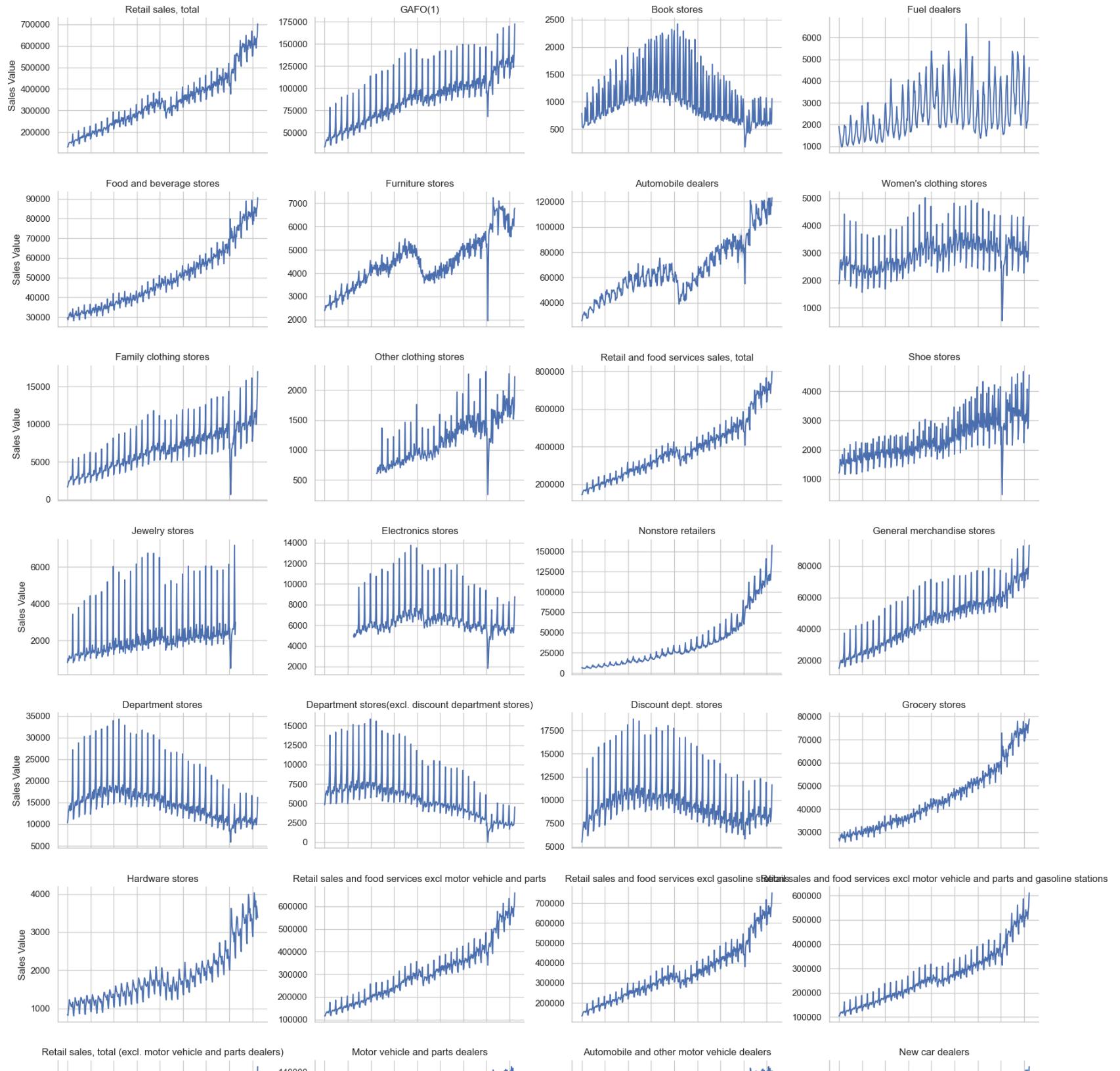
plt.title('Time Series Summary for Each Business')
plt.xlabel('Month')
plt.ylabel('Sales Value')
plt.legend(loc='upper left', bbox_to_anchor=(1, 1)) # Move Legend outside the plot

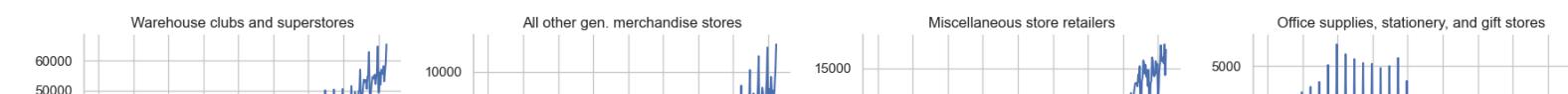
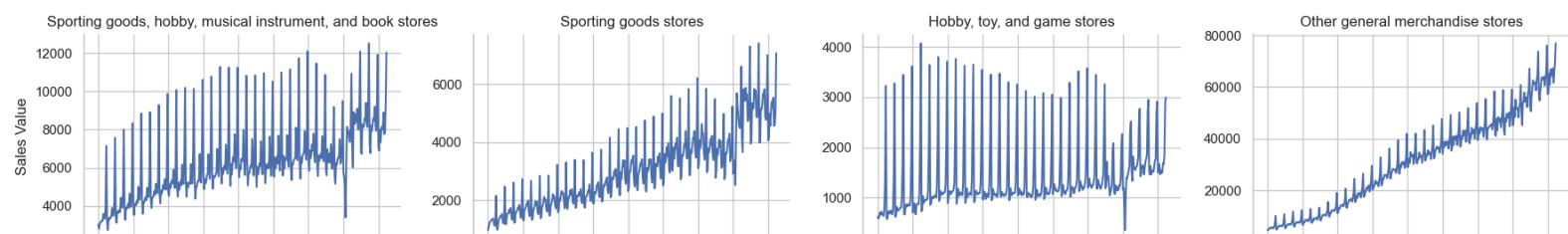
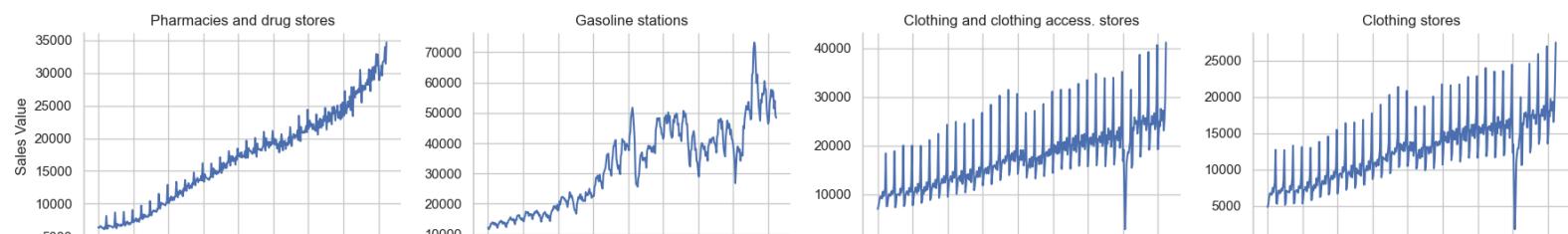
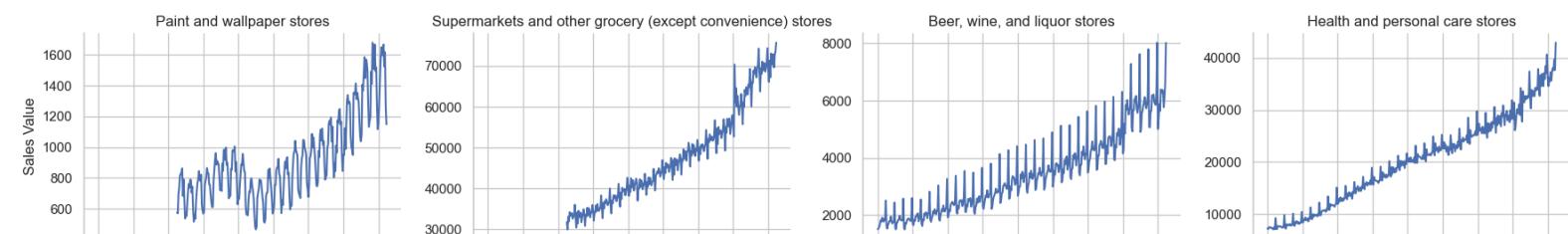
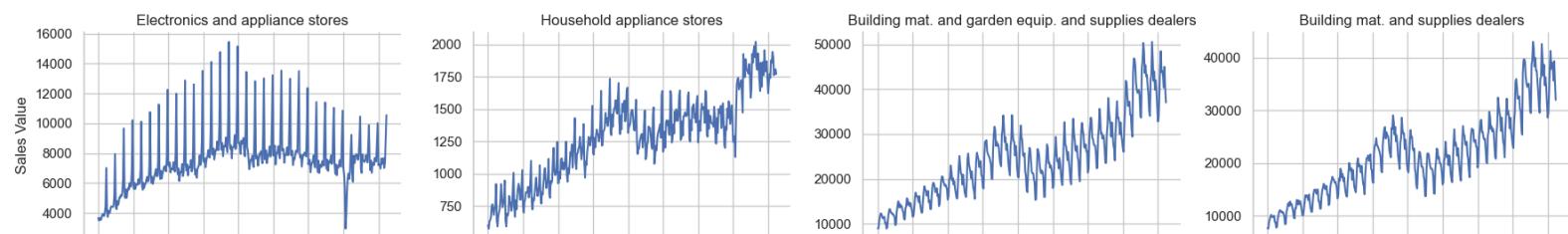
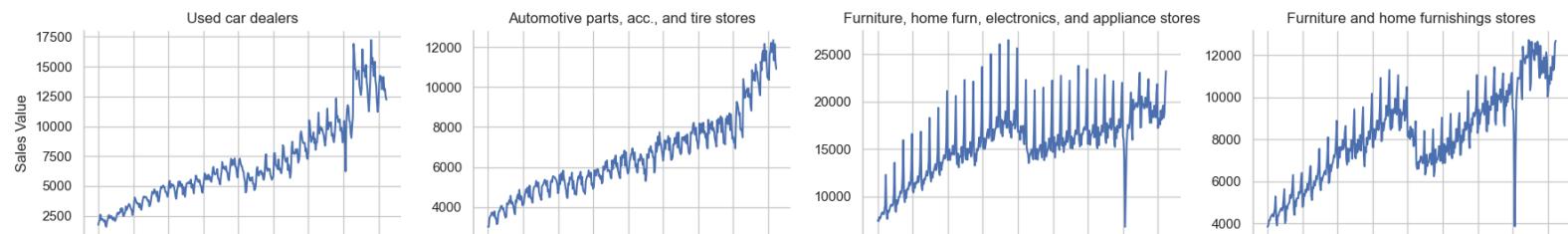
plt.show()
```

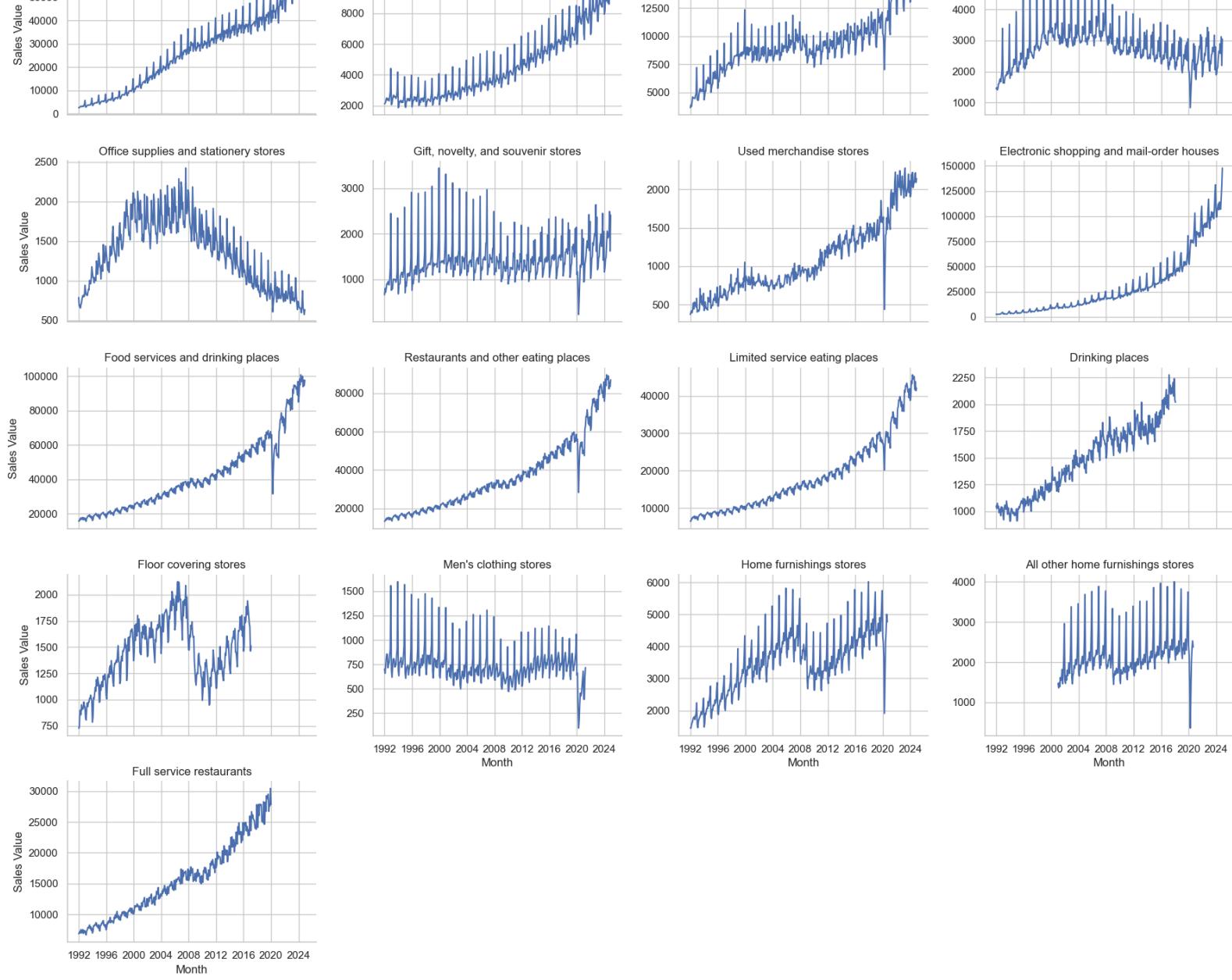


```
In [52]: # Create a faceted grid of time series plots
g = sns.FacetGrid(retail, col="kind_of_business", col_wrap=4, height=3, aspect=1.5, sharey=False)
```

```
g.map(sns.lineplot, "month", "value")
g.set_titles("{col_name}")
g.set_axis_labels("Month", "Sales Value")
g.tight_layout()
plt.show()
```







Sales are distributed differently amongst different business values.

Some business lines show more peaks and dips which can mean more variability in sales which can be attributed to factors such as seasonality. For example, "automobile and other motor vehicle dealers". This differs from lines that are more flat indicating more stable sales as seen in the graph for "All other home furnishing stores".

Also, the range of sales also differs between business values. Some business values have a maximum of 2000 dollars in sales values.

For example "floor covering stores". Whereas, the business of "food services and drinking places" has sales up to 100,000 dollars.

```
In [54]: # What is the range of values assigned to Time Series ID? Based on the possible values for the three Business ID var  
retail.columns
```

```
Out[54]: Index(['month', 'naics_code', 'kind_of_business', 'value'], dtype='object')
```

```
In [55]: from datetime import datetime  
  
start_date = retail['month'].min()  
end_date = retail['month'].max()  
print(f"The earliest date (start date) in the dataset is: {start_date.strftime('%Y-%m-%d')}")  
print(f"The latest date (end date) in the dataset is: {end_date.strftime('%Y-%m-%d')}")  
  
number_of_years = (end_date - start_date).days / 365.25 # approximation using 365.25 days per year to account for Le  
print(f"The dataset covers a period of approximately {number_of_years:.2f} years.")
```

The earliest date (start date) in the dataset is: 1992-01-01

The latest date (end date) in the dataset is: 2024-12-01

The dataset covers a period of approximately 32.92 years.

```
In [56]: # Clean 'naics_code': remove NaN, convert to string, and strip whitespace  
clean_codes = retail['naics_code'].dropna().astype(str).str.strip()  
  
# Get unique and sorted Time Series IDs  
unique_codes = clean_codes.unique()  
sorted_codes = sorted(unique_codes)  
  
# Print results  
print("Number of unique Time Series IDs (naics_code):", len(unique_codes))  
print("Sorted Time Series IDs:", sorted_codes)
```

Number of unique Time Series IDs (naics_code): 60

Sorted Time Series IDs: ['44,114,412', '441', '4411', '44111', '44112', '4413', '442', '442,443', '4421', '4422', '44221', '442299', '443', '443141', '443142', '444', '4441', '44412', '44413', '445', '4451', '44511', '4453', '446', '44611', '447', '448', '4481', '44811', '44812', '44814', '44819', '4482', '44831', '451', '45111', '45112', '451211', '452', '4521', '452111', '452112', '4529', '45291', '45299', '453', '4532', '45321', '45322', '45330', '454', '4541', '45431', '722', '7224', '7225', '722511', '722513', '722514', '722515', 'no code', 'no code']

The range of values assigned to Time Series ID consists of 60 unique IDs. It ranged from short IDs such as "443" to longer IDs including "44112" and also includes "no code" as an ID. Based on the three Business ID variables this is mostly what the group

expected. Each naics_code is directly associated with a kind_of_business value. The group expected numeric codes as seen in the naics_code but did not expect to see "no code" so that came as a surprise.

naics_code: The presence of detailed NAICS codes, including various levels of granularity (e.g., '44819', '451211', 'no code'), is expected in a comprehensive retail sales dataset. This allows for categorization and analysis at different industry levels.

kind_of_business: The detailed textual descriptions for kind_of_business are also expected. They provide a human-readable understanding of the business types, complementing the numerical NAICS codes. The variety of business types, from broad categories like 'Retail sales, total' to specific ones like 'Book stores' or 'Pharmacies and drug stores', reflects the diverse nature of the retail sector.

Q8:

Run your code again and see the results. Paste the Target Time Series Plot (or Multiple Time Series Comparison Plot). How many TSIDs are assigned now?

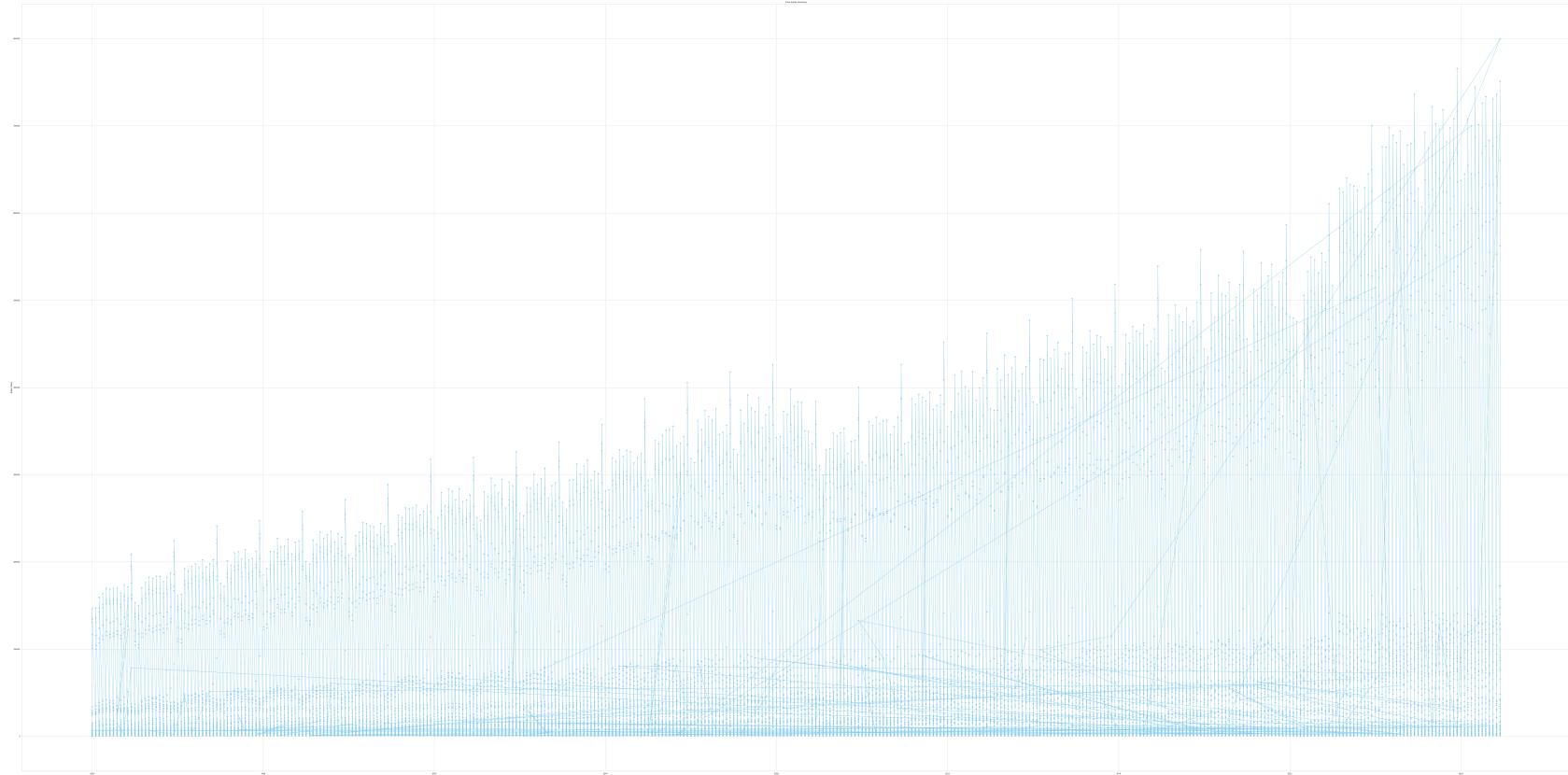
```
In [60]: # Step 2: Remove naics_code, kind_of_business  
retail1 = retail.drop(['naics_code', 'kind_of_business'], axis=1)  
retail1.head()
```

```
Out[60]:
```

	month	value
0	2024-12-01	702255.0
1	2024-06-01	129215.0
2	2024-01-01	111547.0
3	2023-12-01	1073.0
4	2023-07-01	128320.0

```
In [61]: # Run the code again: Generate Time Series Summary  
plt.figure(figsize=(120, 60))  
plt.plot(retail1['month'], retail1['value'], marker='o', linestyle='-', color='skyblue')  
  
plt.title('Time Series Summary')  
plt.xlabel('Month')  
plt.ylabel('Sales Value')
```

```
plt.grid(True)
plt.tight_layout()
plt.show()
print(f"Number of TSIDs: {len( retail1.month.unique())}")
```



Number of TSIDs: 396

There are 396 Time Series IDs assigned, representing the distinct reported months with their business values, after removing naics_code and kind_of_business.

In [63]: pip install tslearn

Requirement already satisfied: tslearn in c:\users\admin\anaconda3\lib\site-packages (0.6.3)
Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: numpy in c:\users\admin\anaconda3\lib\site-packages (from tslearn) (1.26.4)
Requirement already satisfied: scipy in c:\users\admin\anaconda3\lib\site-packages (from tslearn) (1.13.1)
Requirement already satisfied: scikit-learn in c:\users\admin\anaconda3\lib\site-packages (from tslearn) (1.5.1)
Requirement already satisfied: numba in c:\users\admin\anaconda3\lib\site-packages (from tslearn) (0.60.0)
Requirement already satisfied: joblib in c:\users\admin\anaconda3\lib\site-packages (from tslearn) (1.4.2)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in c:\users\admin\anaconda3\lib\site-packages (from numba->tslearn) (0.43.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\admin\anaconda3\lib\site-packages (from scikit-learn->tslearn) (3.5.0)

```
In [64]: # Convert month to datetime for sorting
retail1['month'] = pd.to_datetime(retail1['month'])

# Group values by month
grouped = retail1.groupby('month')['value'].apply(list)

# Convert to DataFrame where each column is a month
retail1_transposed = pd.DataFrame(dict(grouped))

# Sort columns by month
retail1_transposed = retail1_transposed.sort_index(axis=1)

# Sort value ascending
retail1_transposed = retail1_transposed.fillna(0)
retail1_transposed = retail1_transposed.apply(sorted)

# Format column names as strings
retail1_transposed.columns = retail1_transposed.columns.strftime('%Y-%m-%d')
retail1_transposed.tail()
```

Out[64]:	1992-01-01	1992-02-01	1992-03-01	1992-04-01	1992-05-01	1992-06-01	1992-07-01	1992-08-01	1992-09-01	1992-10-01	...	2024-03-01	2024-04-01
60	104466.0	104282.0	111928.0	115144.0	120159.0	116847.0	118202.0	121103.0	115594.0	123254.0	...	514062.0	506656.0
61	116565.0	115862.0	124200.0	127587.0	133608.0	130274.0	132076.0	134928.0	128734.0	136917.0	...	567110.0	561592.0
62	130683.0	131244.0	142488.0	147175.0	152420.0	151849.0	152586.0	152476.0	148158.0	155987.0	...	609462.0	605419.0
63	134277.0	135499.0	147064.0	151226.0	156619.0	155236.0	156016.0	156539.0	151477.0	159992.0	...	654589.0	645366.0
64	146376.0	147079.0	159336.0	163669.0	170068.0	168663.0	169890.0	170364.0	164617.0	173655.0	...	707637.0	700302.0

5 rows × 396 columns



```
In [65]: from tslearn.metrics import dtw
# Define Sales 1 series
sales_1 = retail1_transposed.iloc[:, 0] # Assuming 'Sales 1' is the first column
print(f"Sales 1 date: {sales_1.name}")

# Compute DTW distance for all series (excluding itself)
similarity_scores = {}
for col in retail1_transposed.columns:
    if col == retail1_transposed.columns[0]:
        continue
    series = retail1_transposed[col].values
    # Compute the DTW distance
    score = dtw(sales_1, series)
    similarity_scores[col] = score

if not similarity_scores:
    raise ValueError("No comparable series found to compare against 'Sales 1'.")
```

```
most_similar = min(similarity_scores, key=similarity_scores.get)
least_similar = max(similarity_scores, key=similarity_scores.get)
```

```
print(f"Most similar to Sales 1: {most_similar}")
print(f"Least similar to Sales 1: {least_similar}")
```

Sales 1 date: 1992-01-01

Most similar to Sales 1: 1992-02-01

Least similar to Sales 1: 2024-12-01

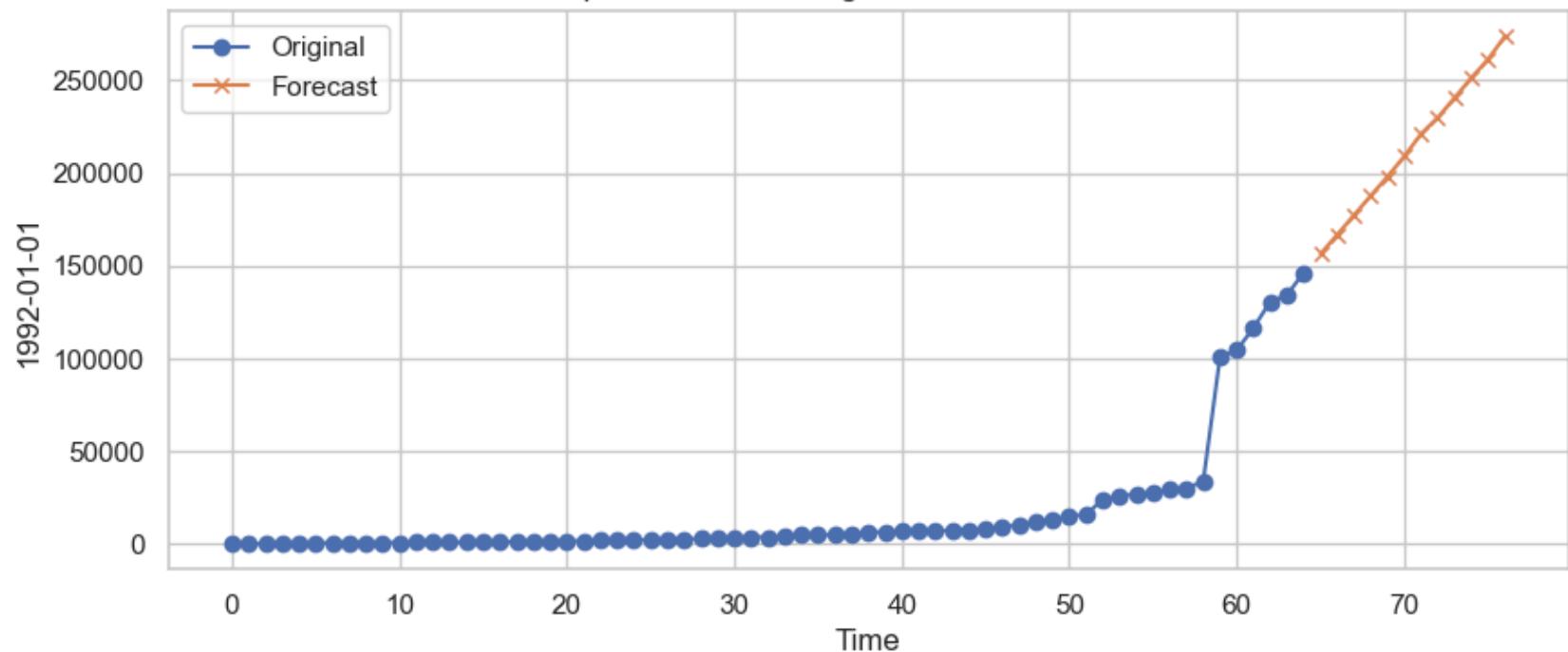
The sales data for January 1, 1992, represented as "Sales 1," exhibits varying degrees of similarity when compared to other time periods, as determined by Dynamic Time Warping (DTW) distance. Notably, the sales pattern on February 1, 1992, is found to be

the most similar to "Sales 1," suggesting that early-year sales trends in 1992 shared significant alignment. Conversely, the sales data from December 1, 2024, stands out as the least similar. This stark dissimilarity over an extended period—more than three decades—highlights potential long-term shifts in sales patterns, seasonal influences, market dynamics, or underlying economic conditions that have diverged significantly from the early 1992 baseline.

```
In [ ]: #Step 4: Apply an “TS Exponential Smoothing” to all “Time Series” and forecast all-time series.  
from statsmodels.tsa.holtwinters import ExponentialSmoothing  
# Define forecast horizon (e.g., 12 periods into the future)  
forecast_steps = 12  
  
# Create an empty dictionary to store forecasts for every series  
forecasts = {}  
  
# Loop through each time series (each column) in the DataFrame  
for col in retail1_transposed.columns:  
    series = retail1_transposed[col]  
  
    # Try to fit a seasonal model (if you expect seasonality, e.g., monthly with seasonal_periods=12)  
    try:  
        model = ExponentialSmoothing(series, trend='add', seasonal='add', seasonal_periods=12)  
        fit = model.fit(optimized=True)  
    except Exception as e:  
        print(f"Seasonal model failed for {col} due to: {e}. Trying non-seasonal model.")  
        # Fallback to a non-seasonal model if fitting fails  
        model = ExponentialSmoothing(series, trend='add', seasonal=None)  
        fit = model.fit(optimized=True)  
  
    # Forecast the future values  
    forecast = fit.forecast(steps=forecast_steps)  
    forecasts[col] = forecast  
  
    # Plot the original series and the forecast  
    plt.figure(figsize=(10, 4))  
    plt.plot(series.index, series, label='Original', marker='o')  
    plt.plot(forecast.index, forecast, label='Forecast', marker='x')  
    plt.title(f"Exponential Smoothing Forecast for {col}")  
    plt.xlabel("Time")  
    plt.ylabel(col)  
    plt.legend()  
    plt.show()
```

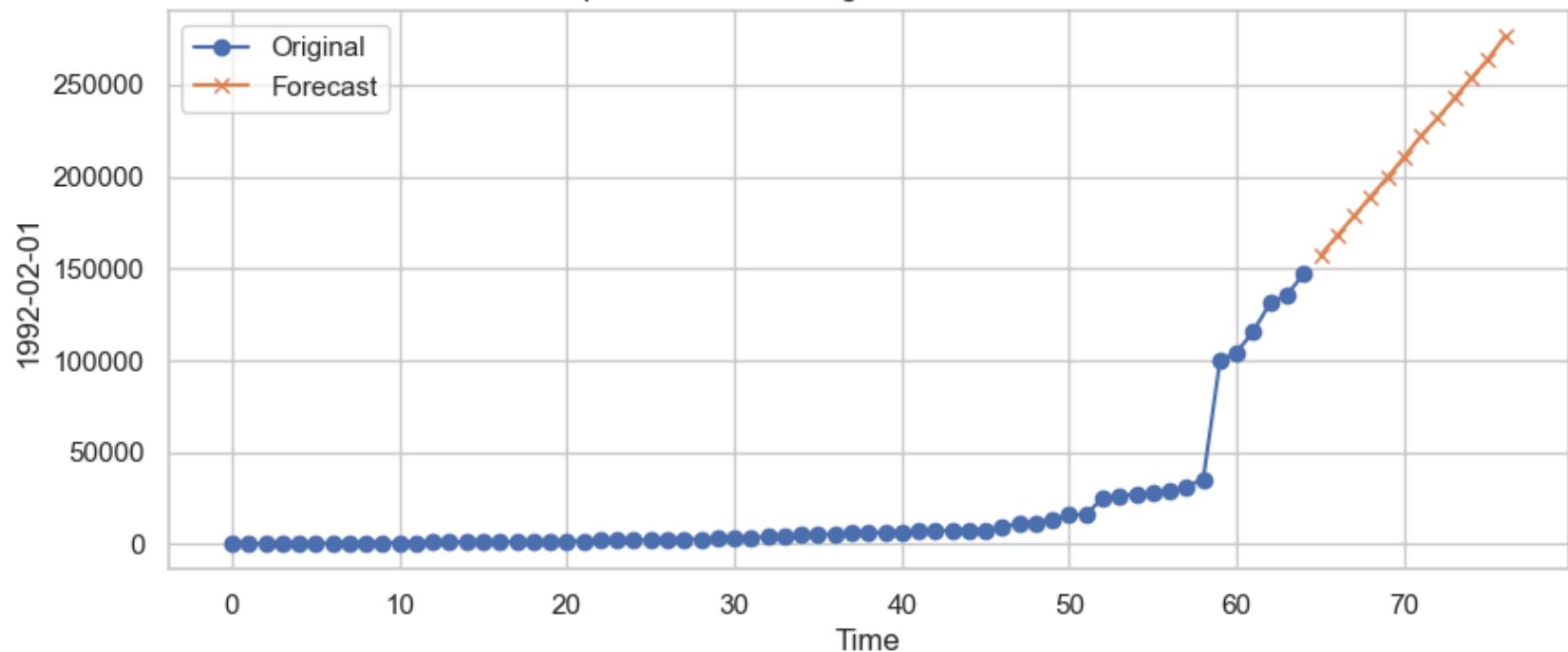
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retrvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1992-01-01



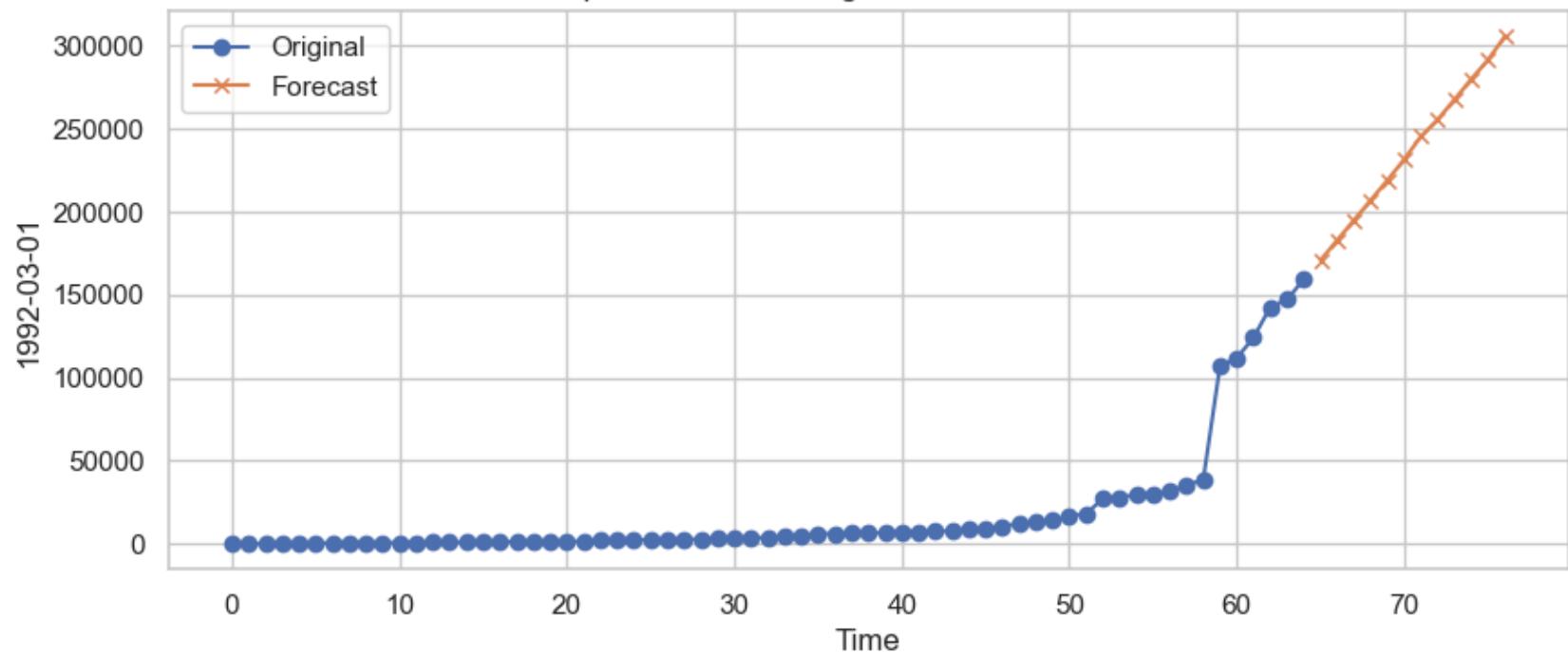
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1992-02-01



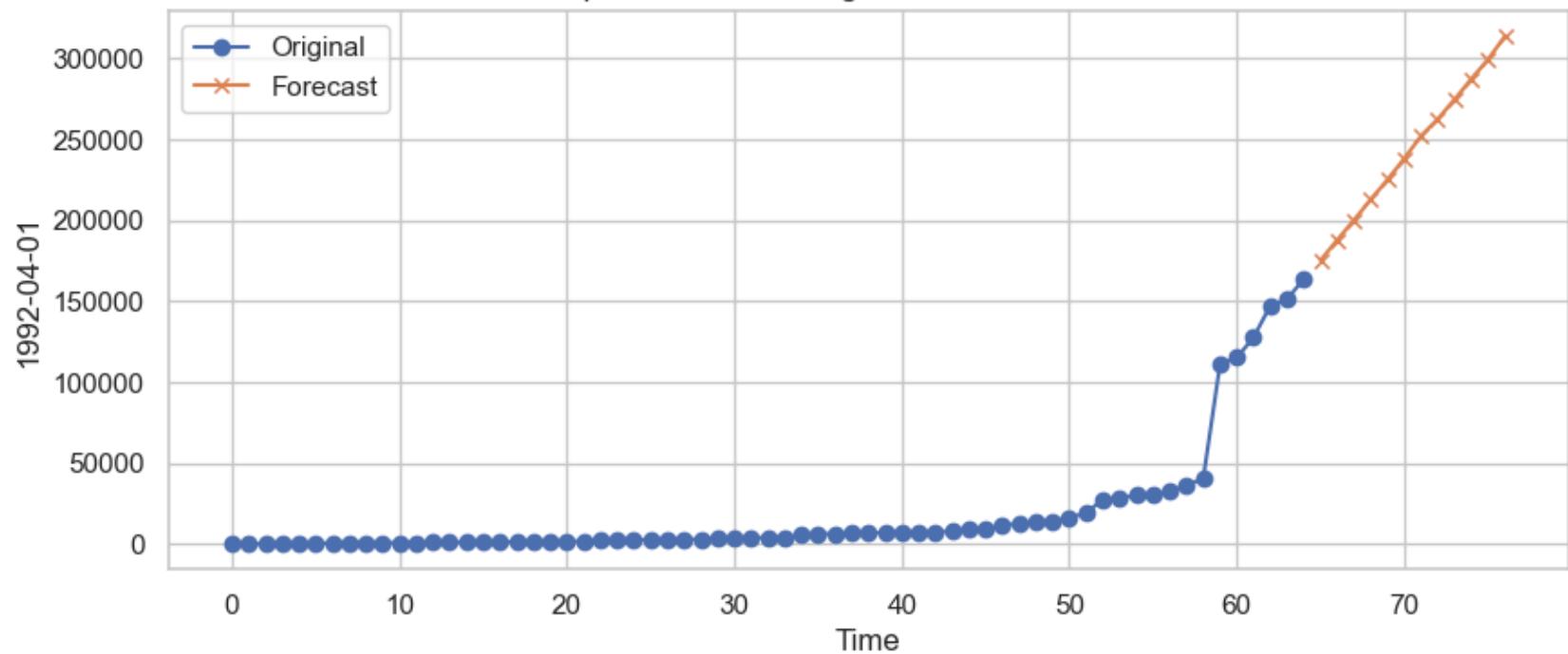
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1992-03-01



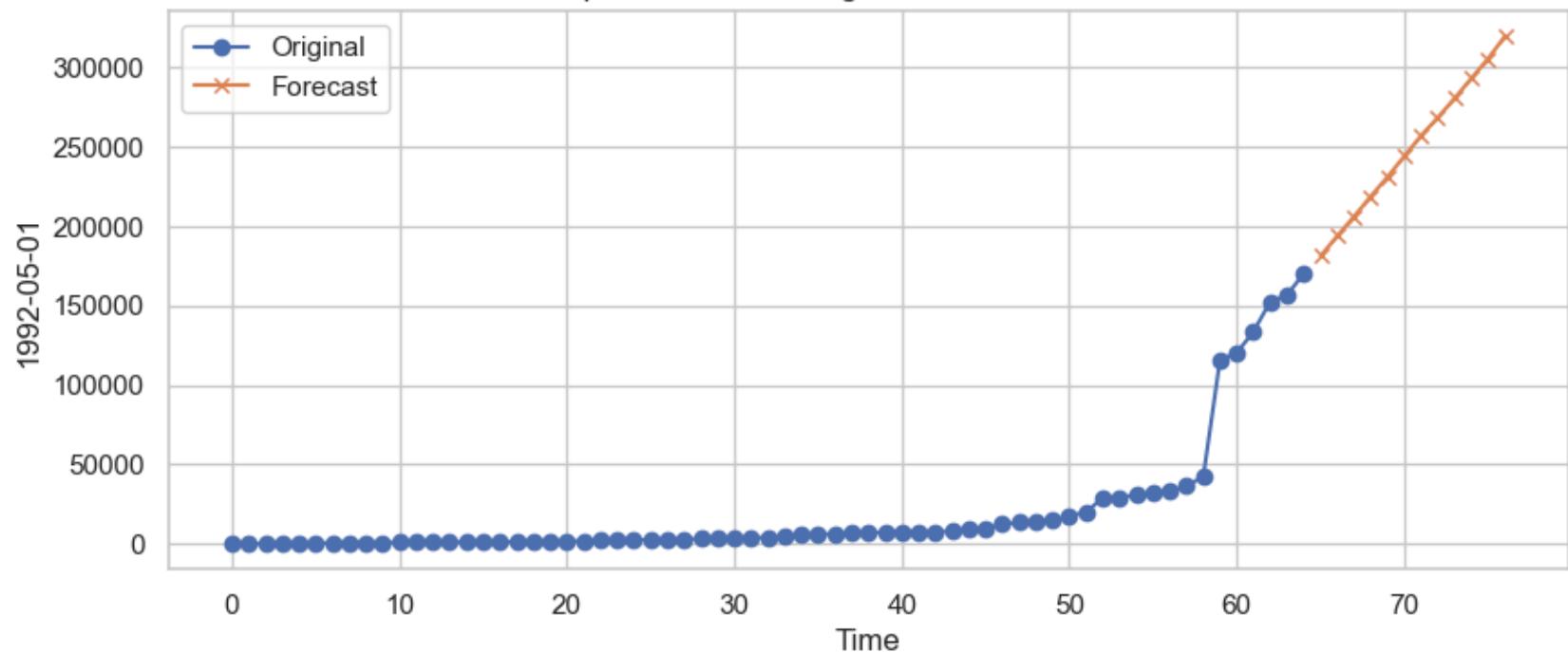
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1992-04-01



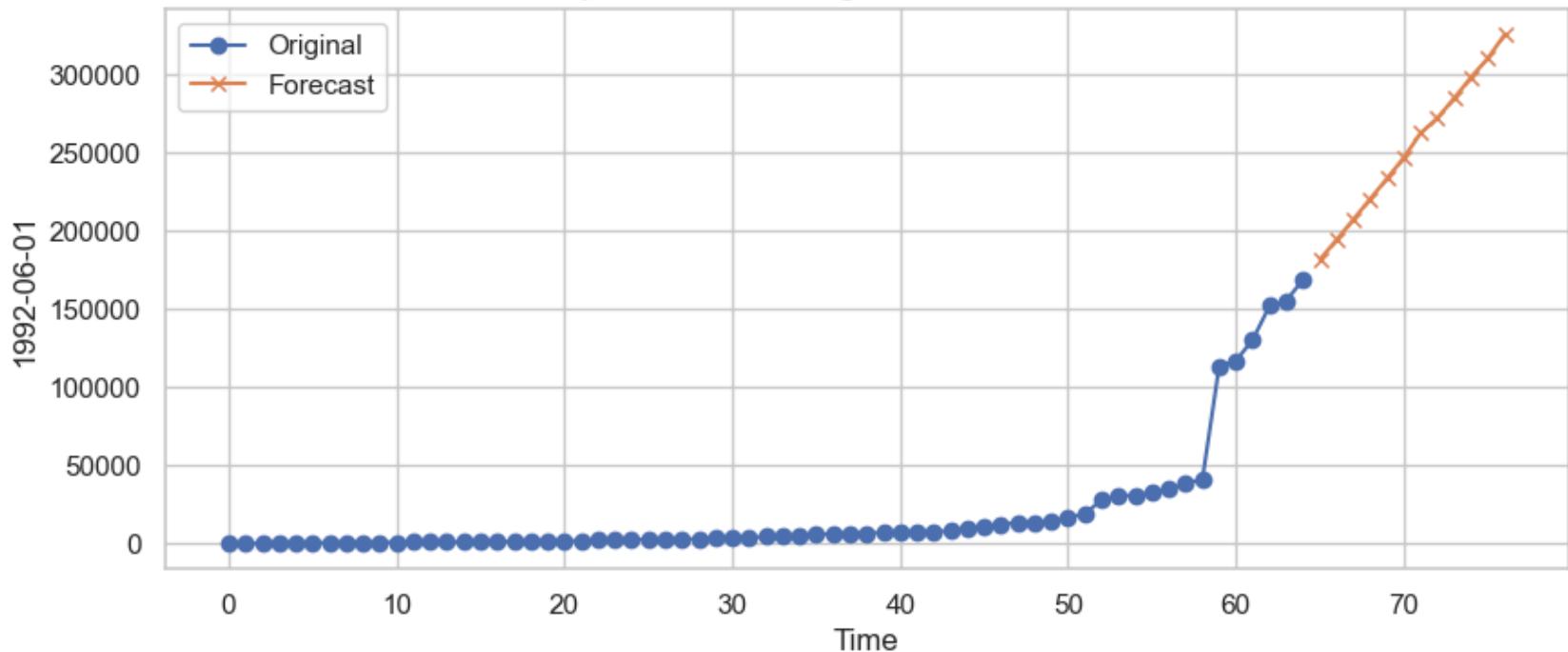
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1992-05-01



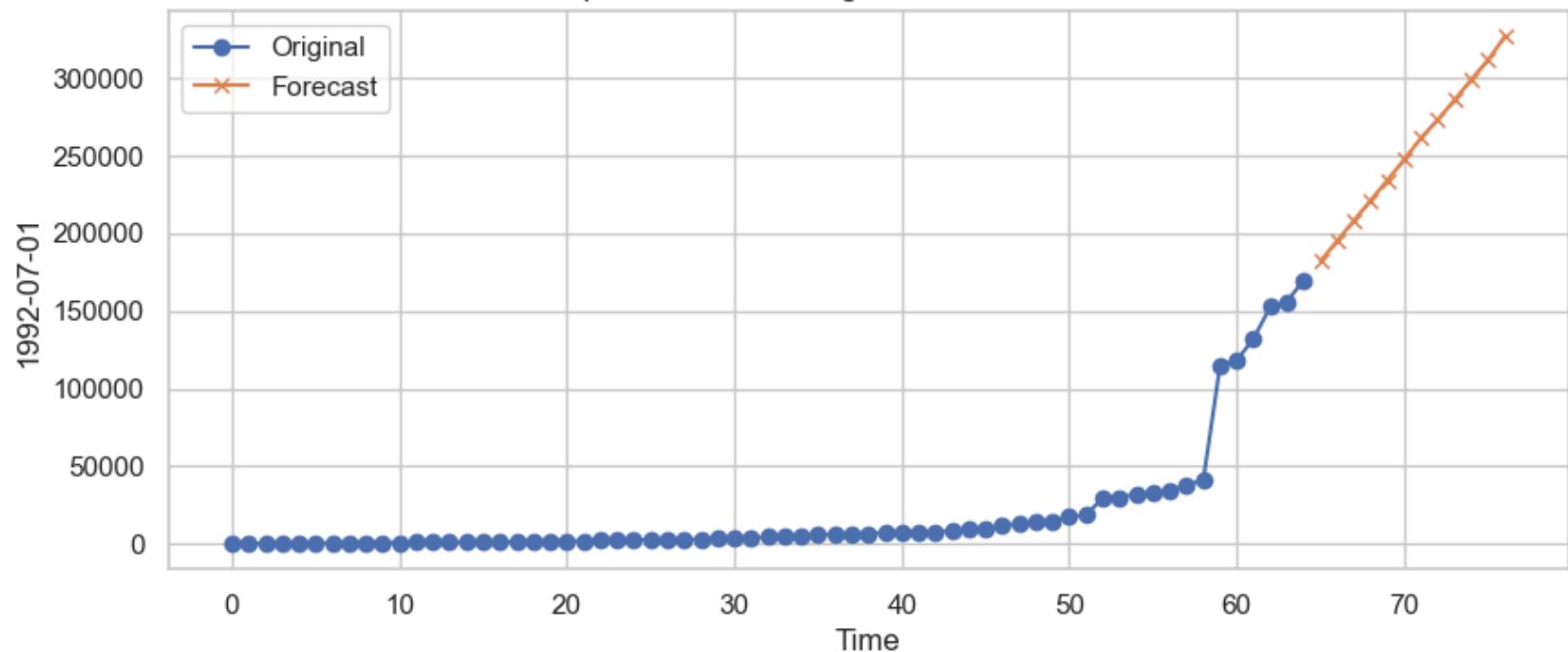
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1992-06-01



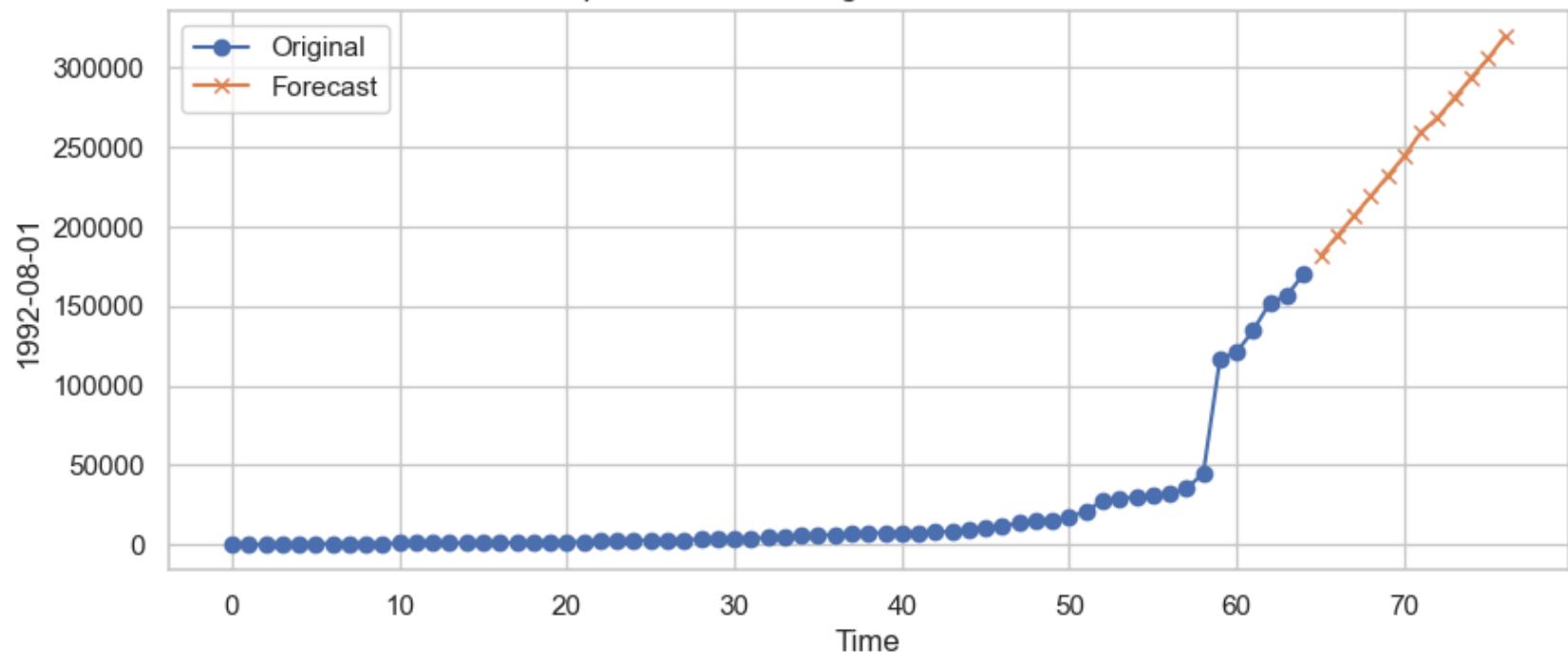
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1992-07-01



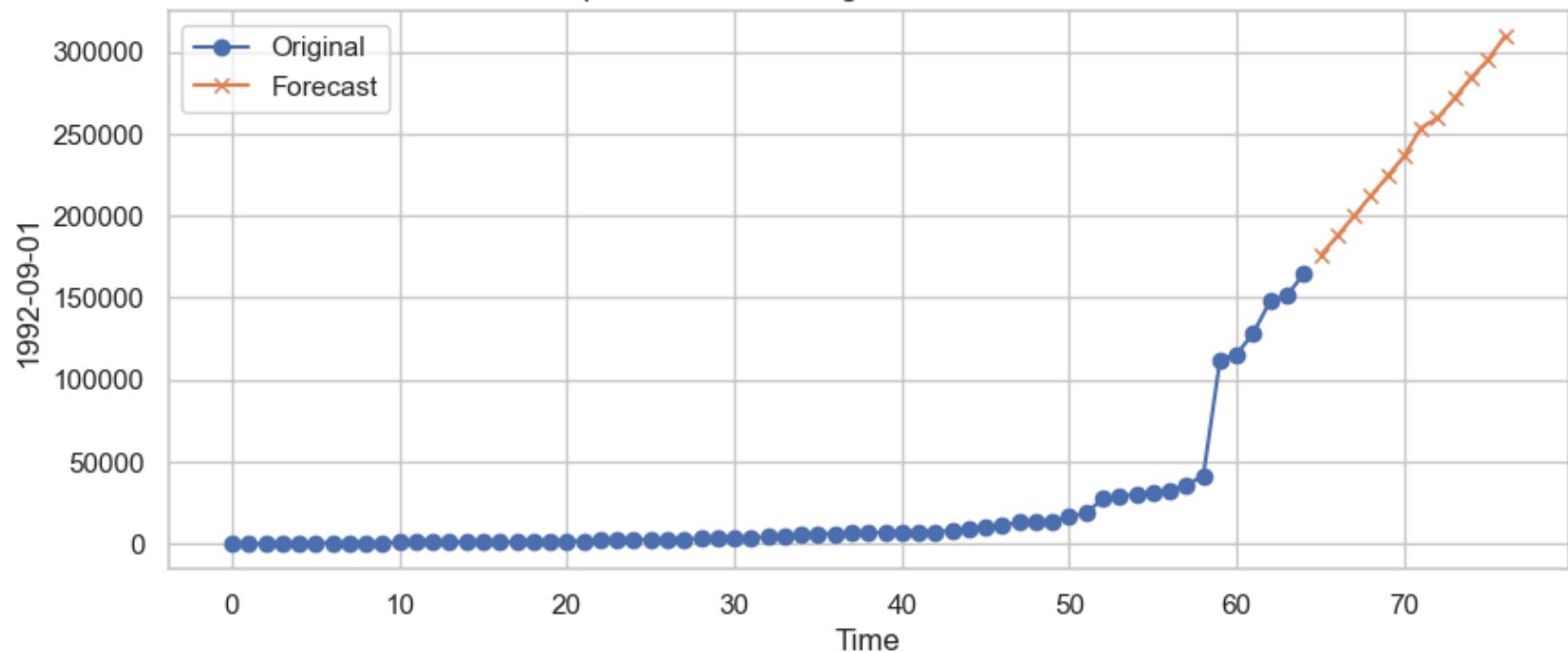
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1992-08-01



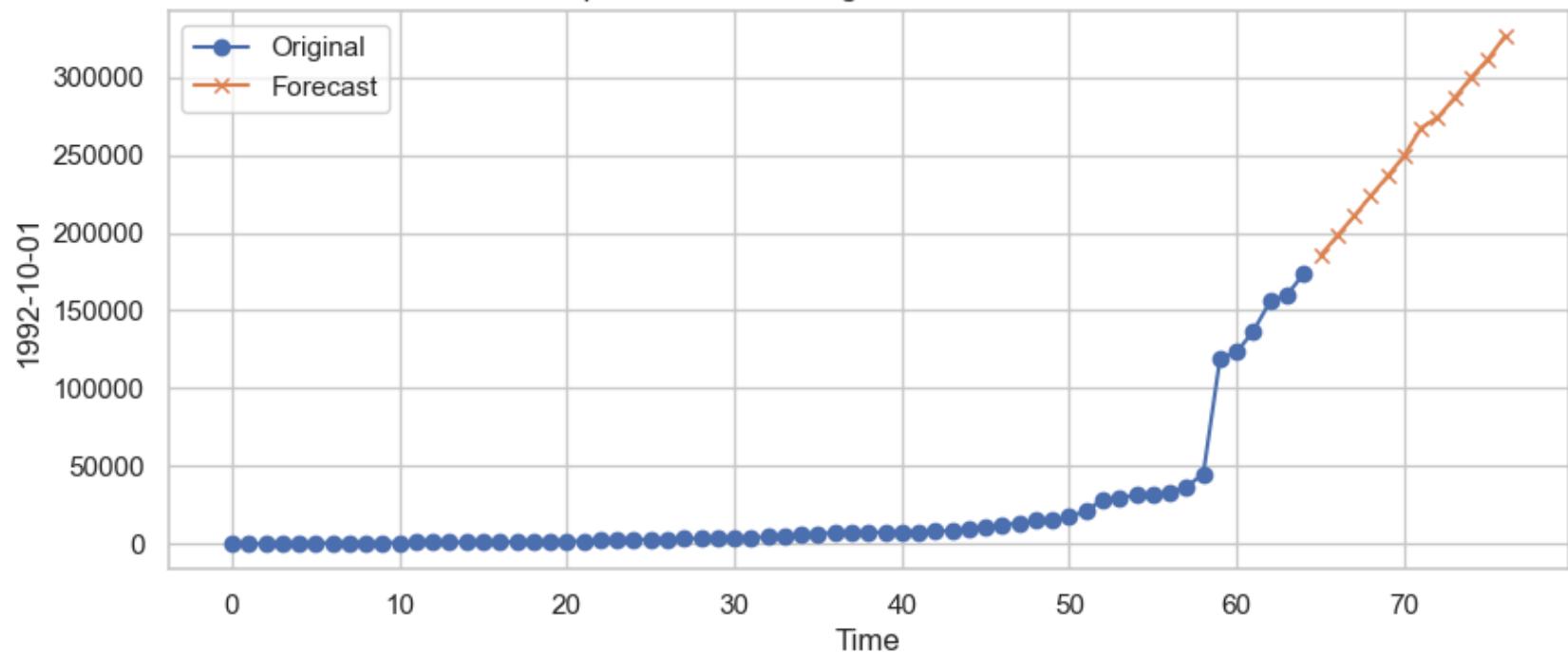
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1992-09-01



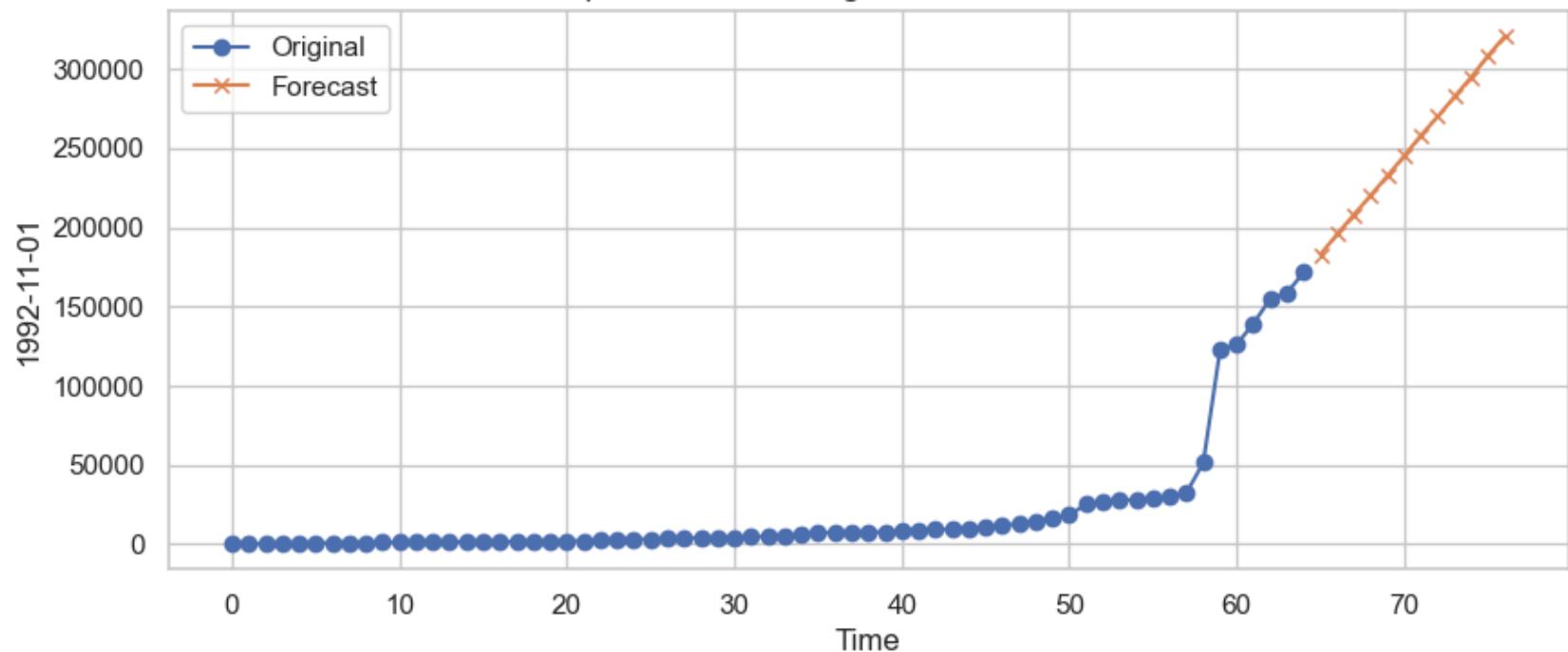
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1992-10-01

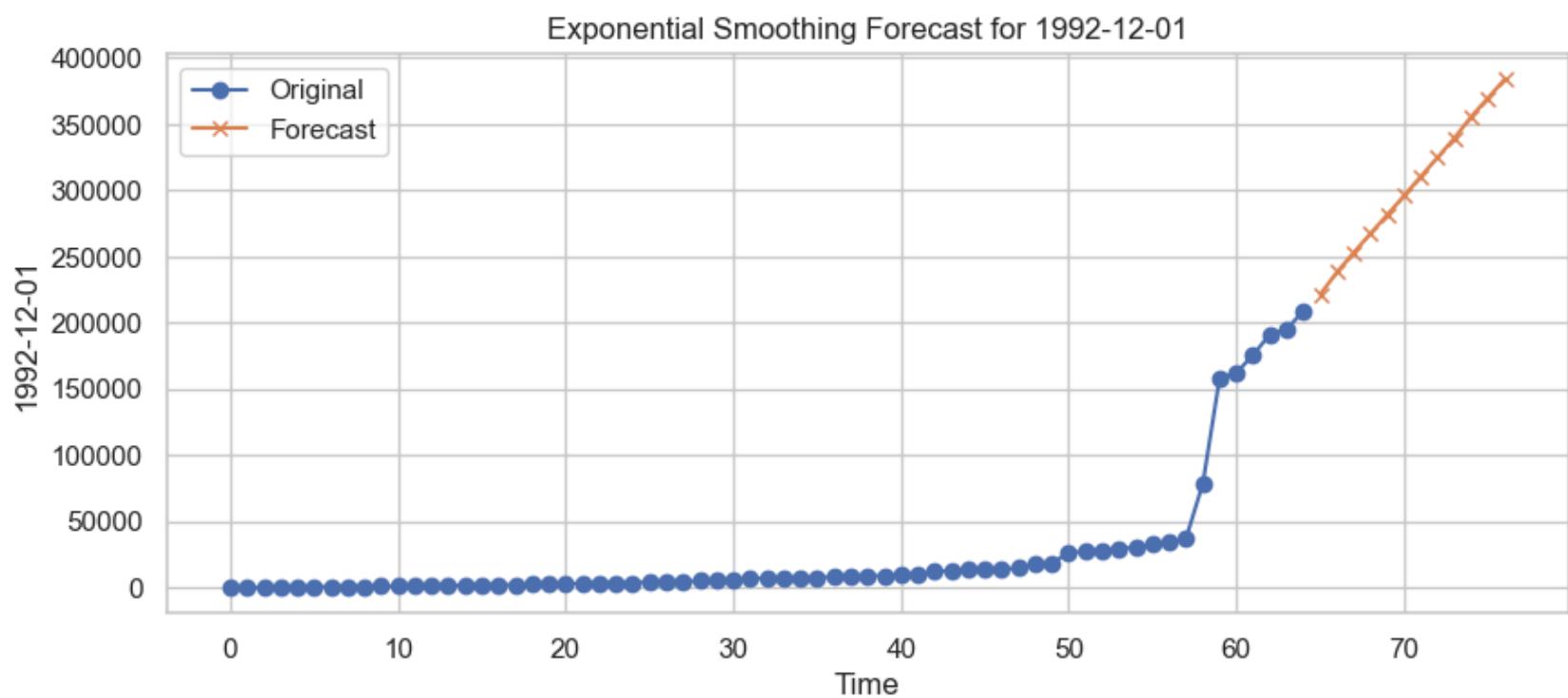


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

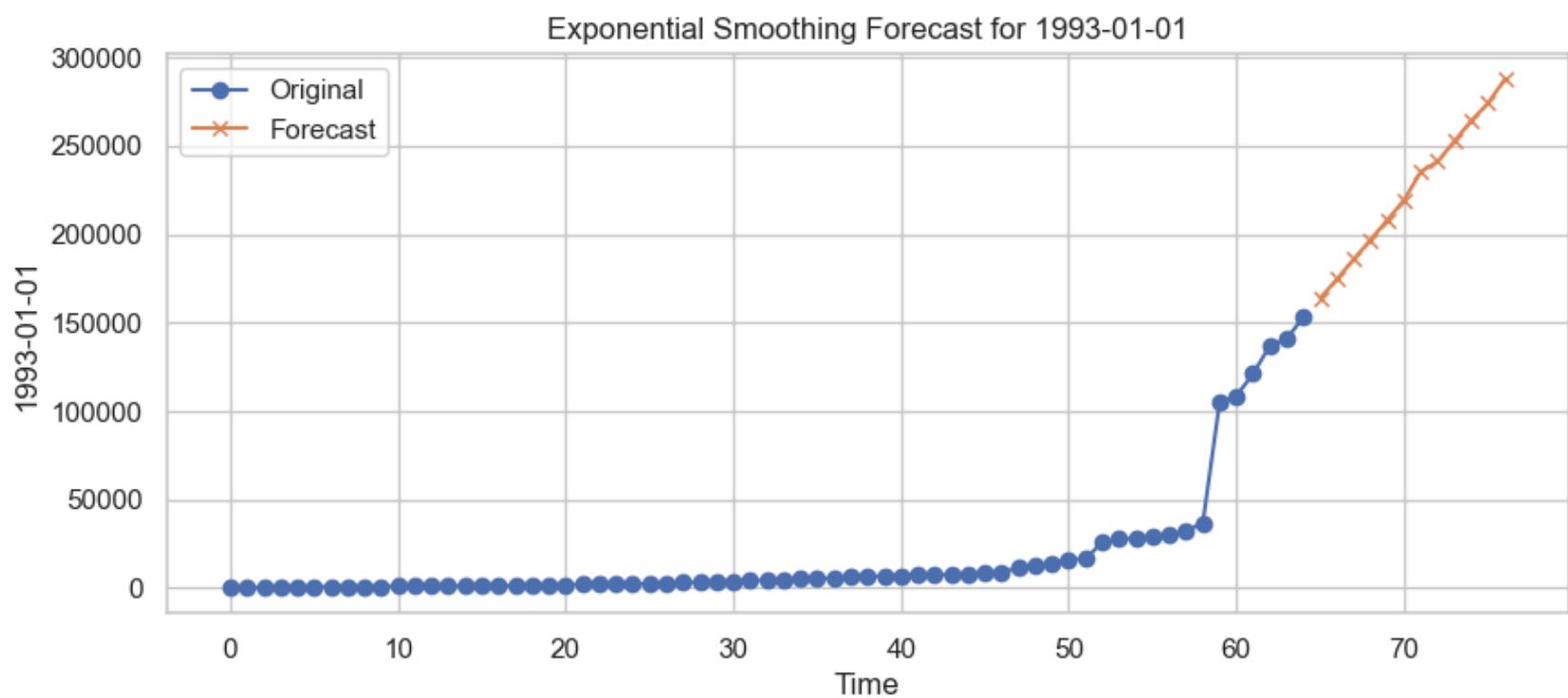
Exponential Smoothing Forecast for 1992-11-01



```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

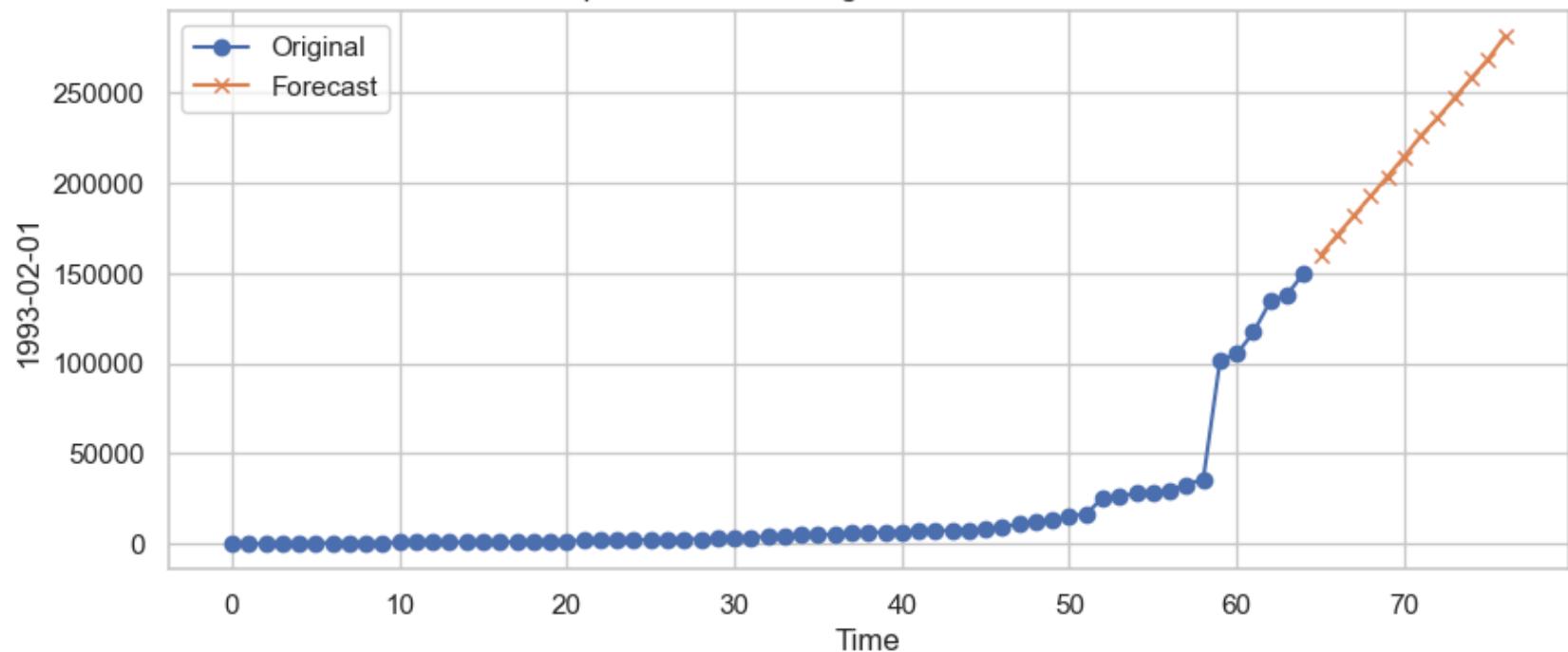


```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



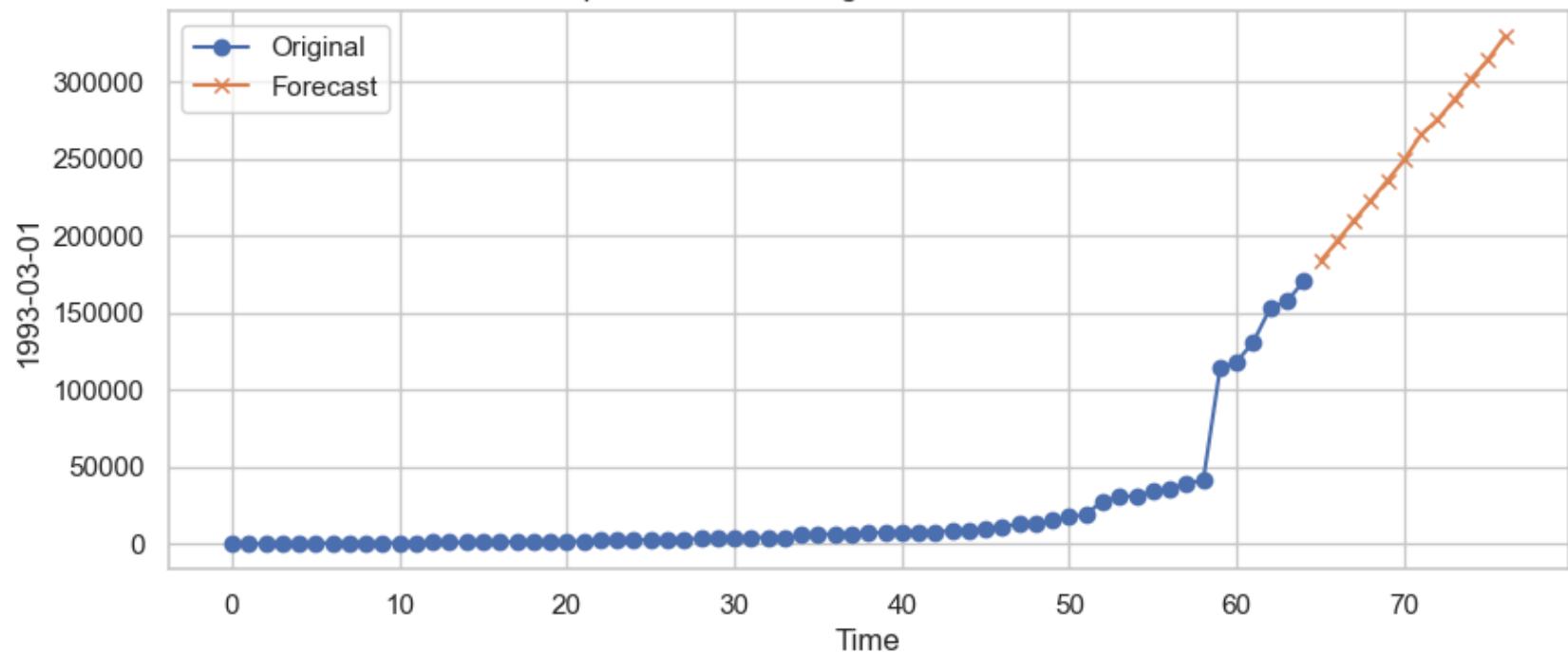
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1993-02-01



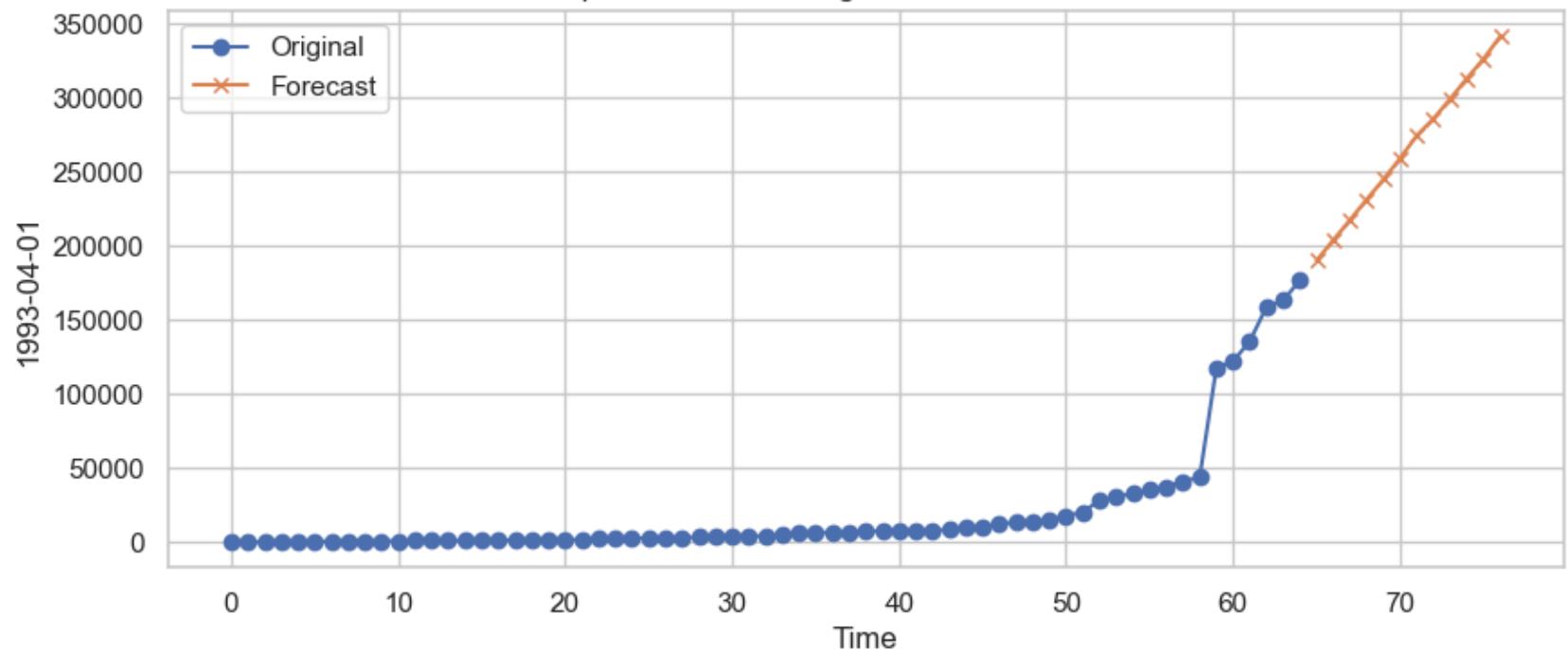
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1993-03-01



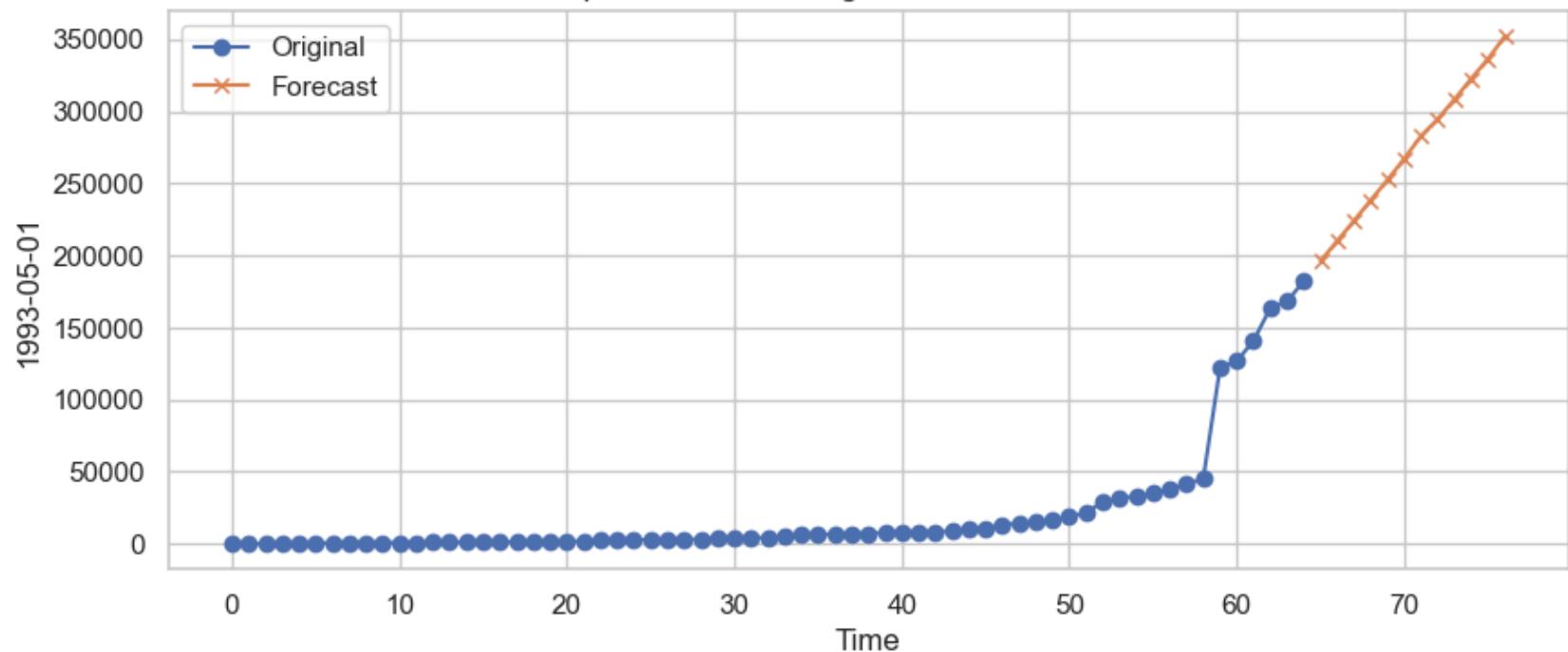
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1993-04-01



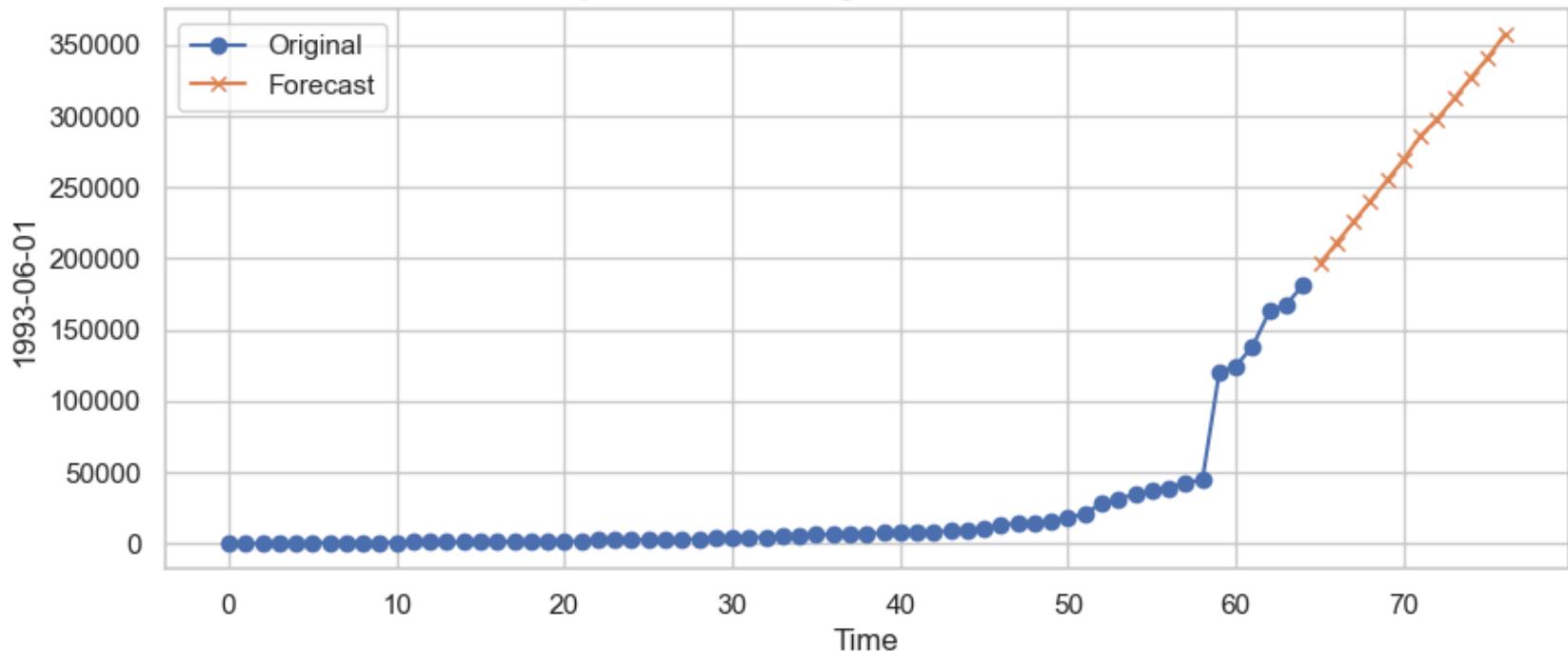
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1993-05-01



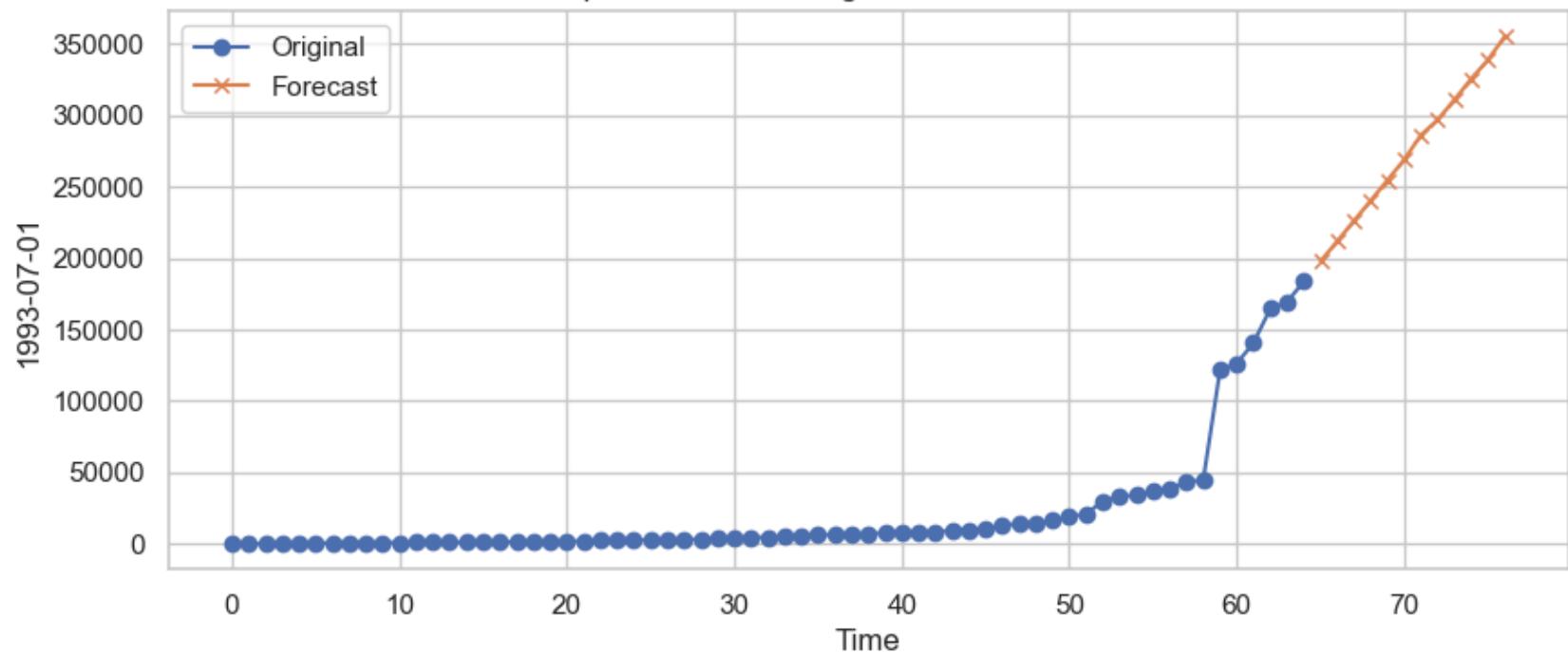
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1993-06-01



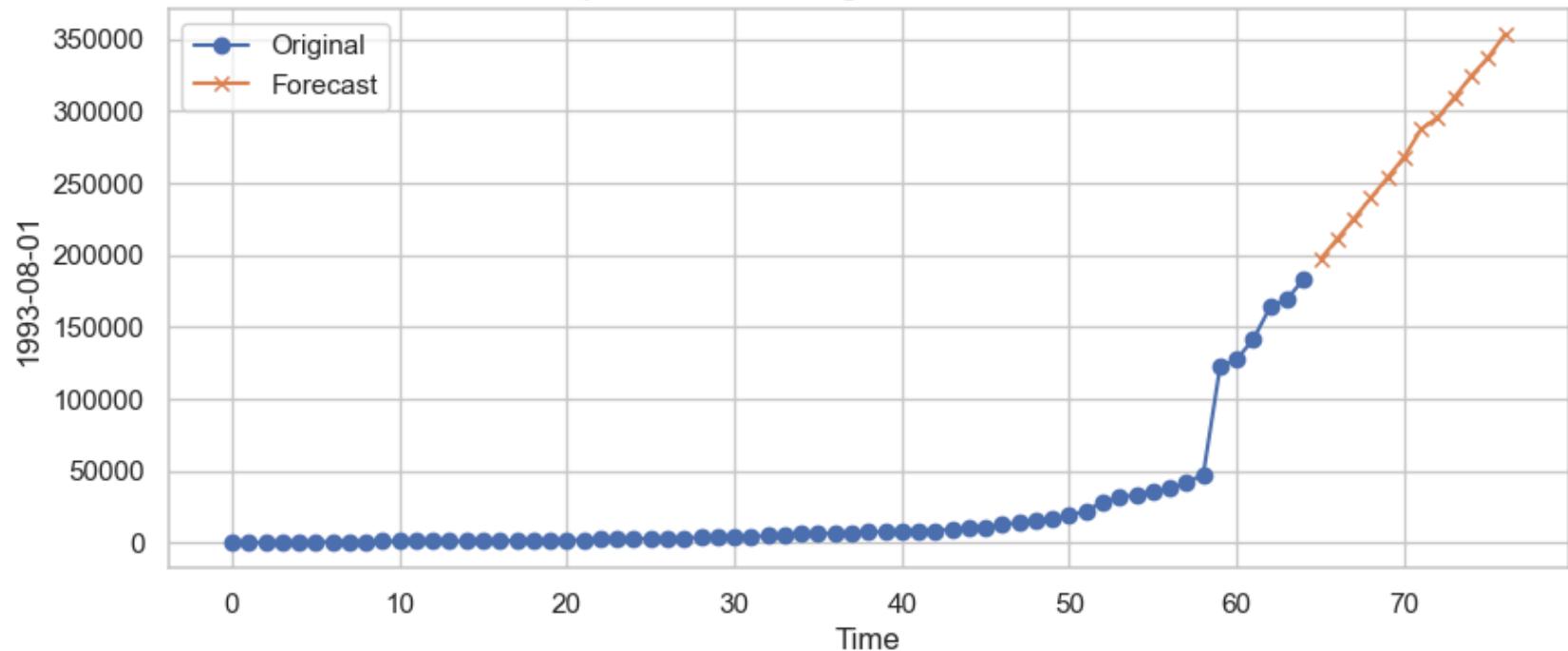
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1993-07-01



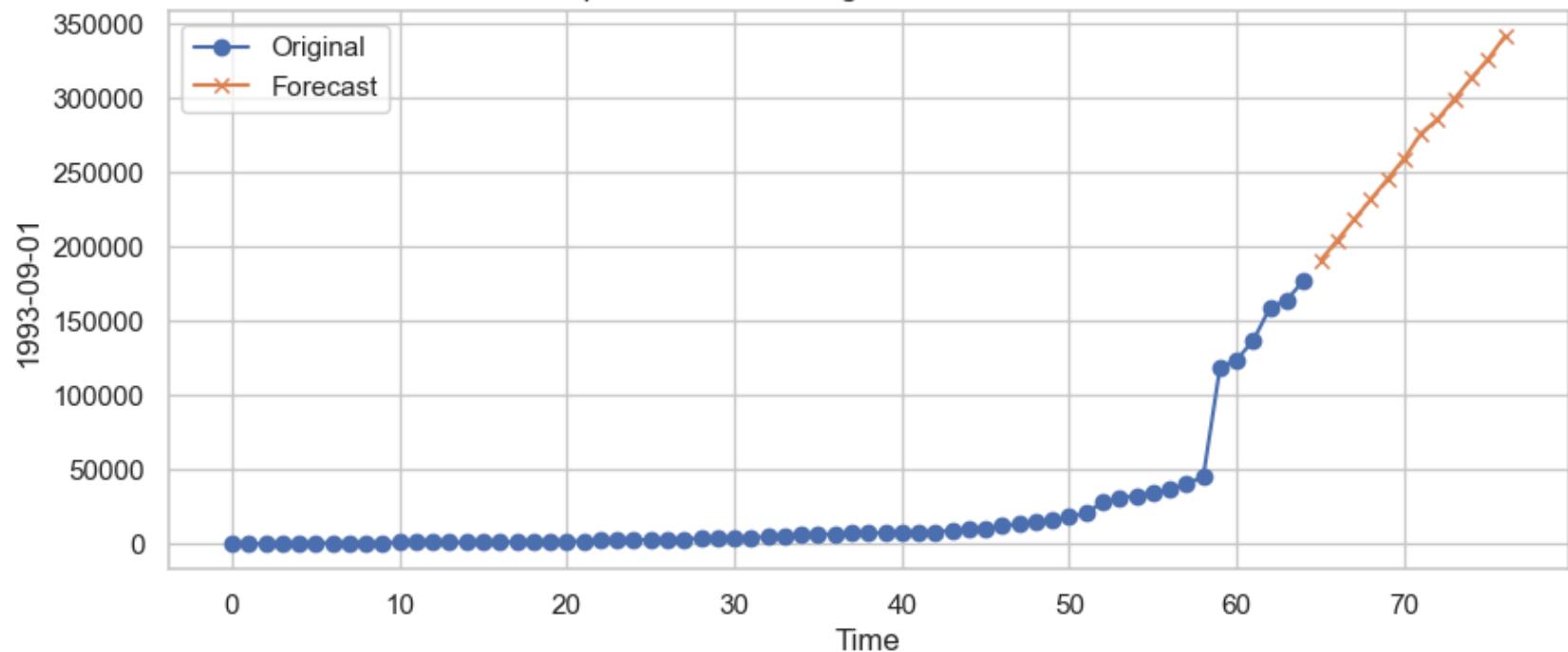
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1993-08-01



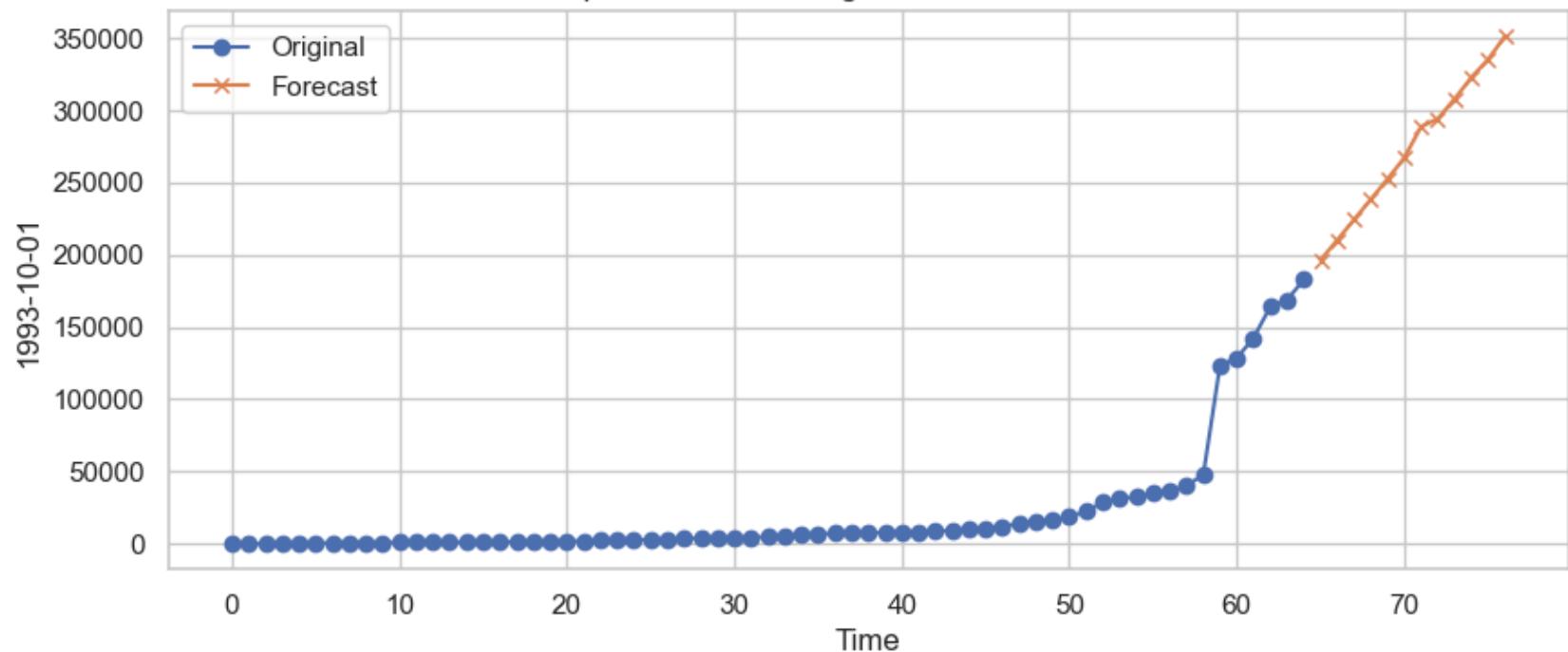
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1993-09-01



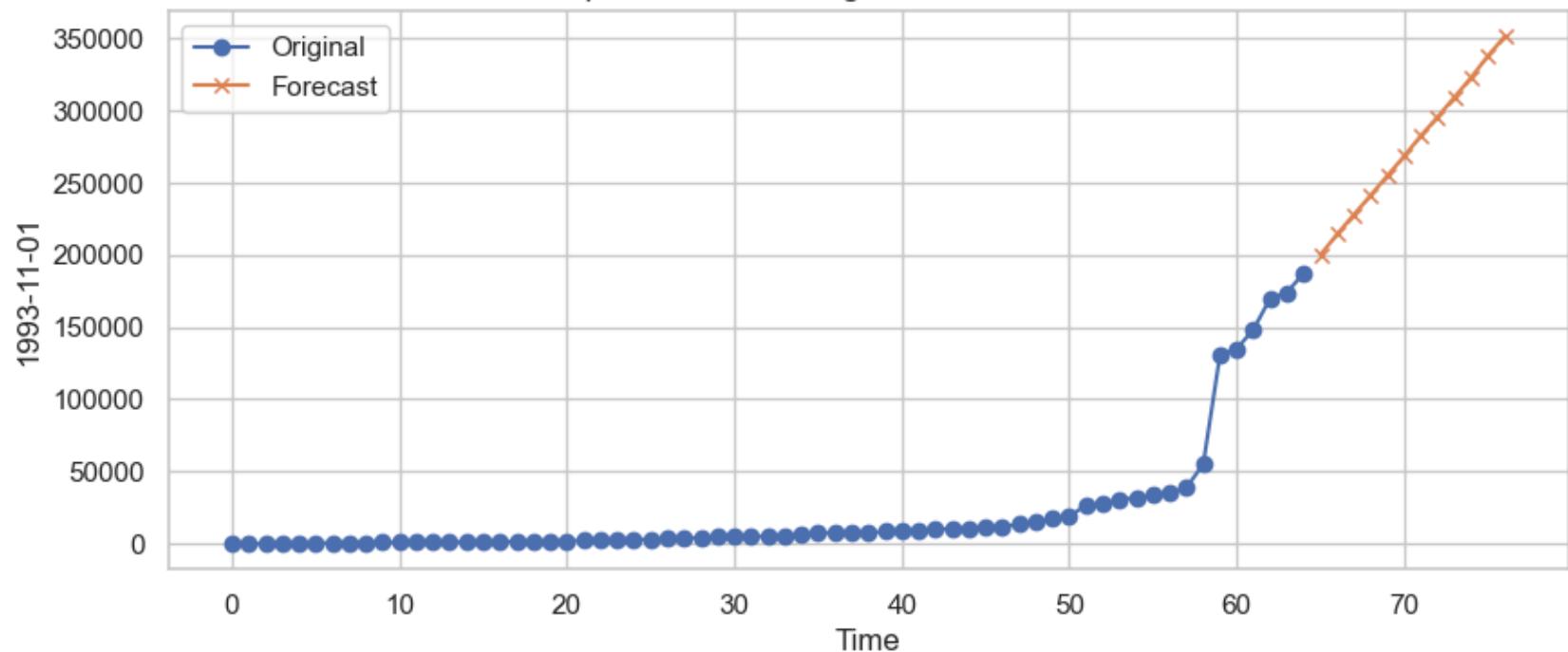
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1993-10-01



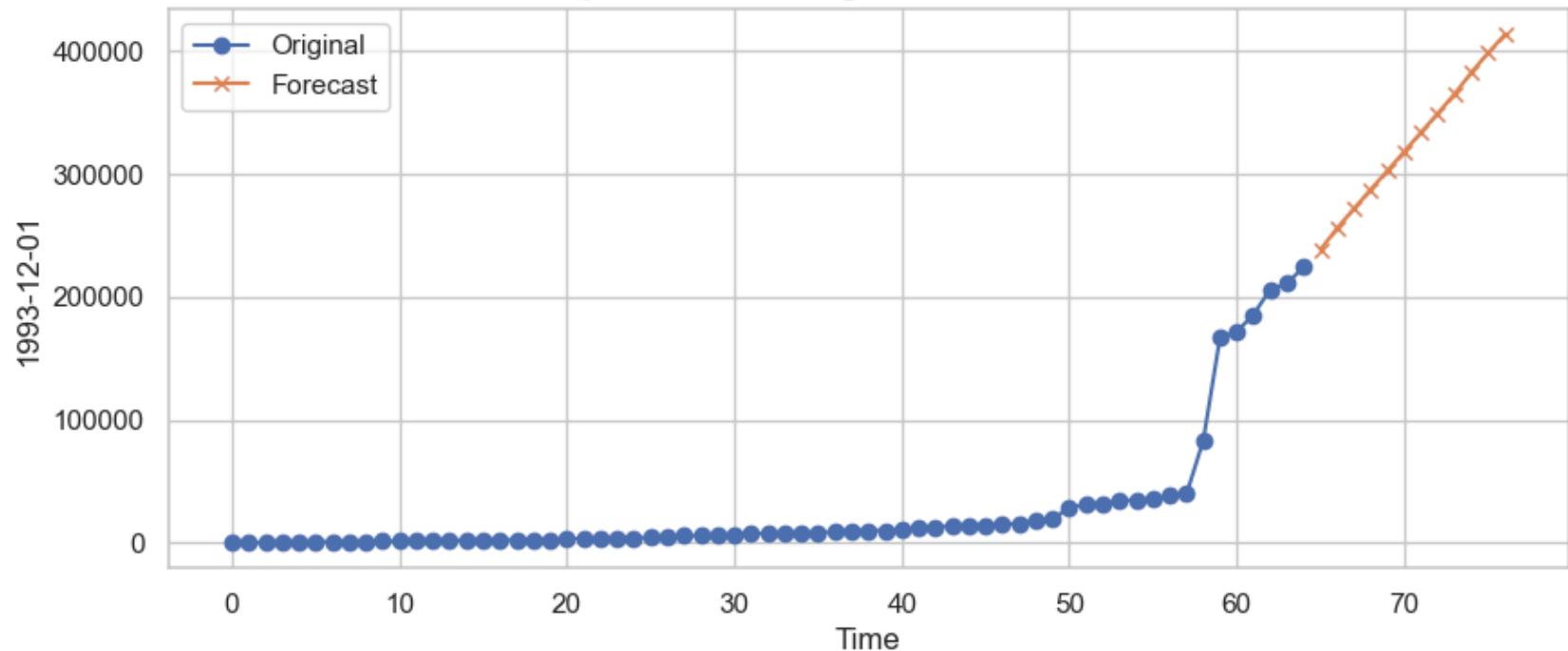
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1993-11-01



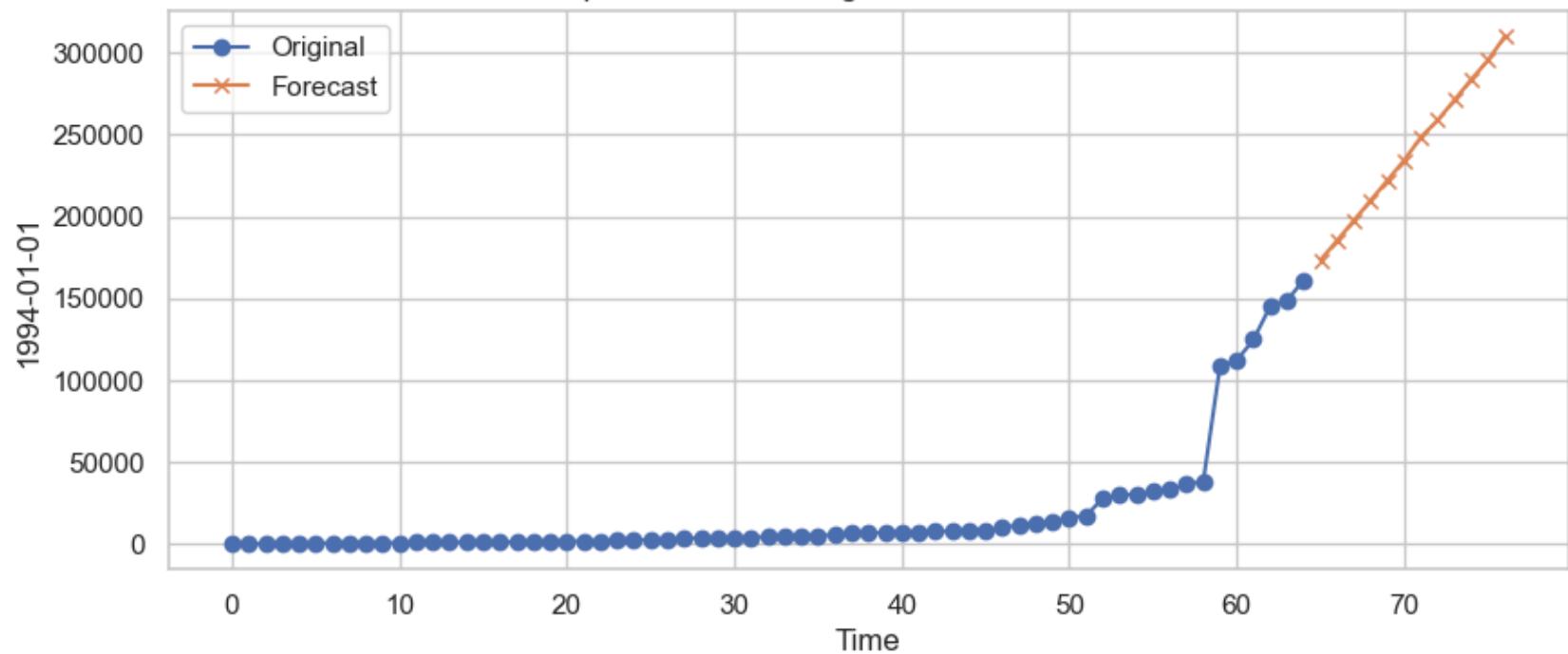
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1993-12-01



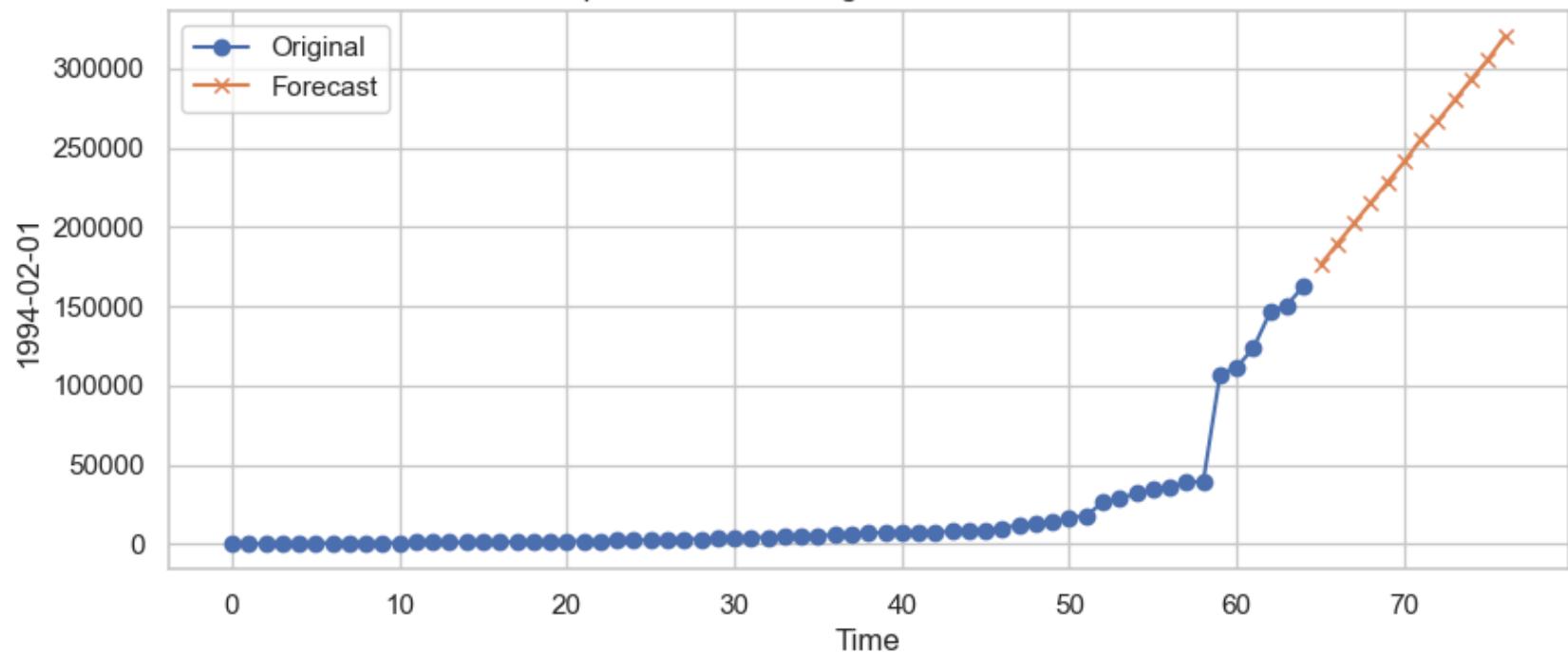
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1994-01-01



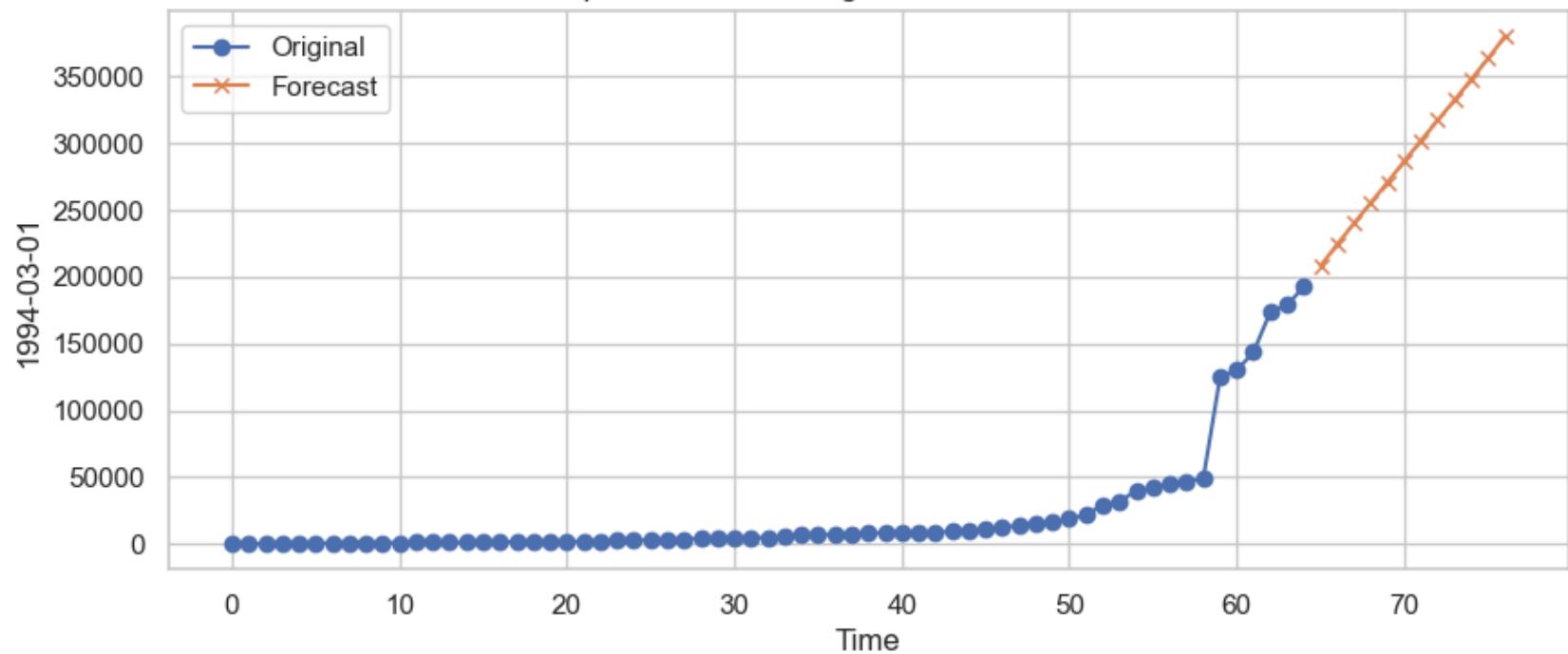
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1994-02-01



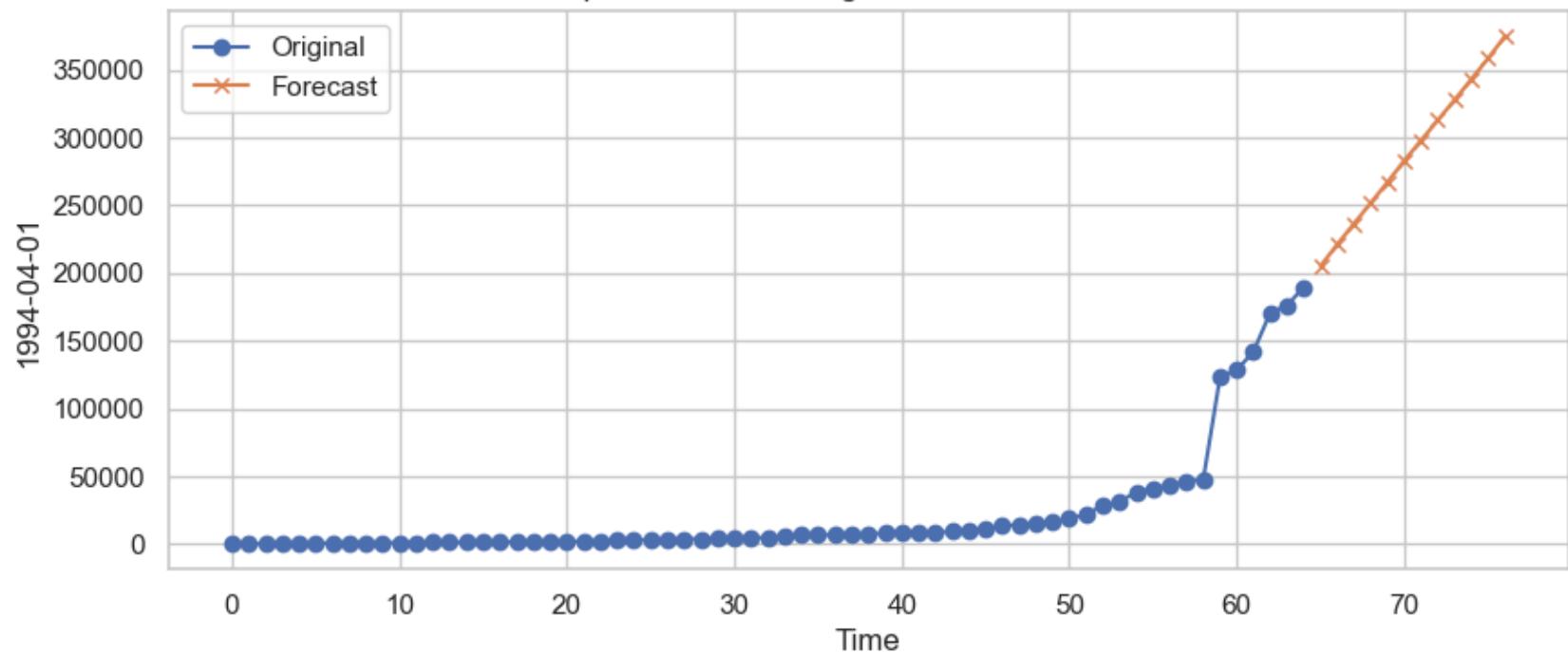
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1994-03-01

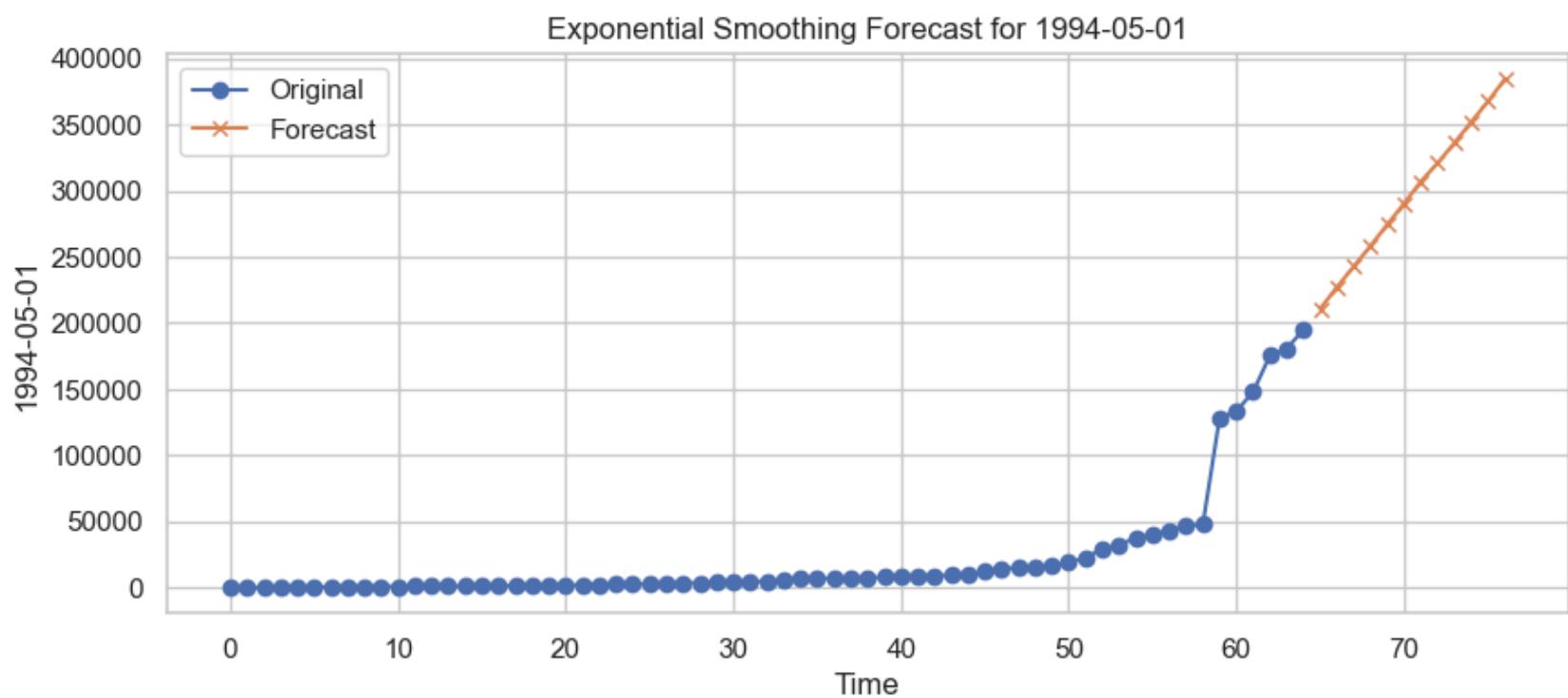


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1994-04-01

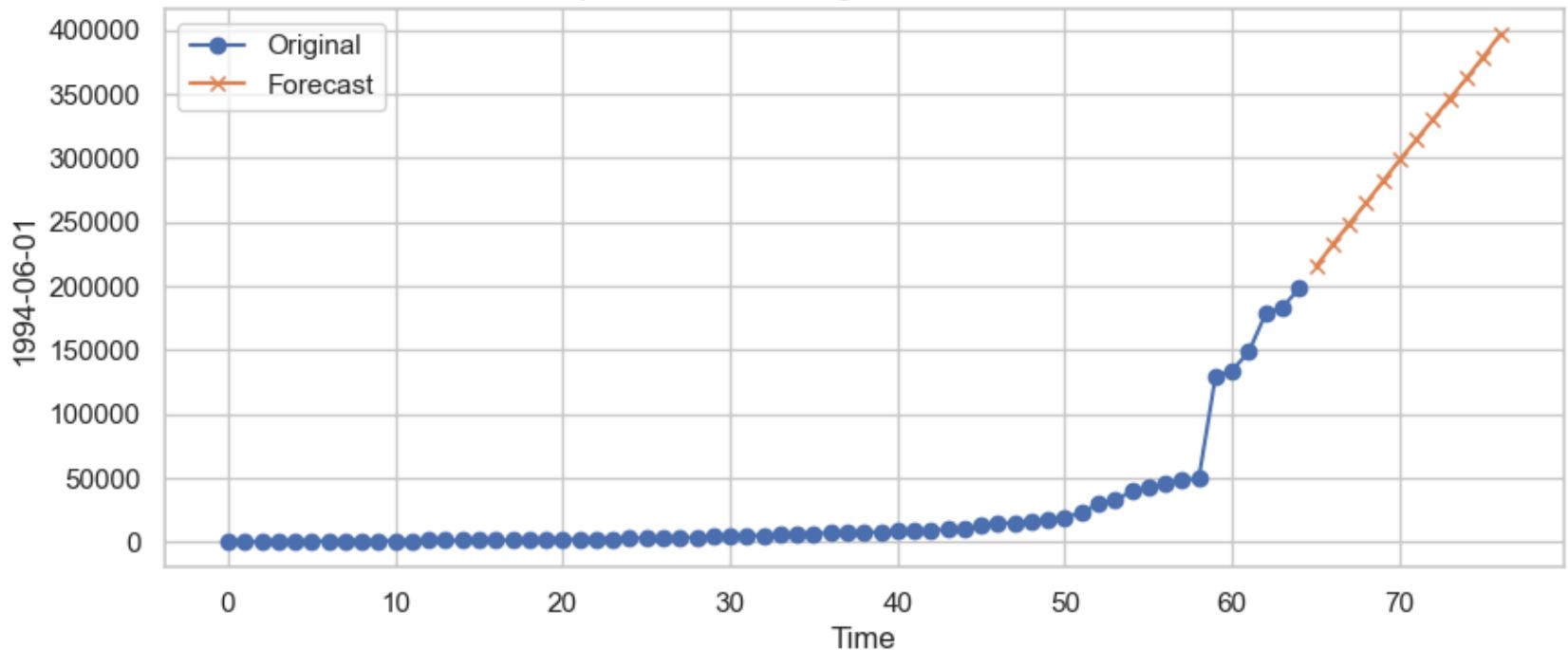


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



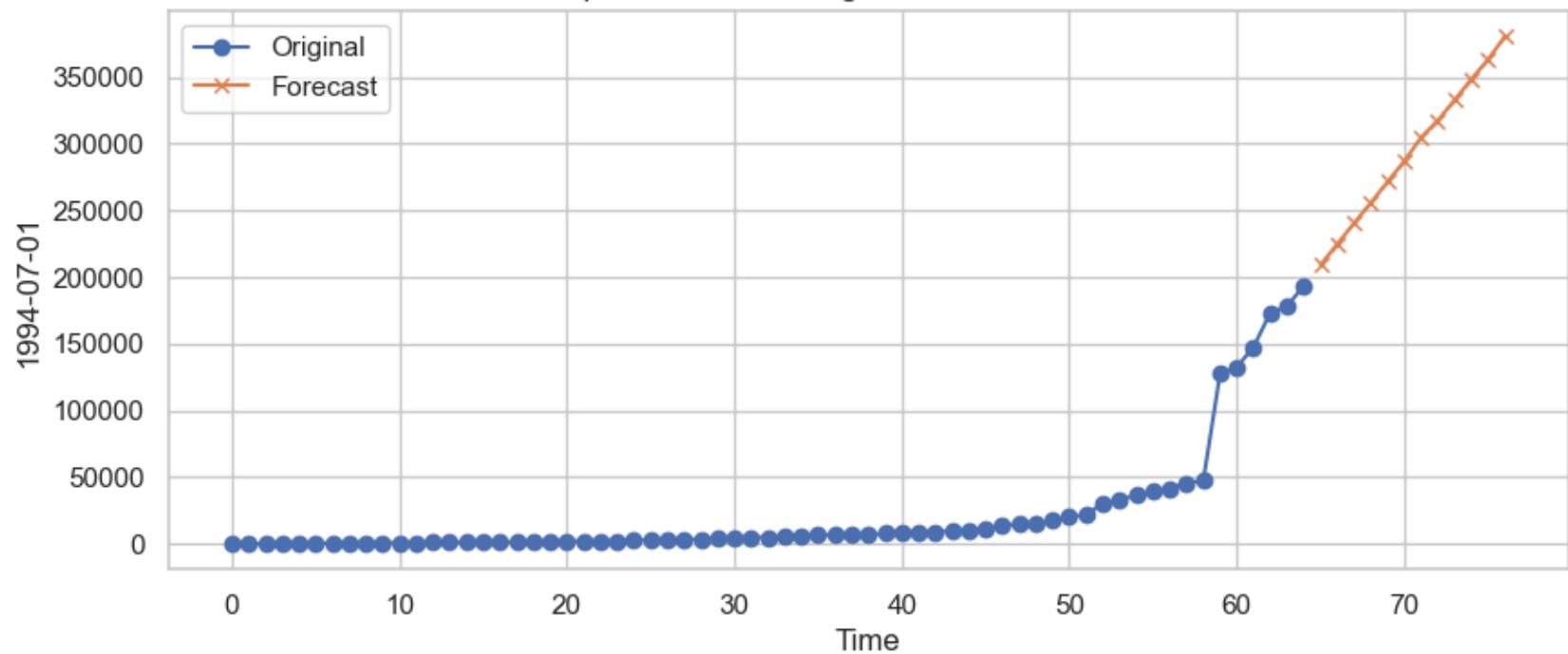
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1994-06-01



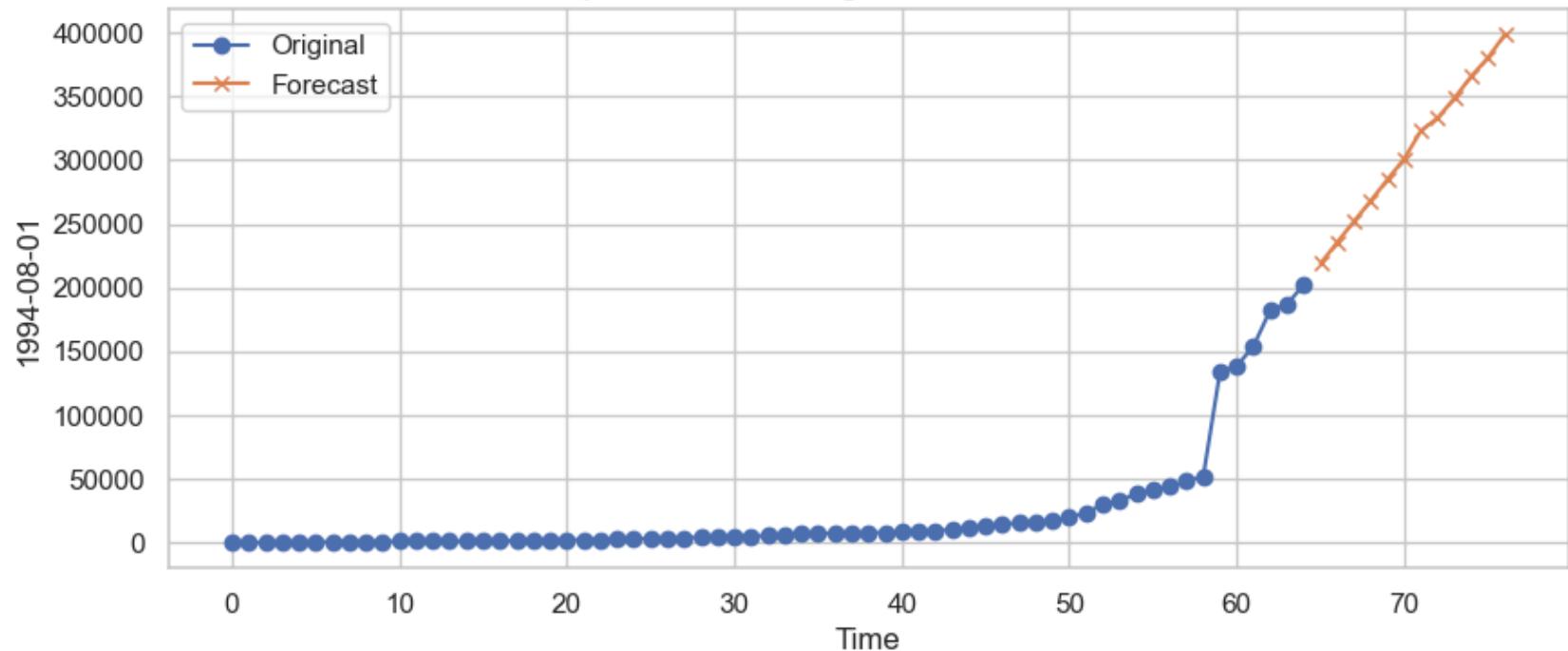
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1994-07-01

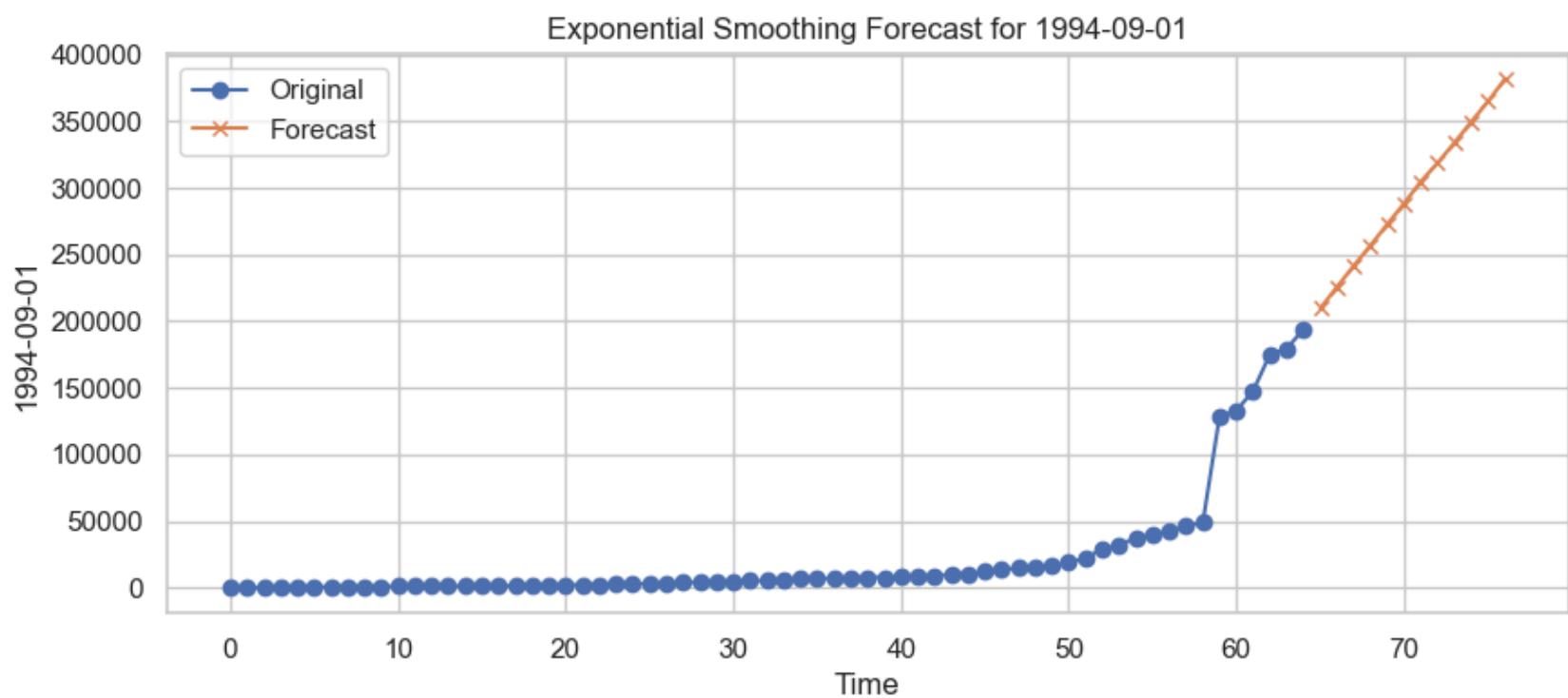


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

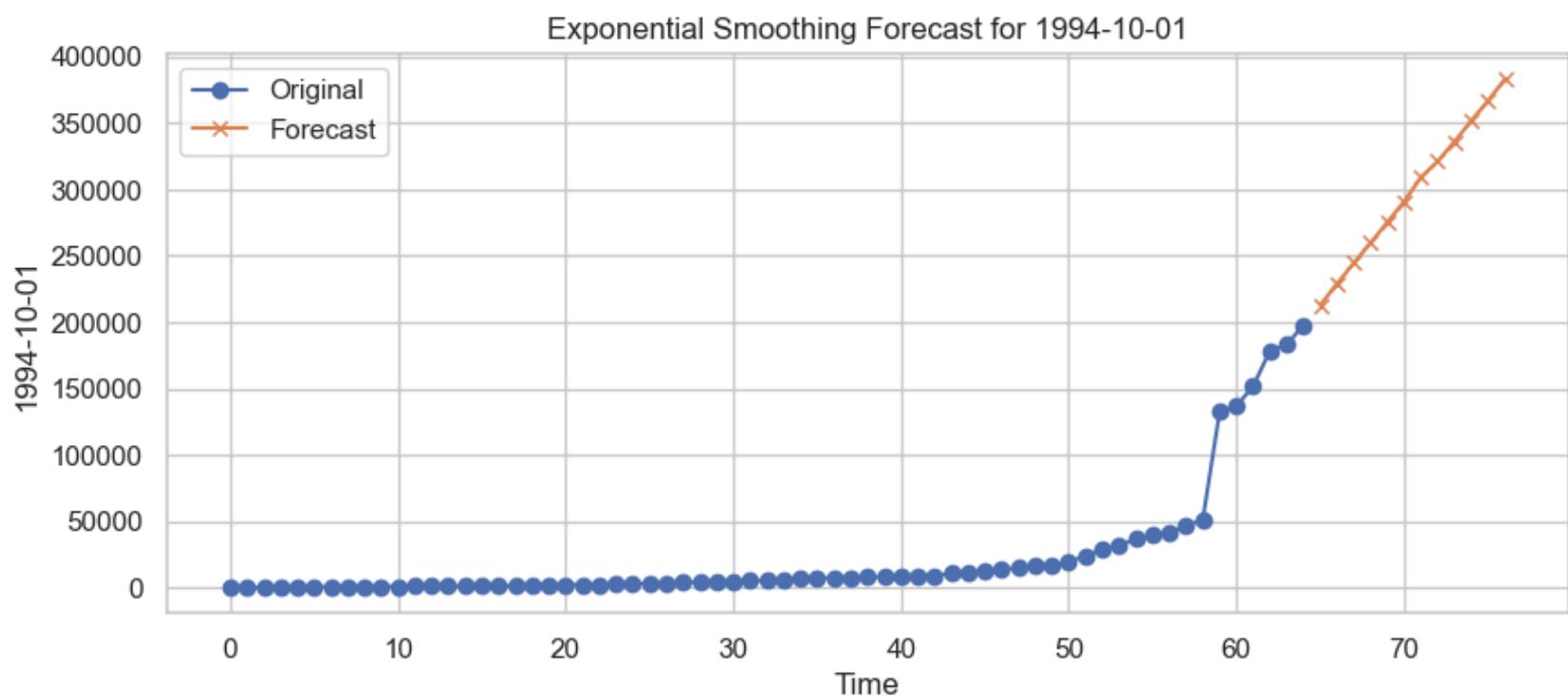
Exponential Smoothing Forecast for 1994-08-01



```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

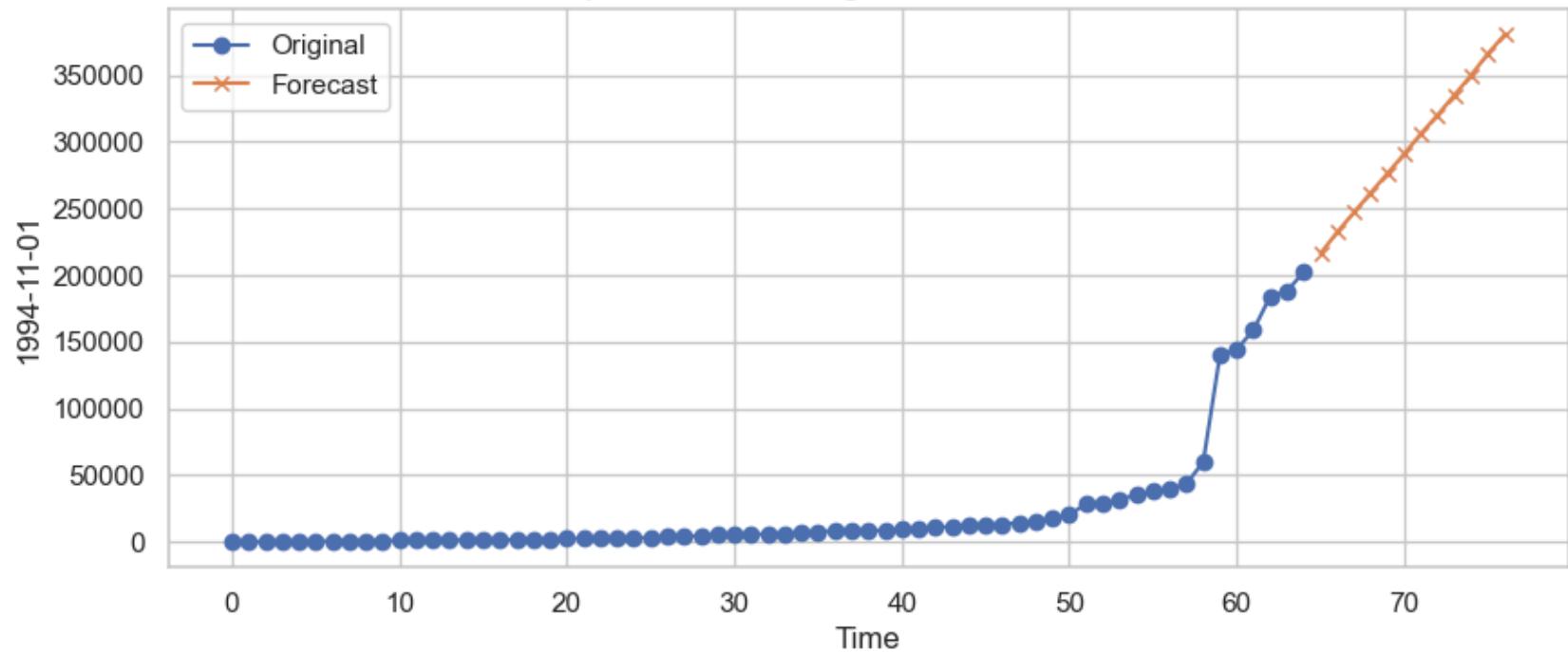


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



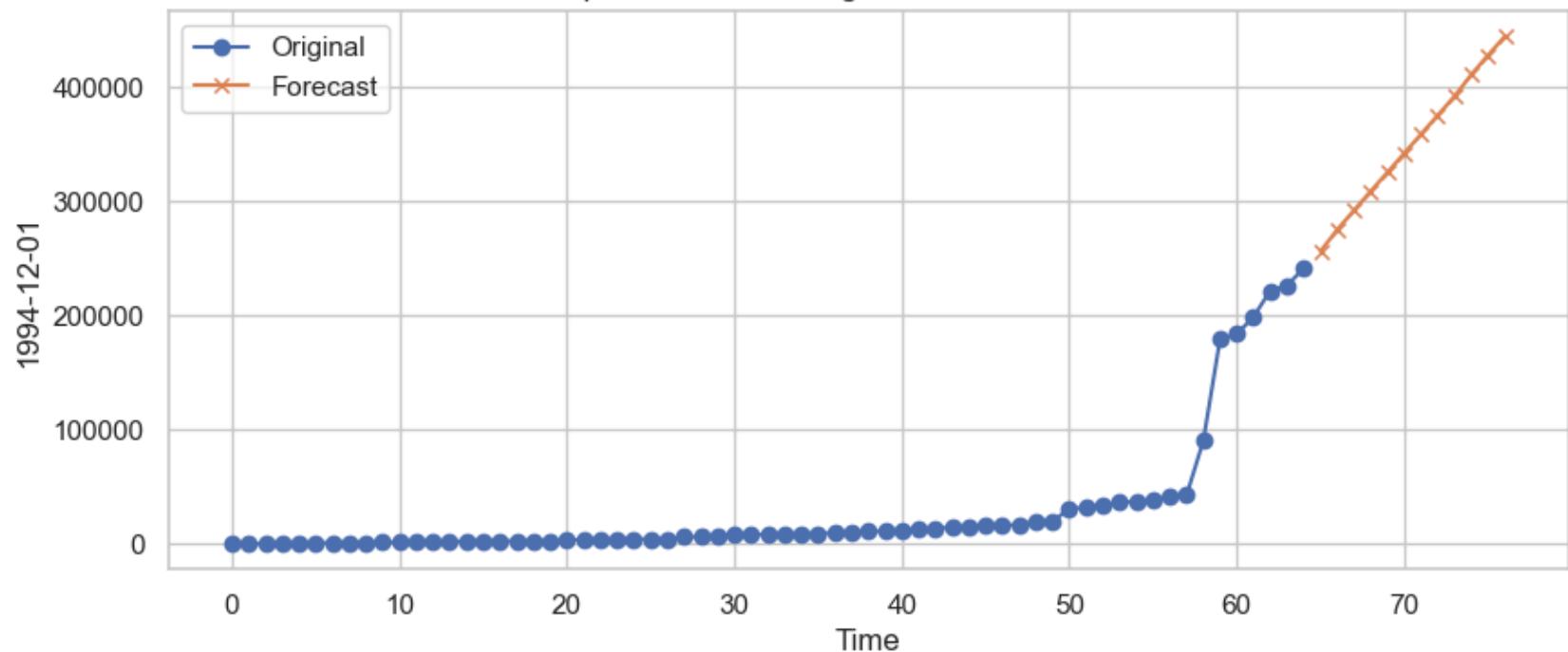
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1994-11-01

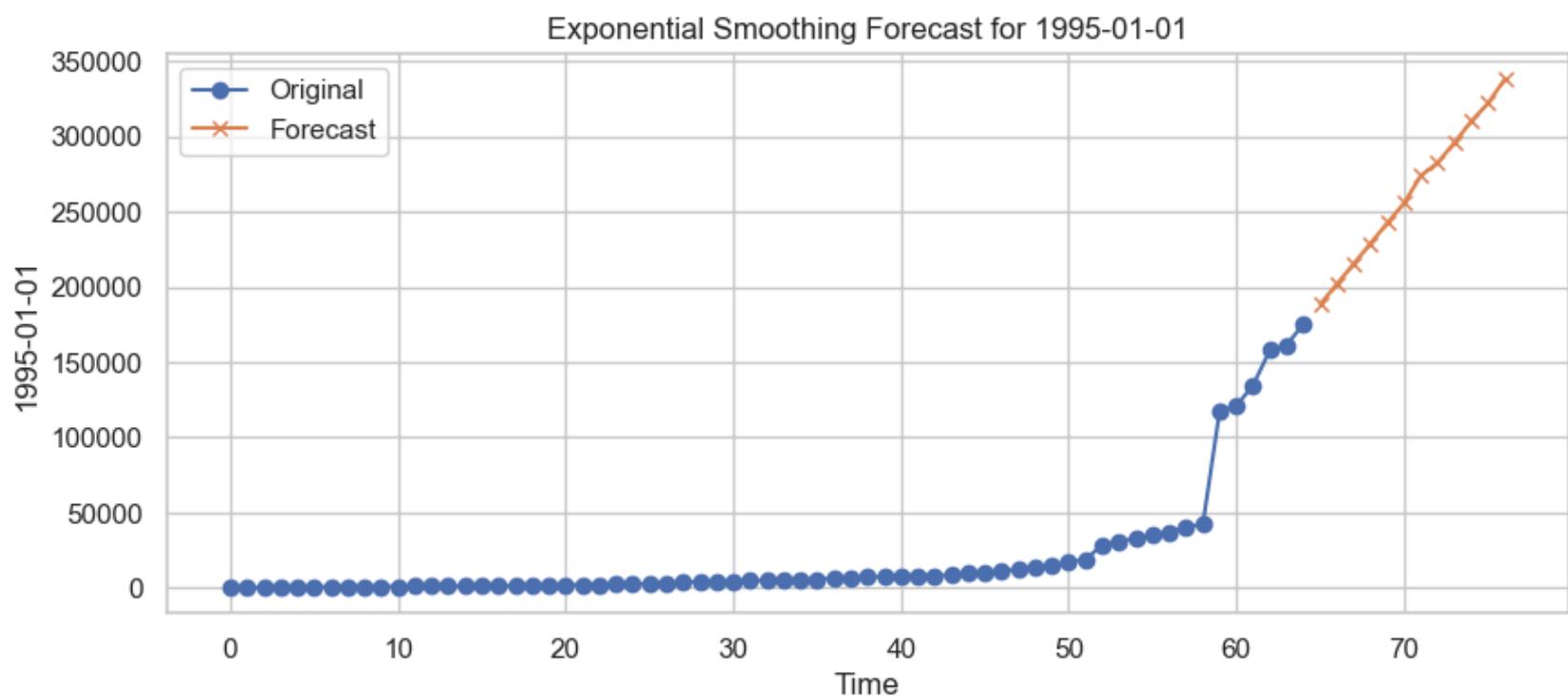


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

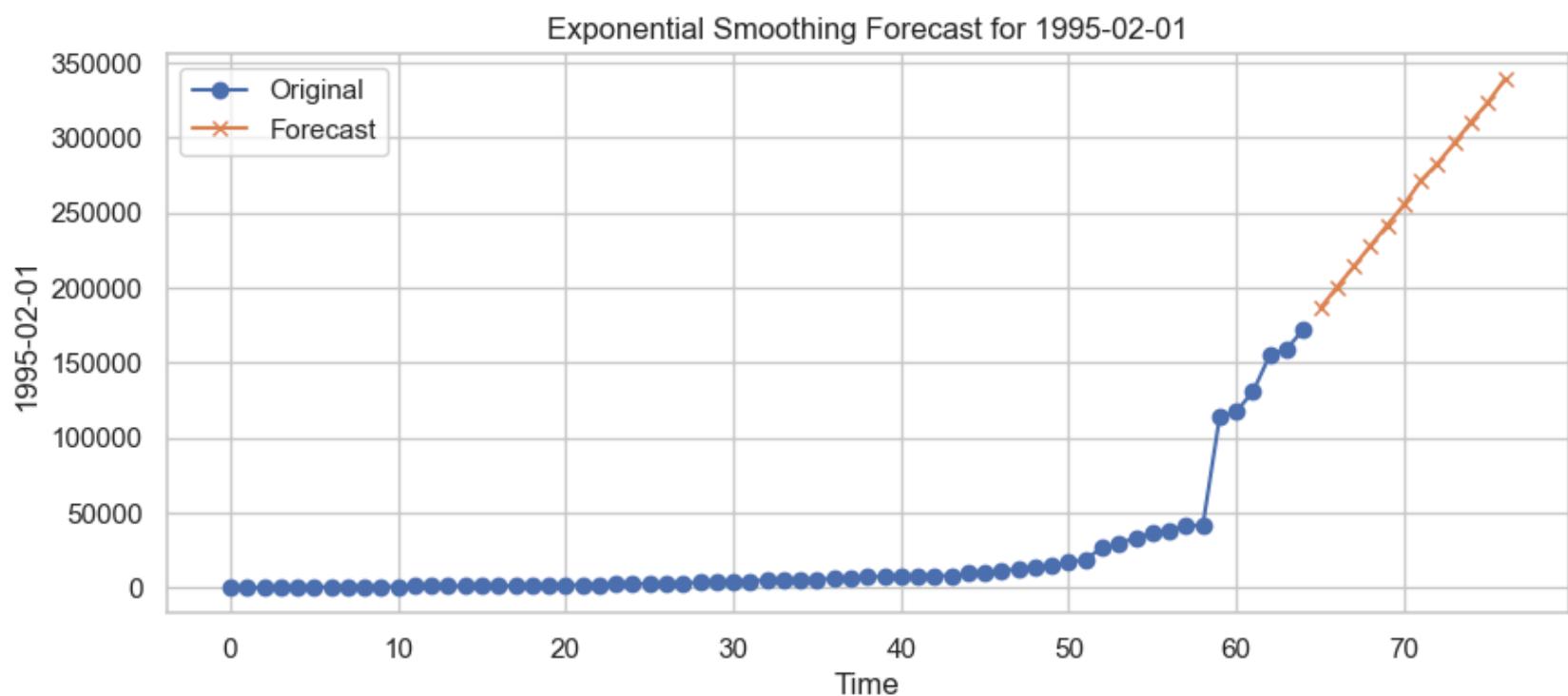
Exponential Smoothing Forecast for 1994-12-01



```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

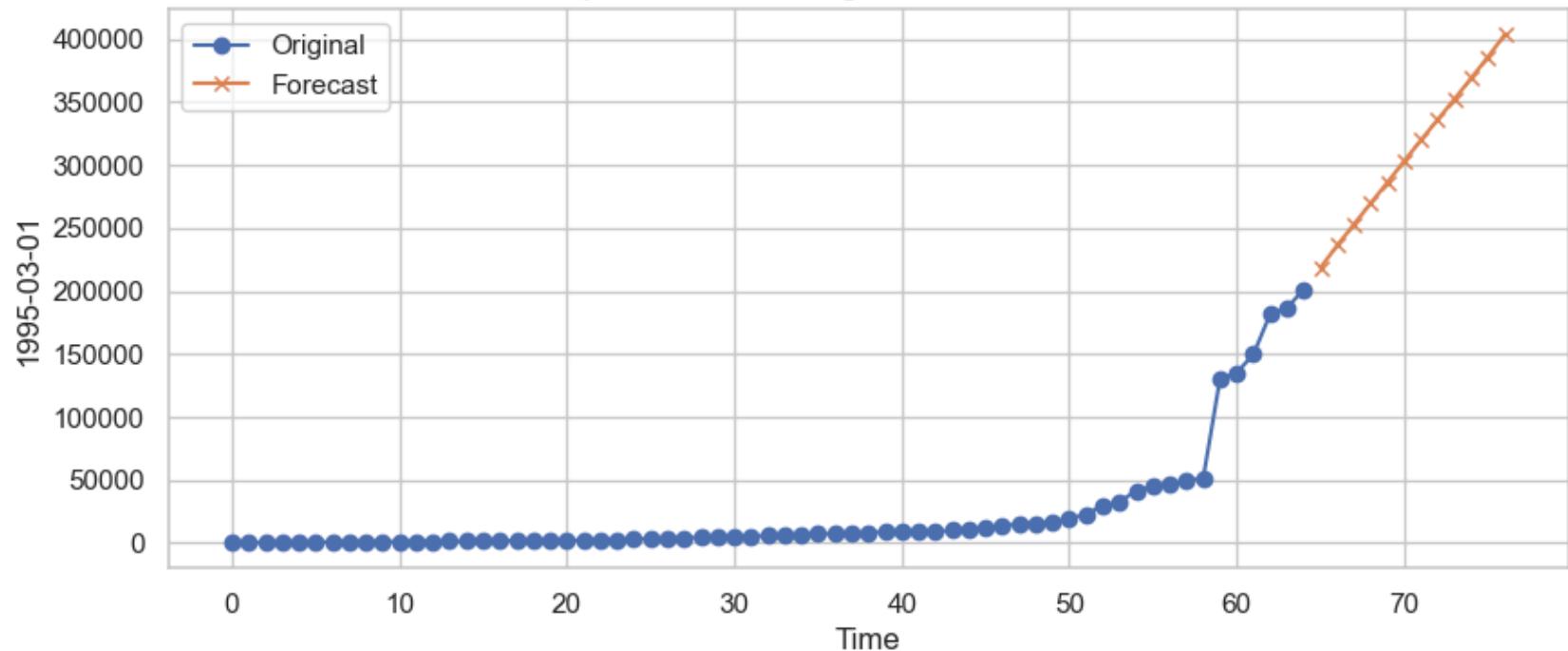


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

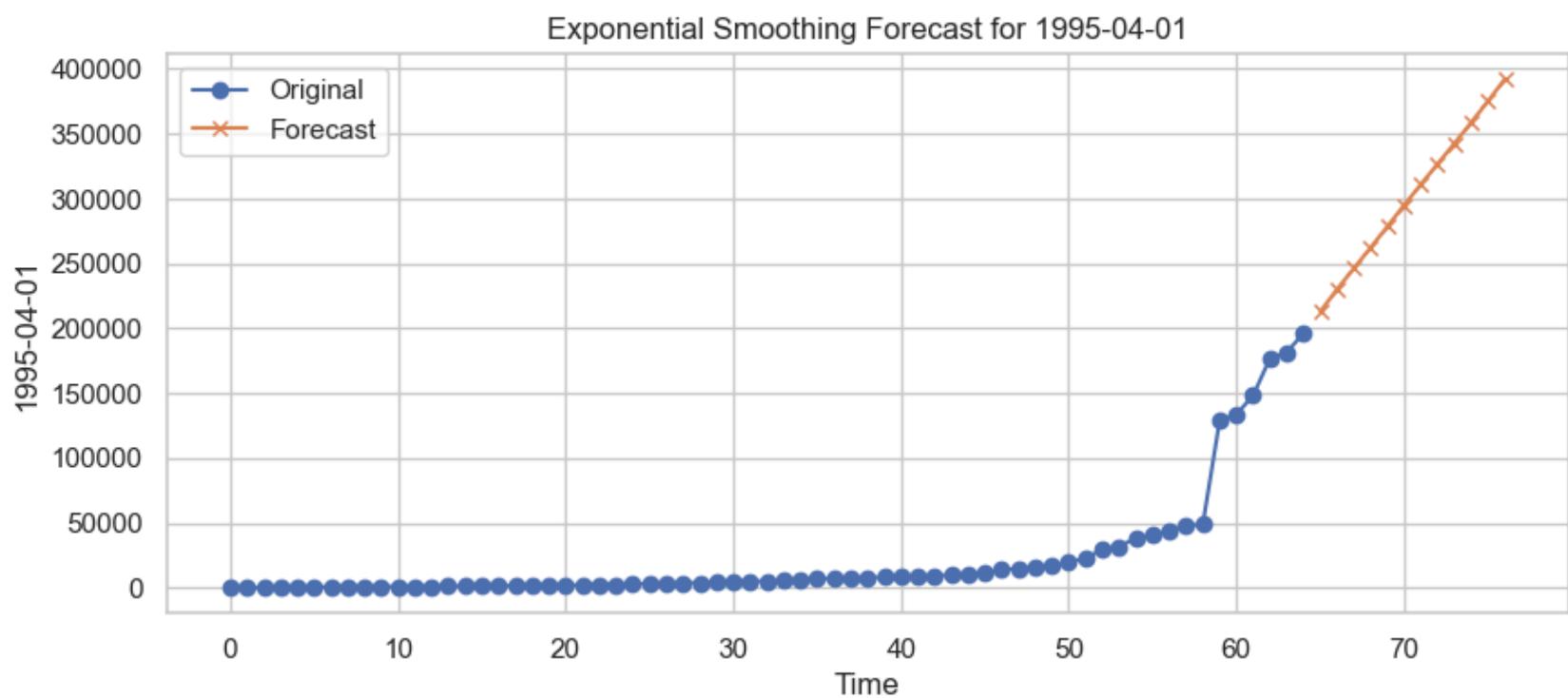


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1995-03-01

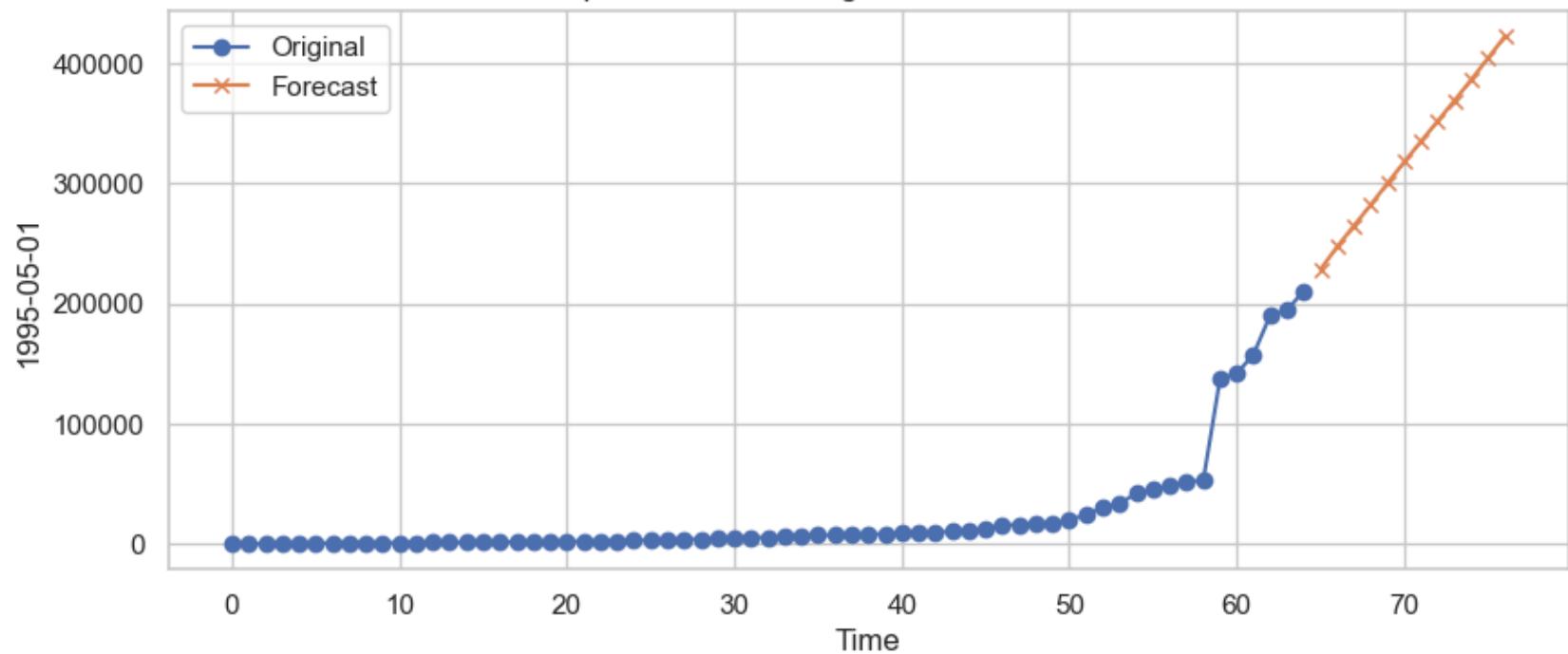


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



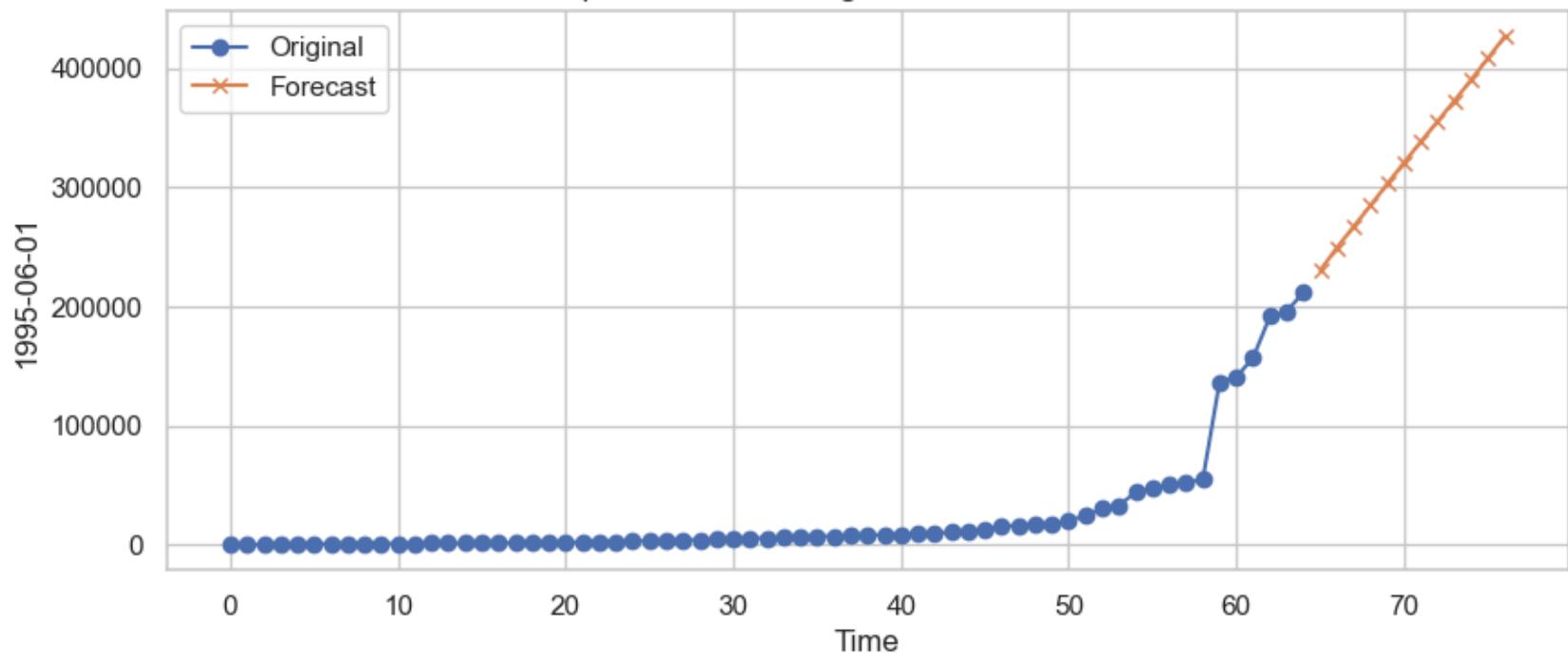
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1995-05-01



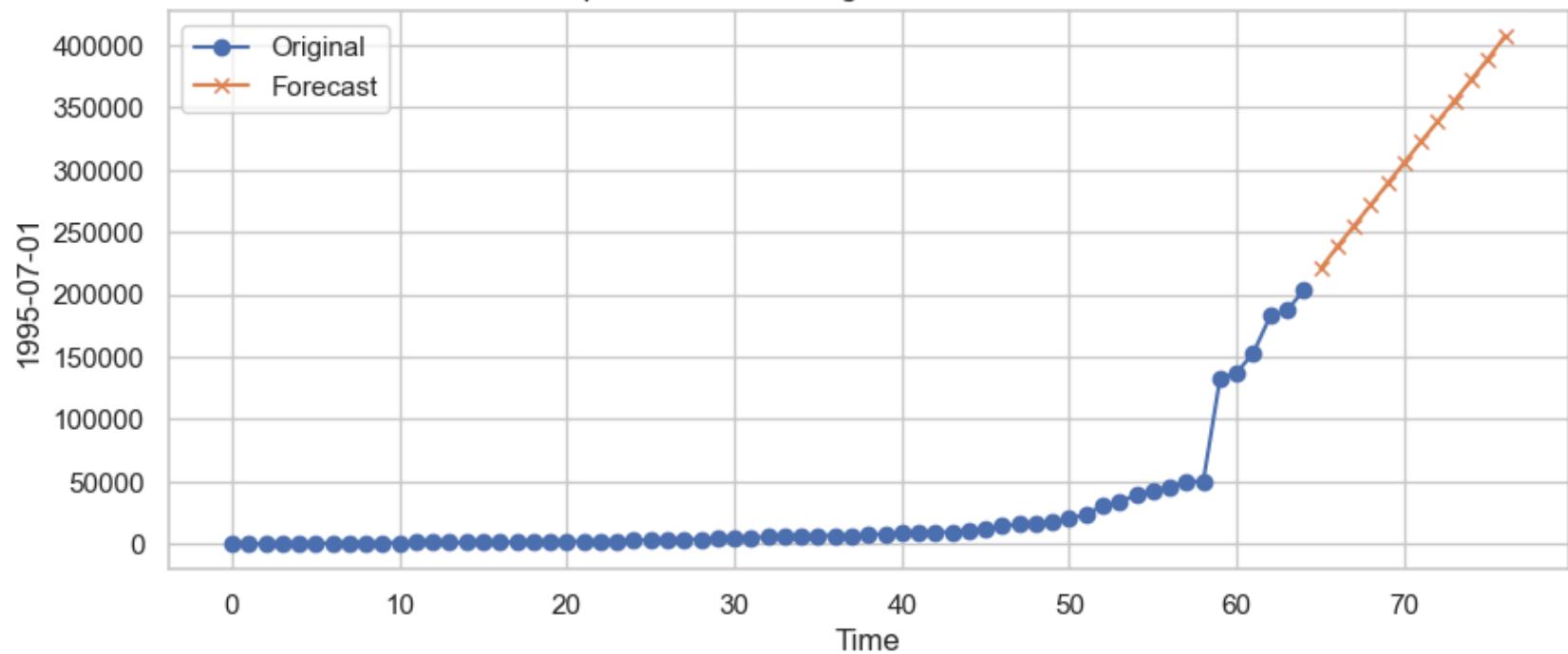
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1995-06-01



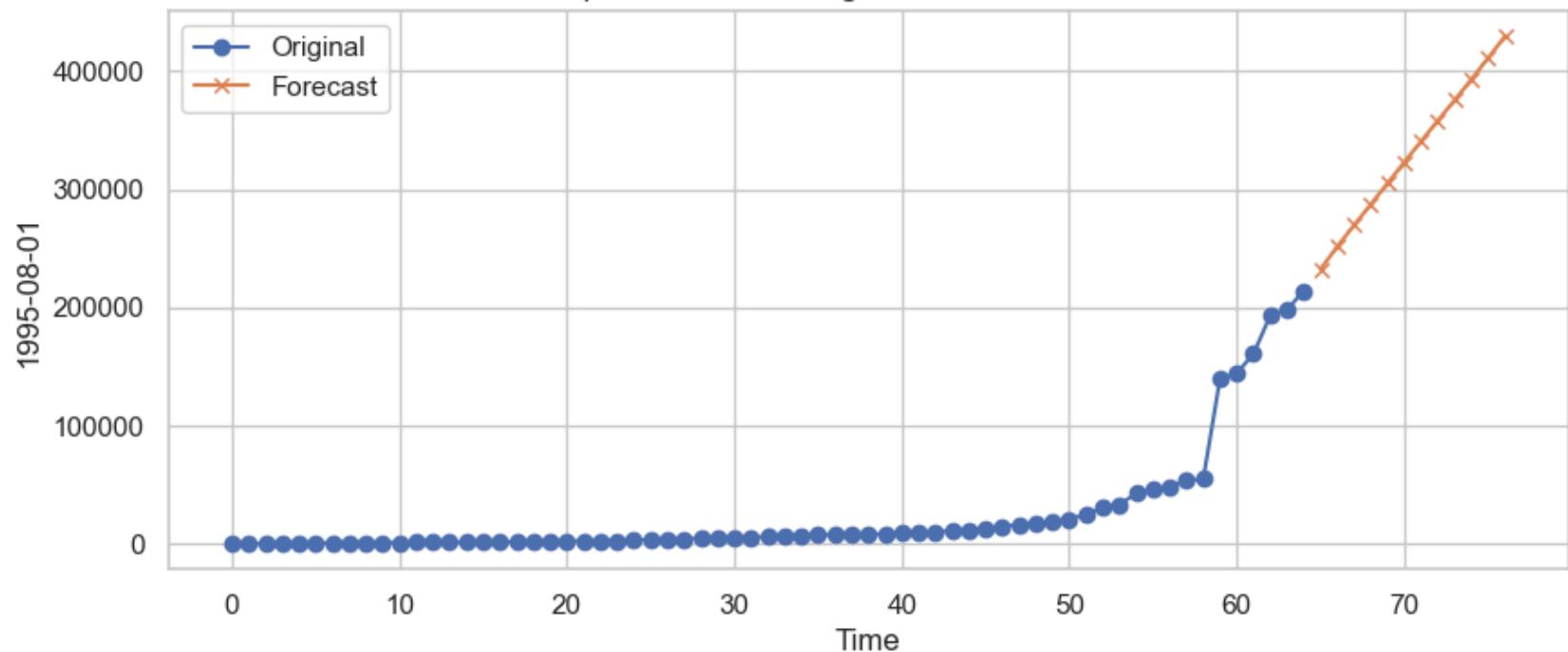
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1995-07-01



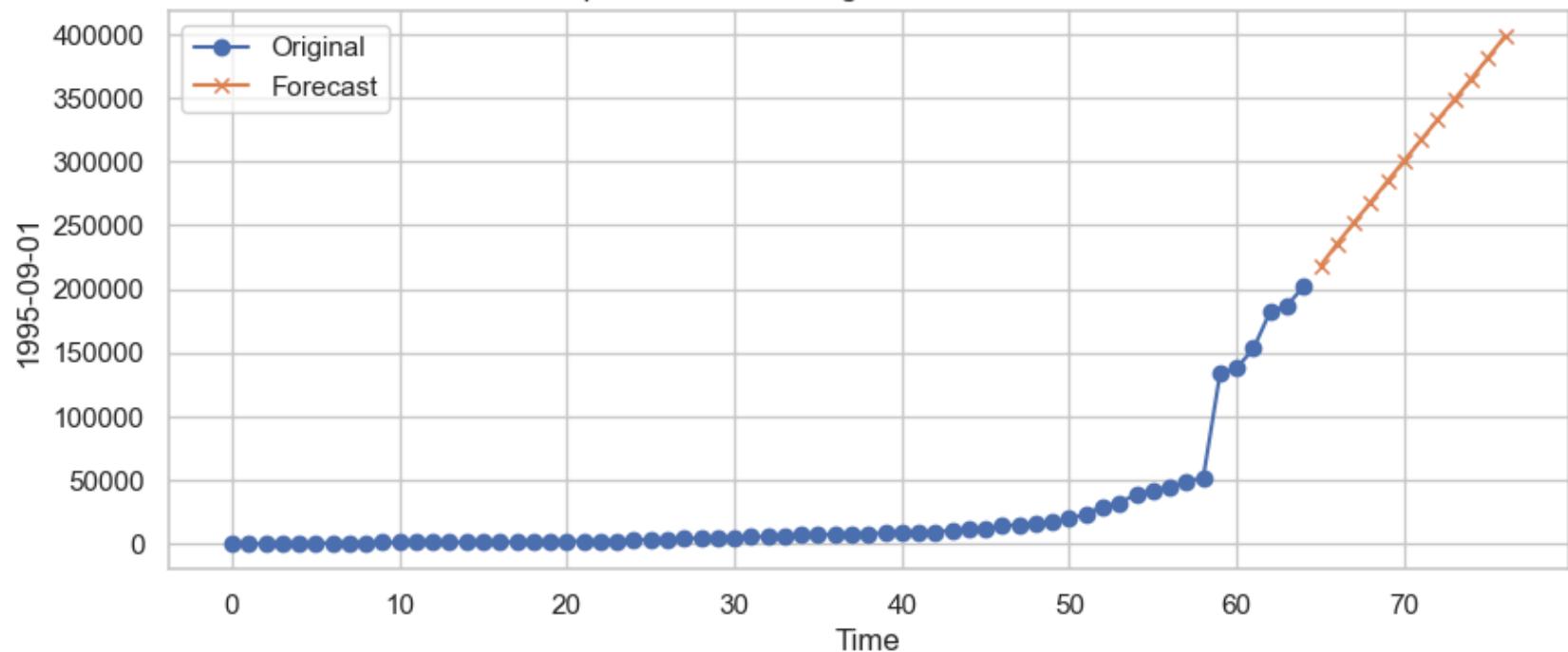
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1995-08-01



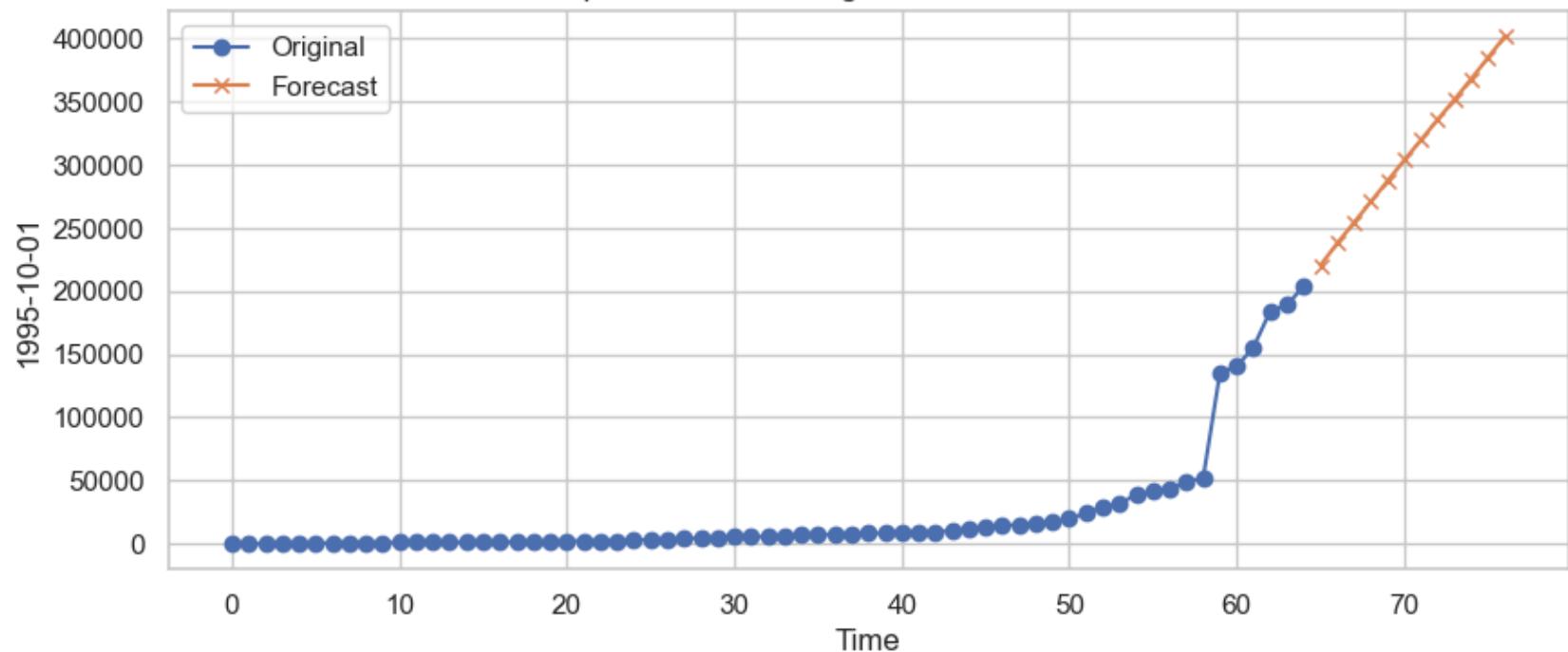
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1995-09-01



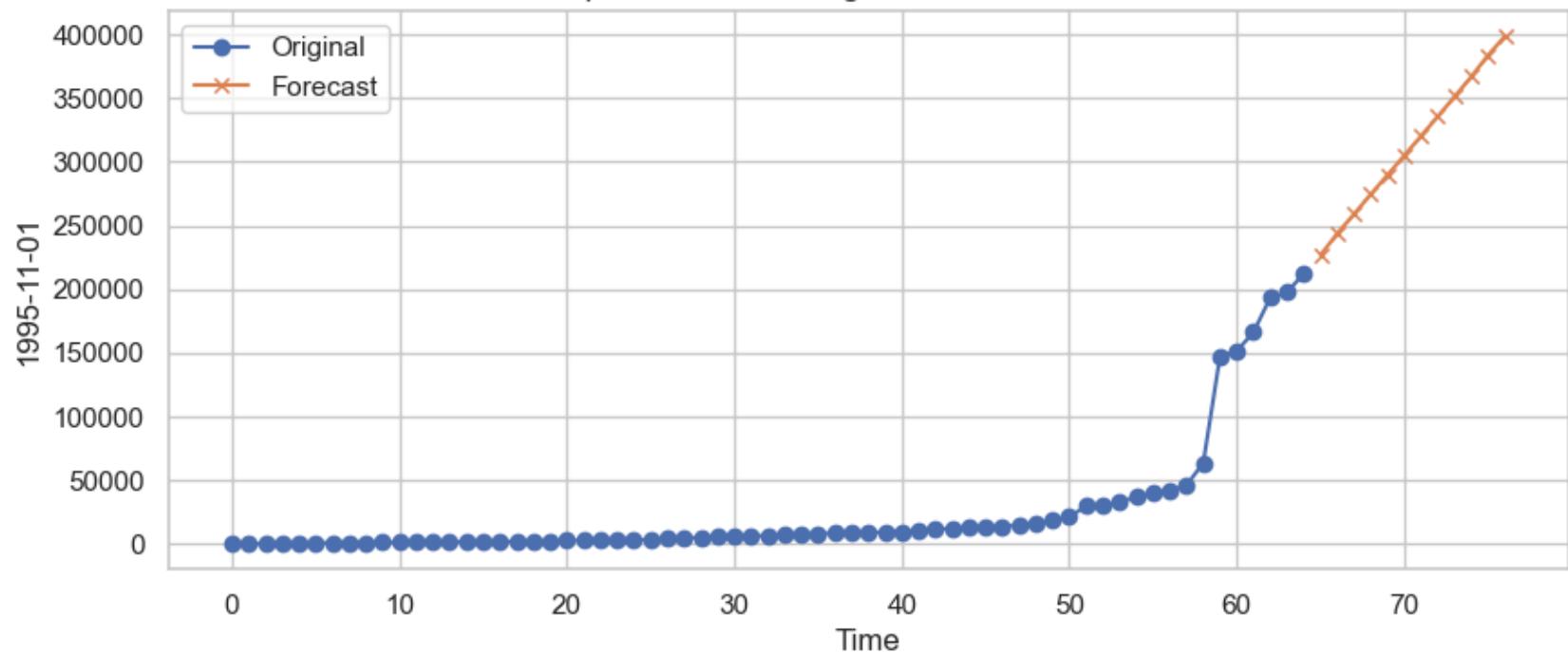
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1995-10-01



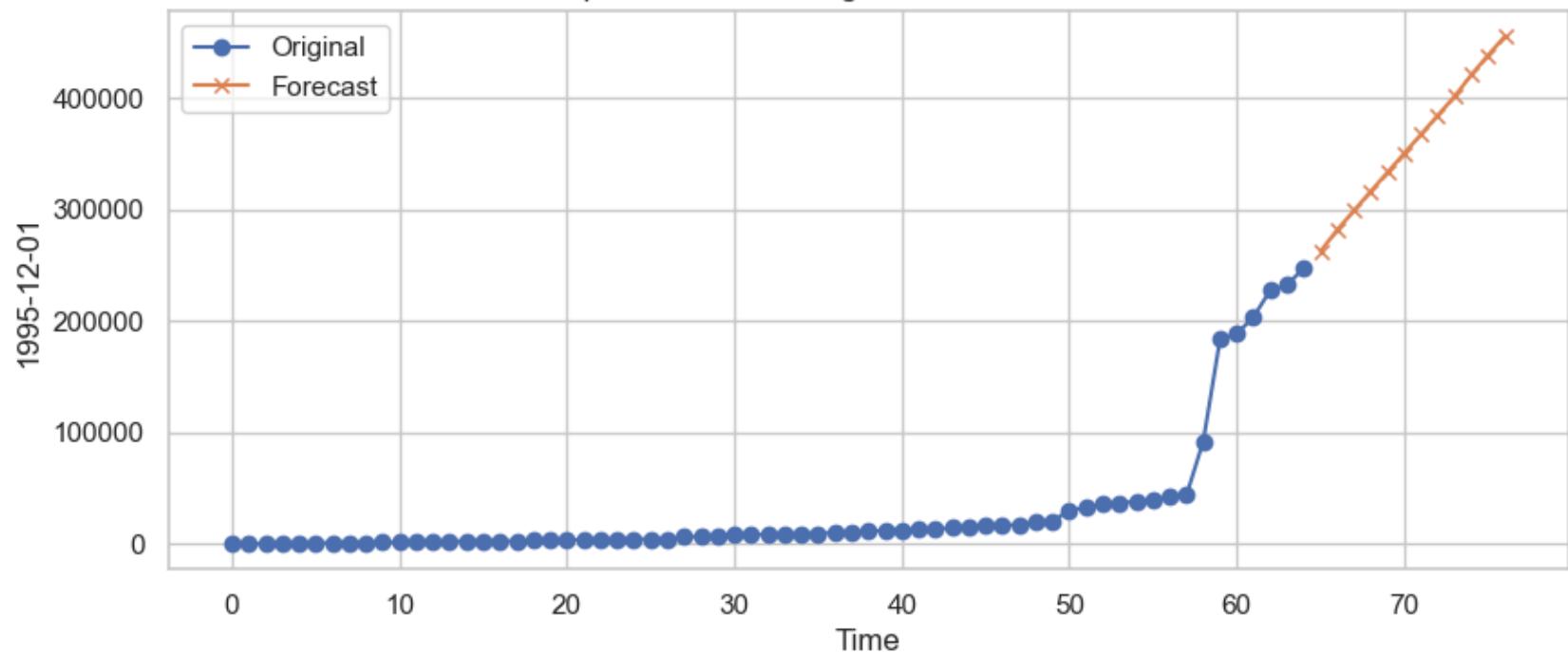
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1995-11-01



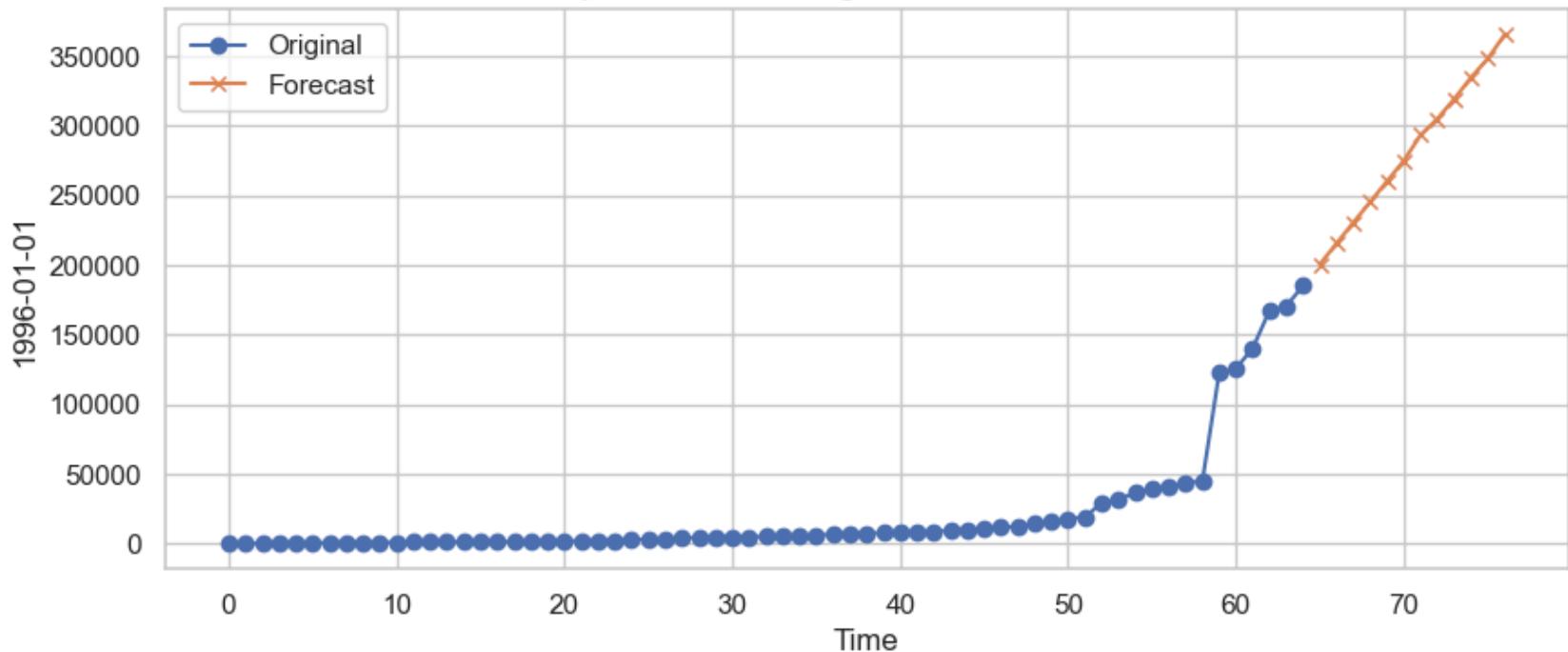
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1995-12-01



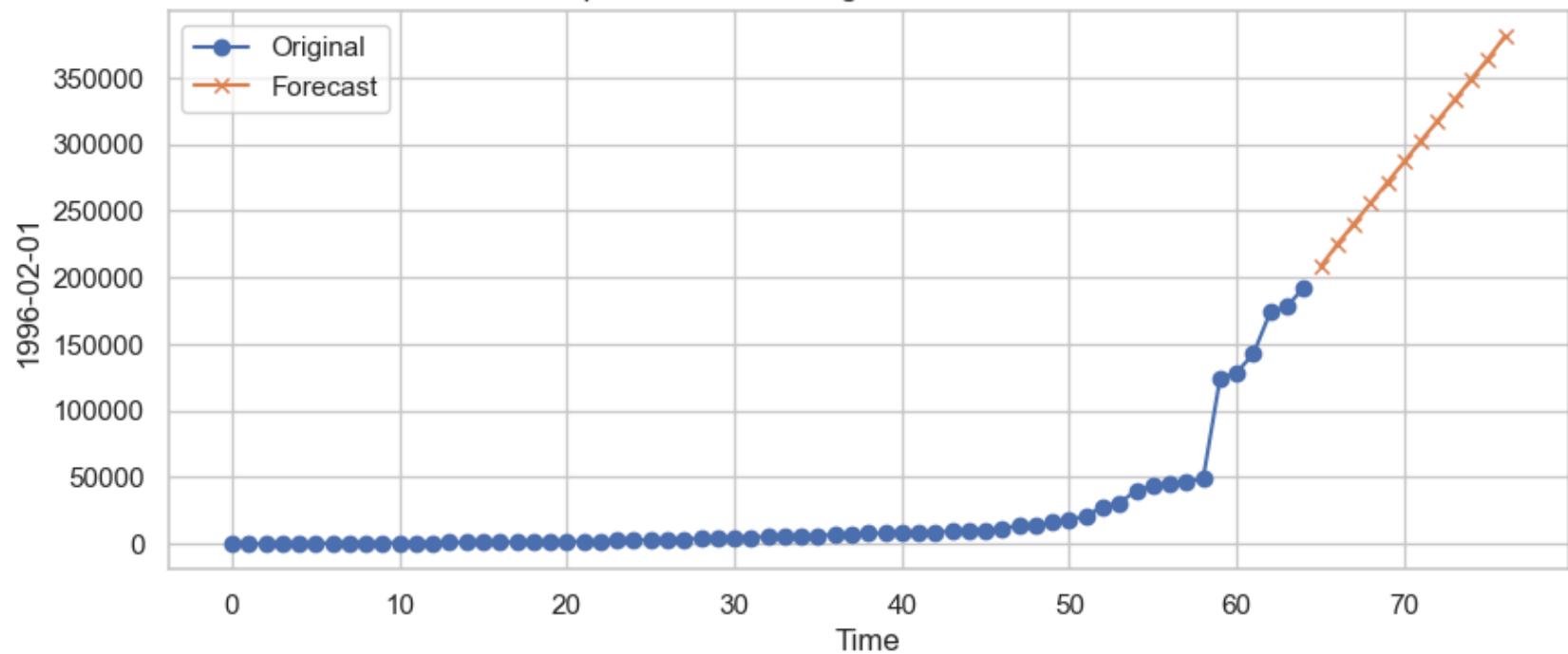
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-01-01



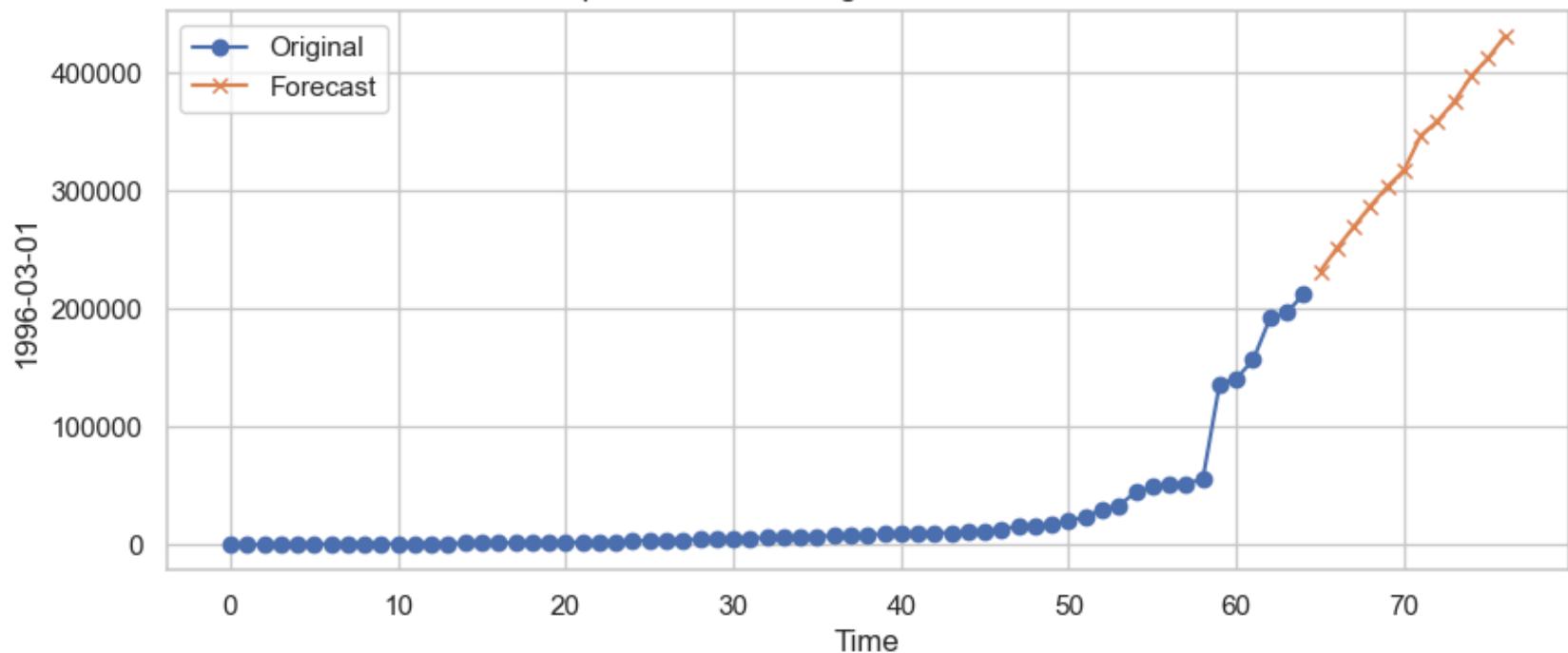
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-02-01



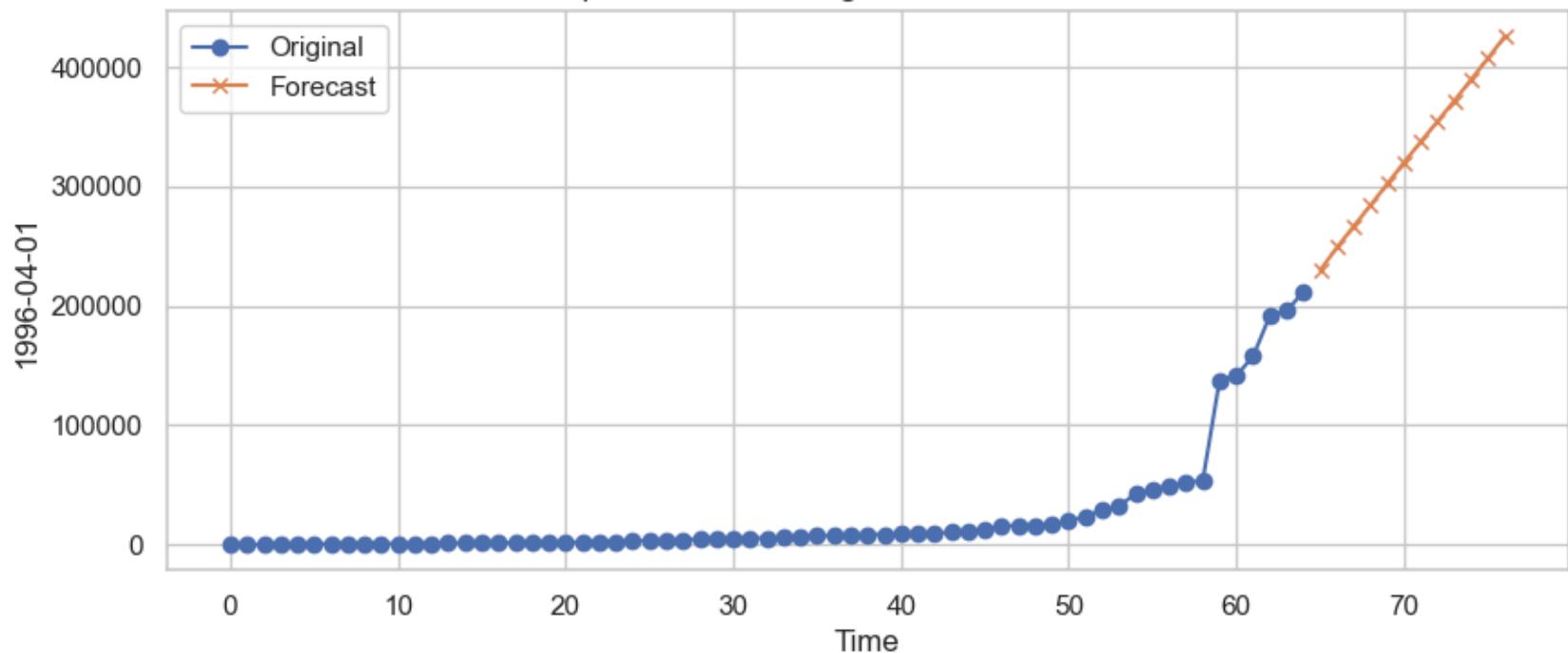
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-03-01



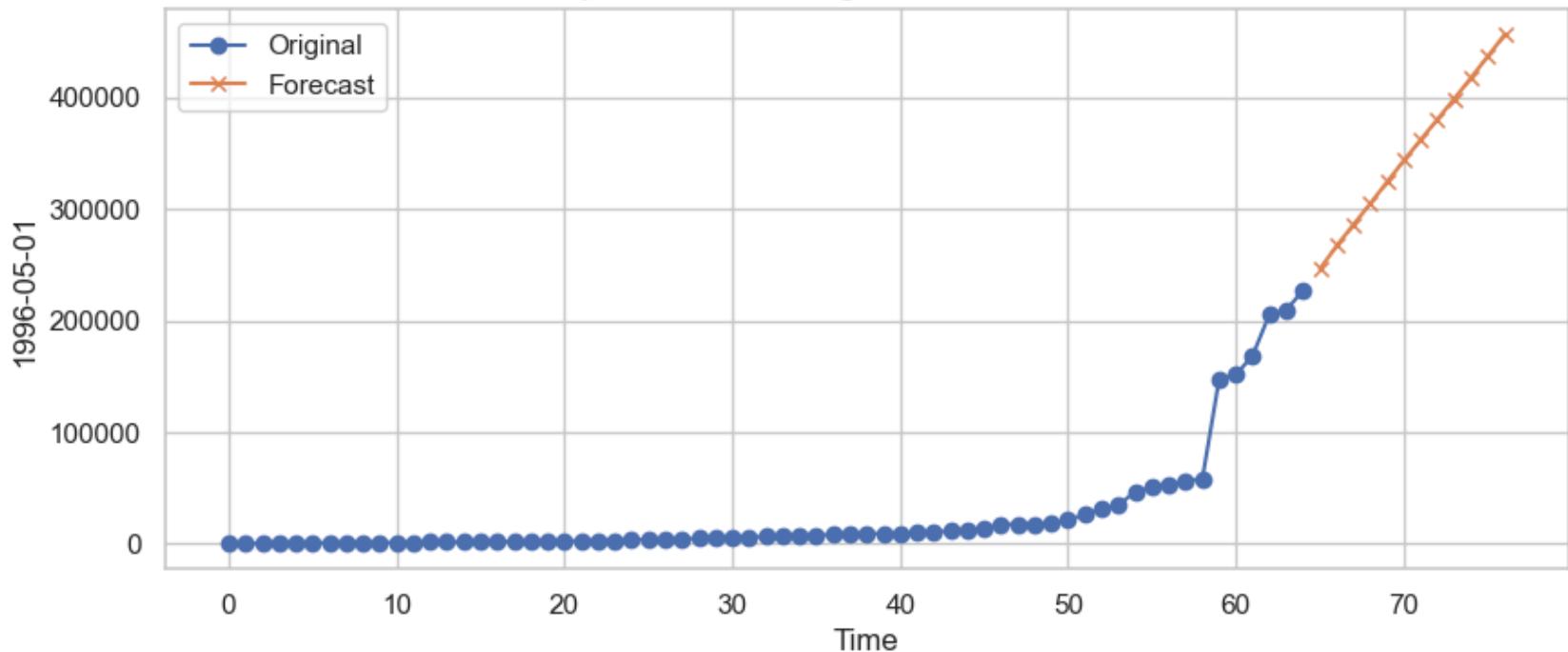
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-04-01



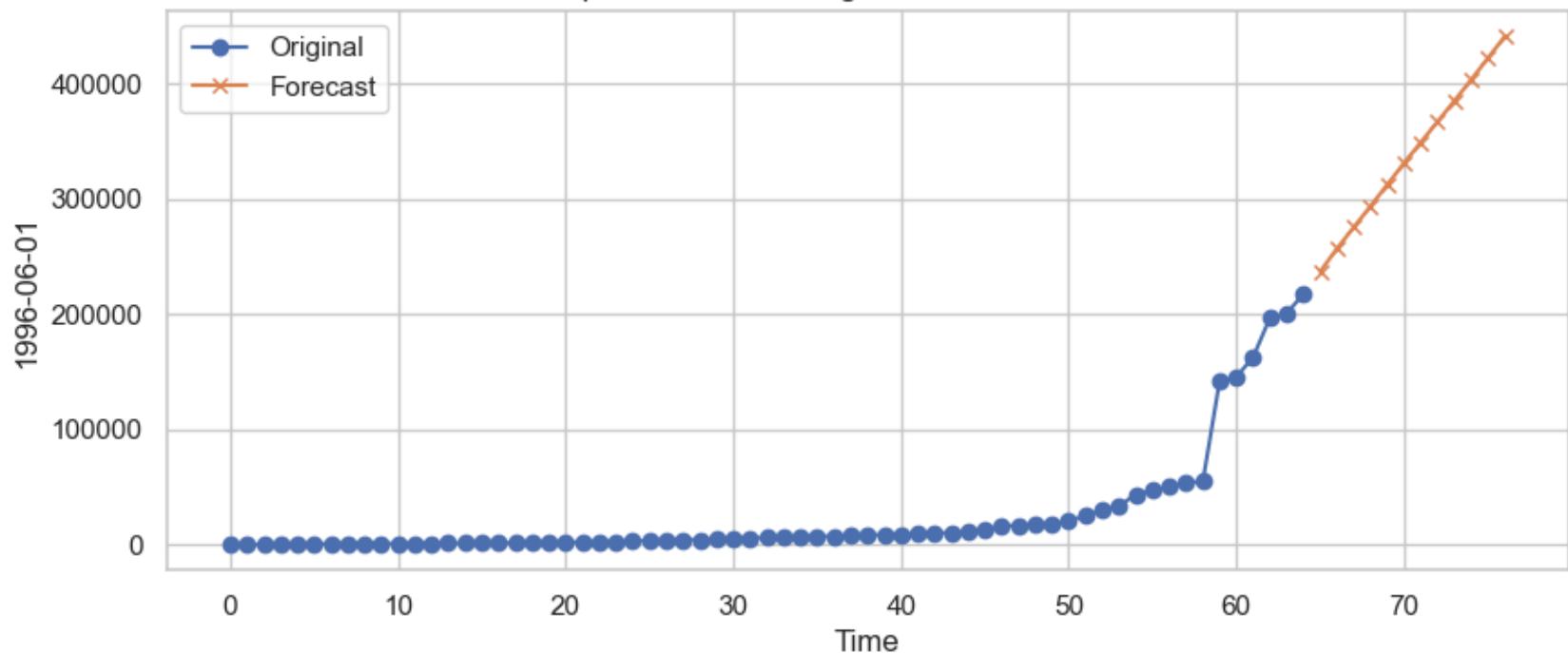
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-05-01



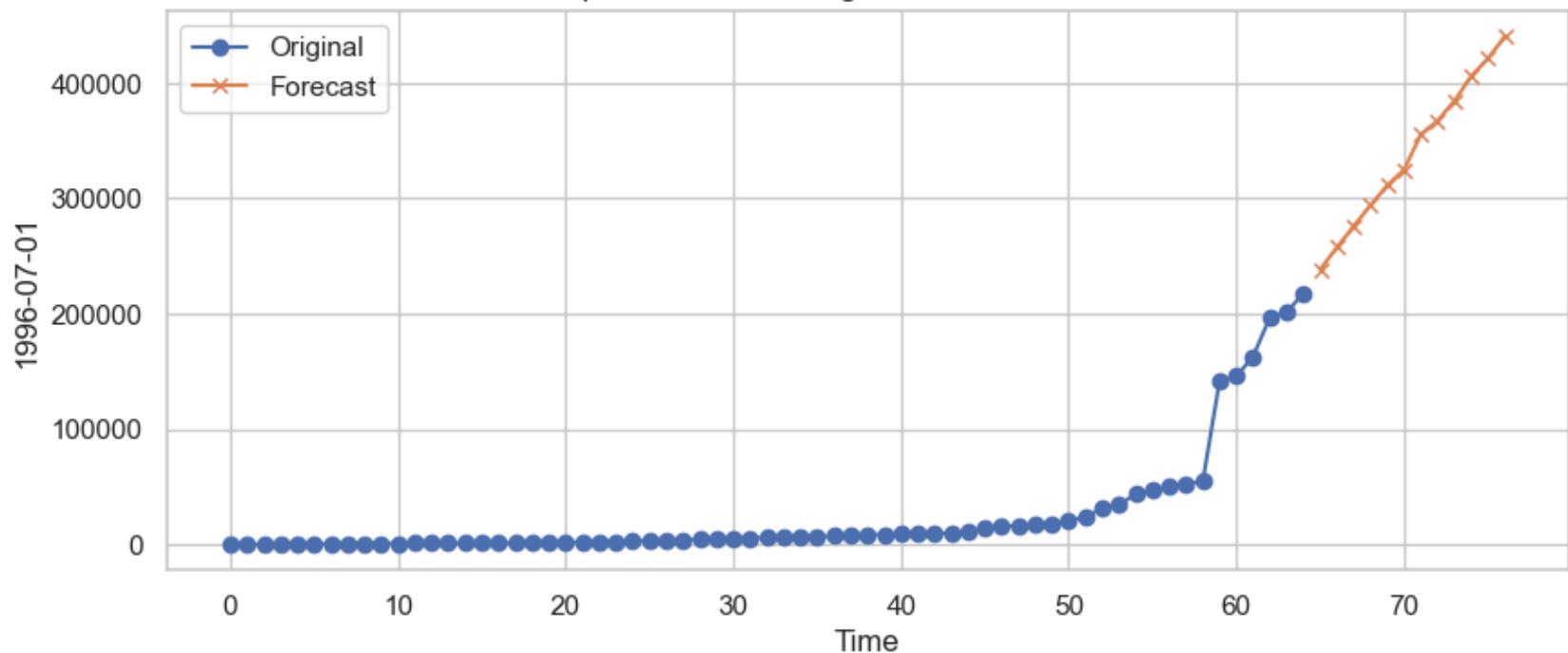
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-06-01



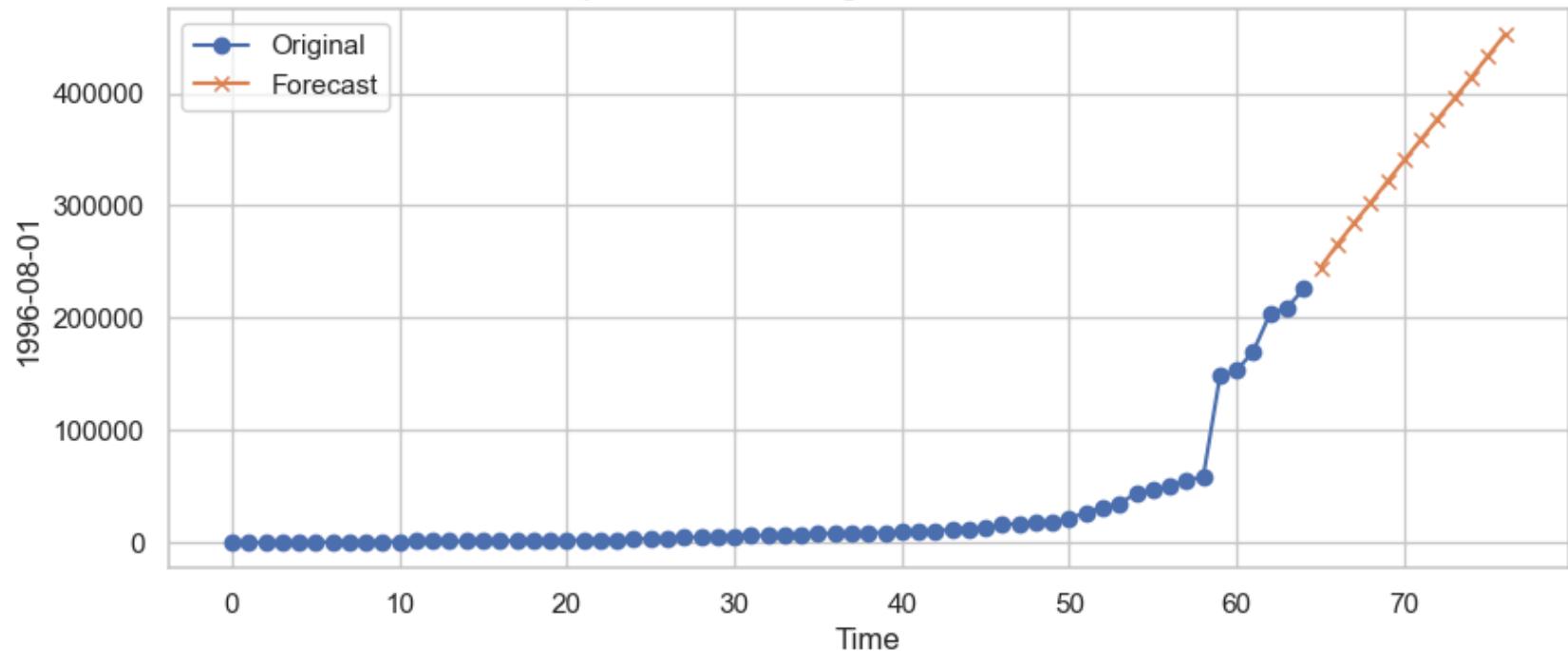
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-07-01



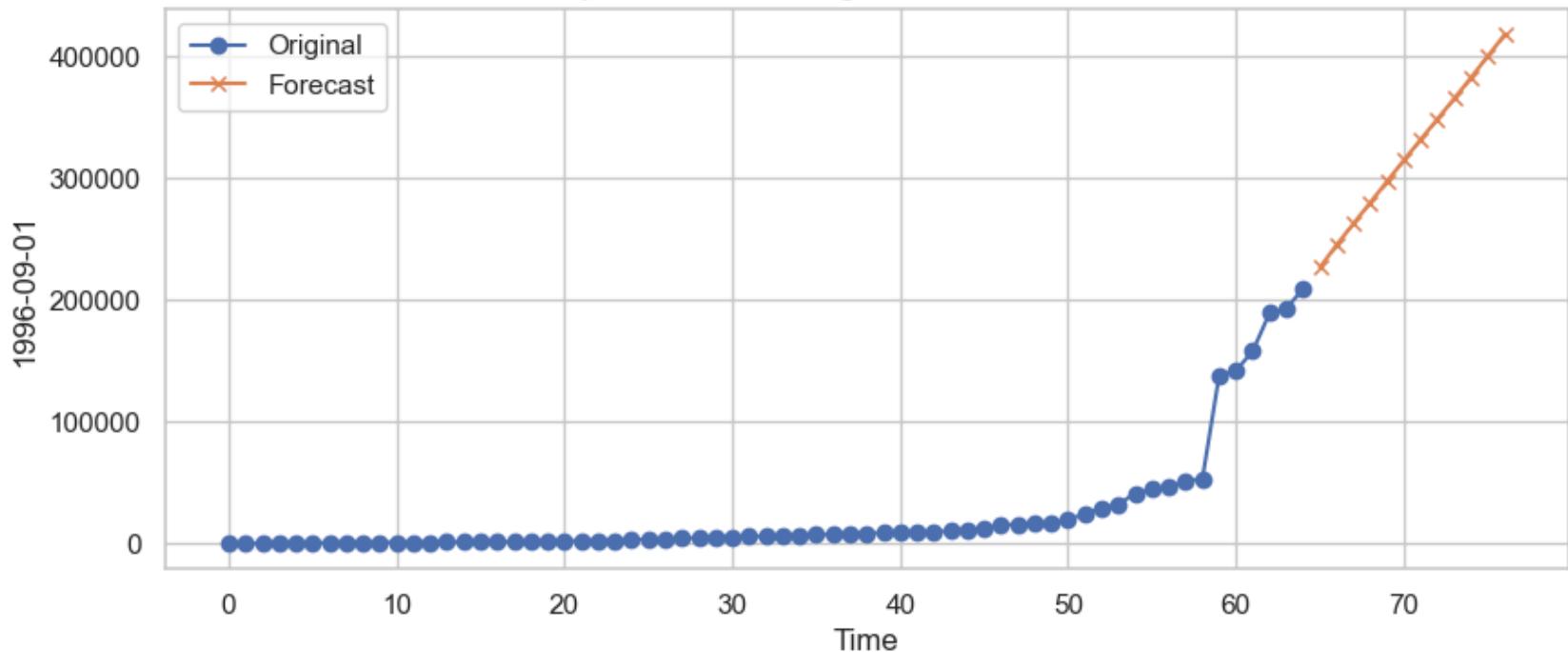
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-08-01



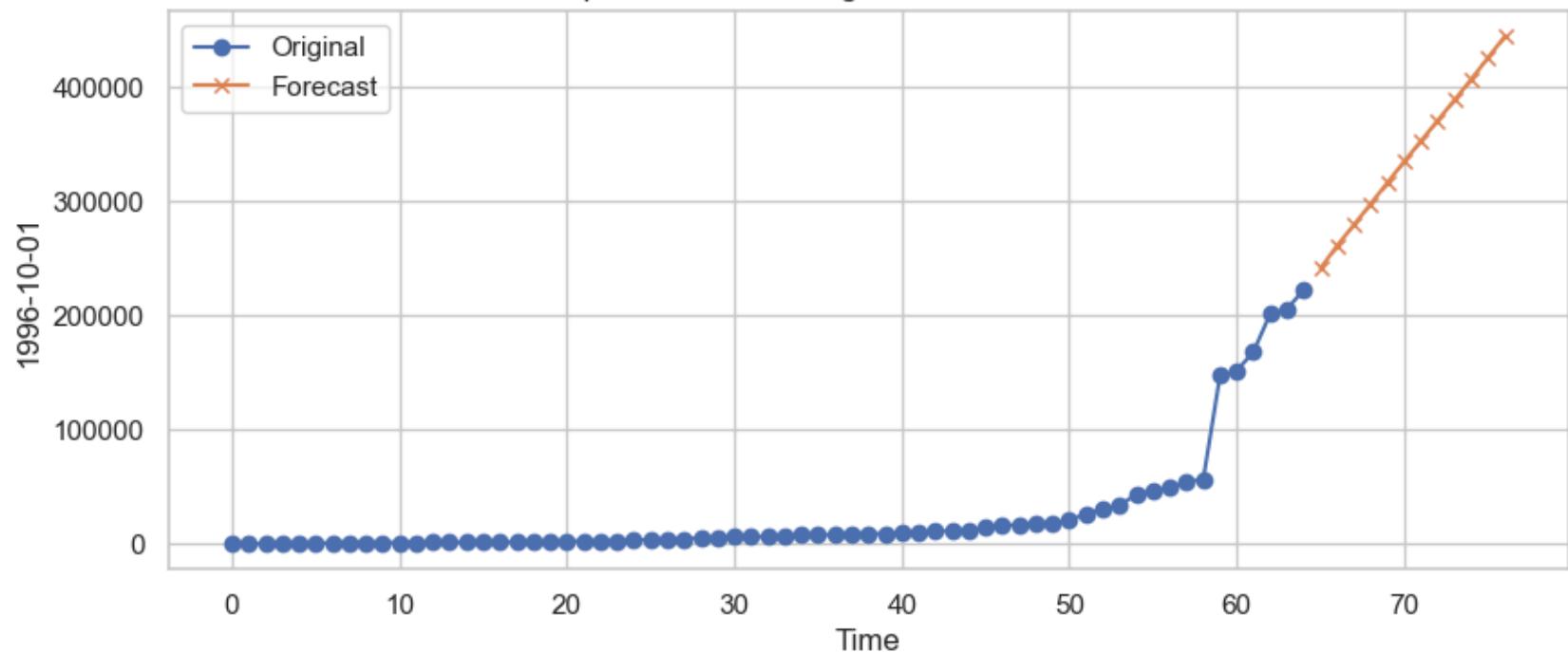
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-09-01



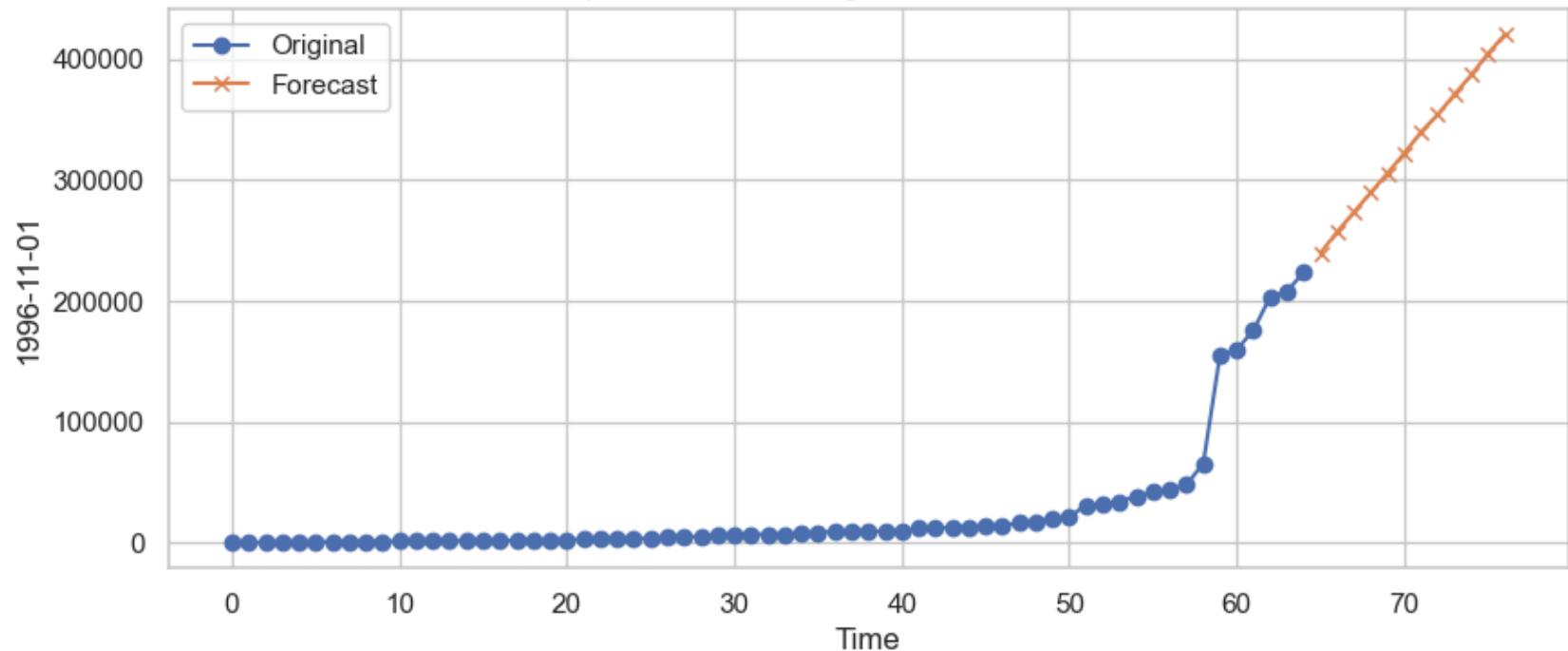
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-10-01



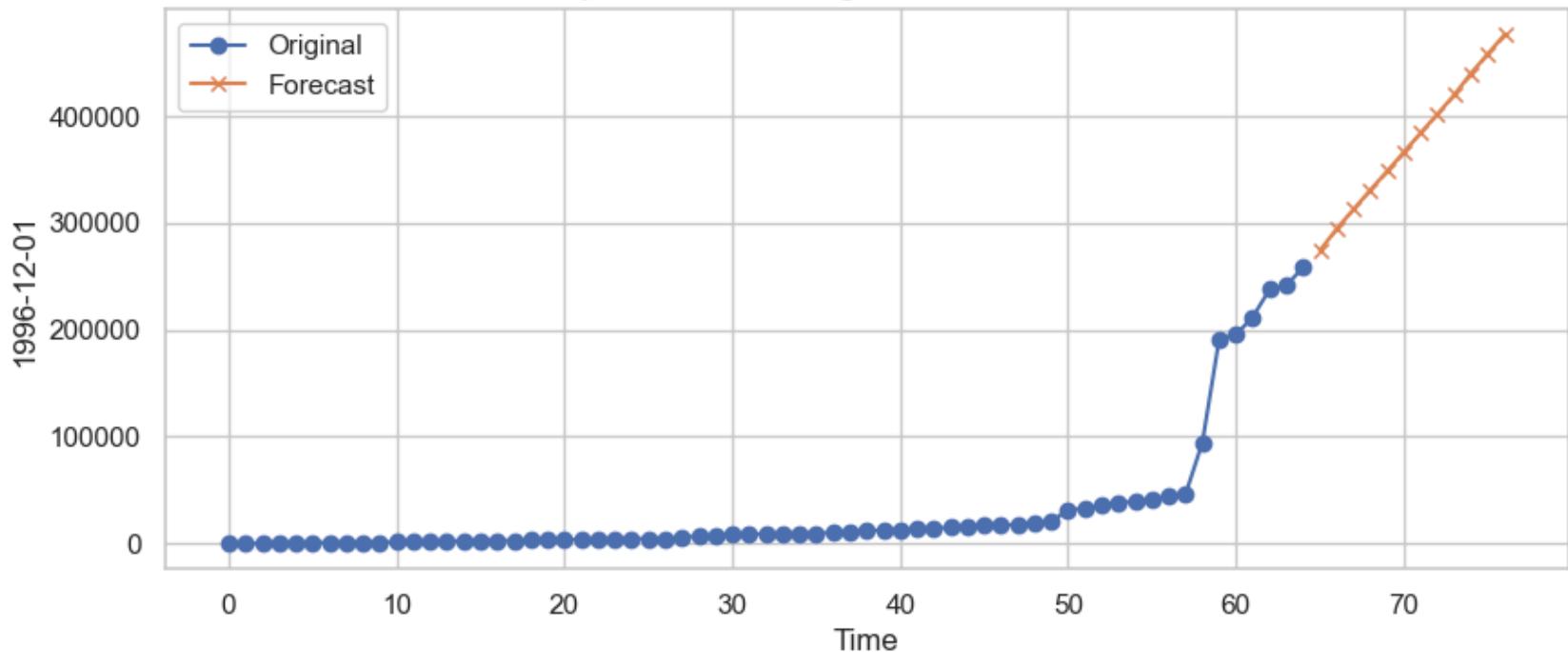
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-11-01

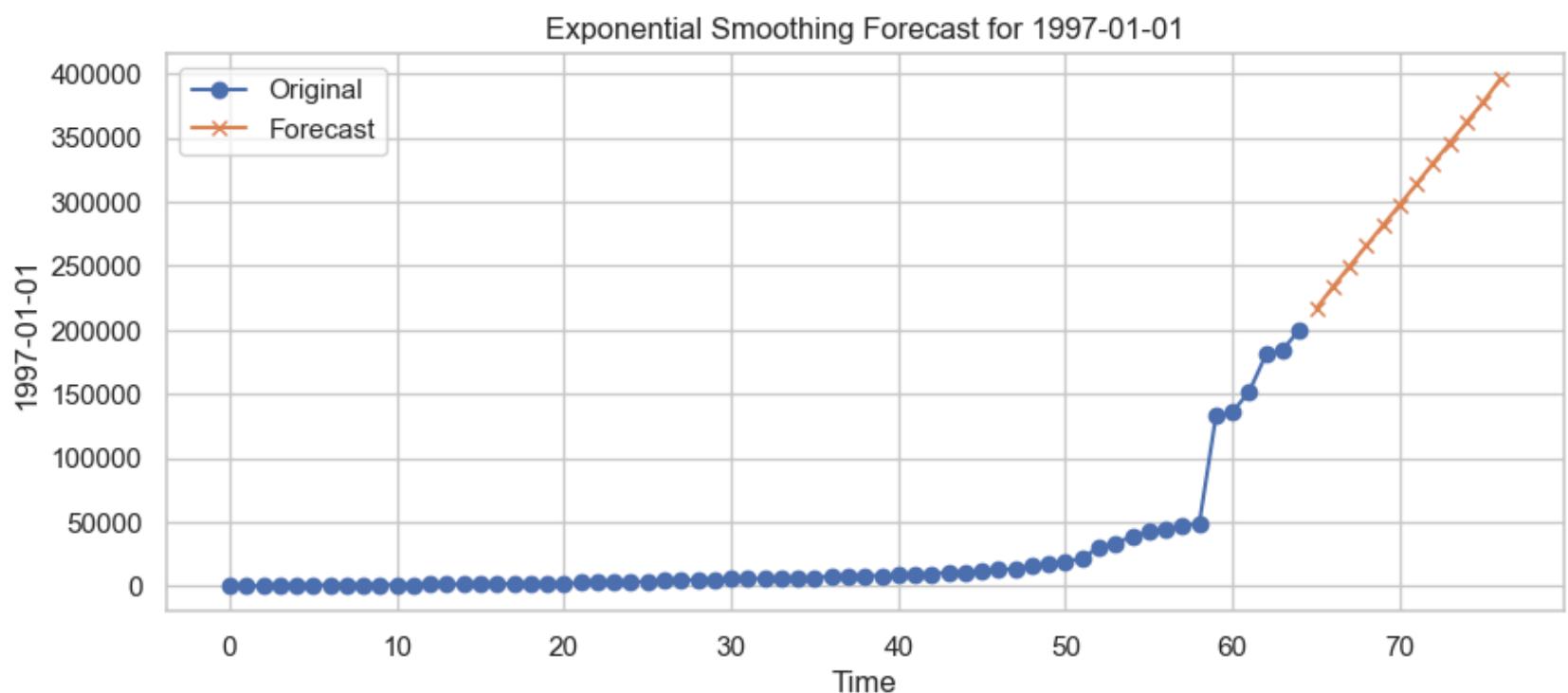


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1996-12-01

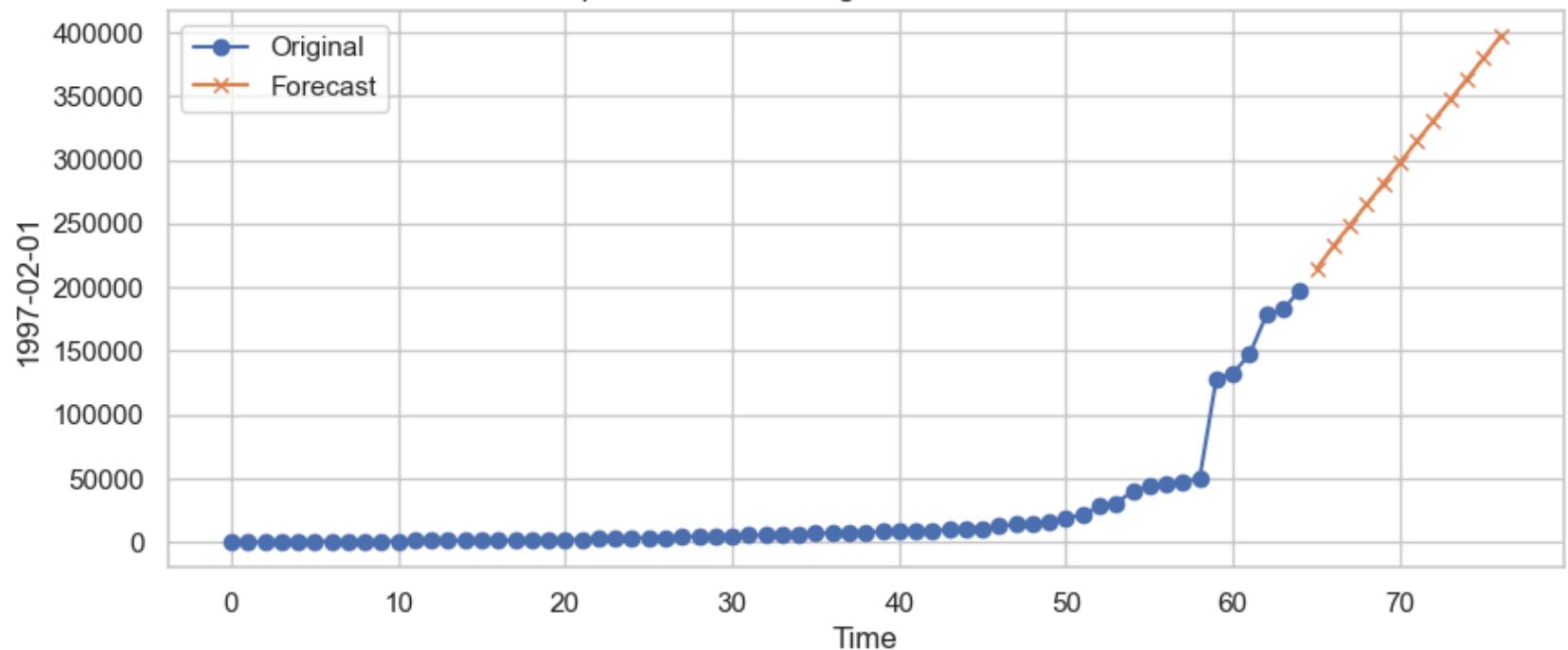


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



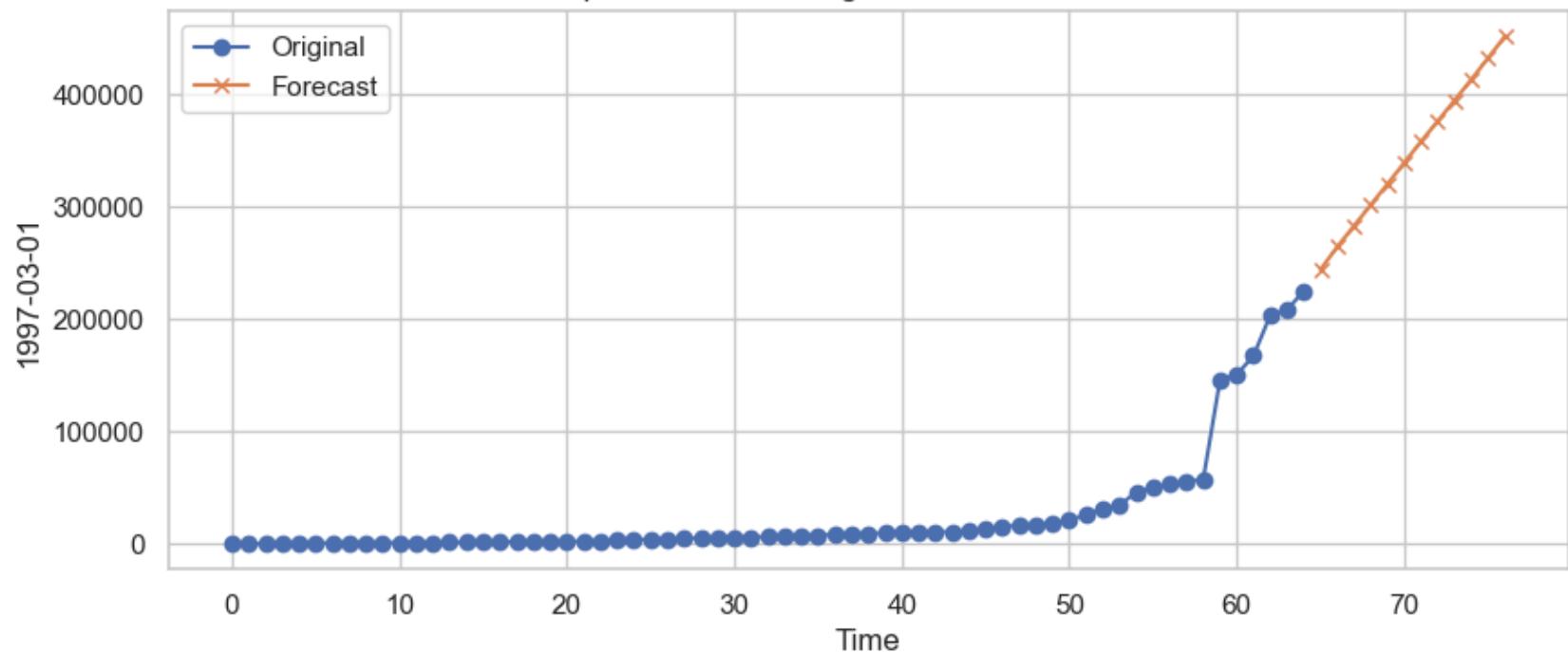
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1997-02-01



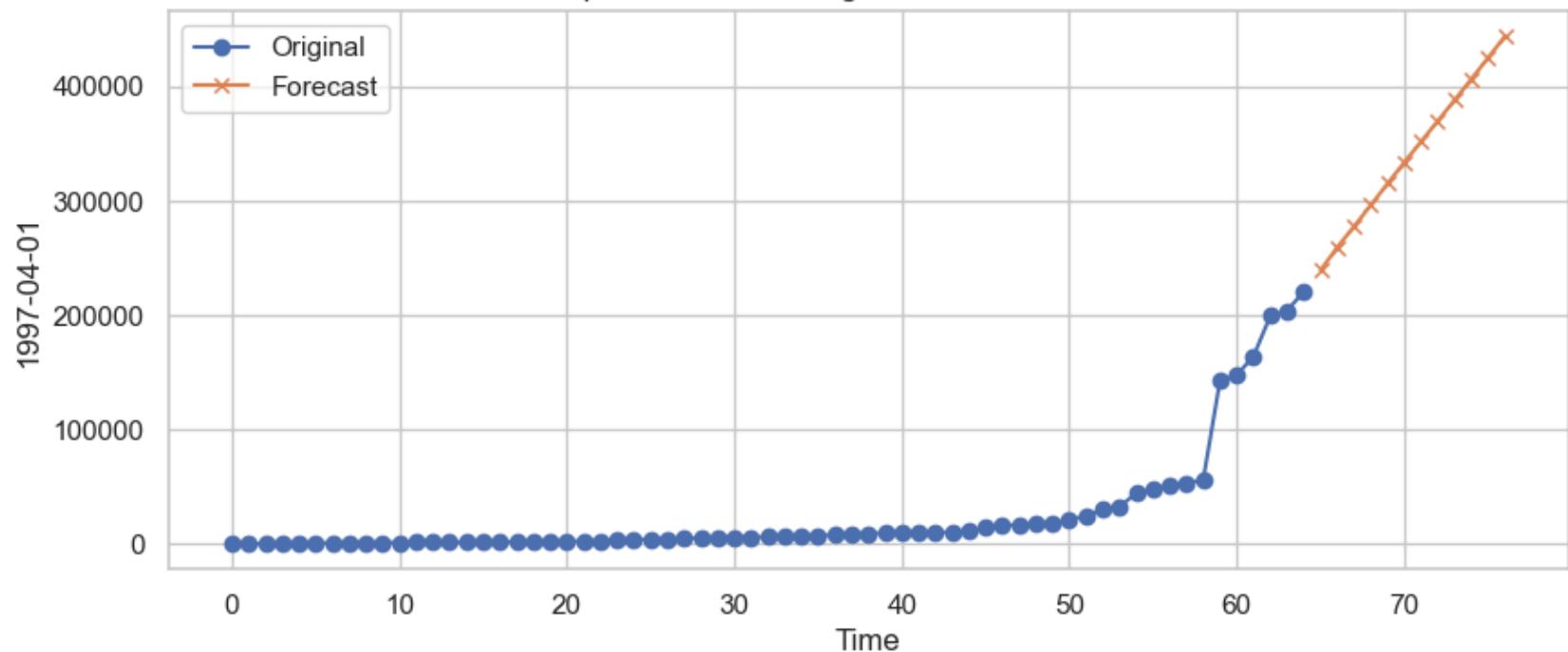
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1997-03-01



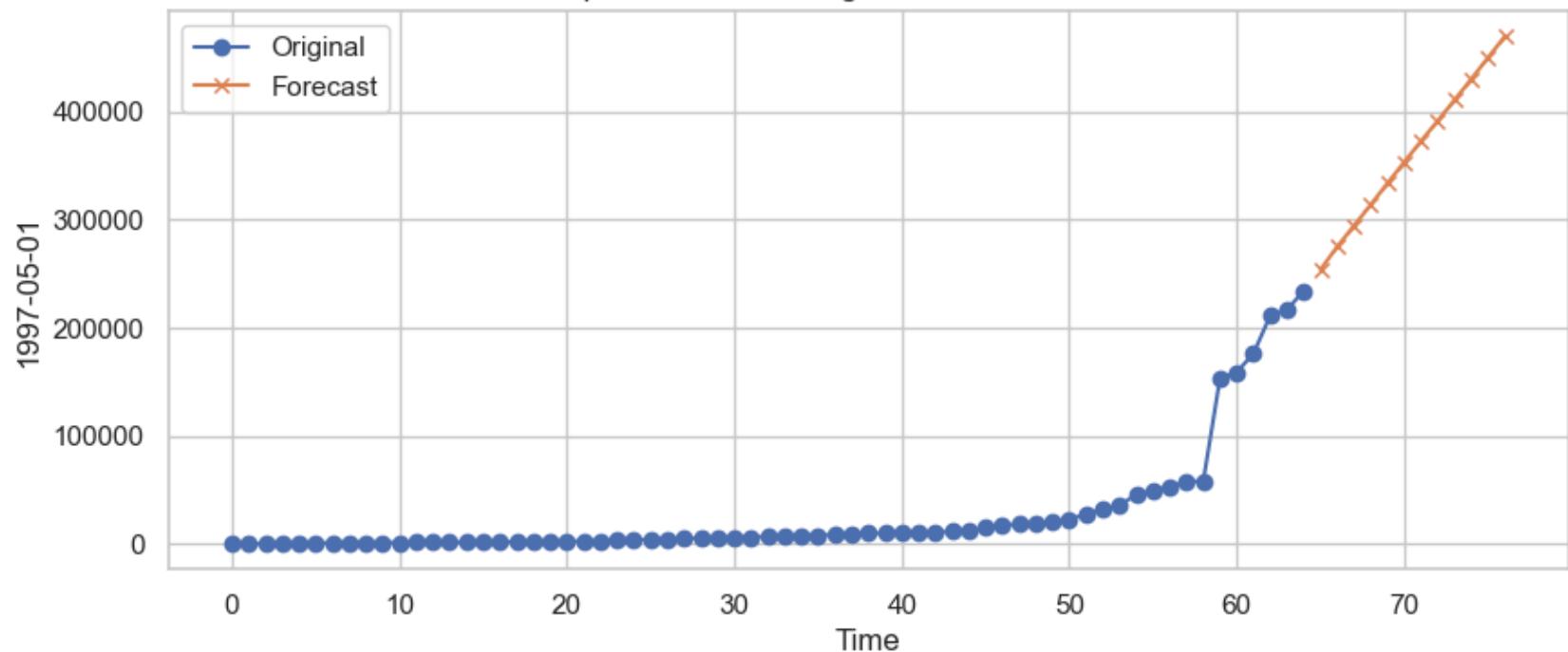
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1997-04-01



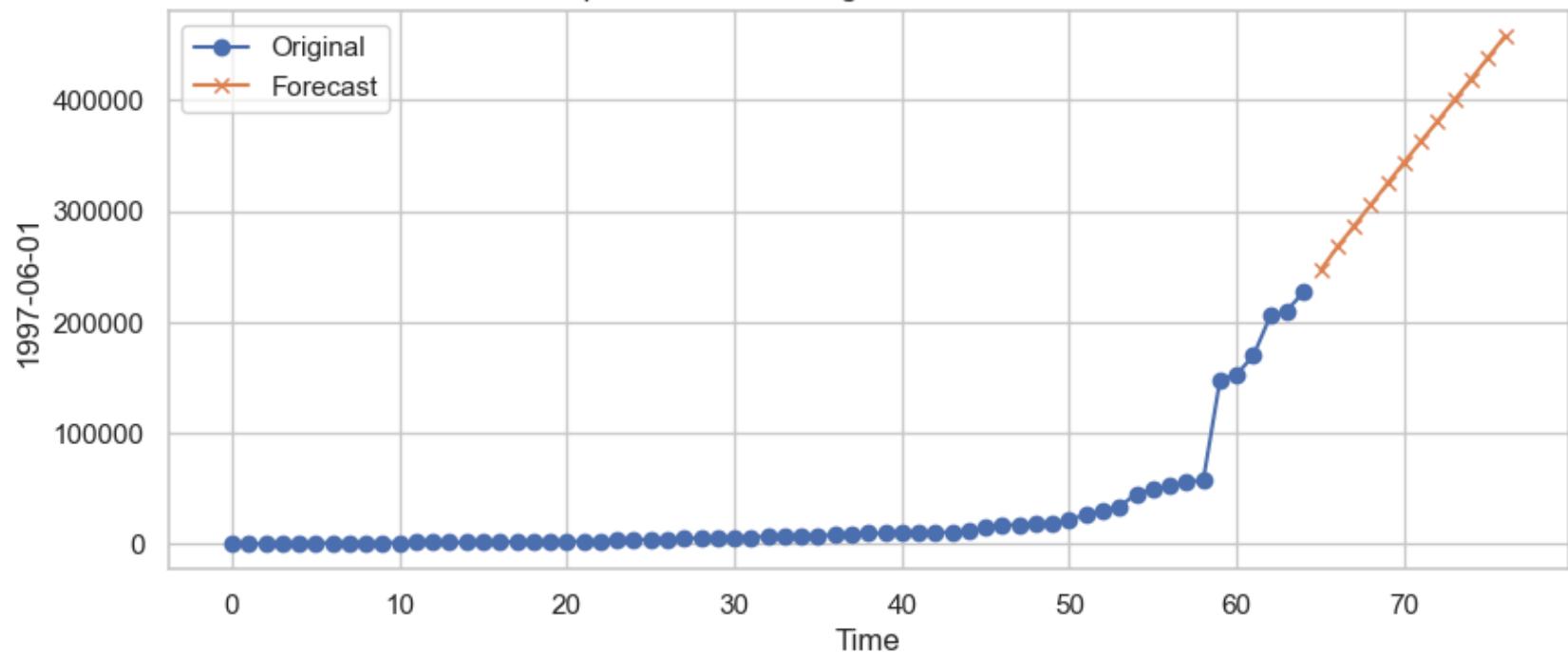
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1997-05-01



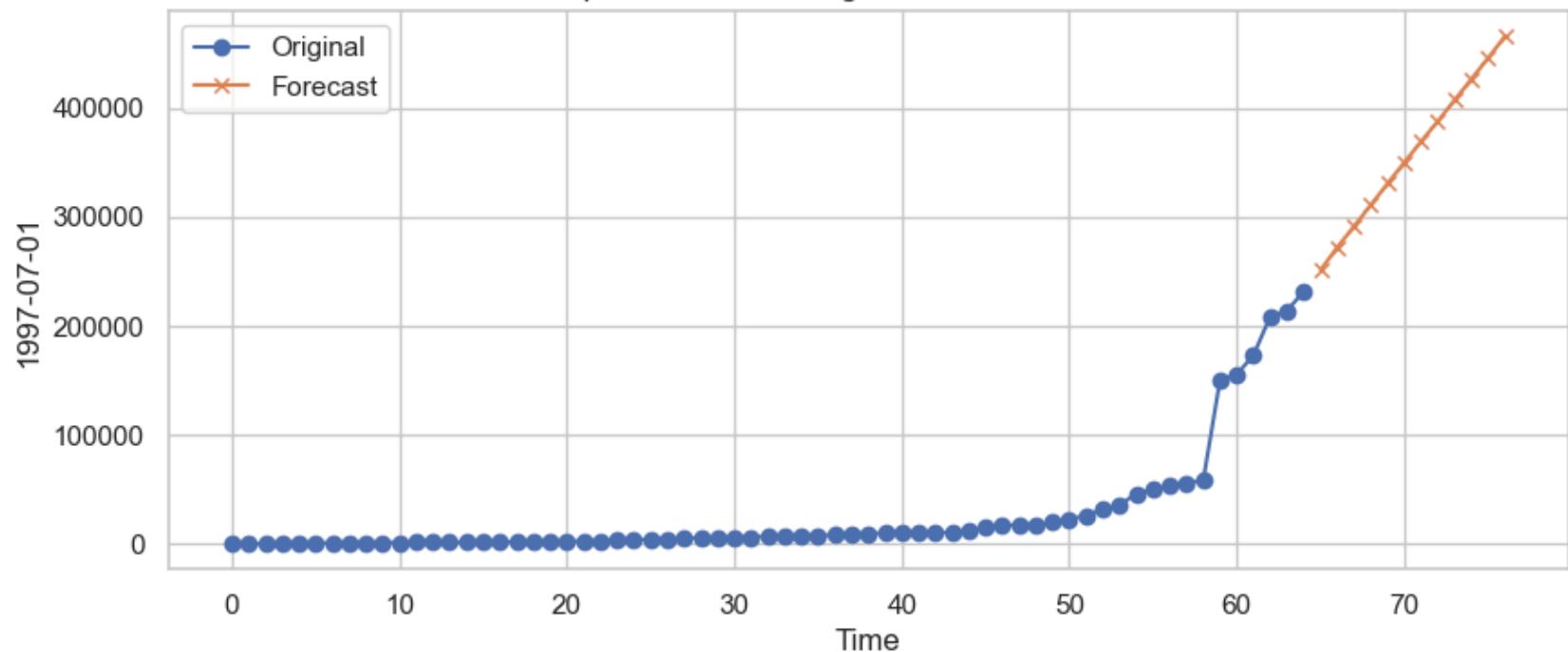
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1997-06-01



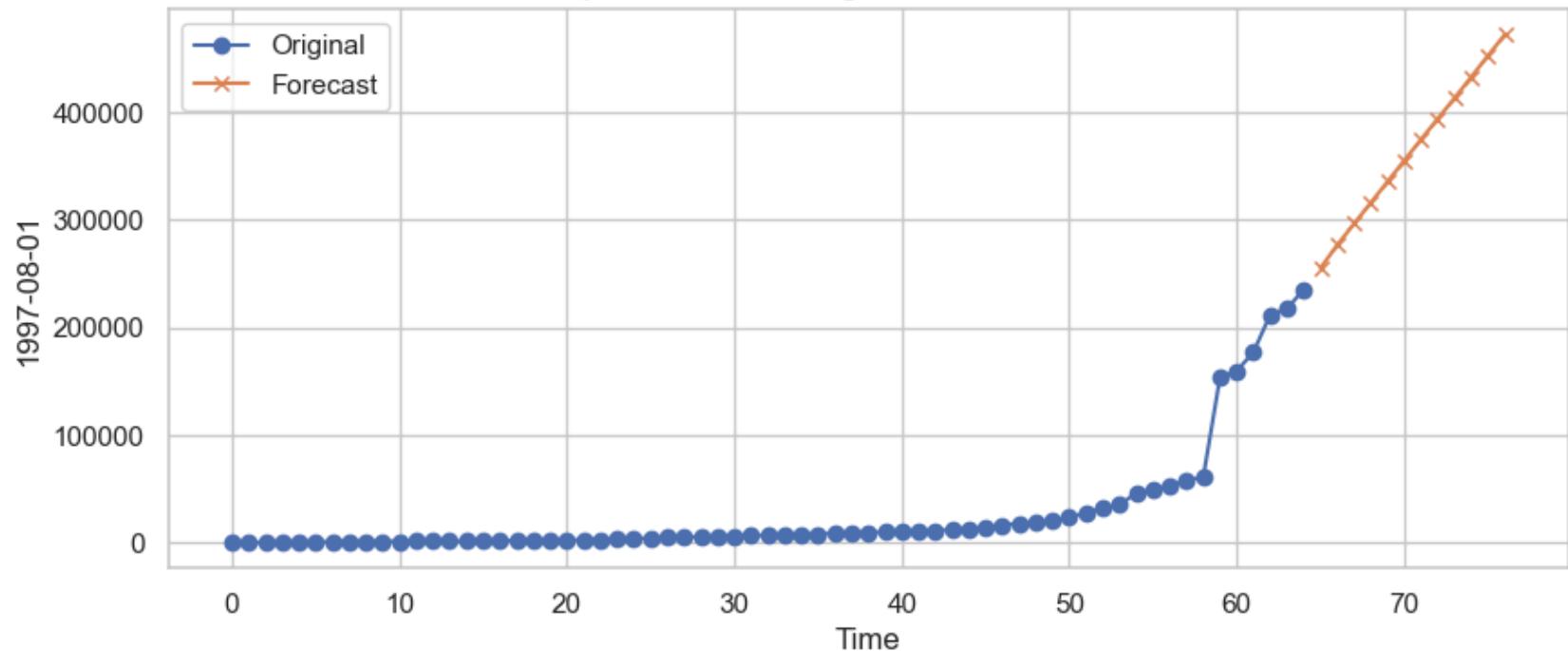
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1997-07-01



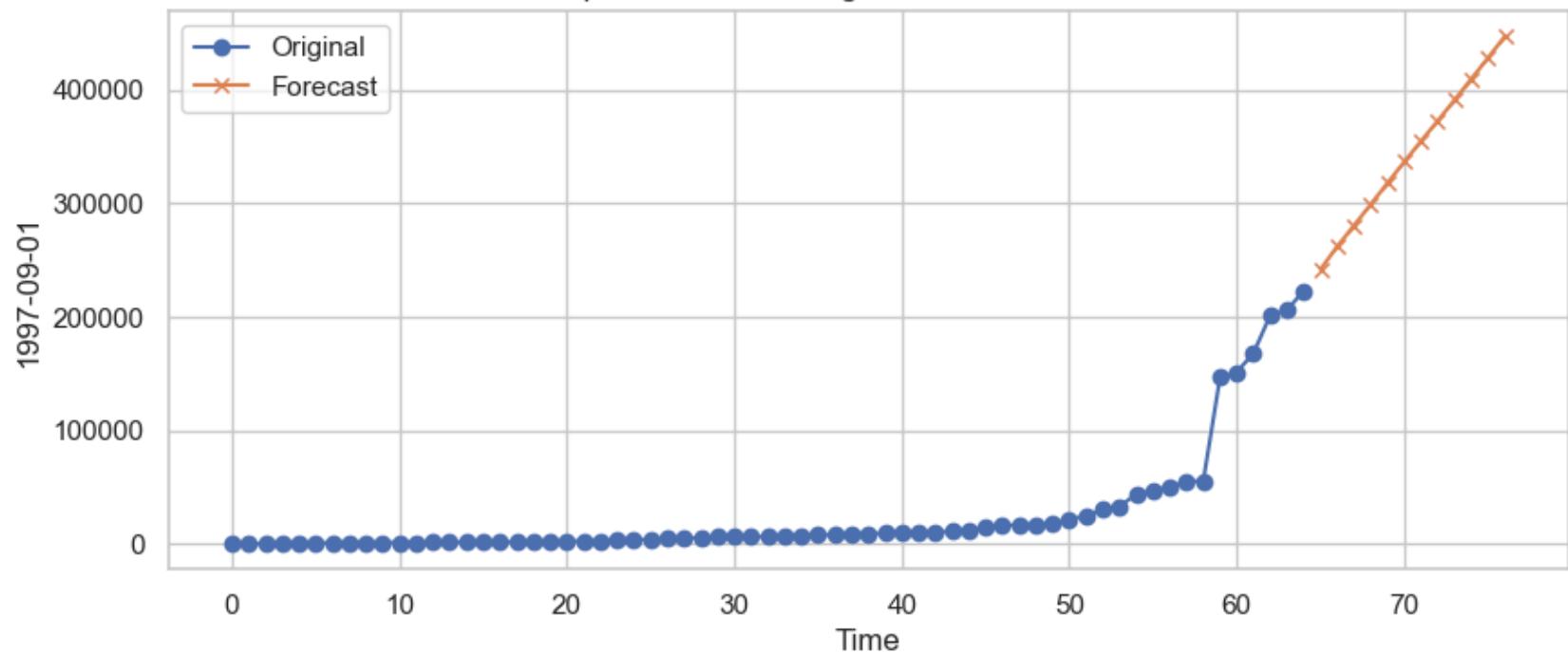
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1997-08-01



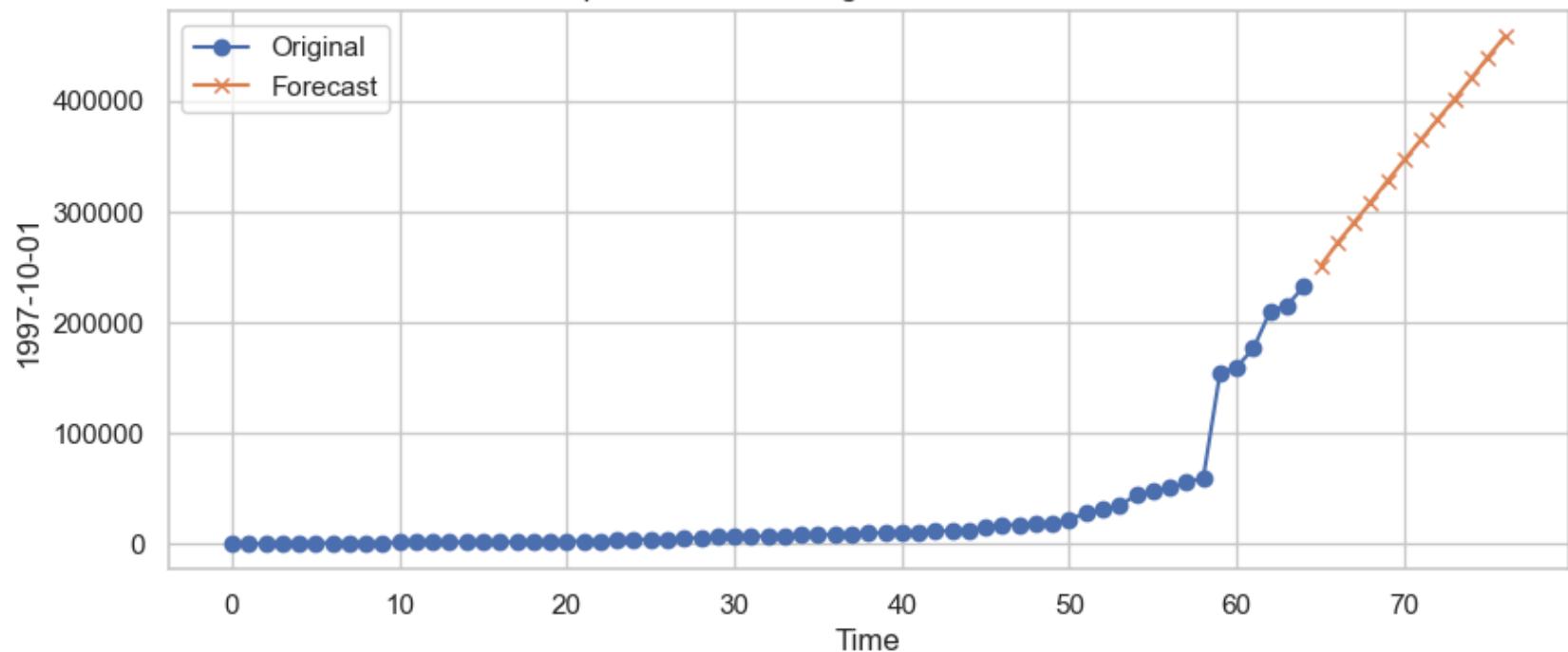
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1997-09-01



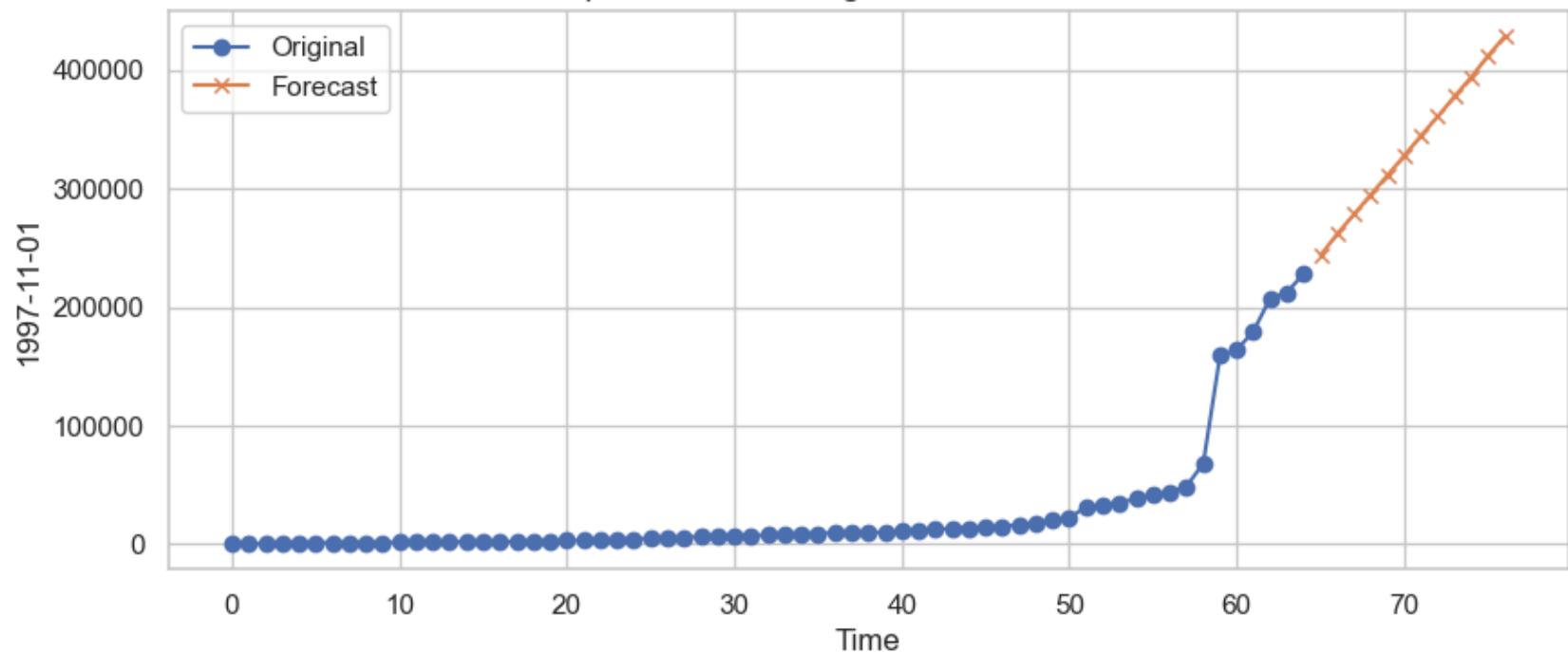
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1997-10-01



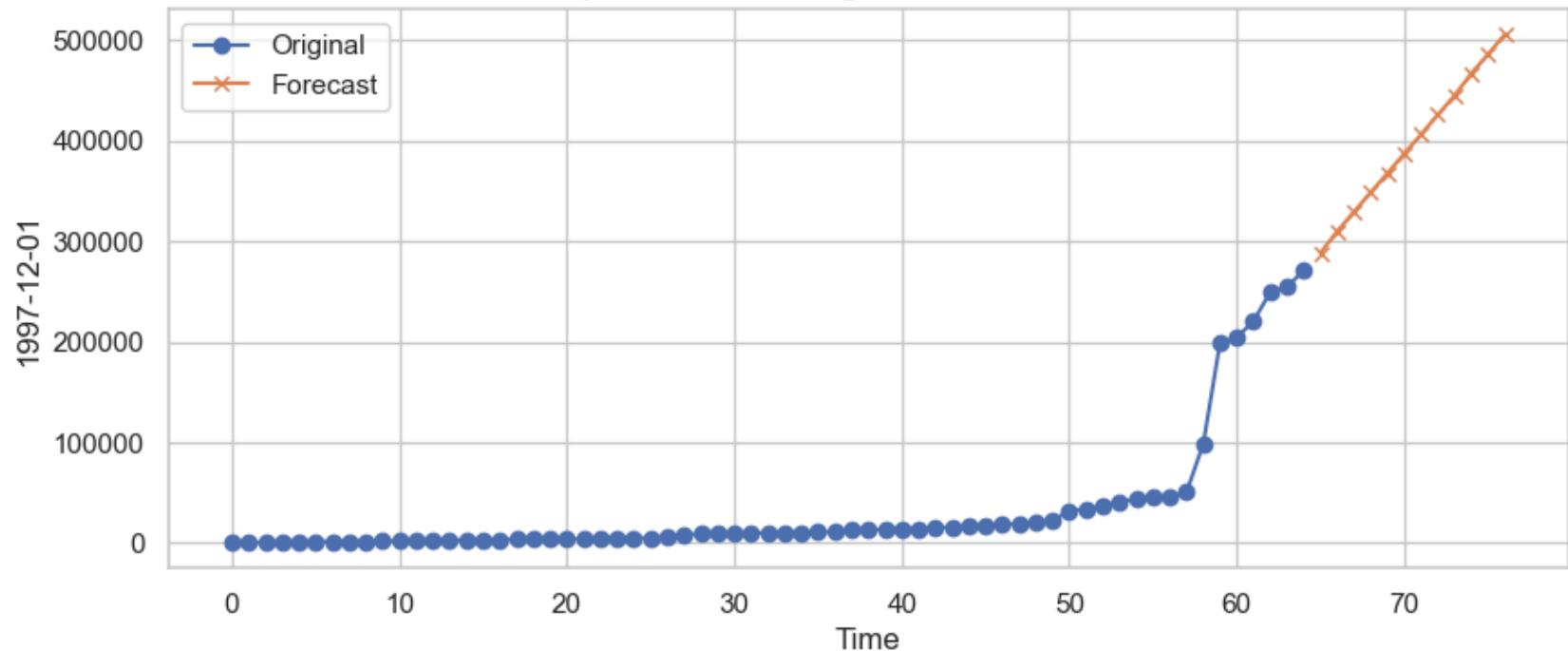
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1997-11-01



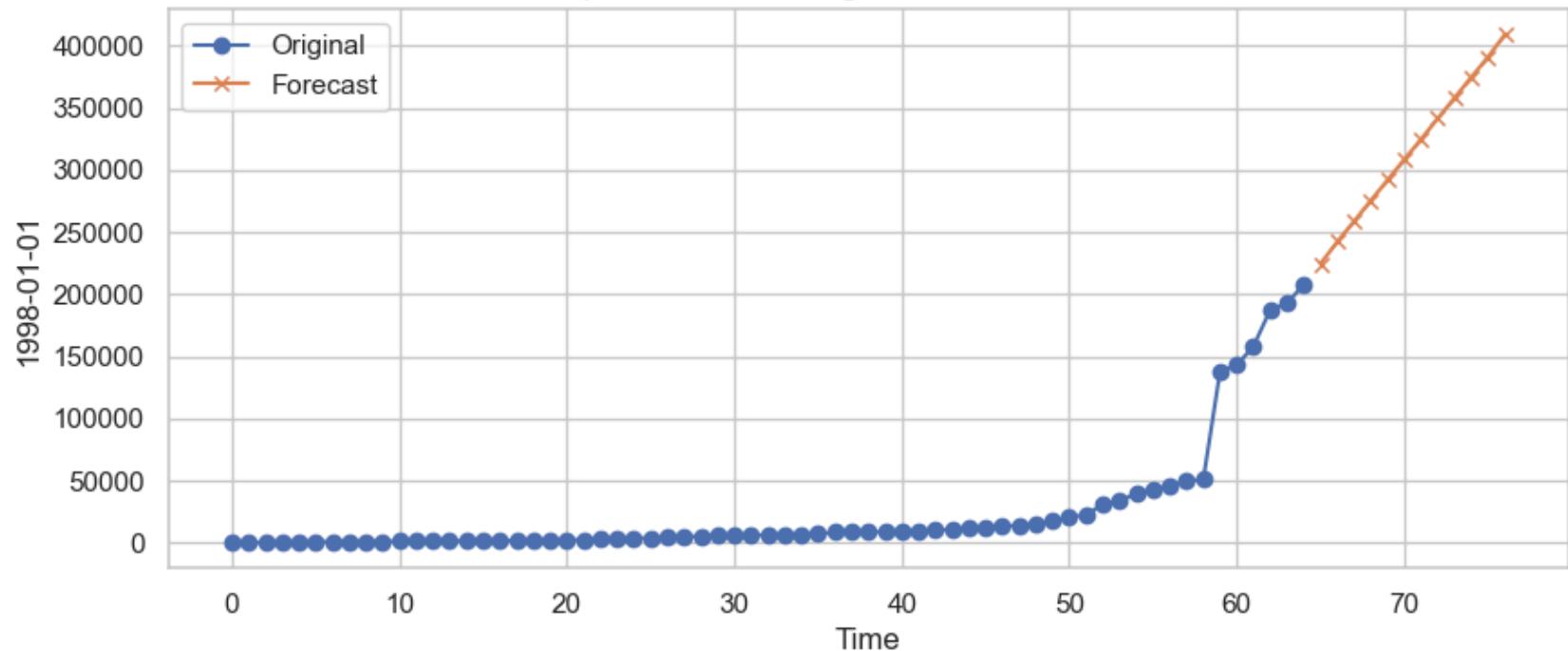
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1997-12-01



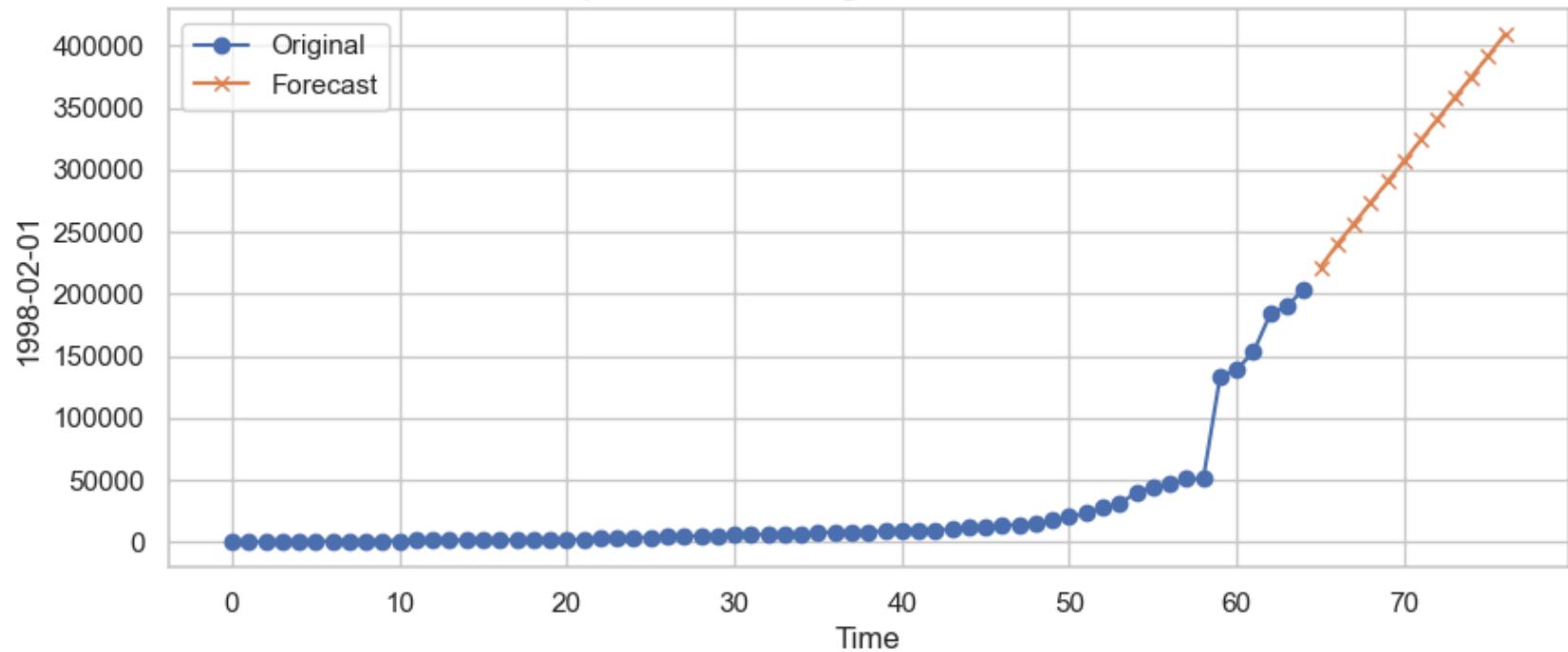
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1998-01-01



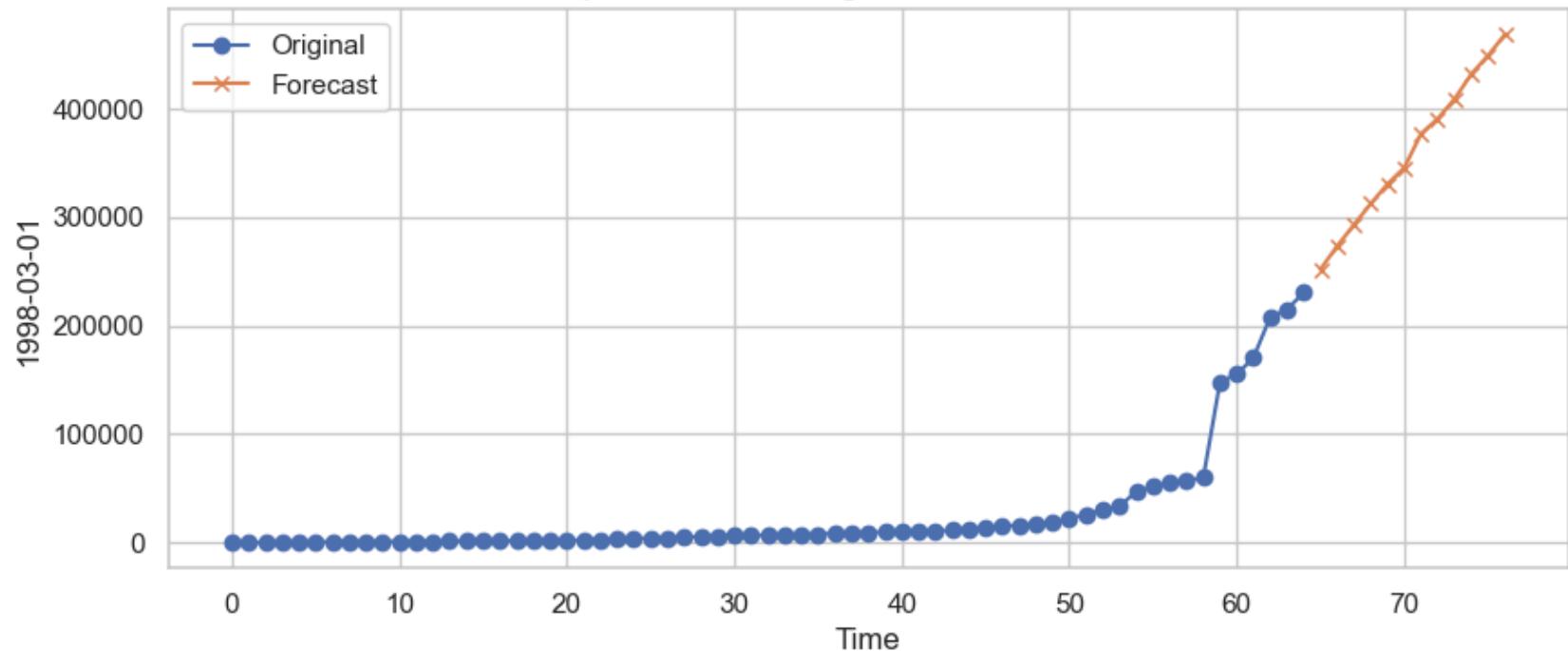
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1998-02-01



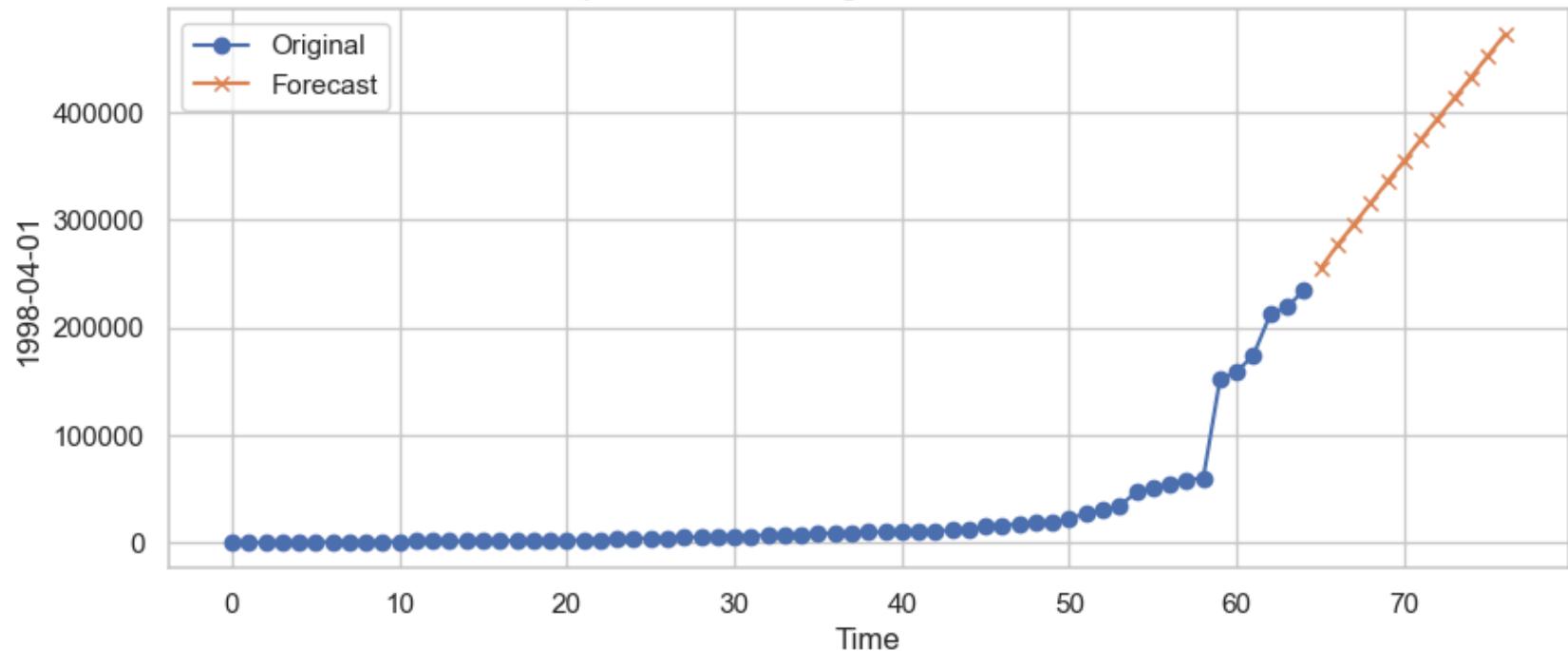
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1998-03-01



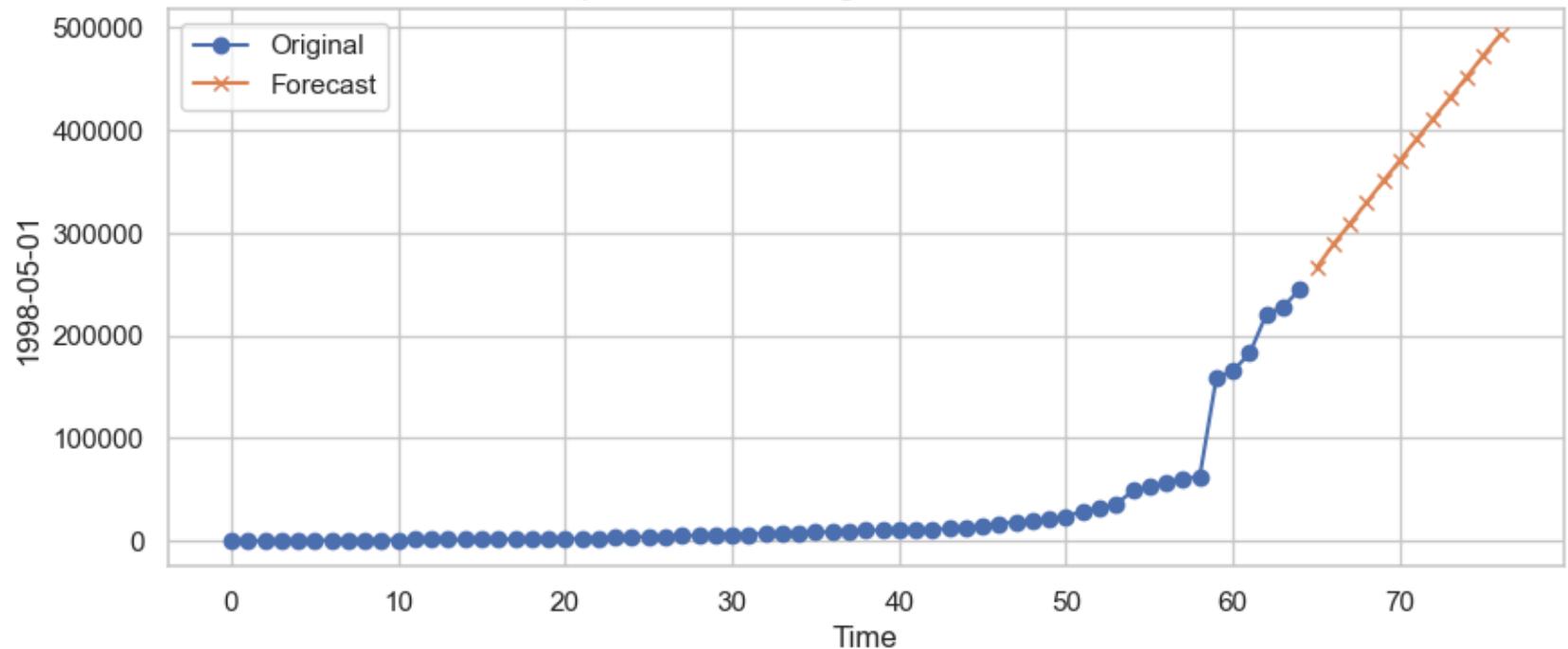
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1998-04-01



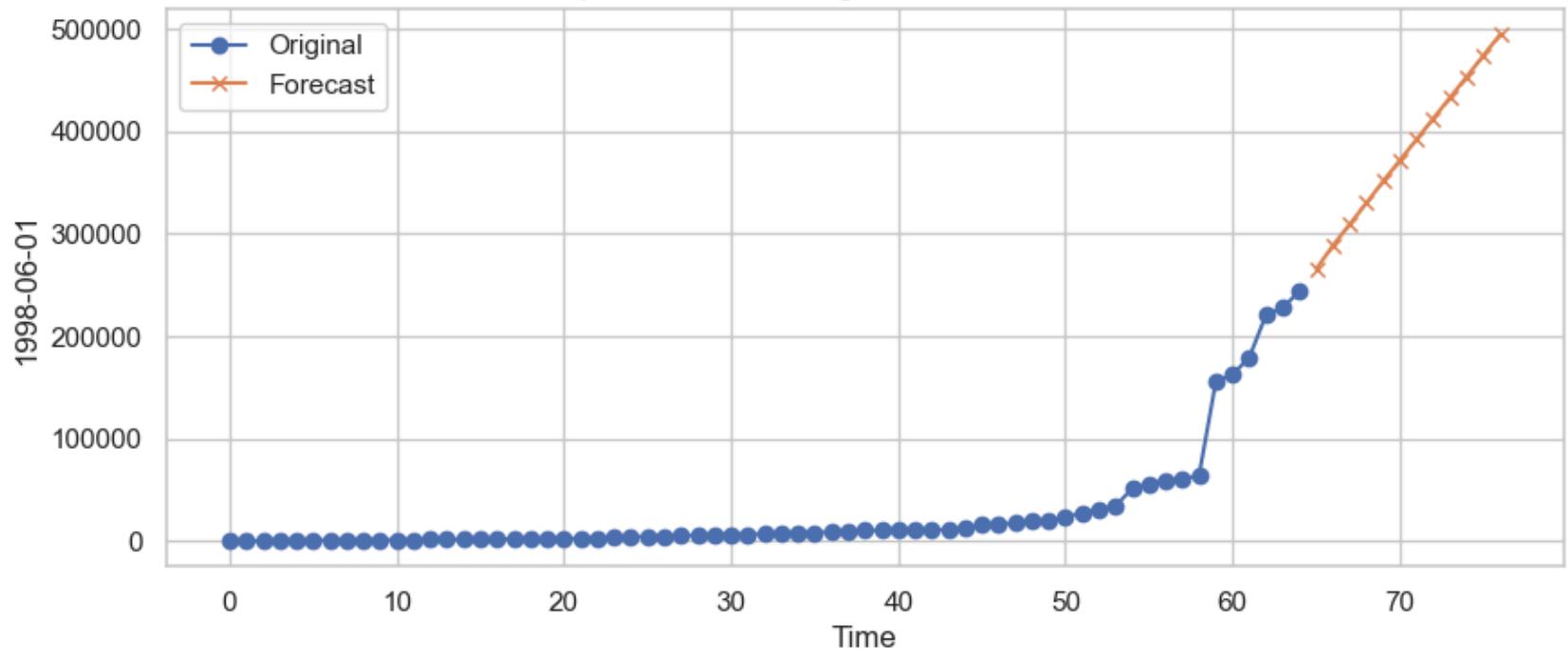
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1998-05-01

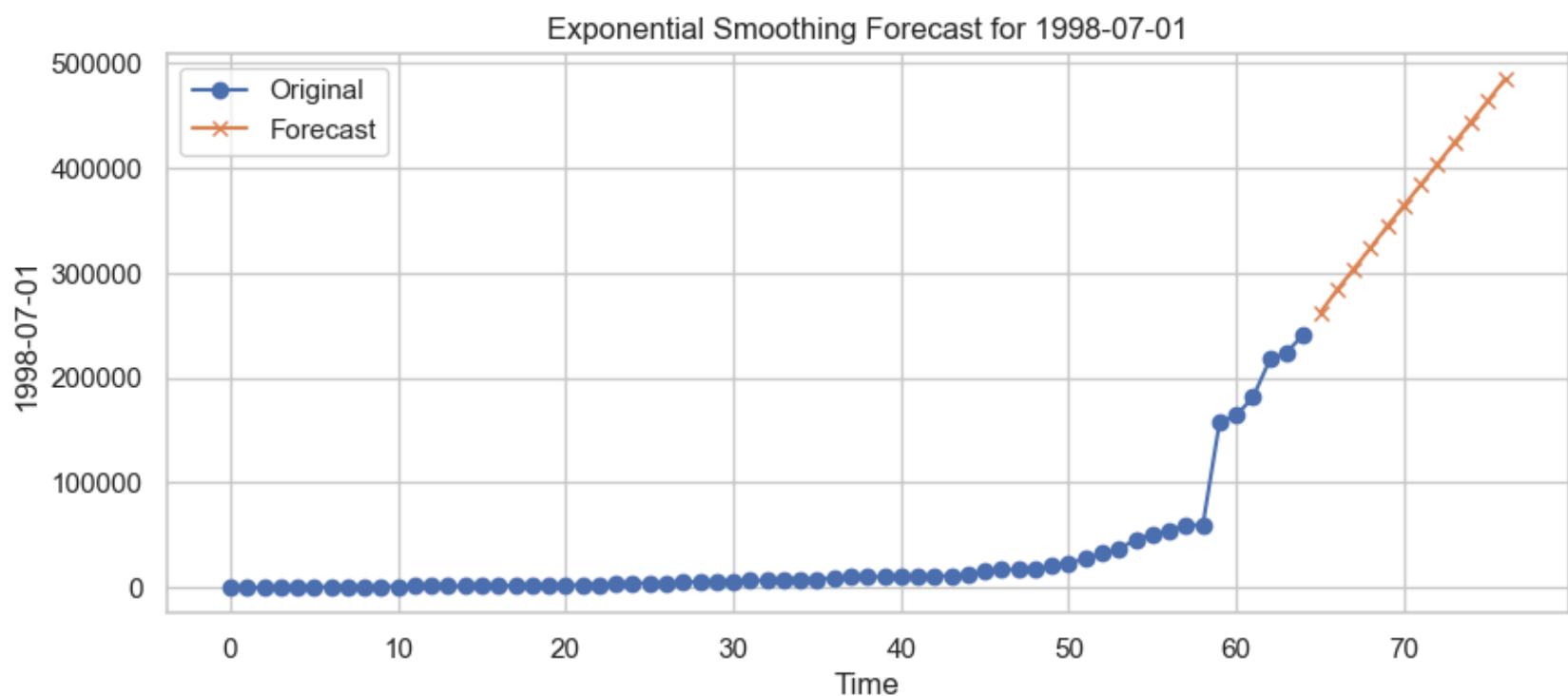


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1998-06-01

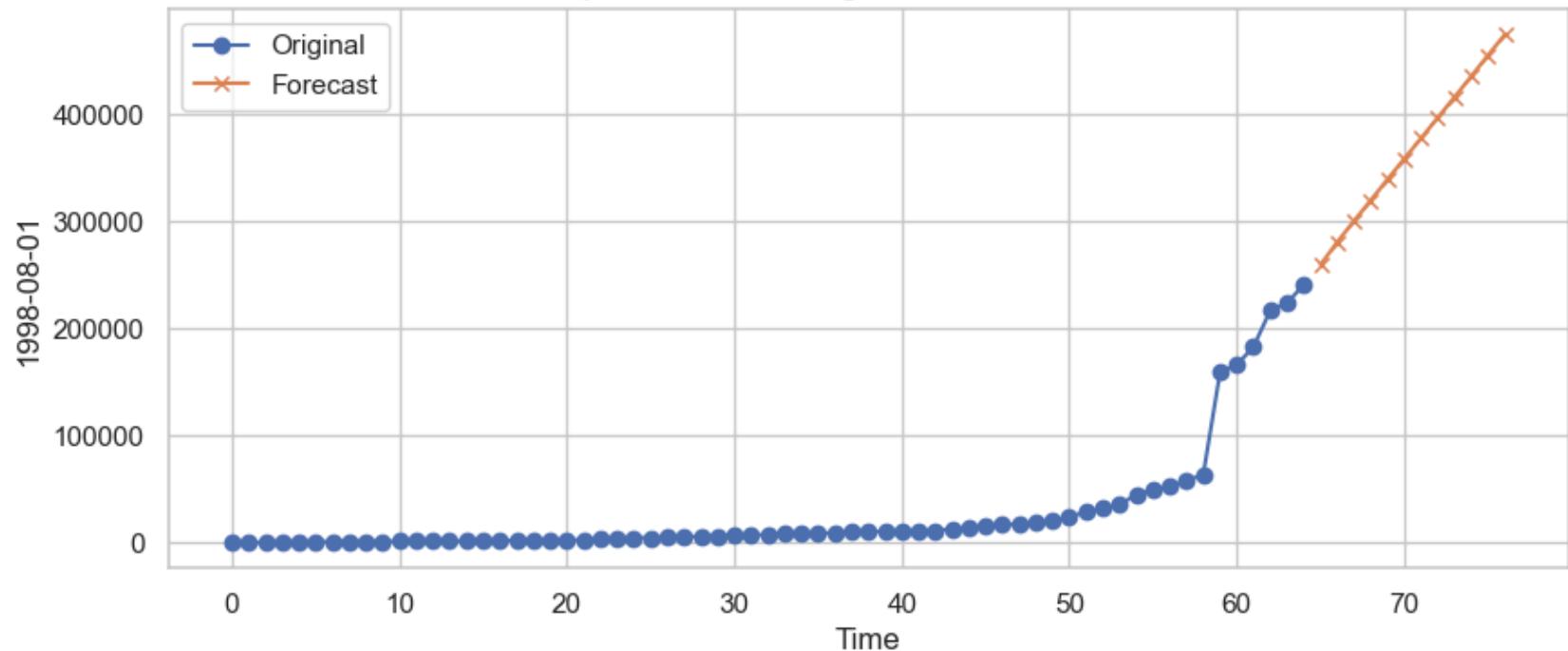


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



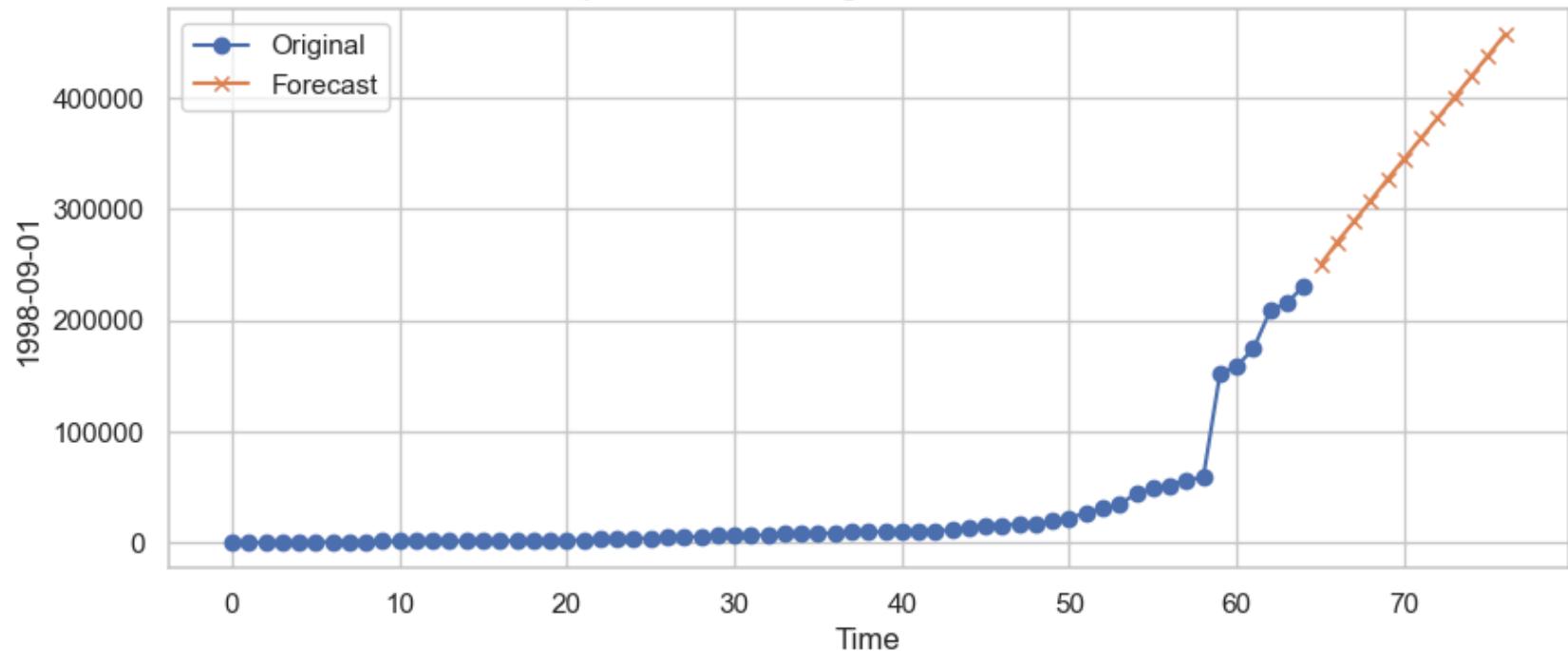
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1998-08-01

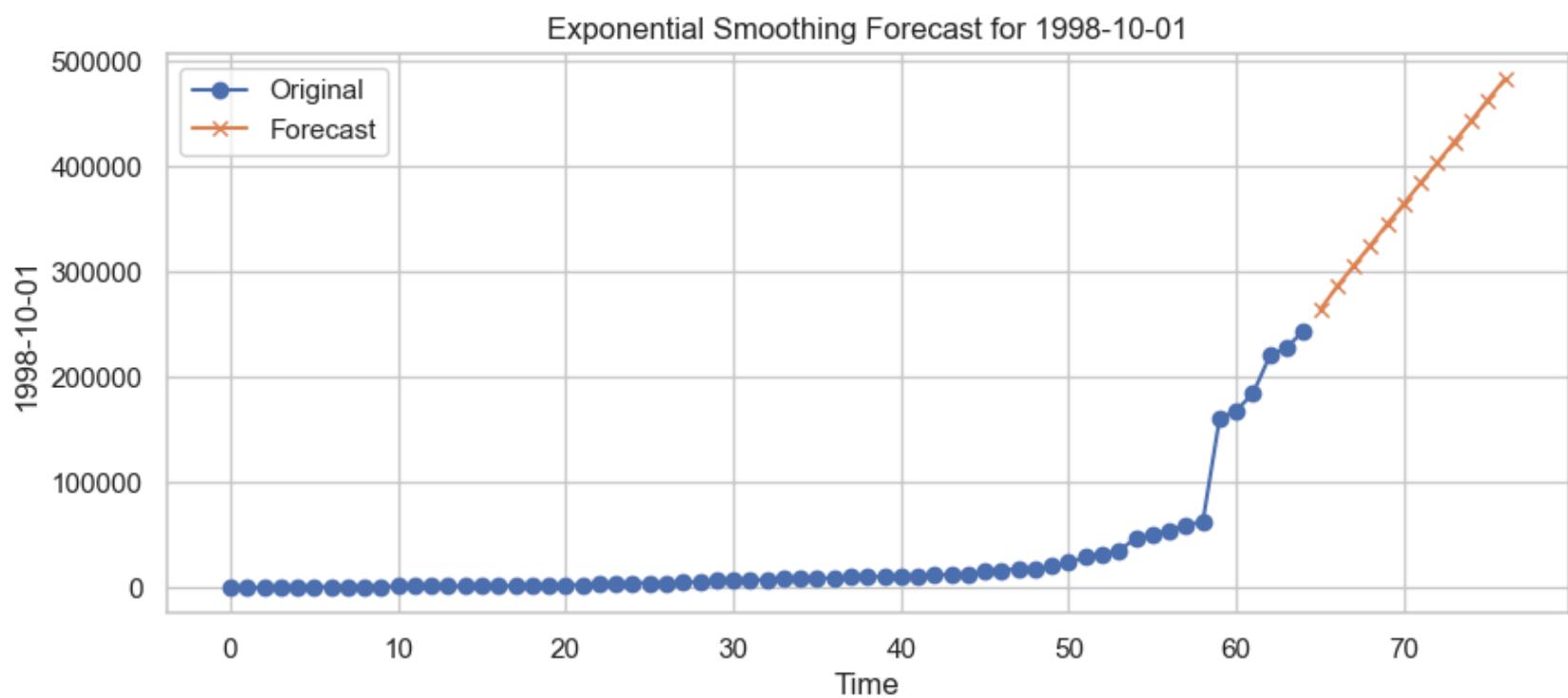


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1998-09-01

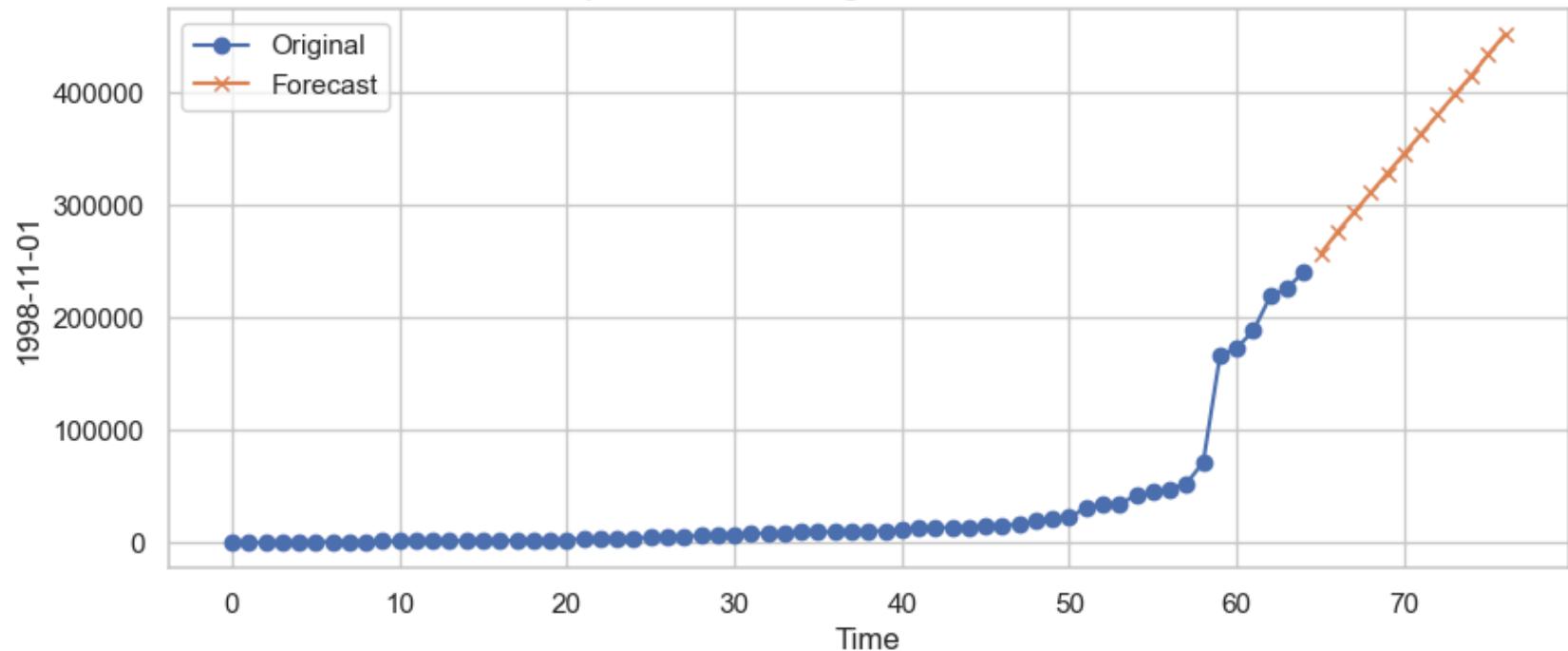


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



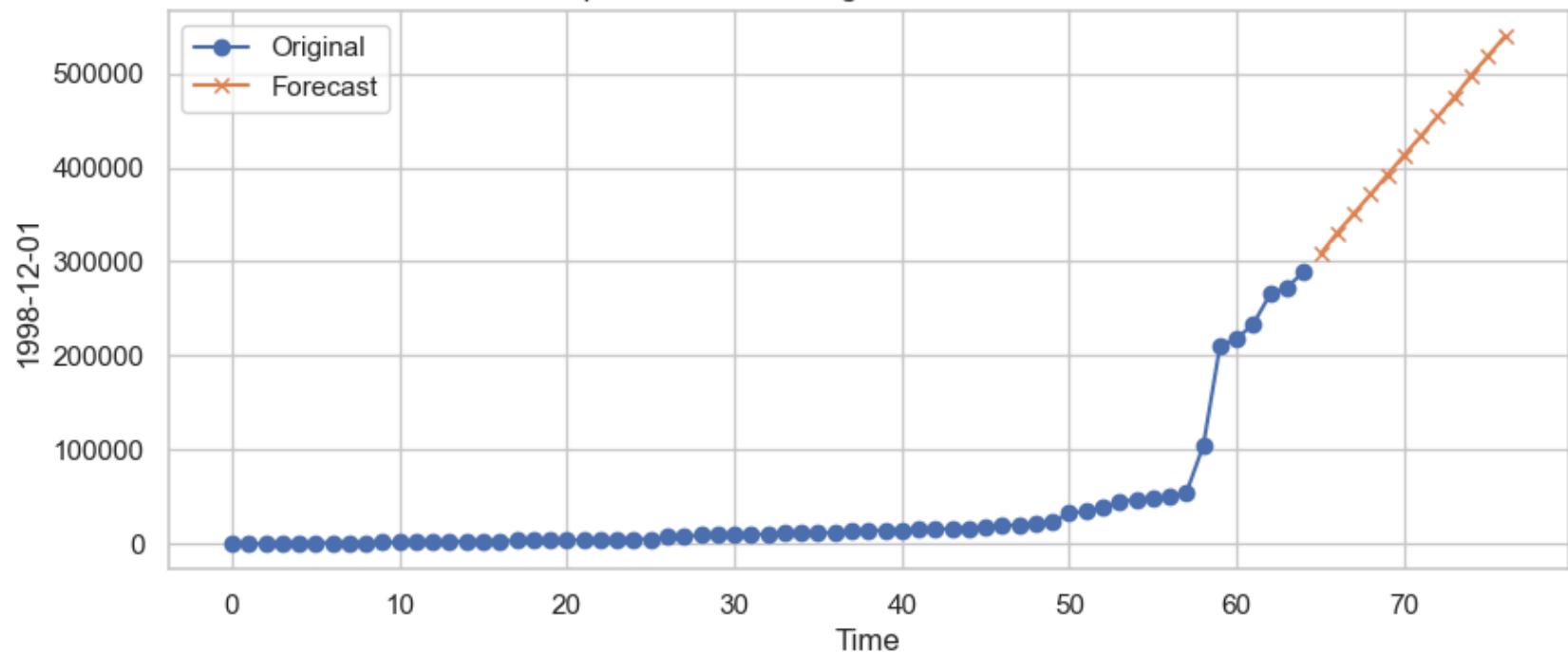
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1998-11-01



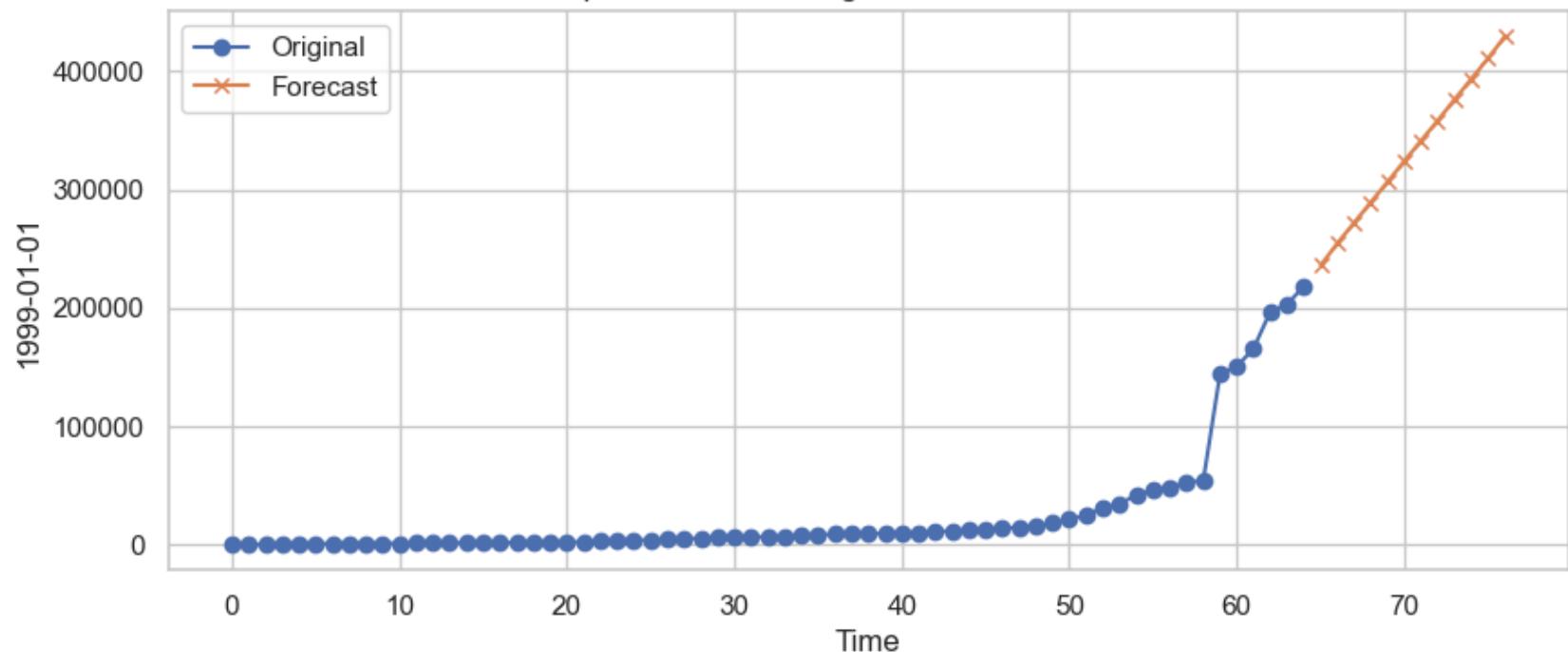
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1998-12-01



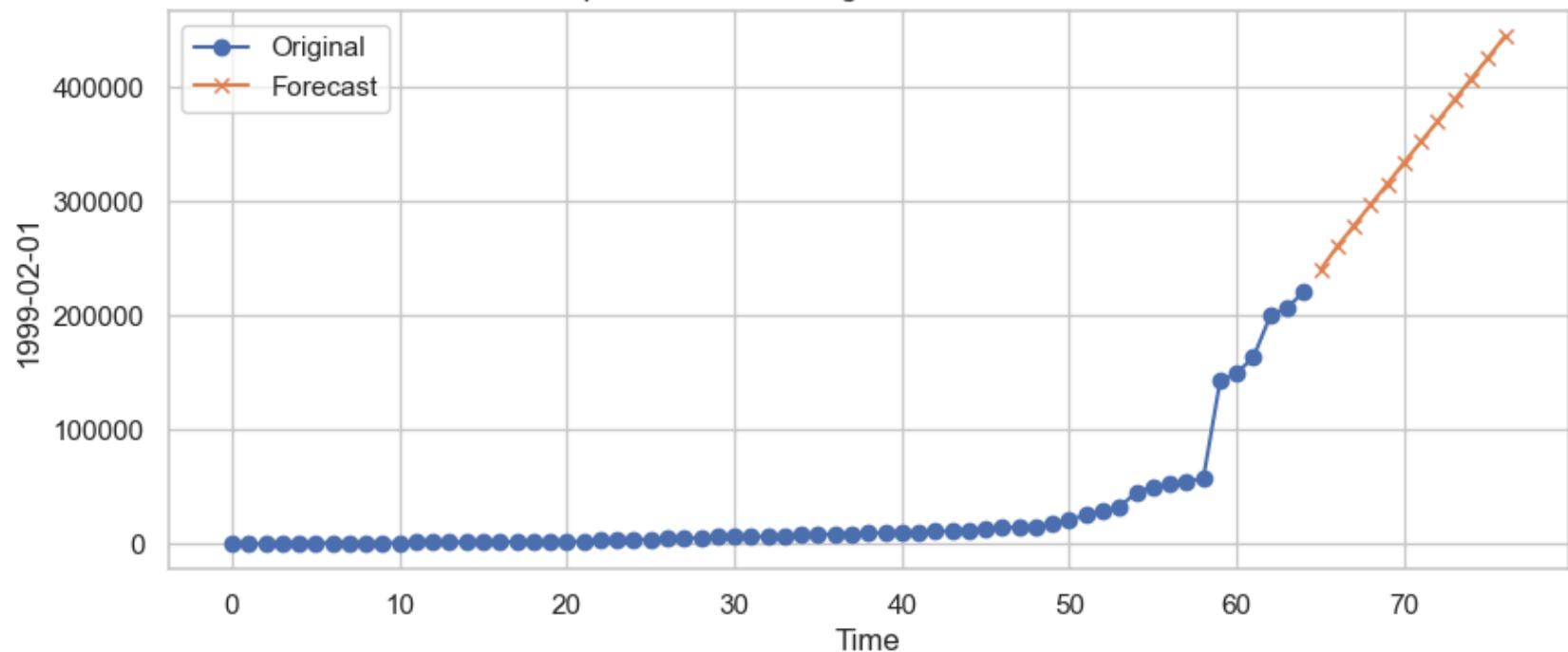
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1999-01-01



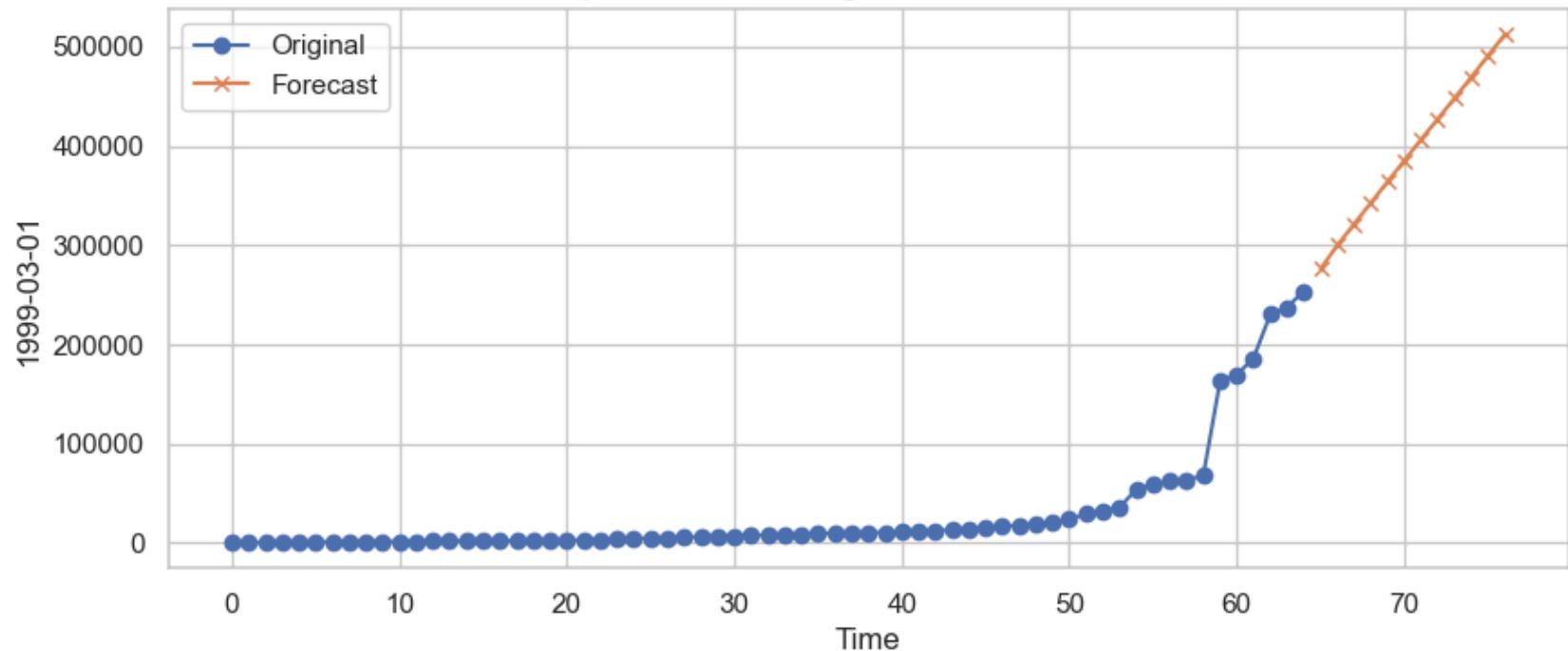
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1999-02-01



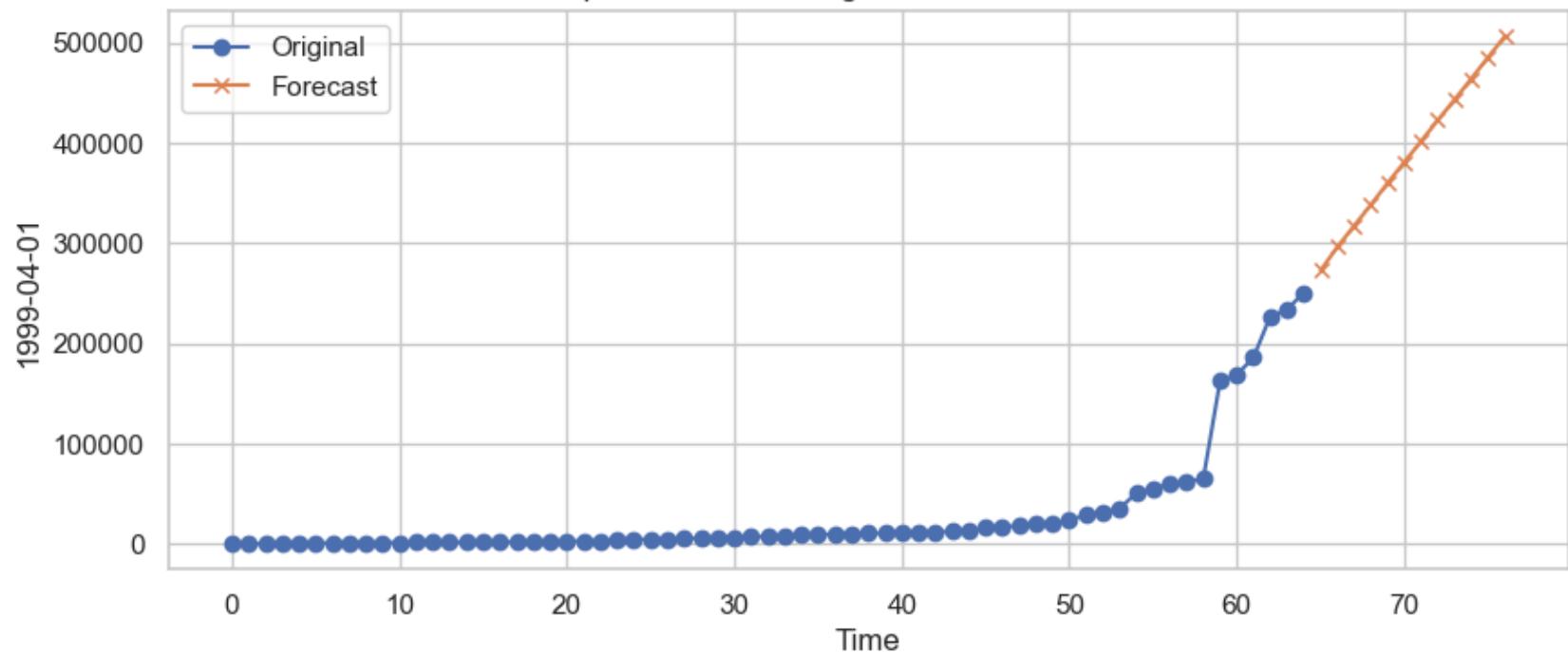
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1999-03-01



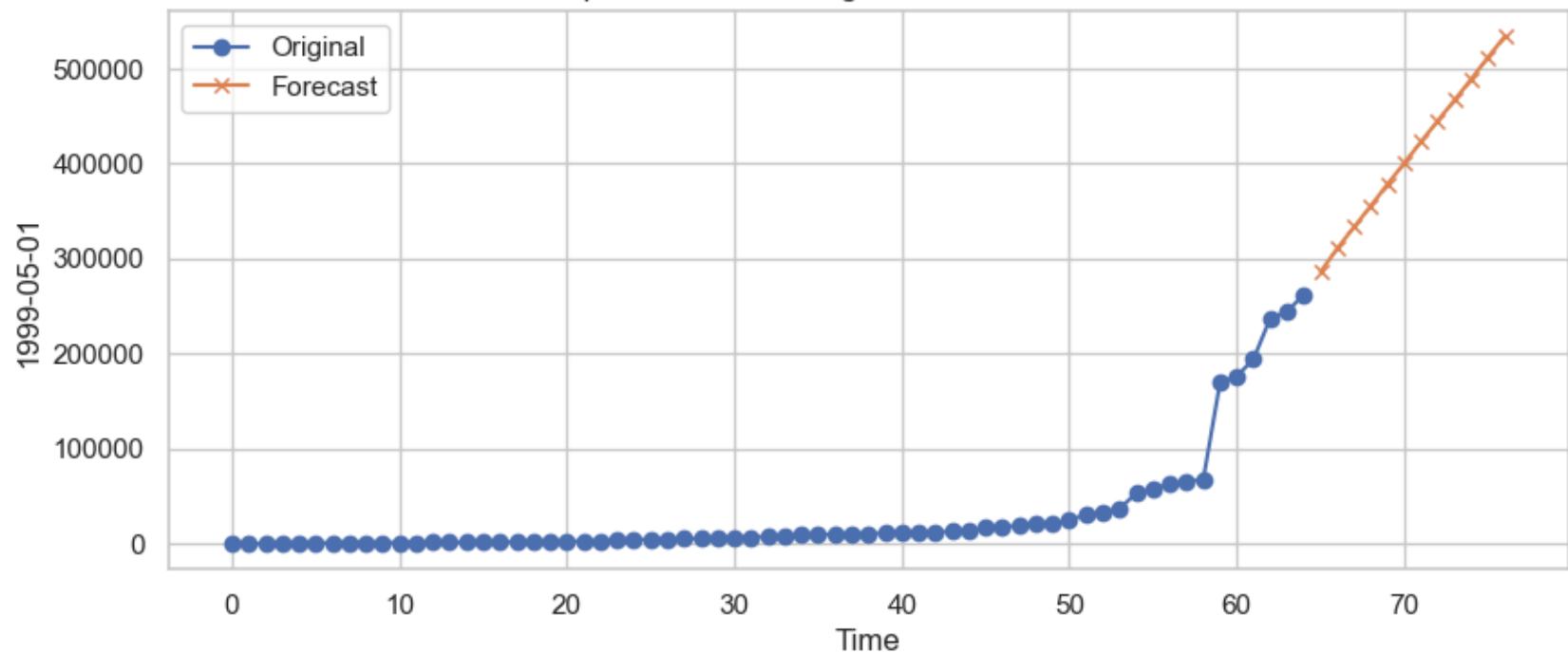
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1999-04-01



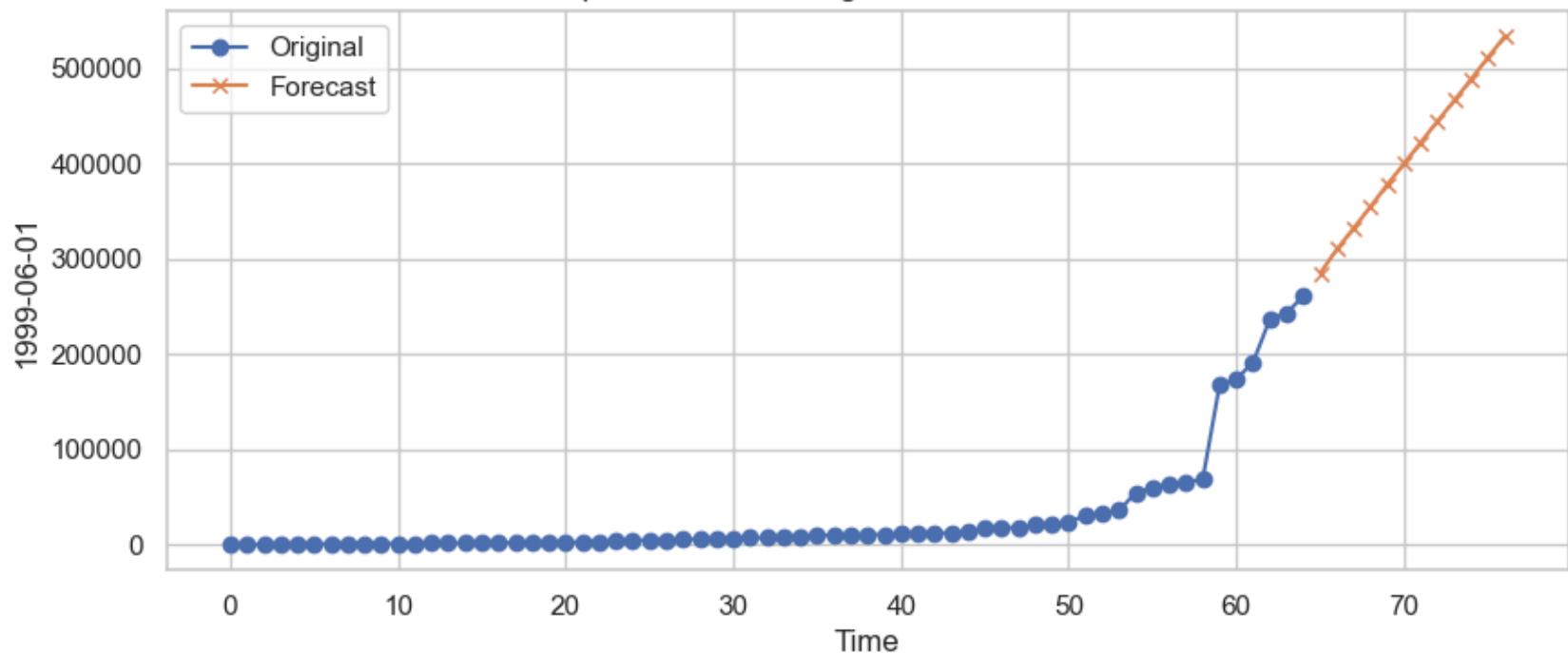
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1999-05-01



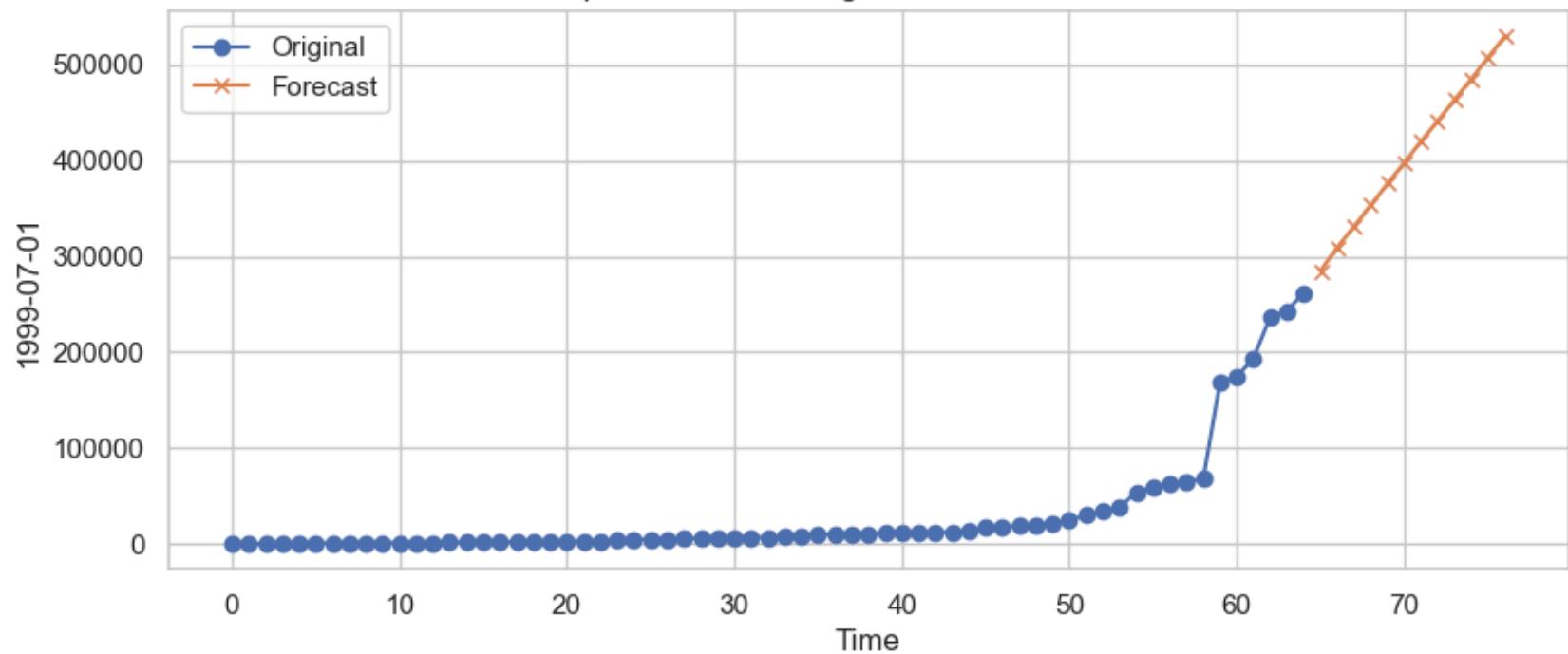
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1999-06-01



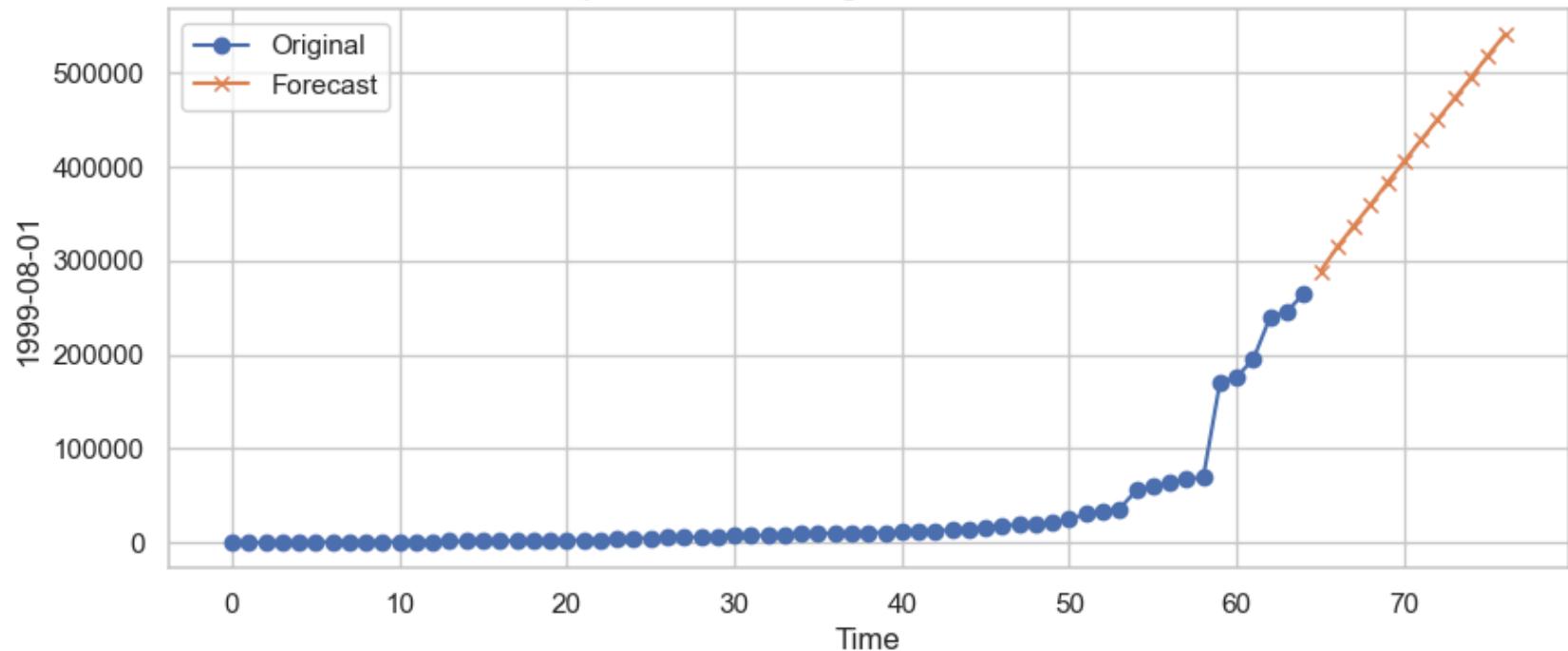
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1999-07-01



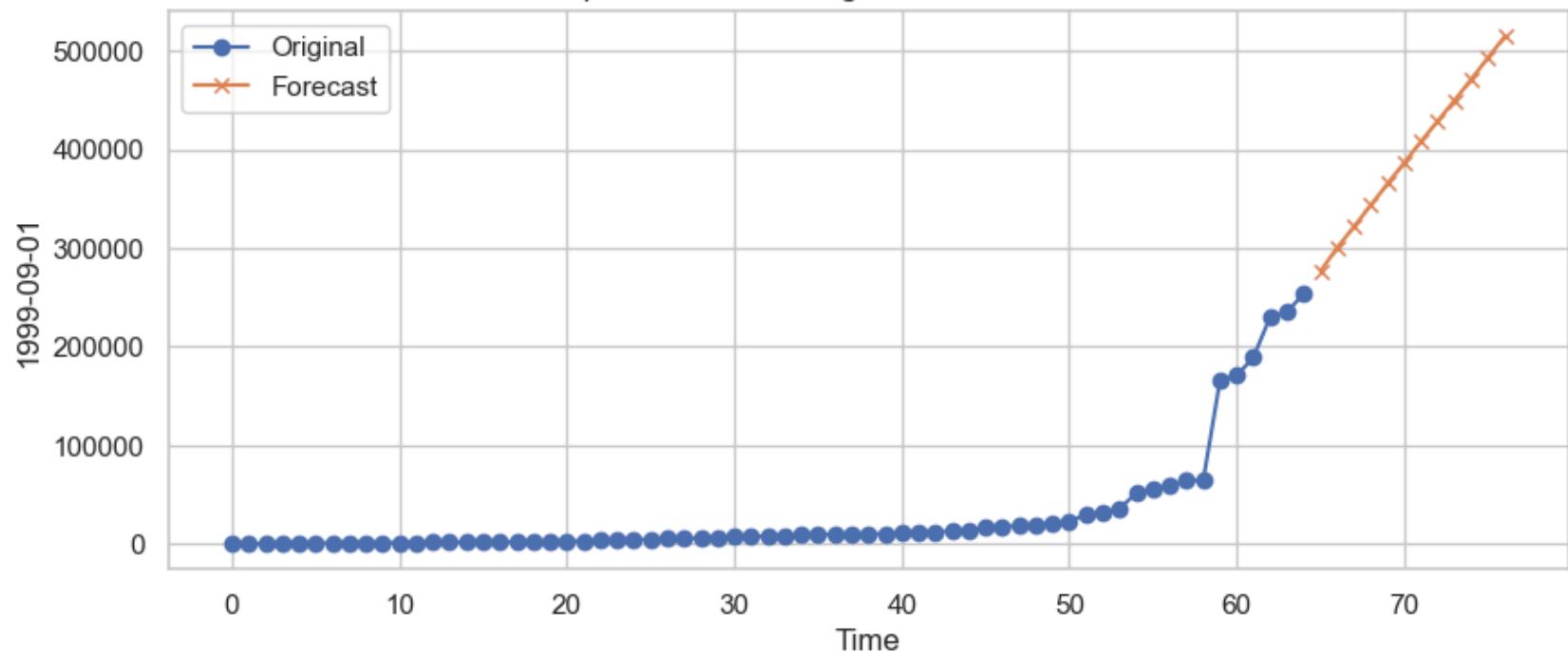
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1999-08-01



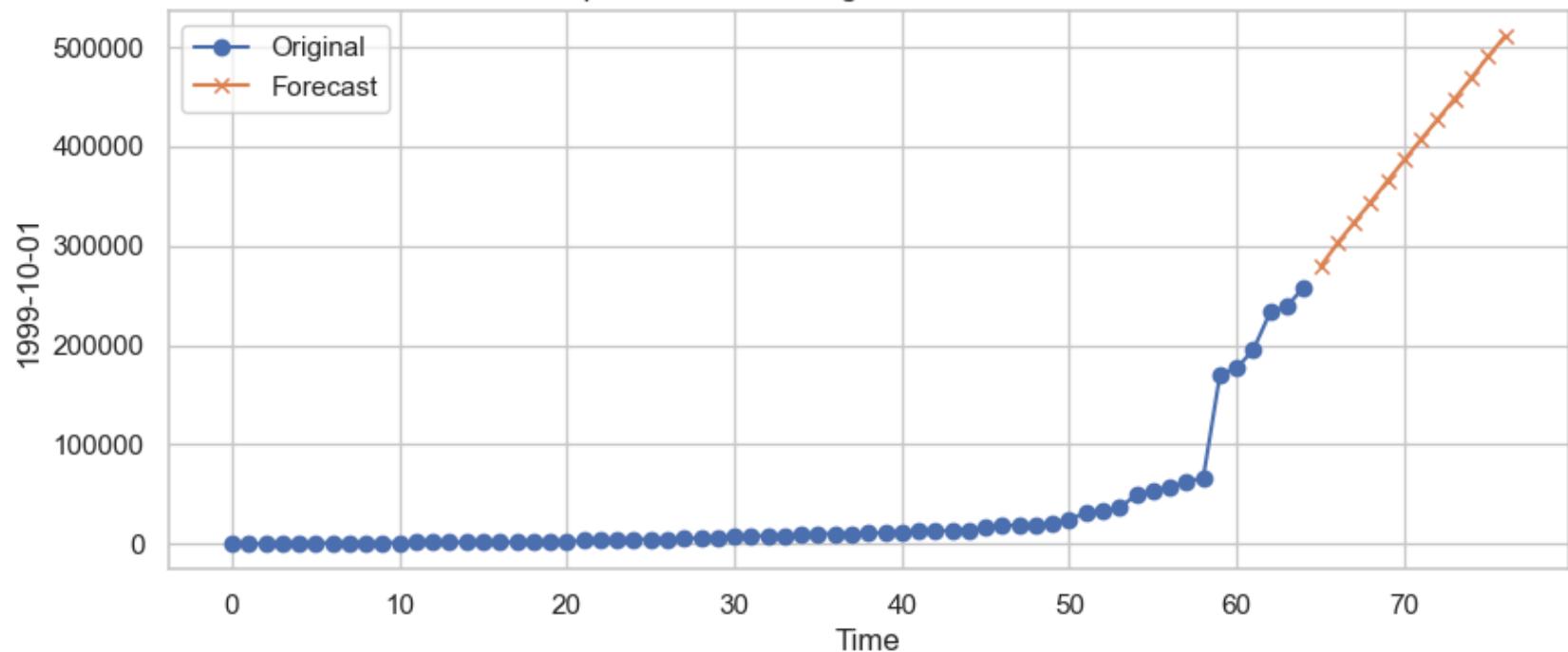
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1999-09-01



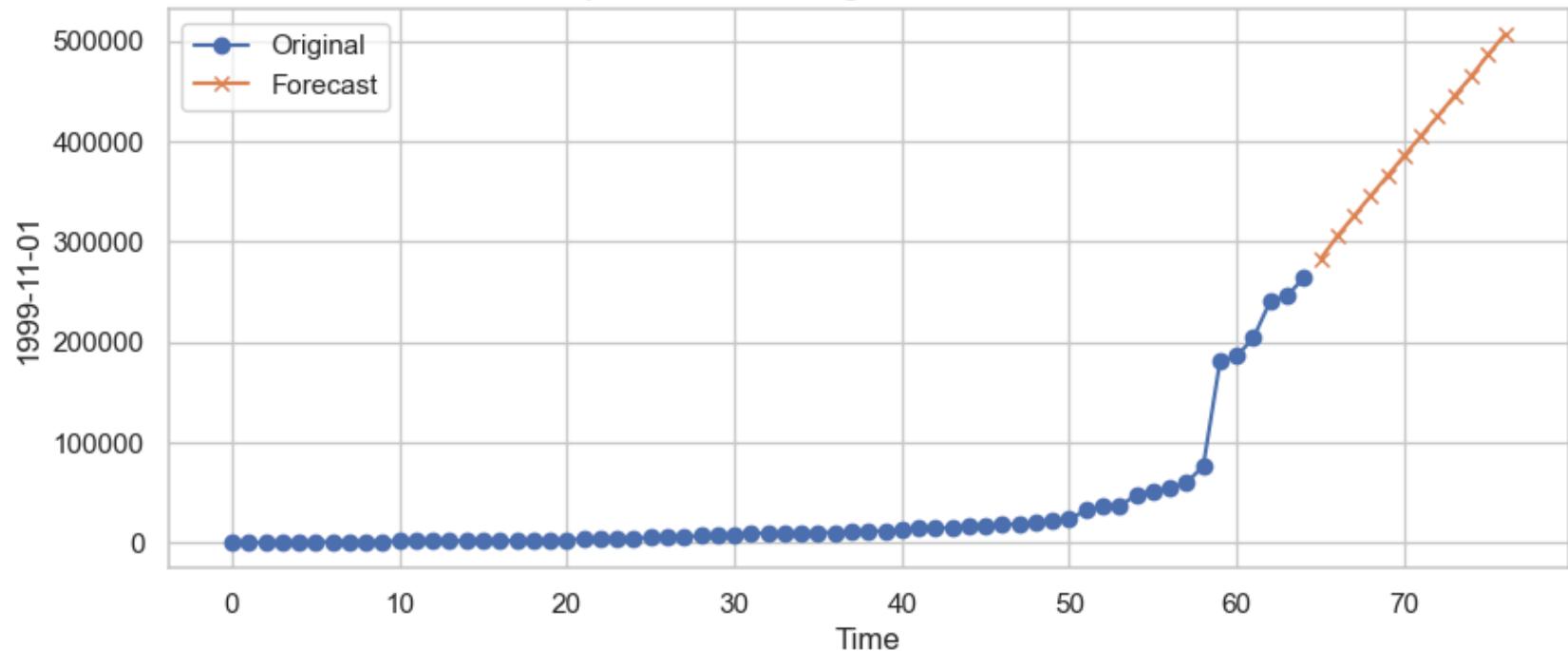
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1999-10-01

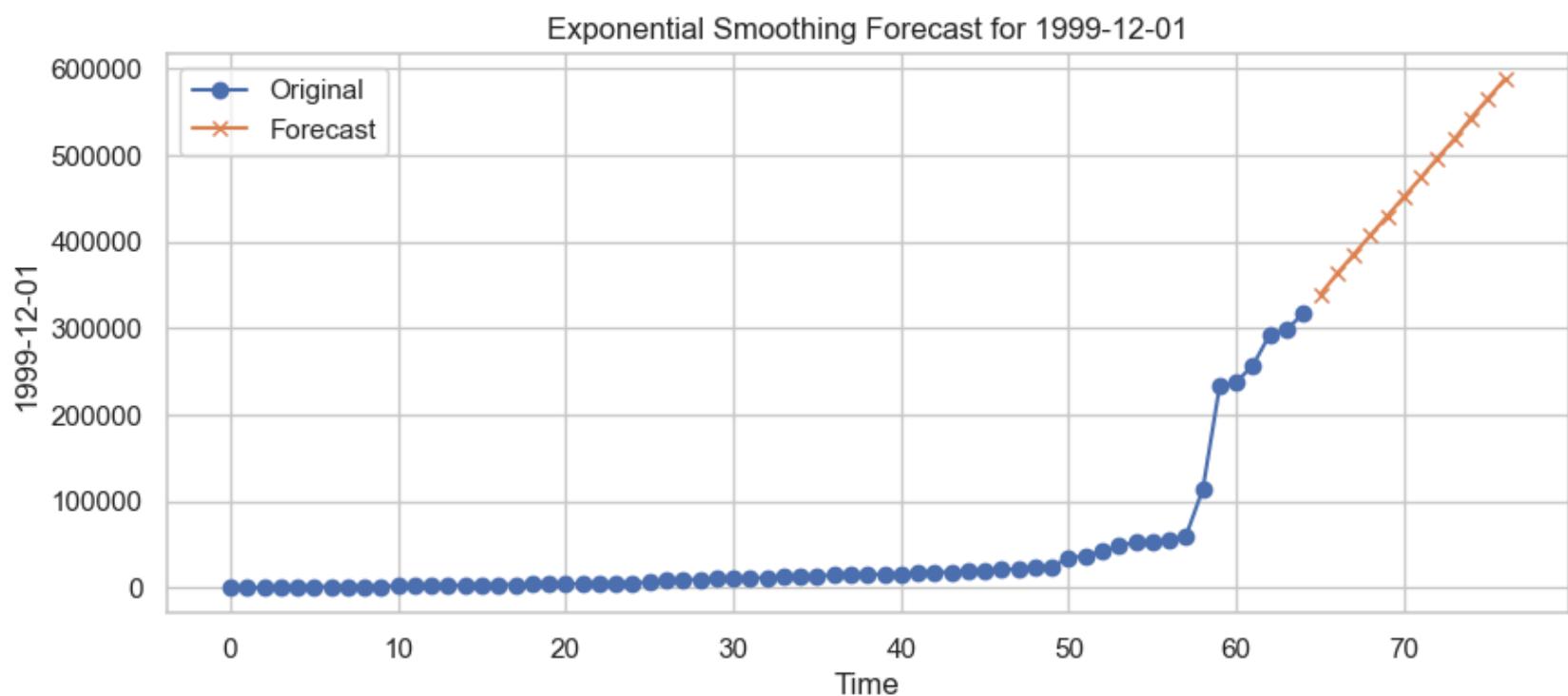


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 1999-11-01

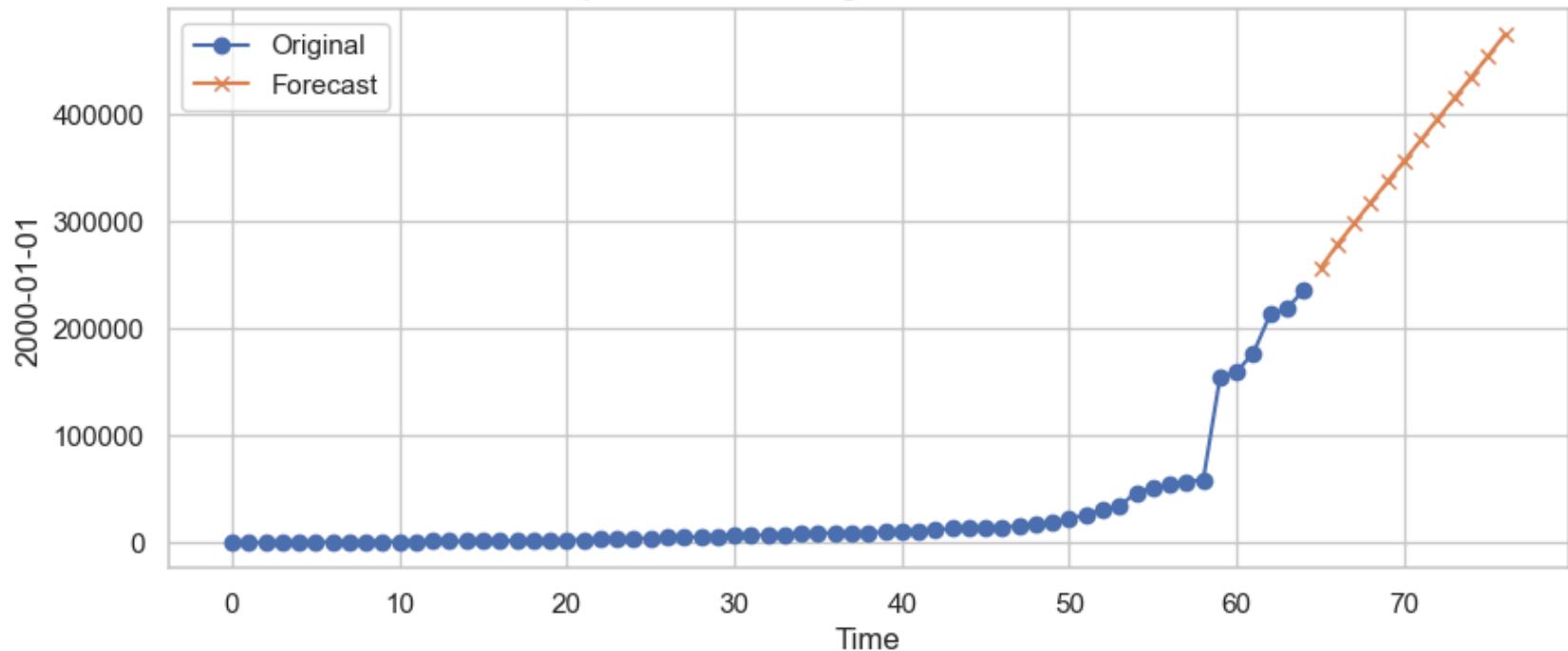


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



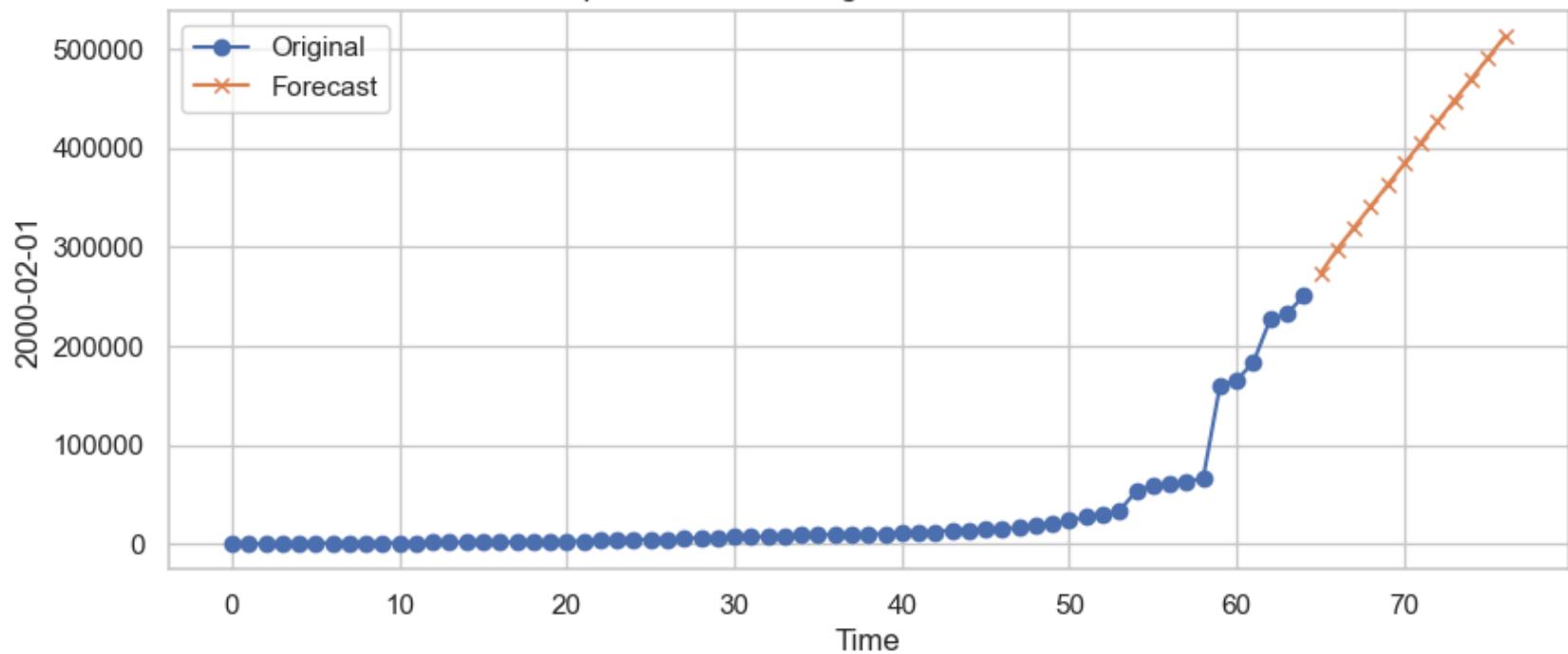
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2000-01-01

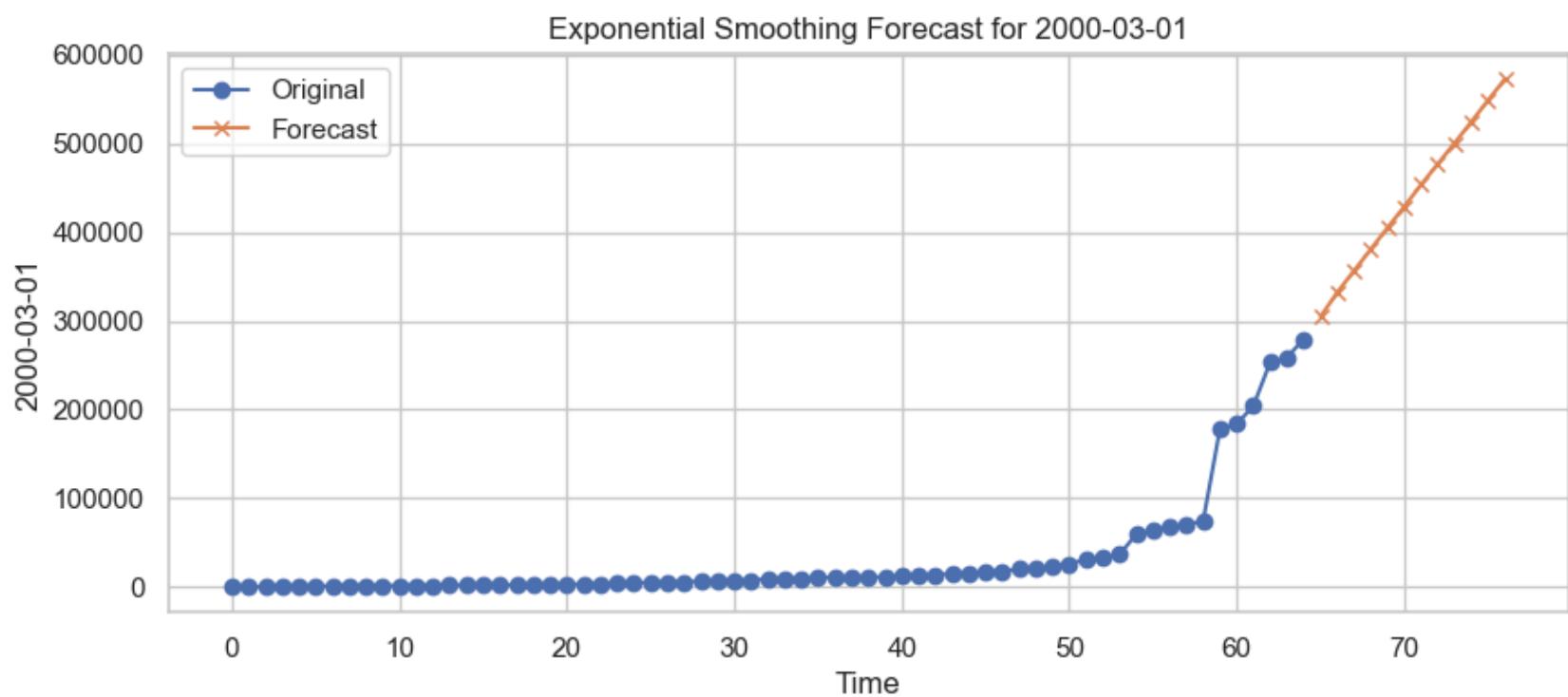


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2000-02-01

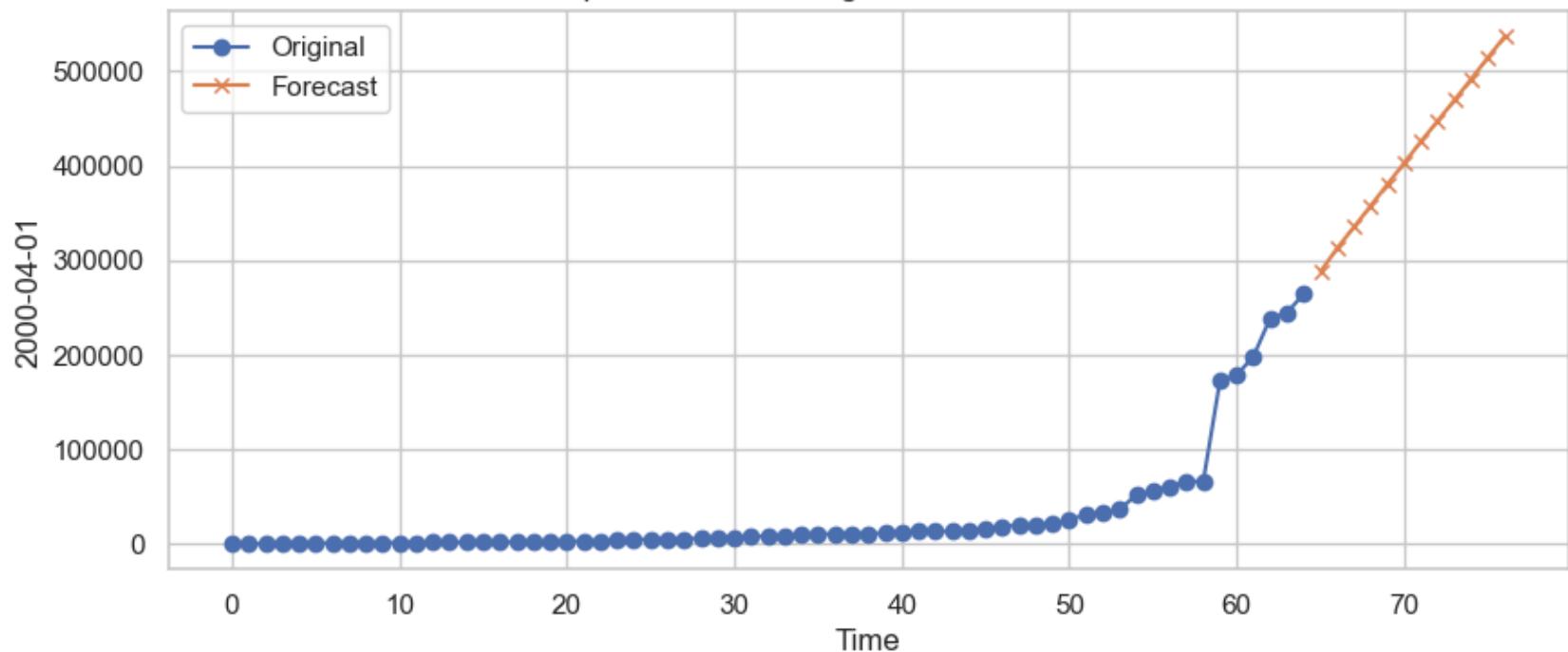


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

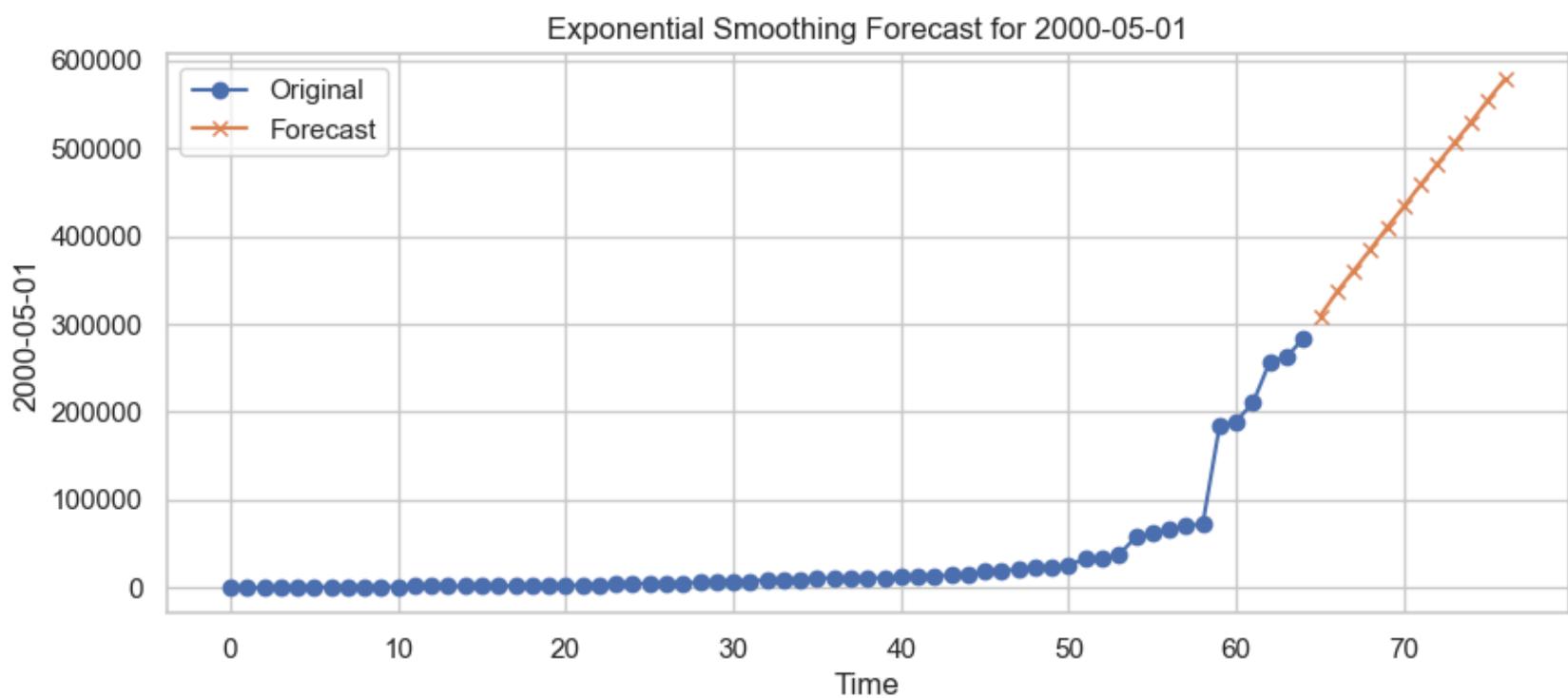


```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

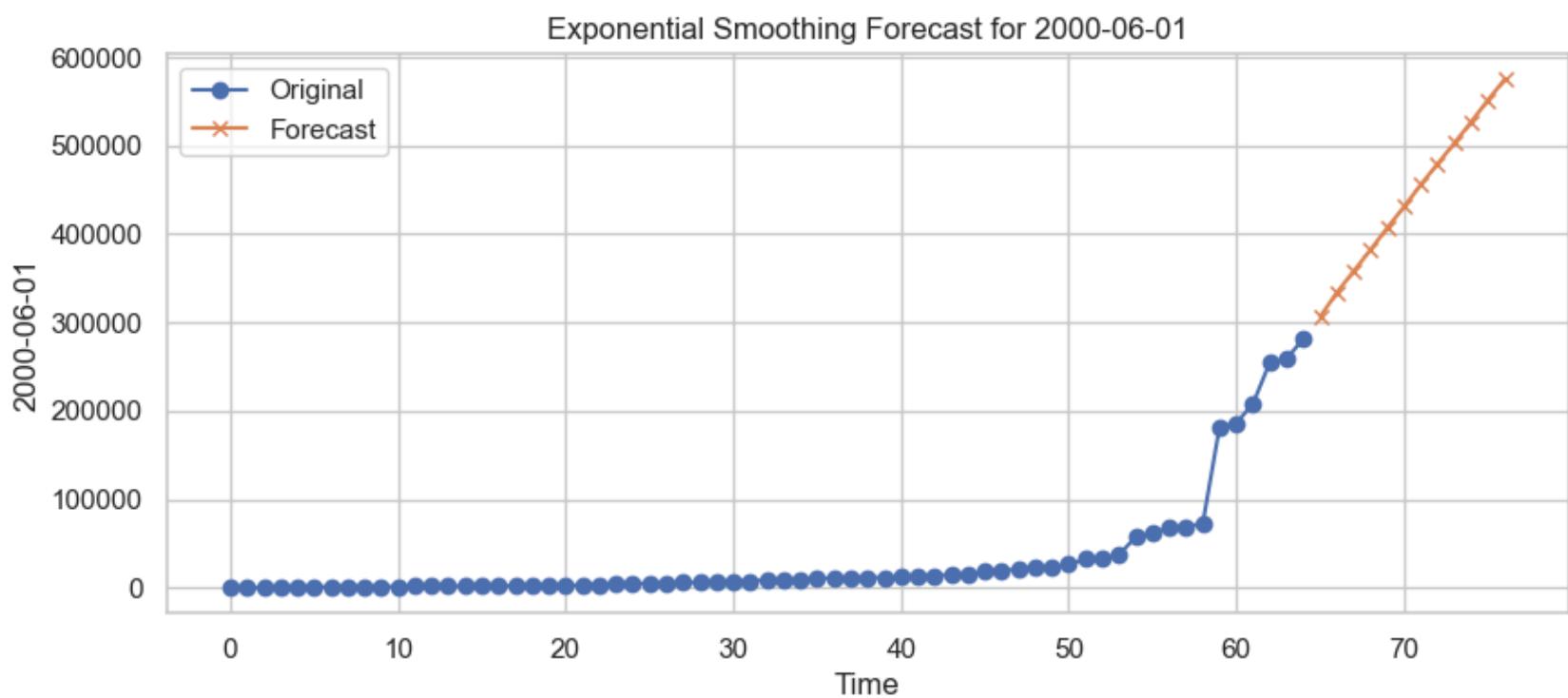
Exponential Smoothing Forecast for 2000-04-01



```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

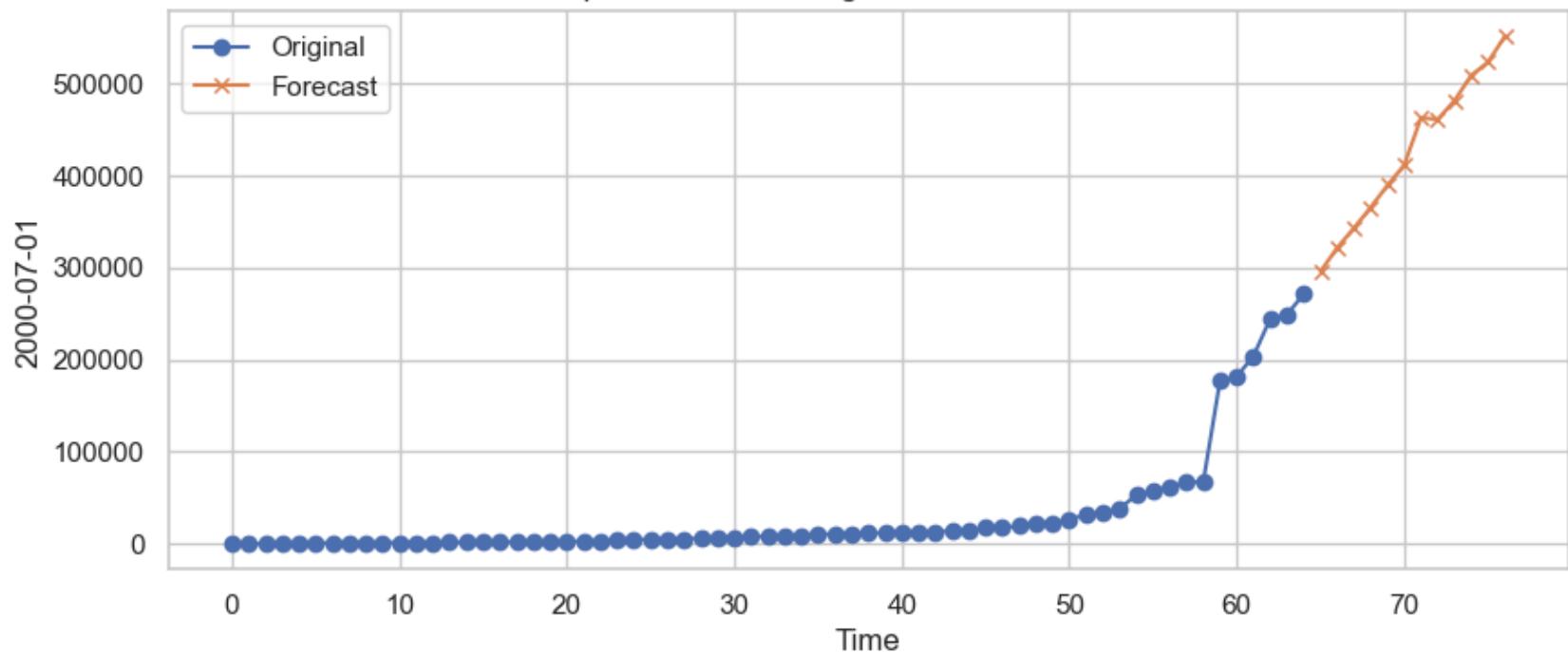


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

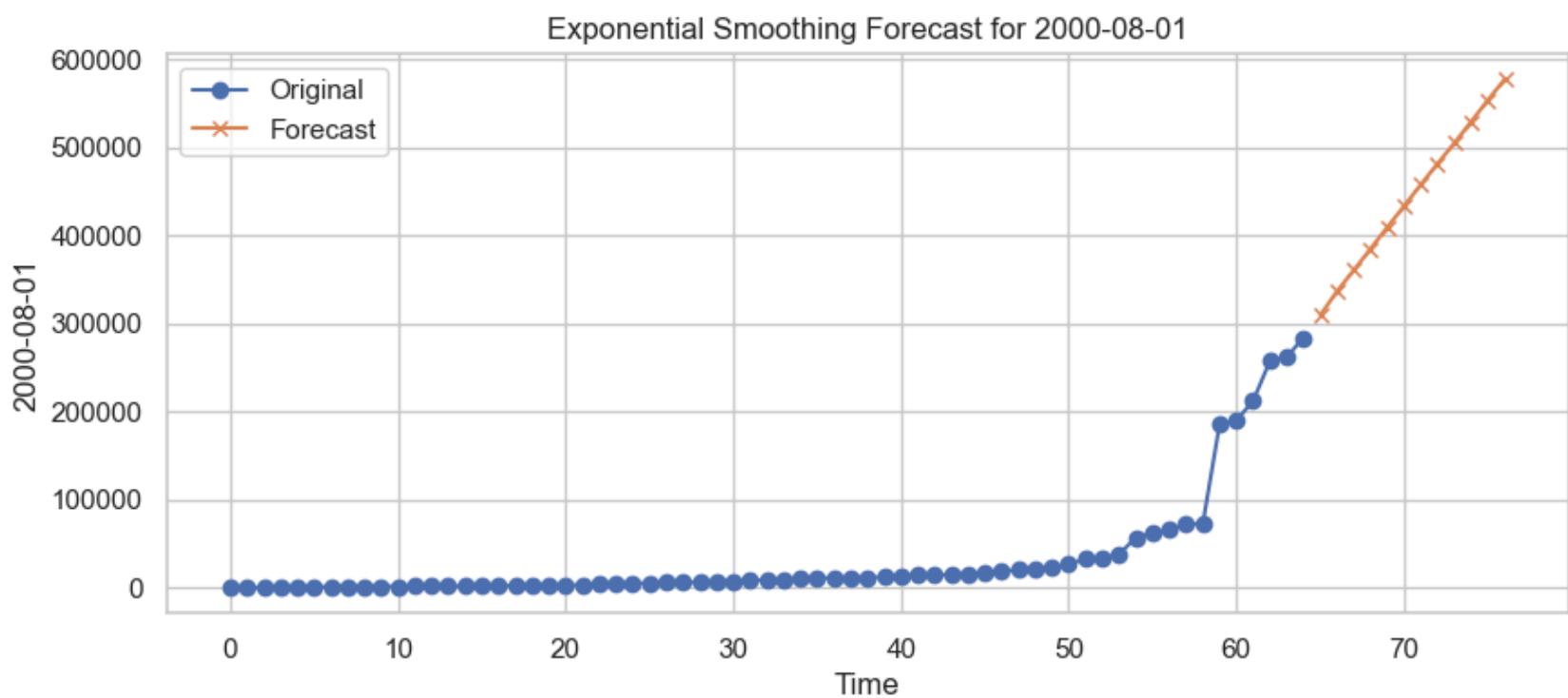


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2000-07-01

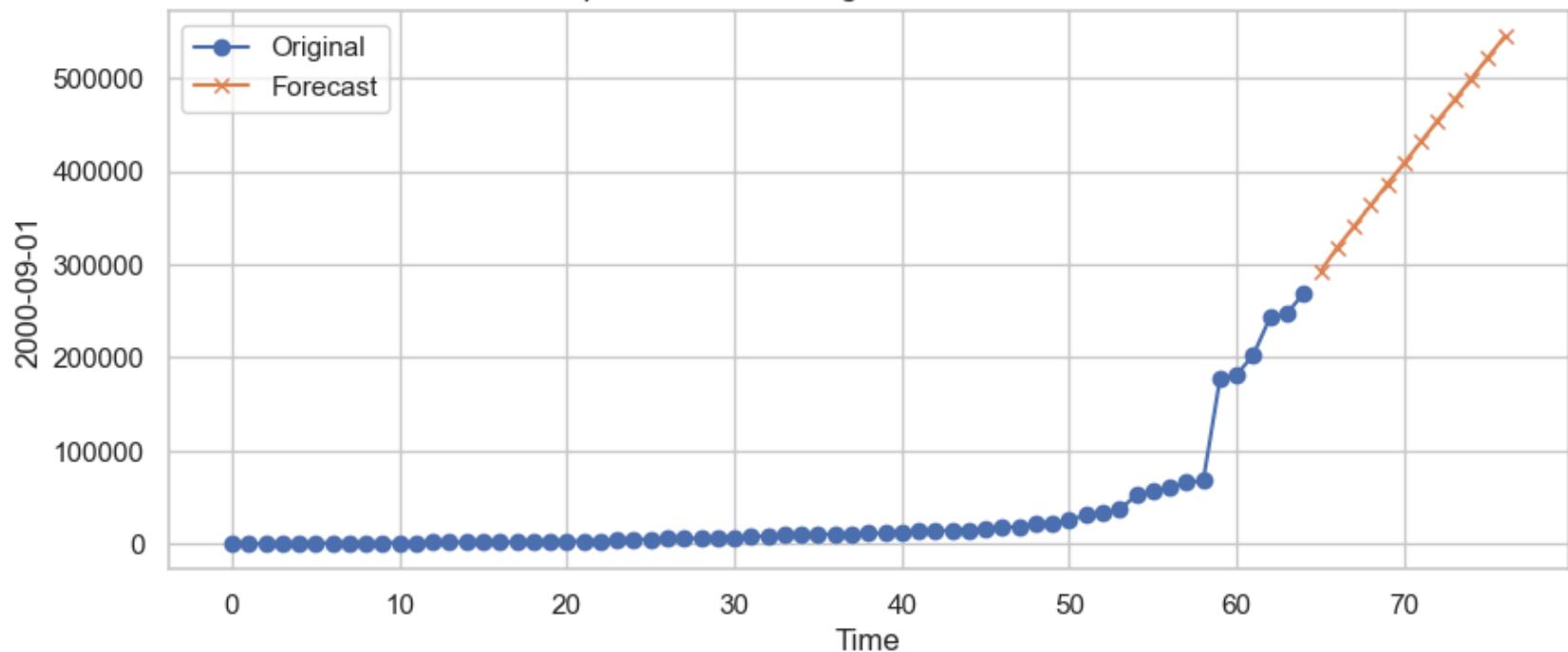


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



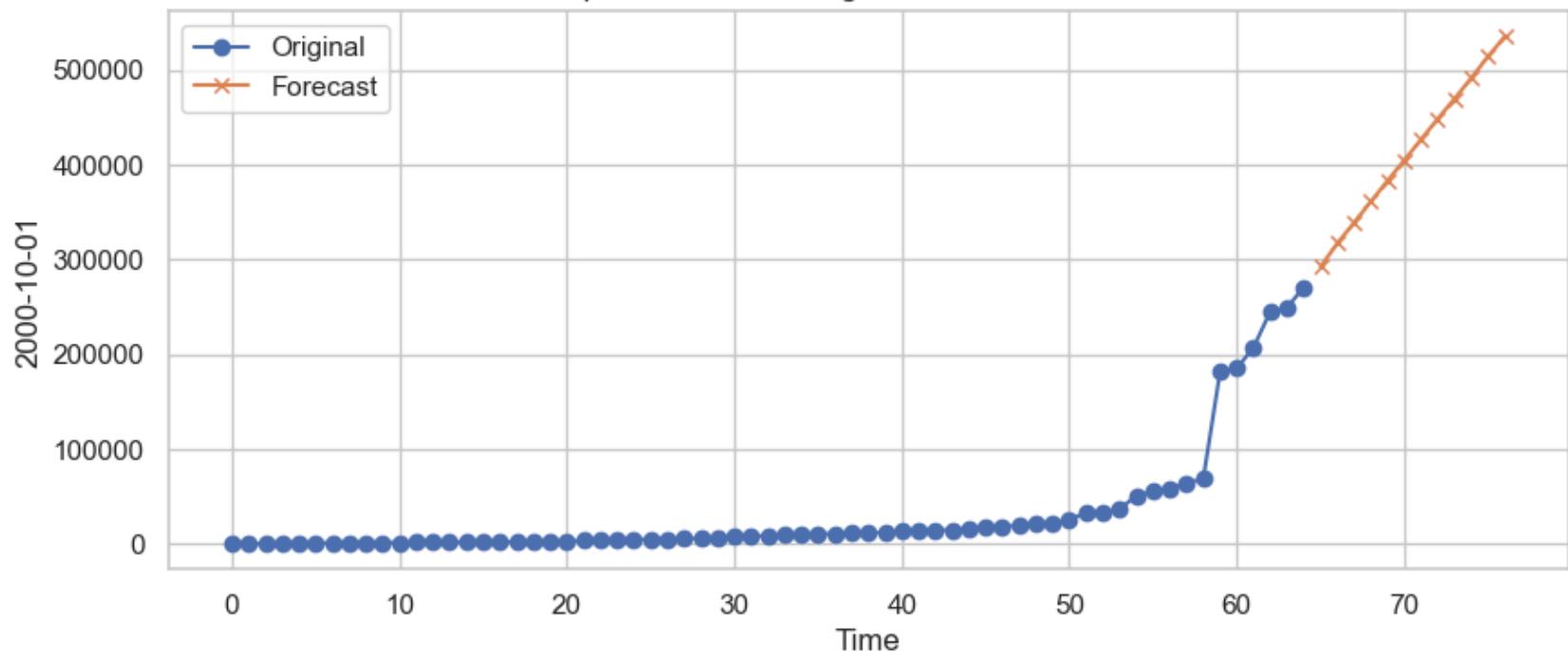
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2000-09-01



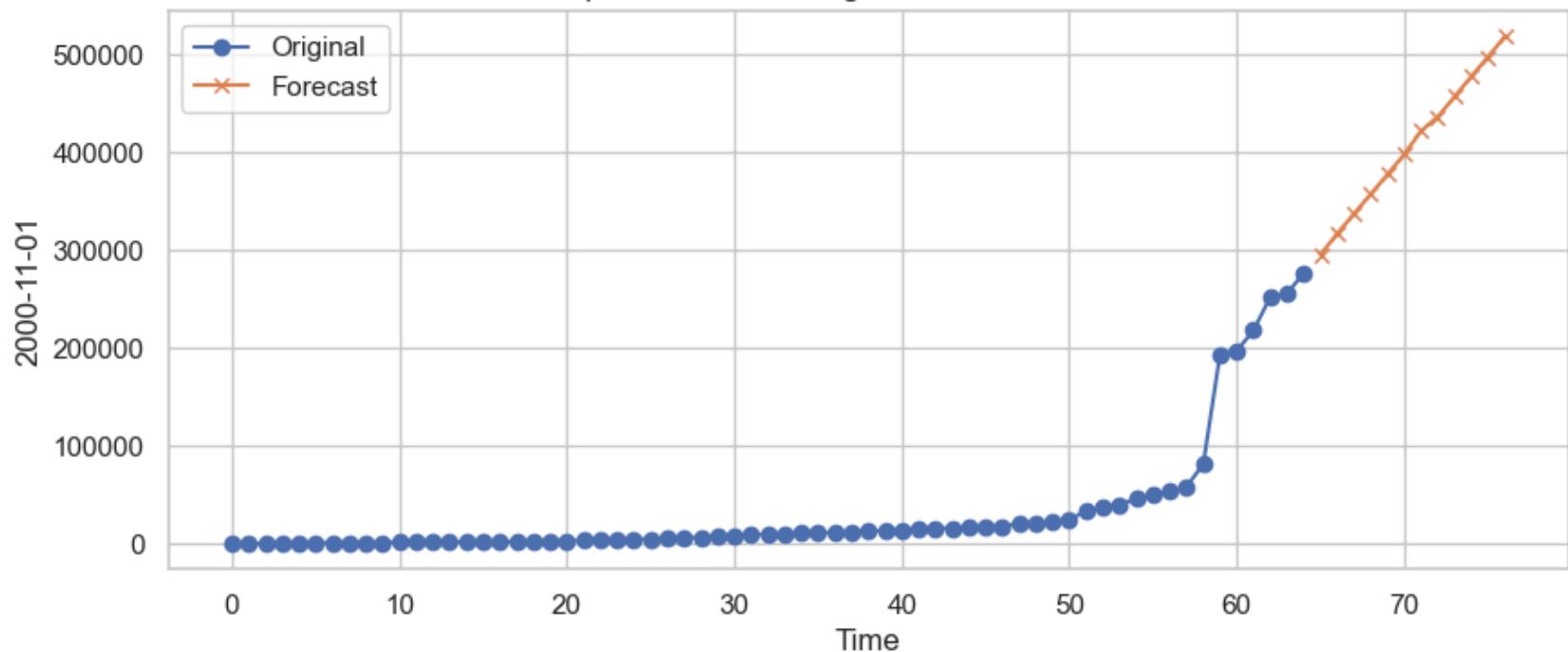
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2000-10-01

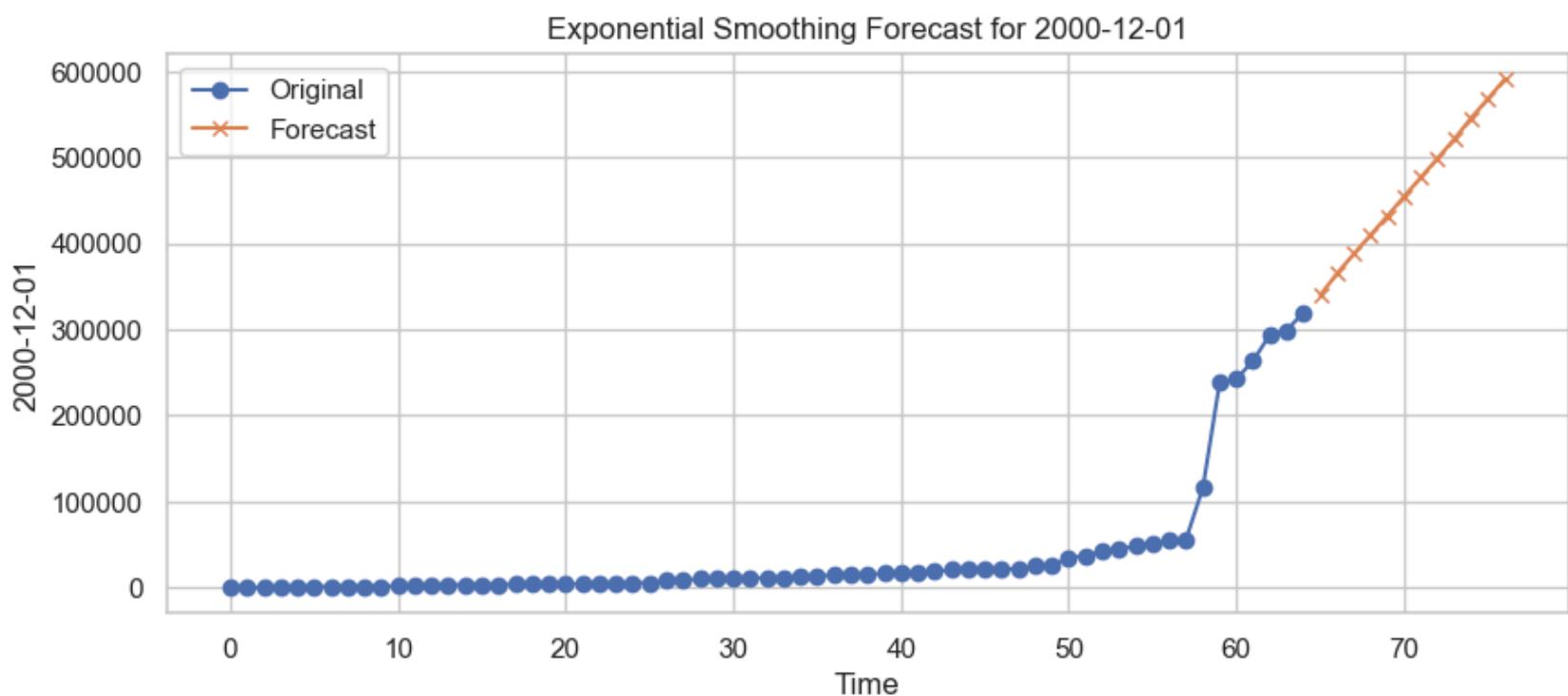


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2000-11-01

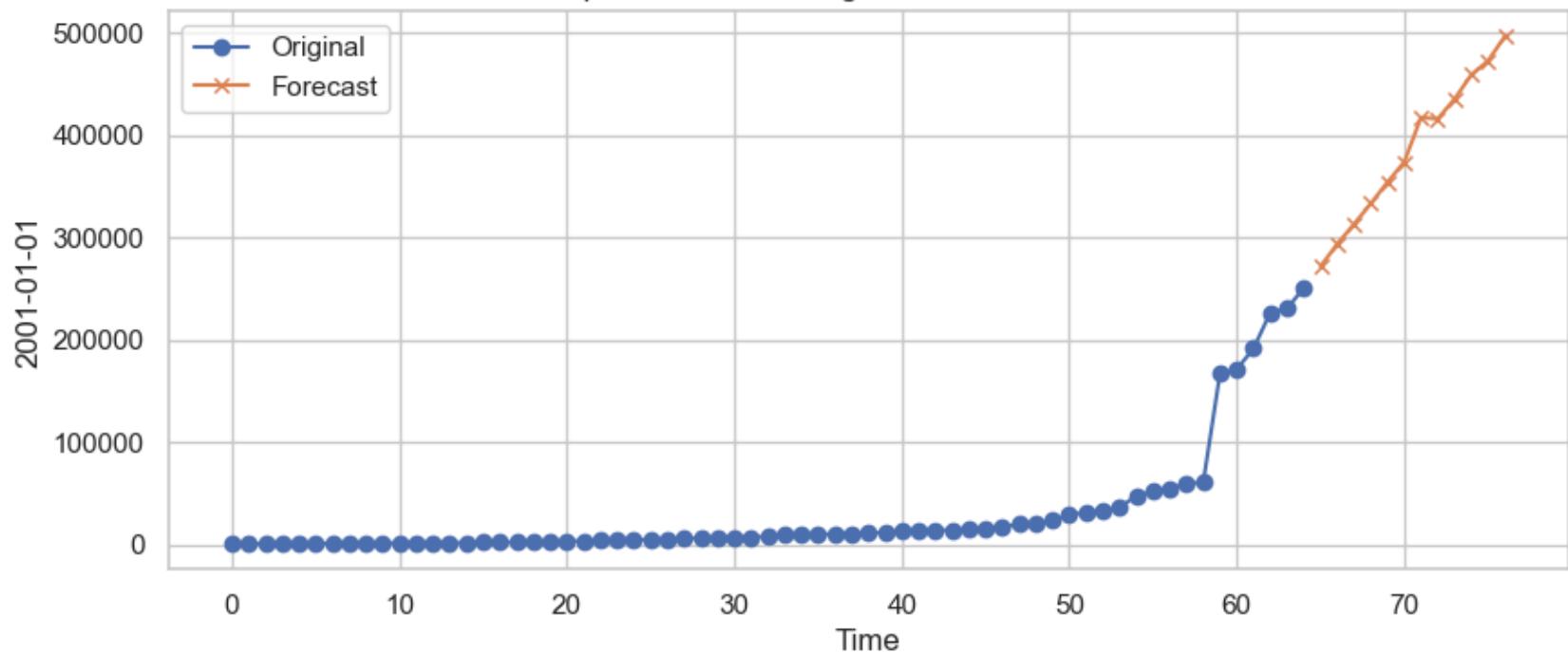


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



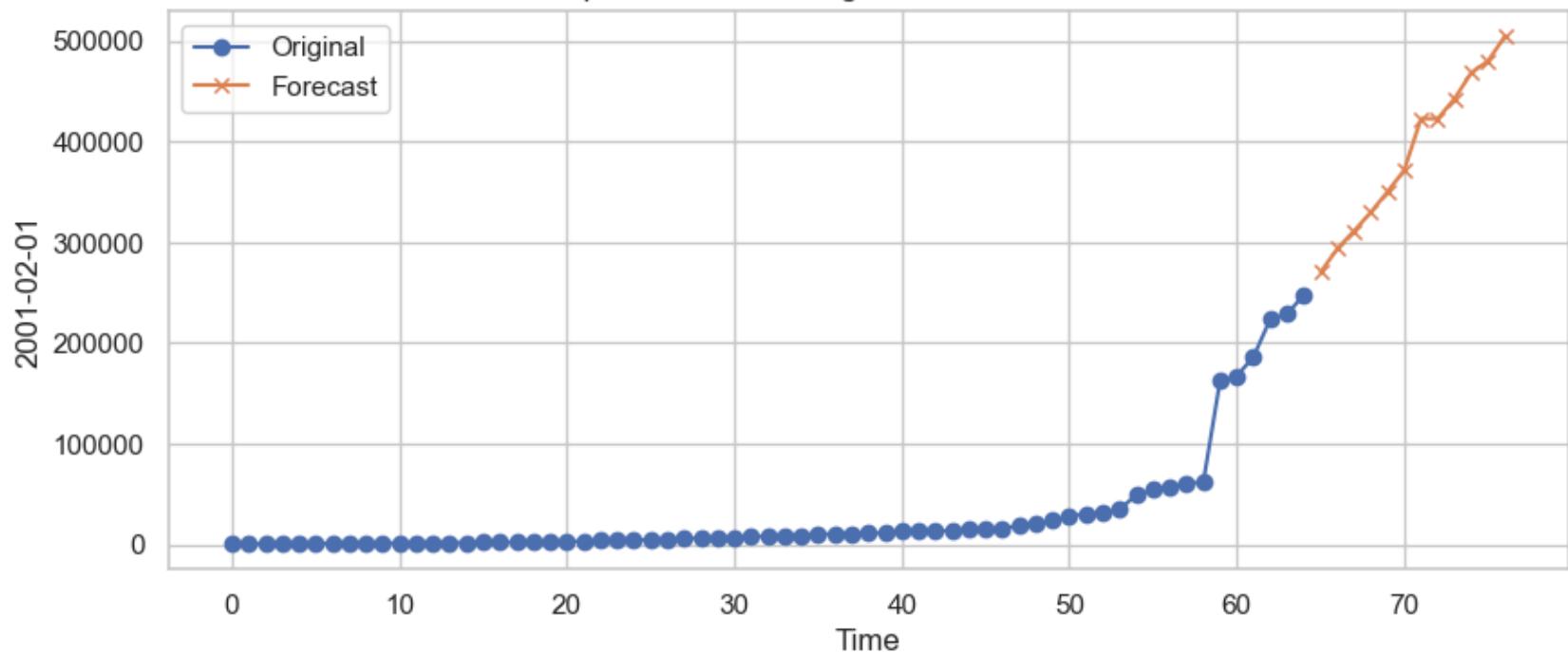
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2001-01-01

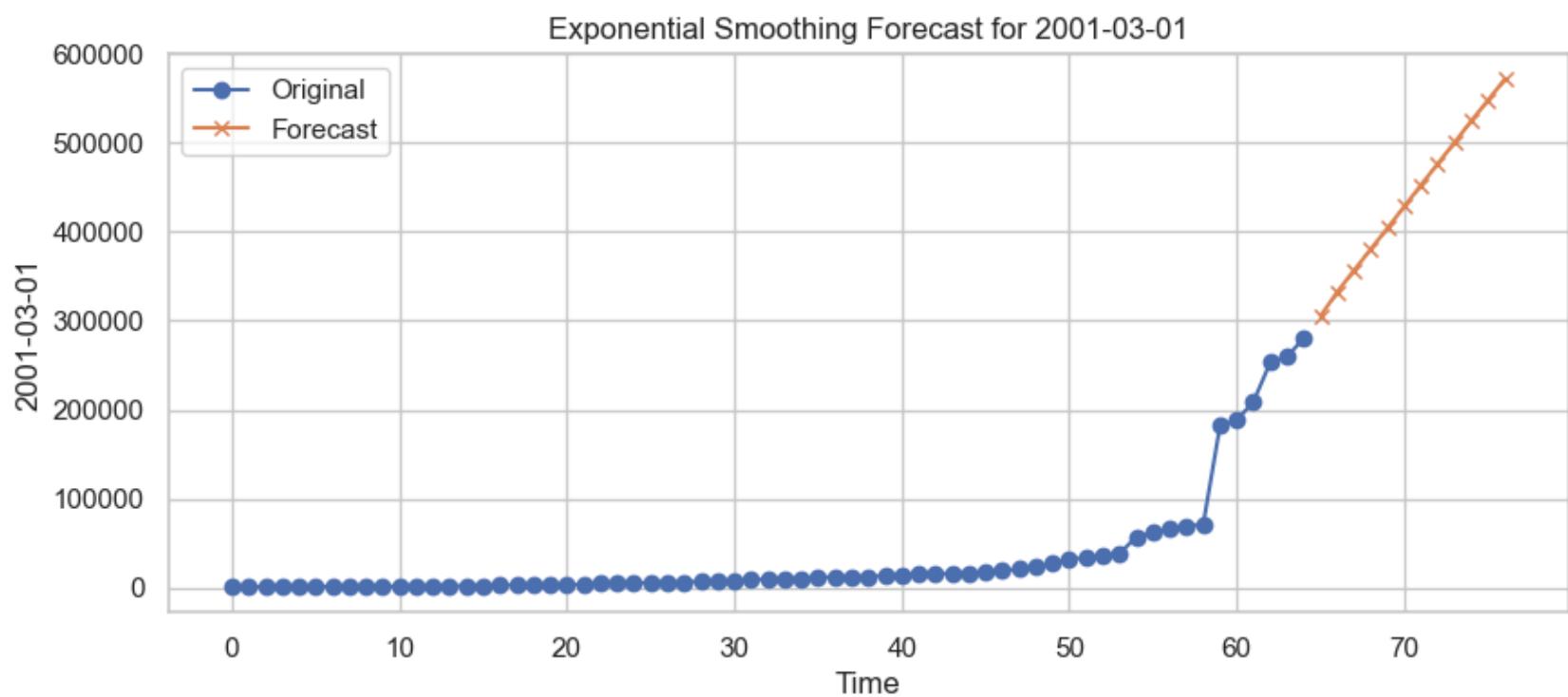


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2001-02-01

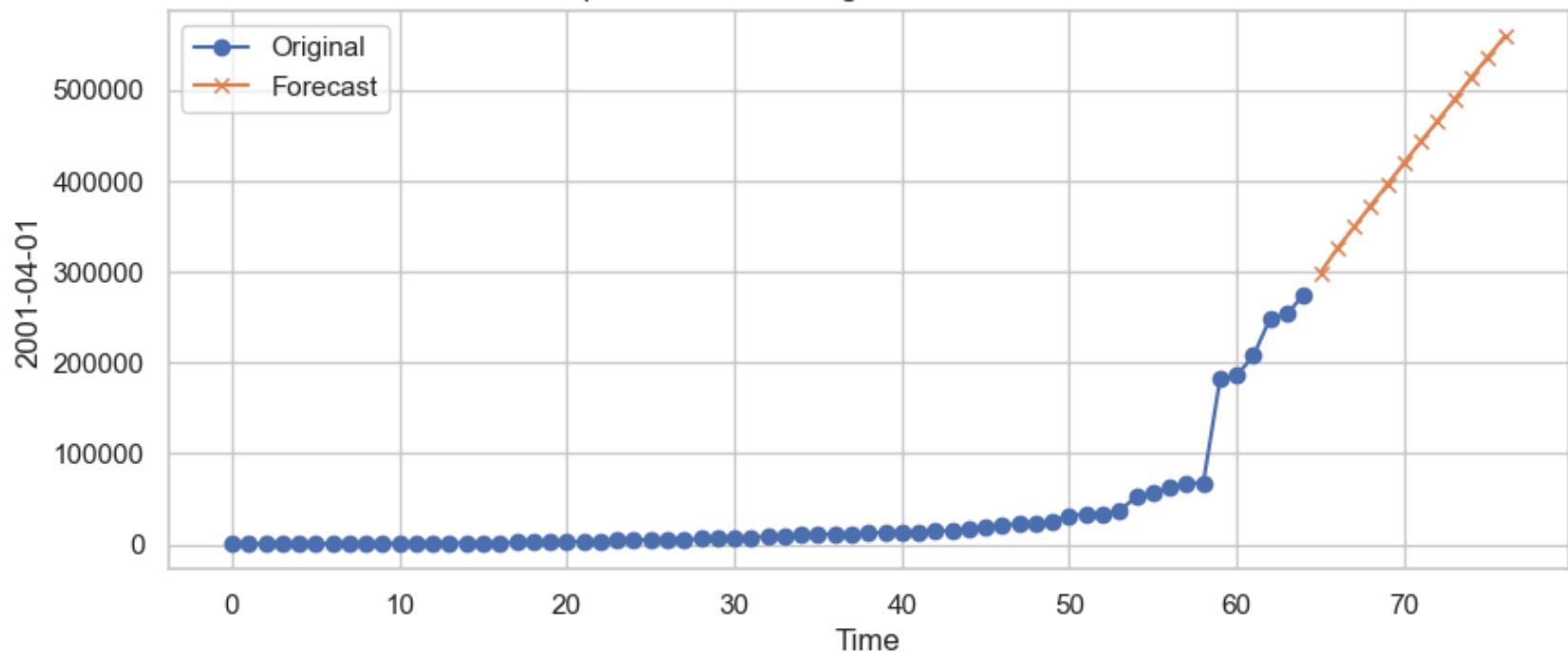


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



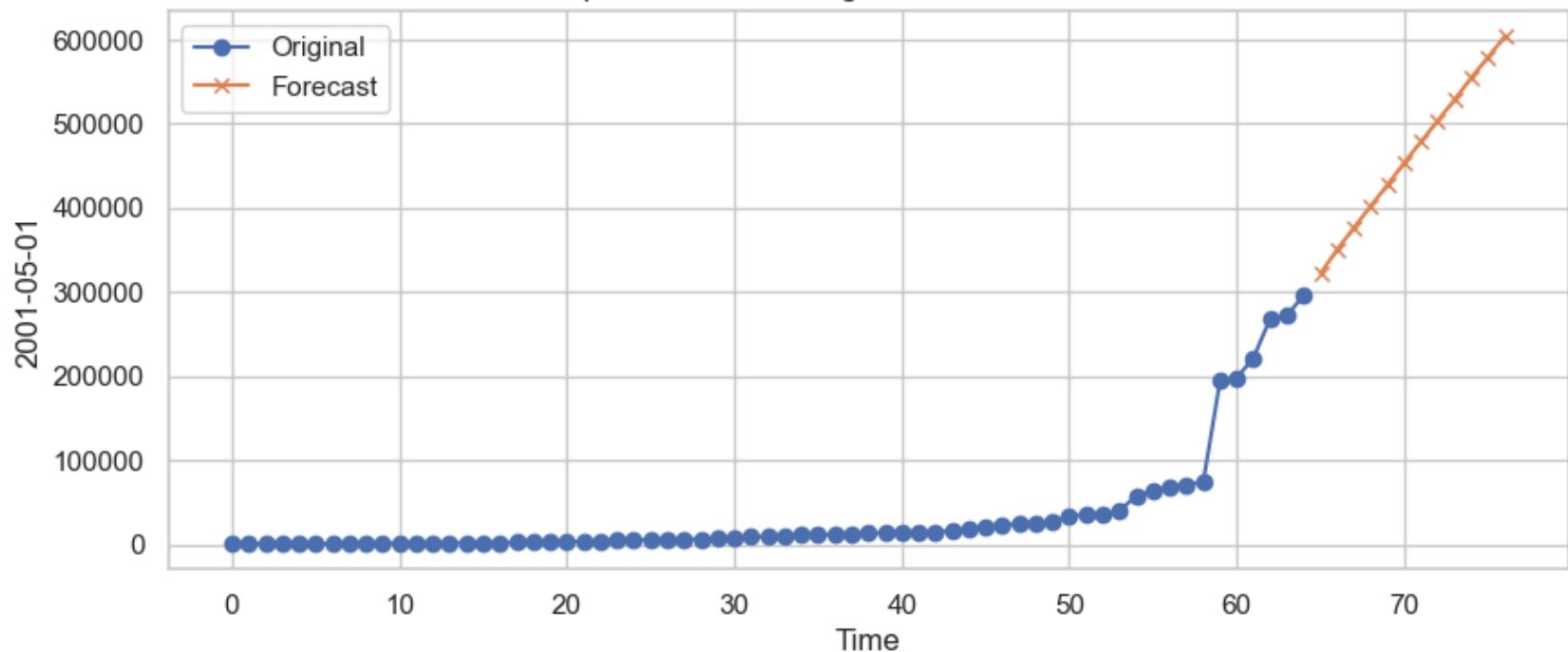
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2001-04-01

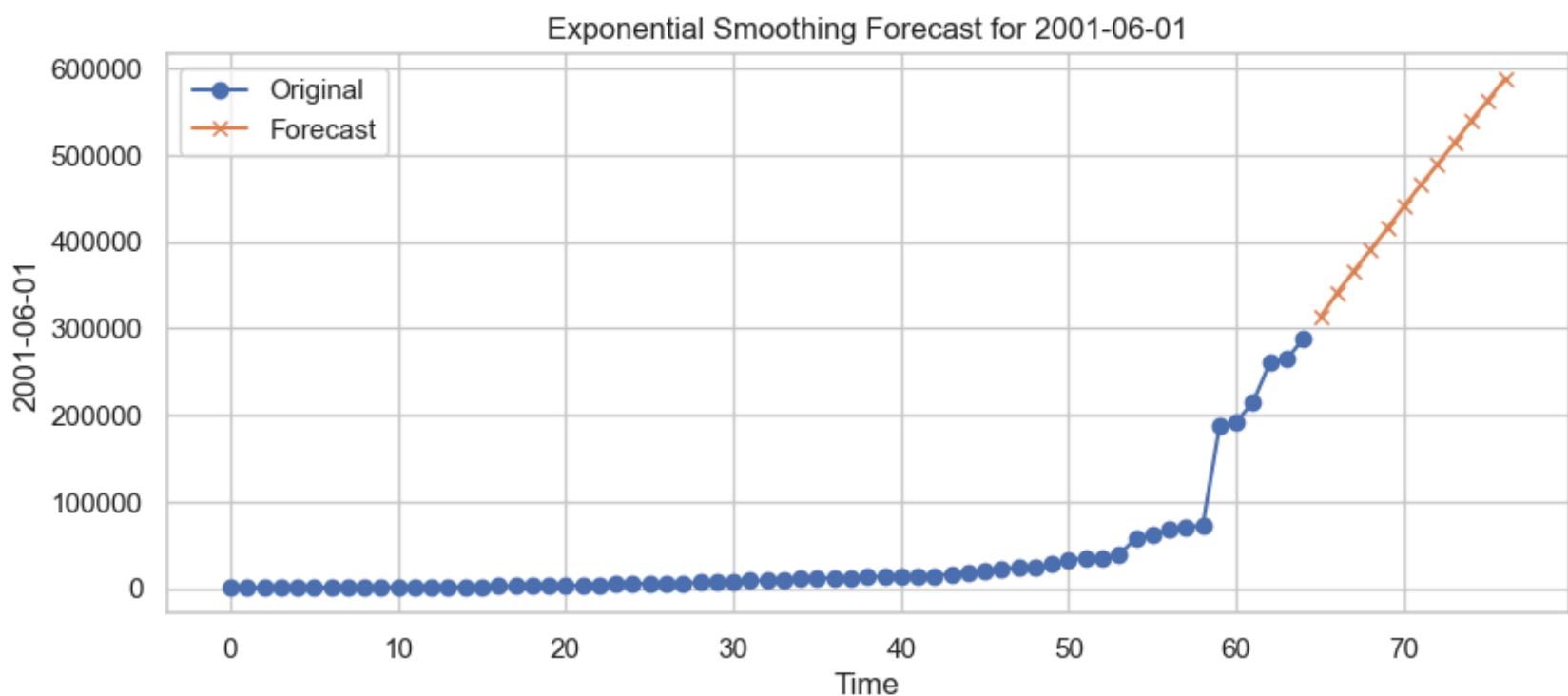


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2001-05-01

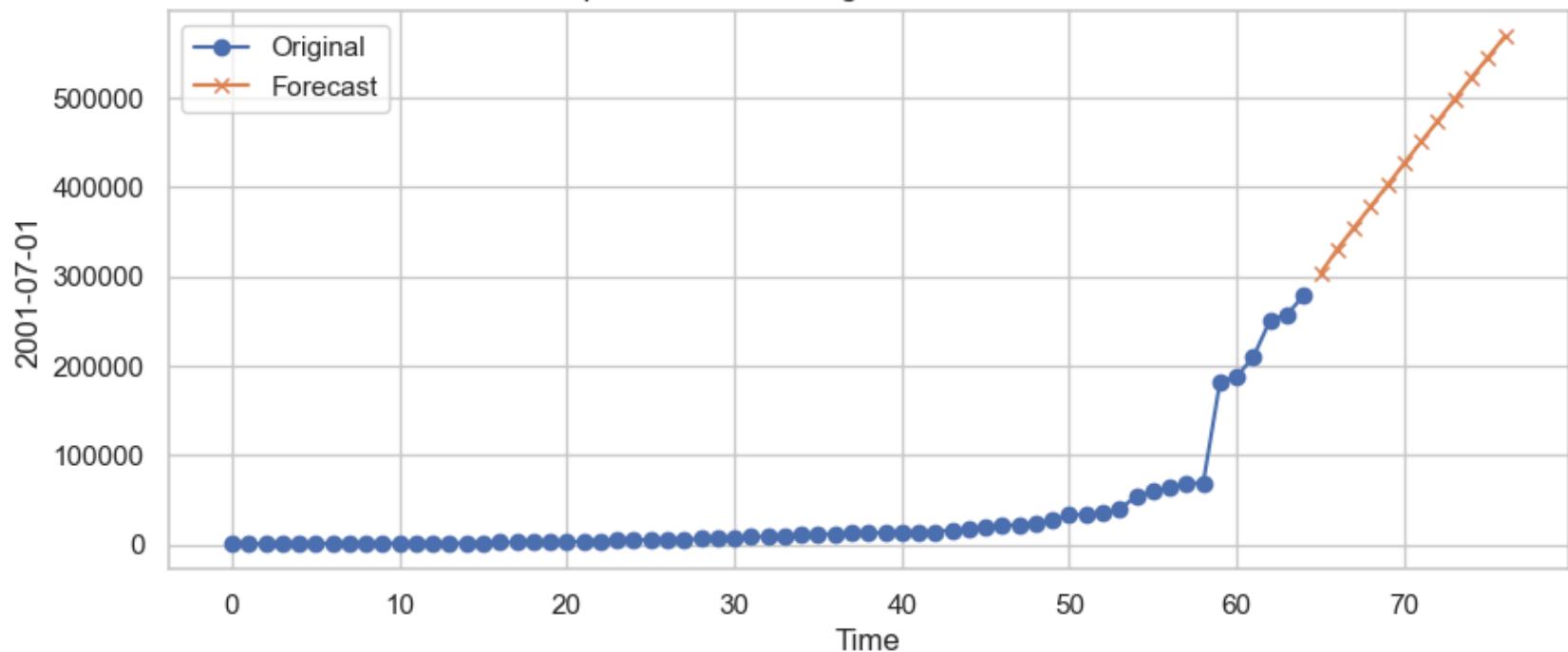


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



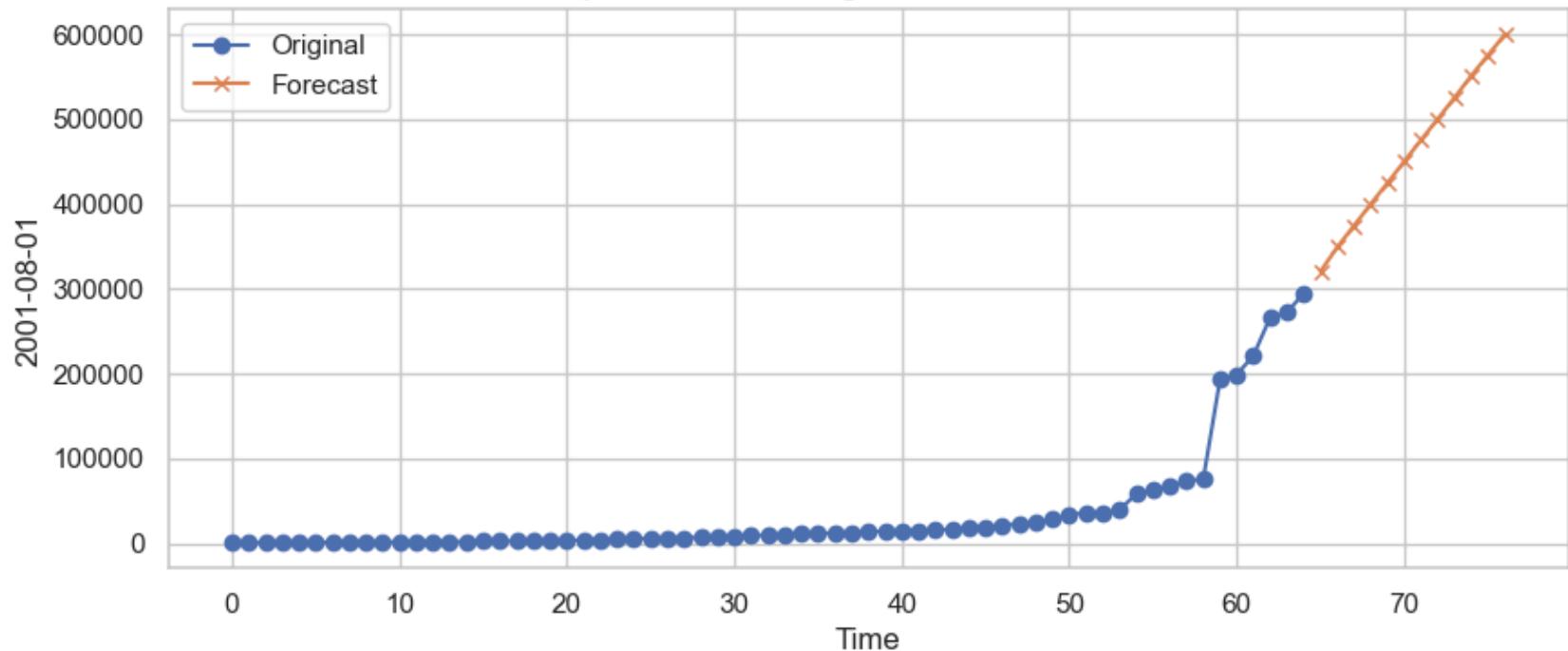
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2001-07-01



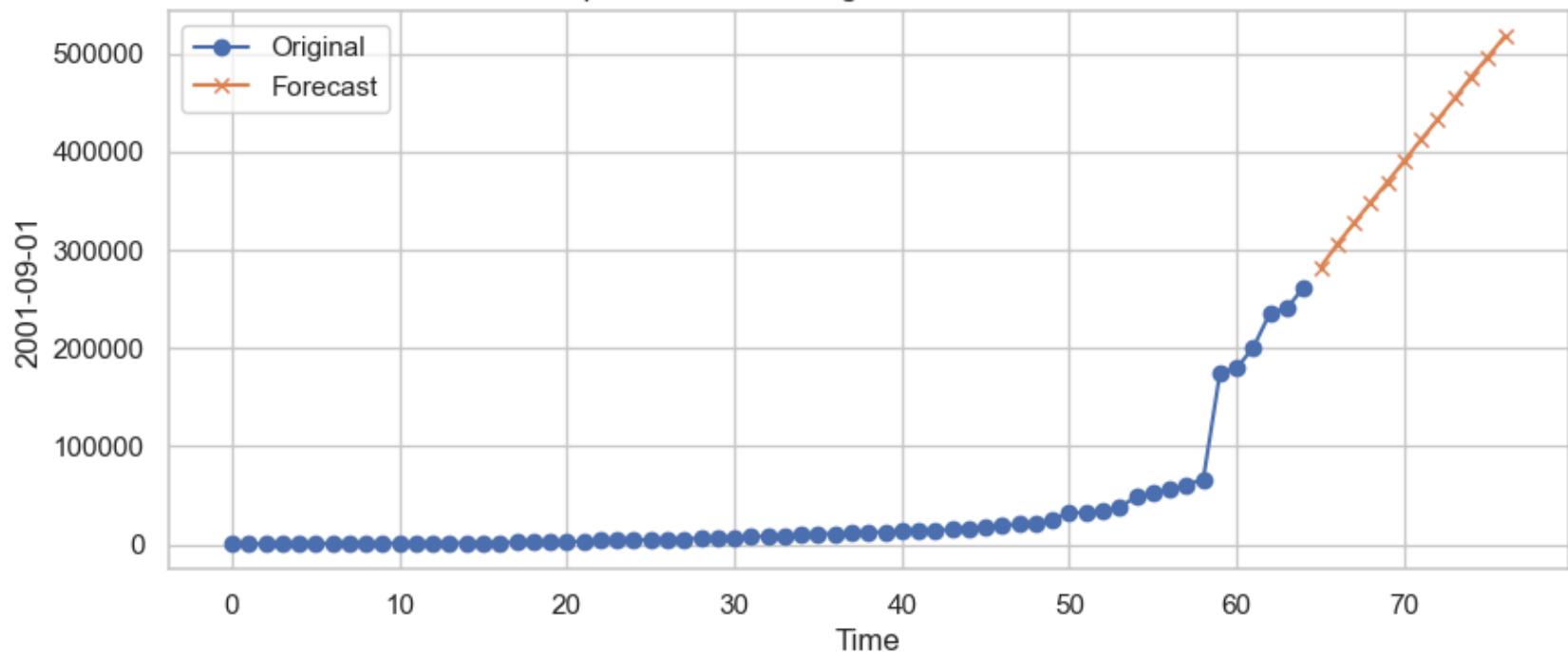
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2001-08-01



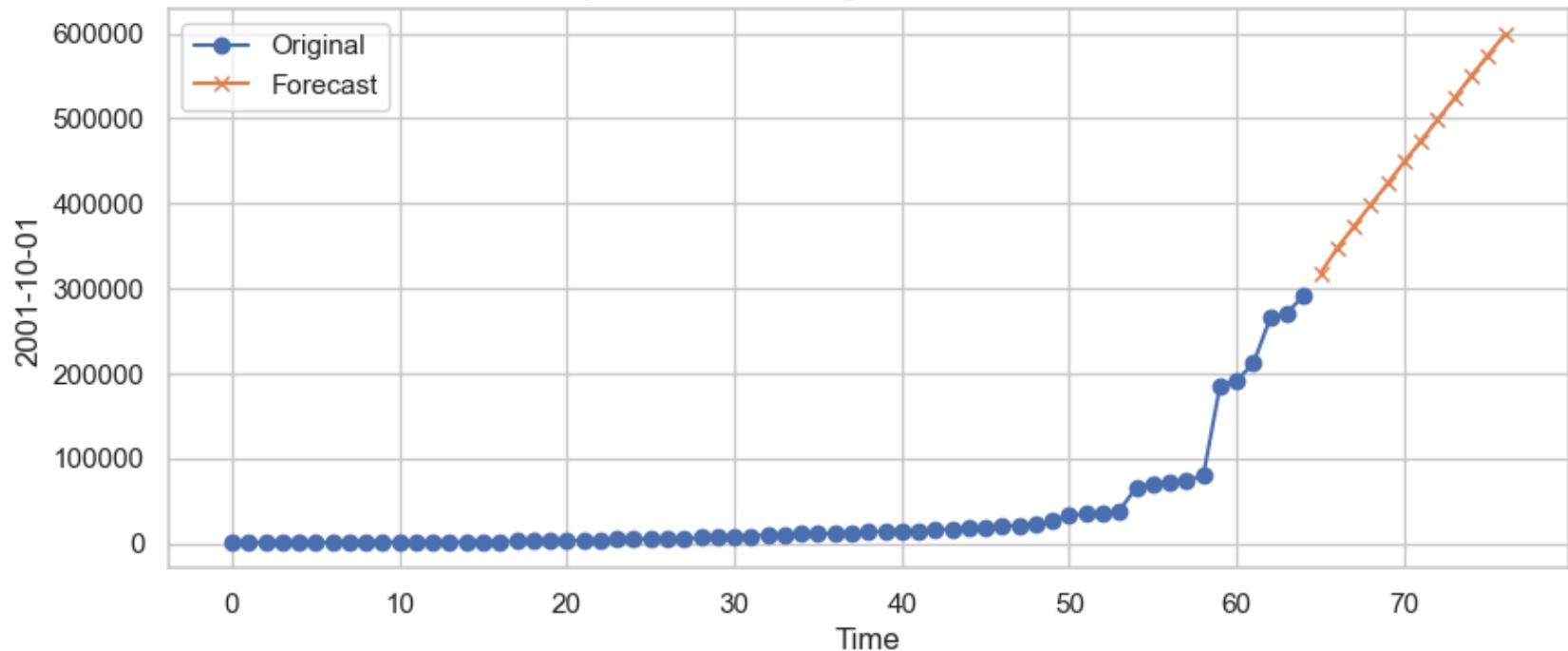
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2001-09-01



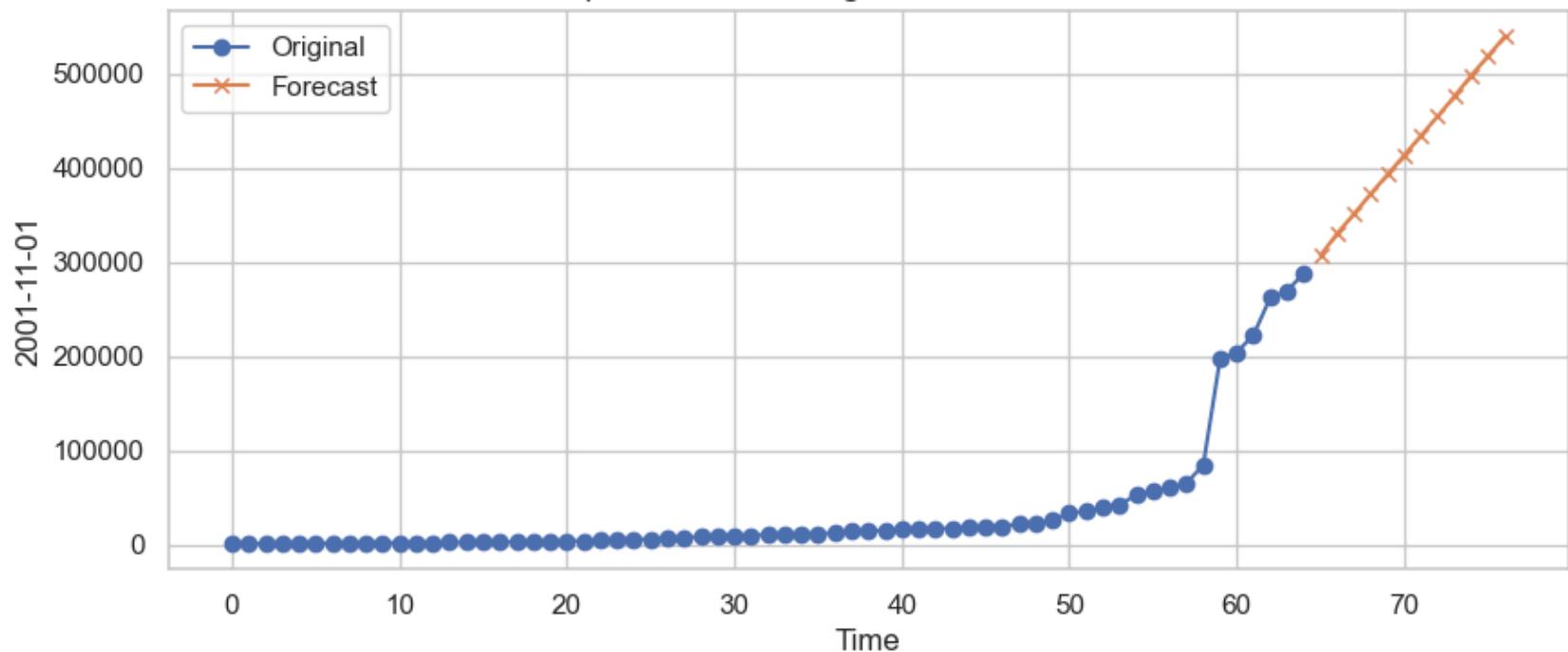
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2001-10-01



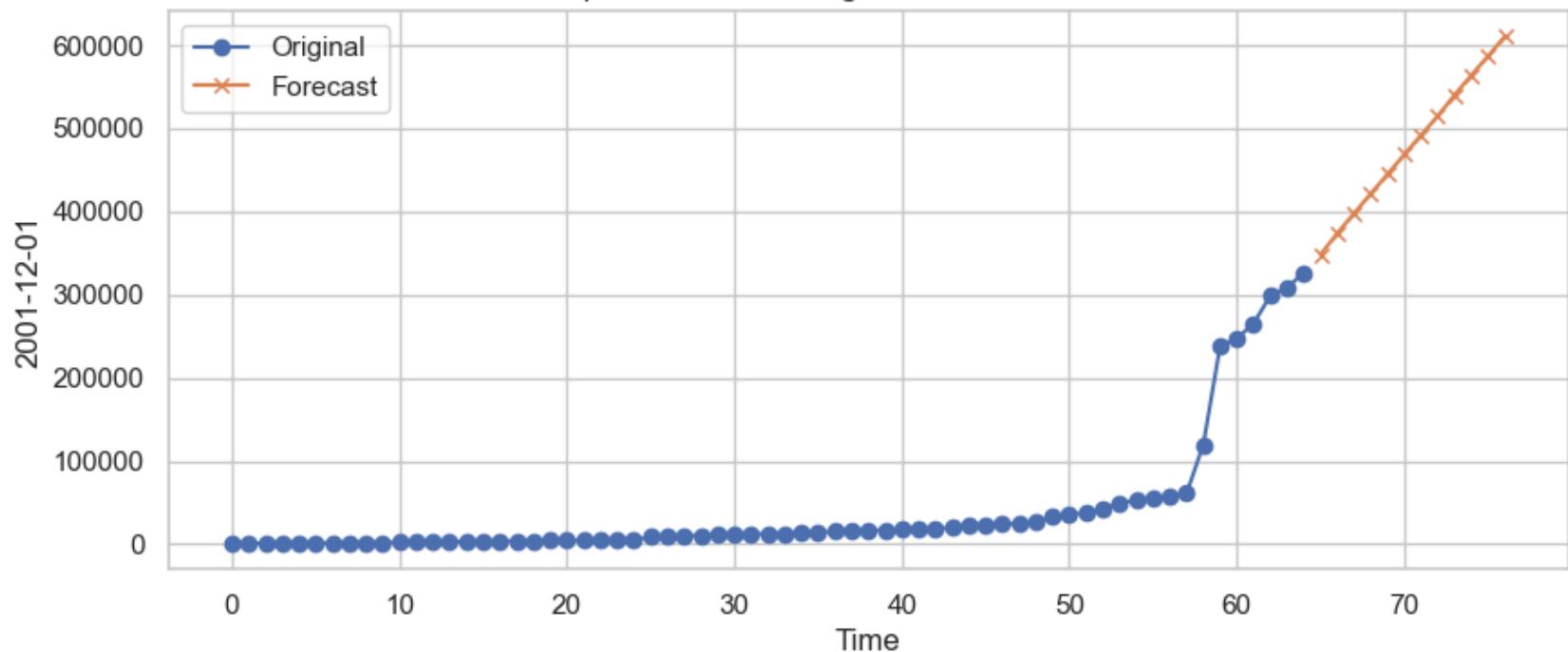
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2001-11-01



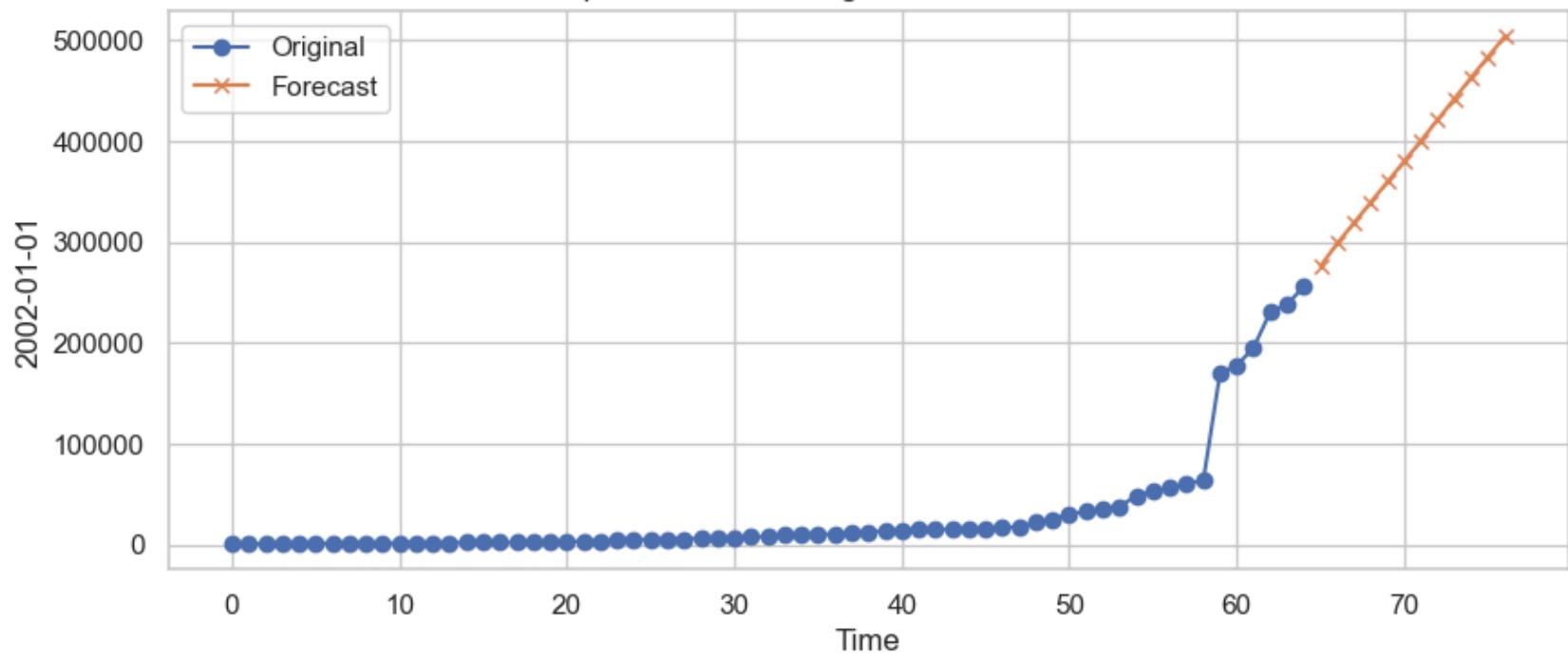
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2001-12-01



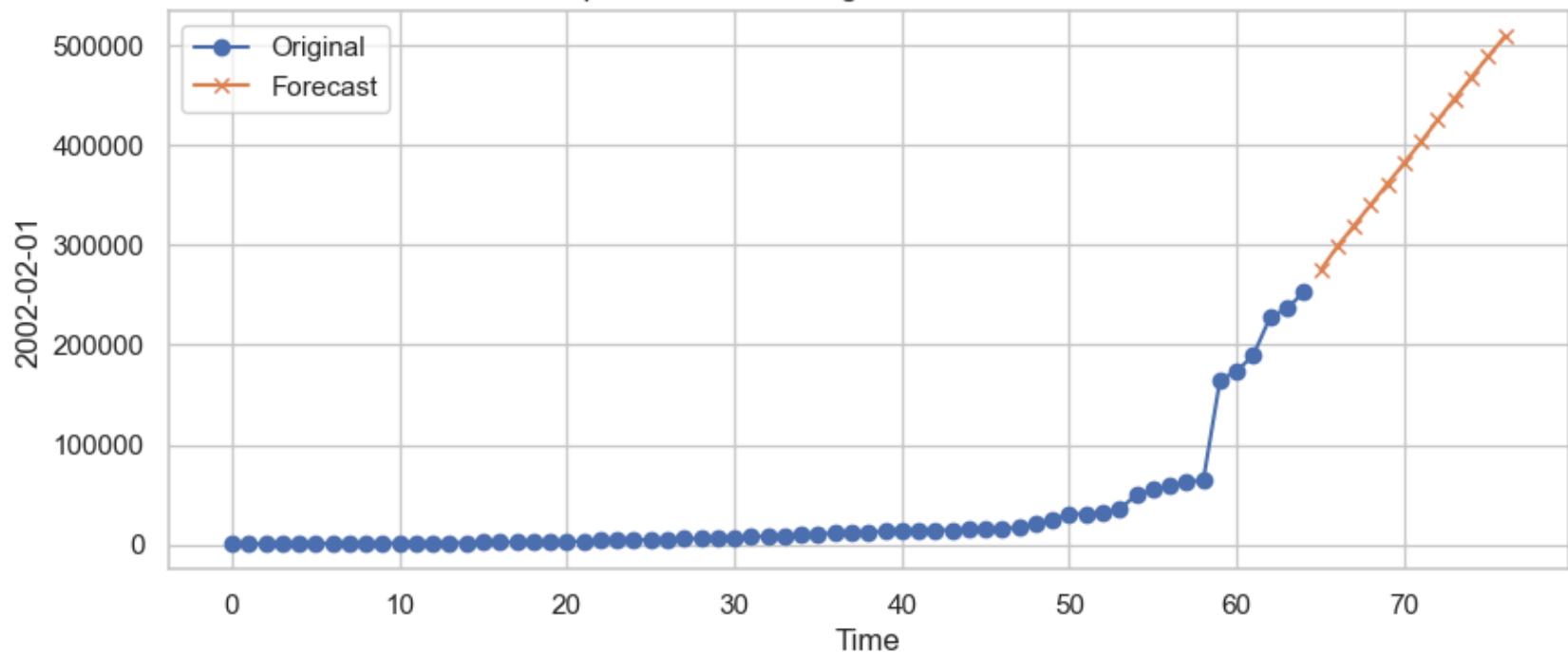
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2002-01-01

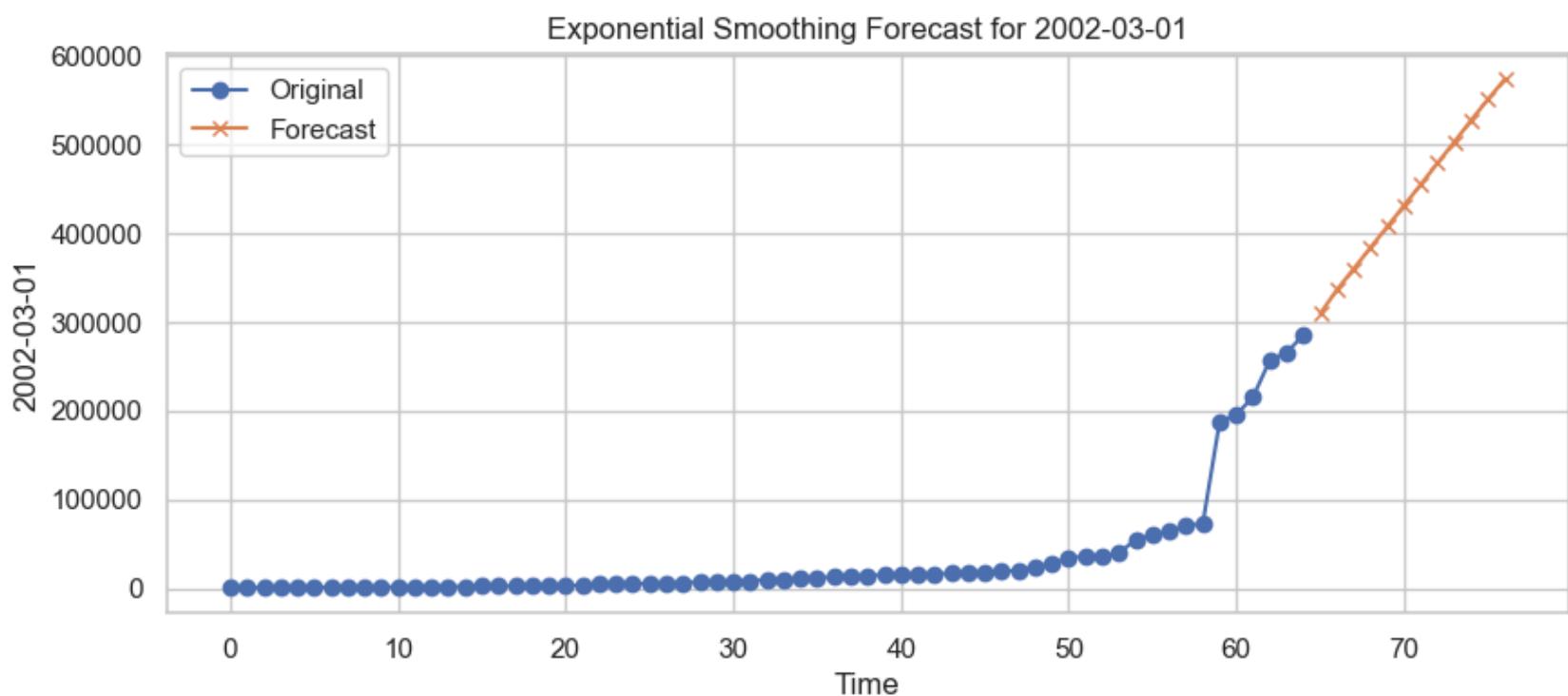


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

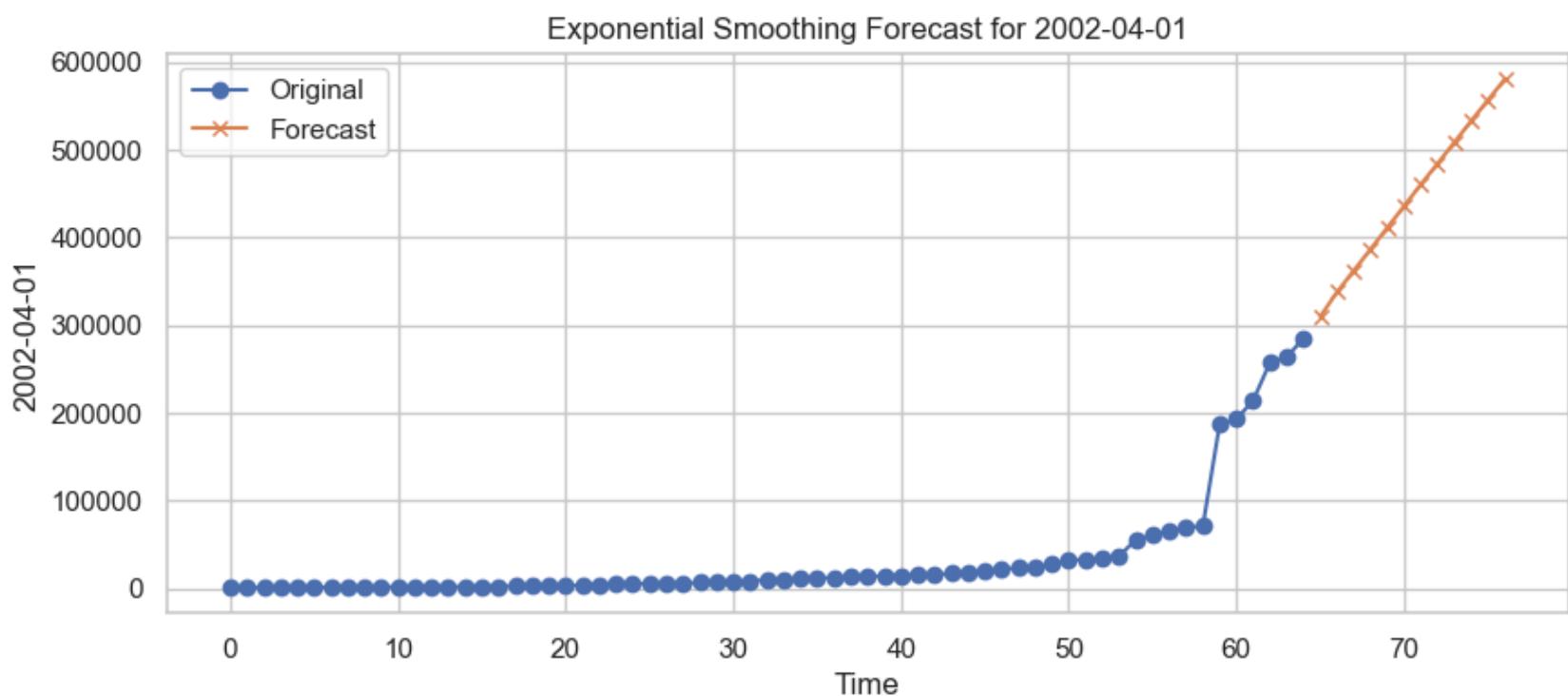
Exponential Smoothing Forecast for 2002-02-01



```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

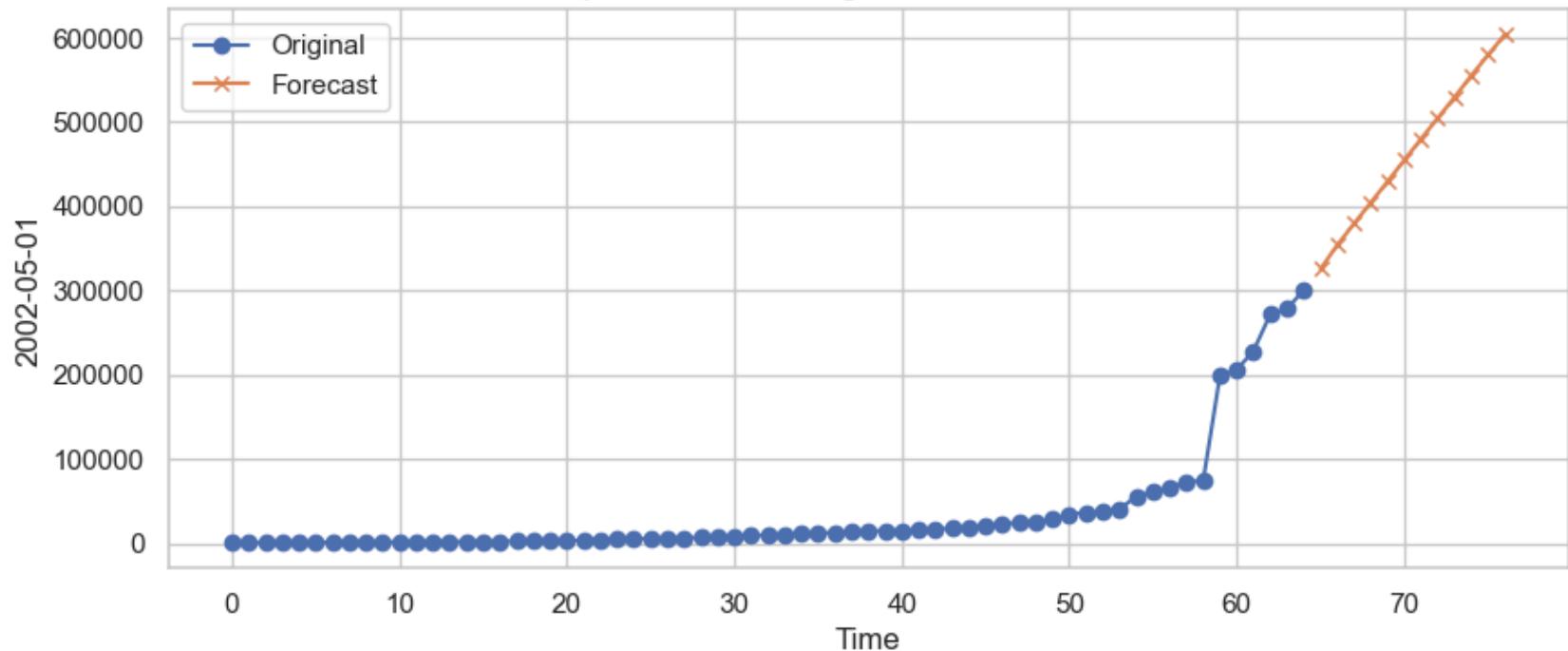


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

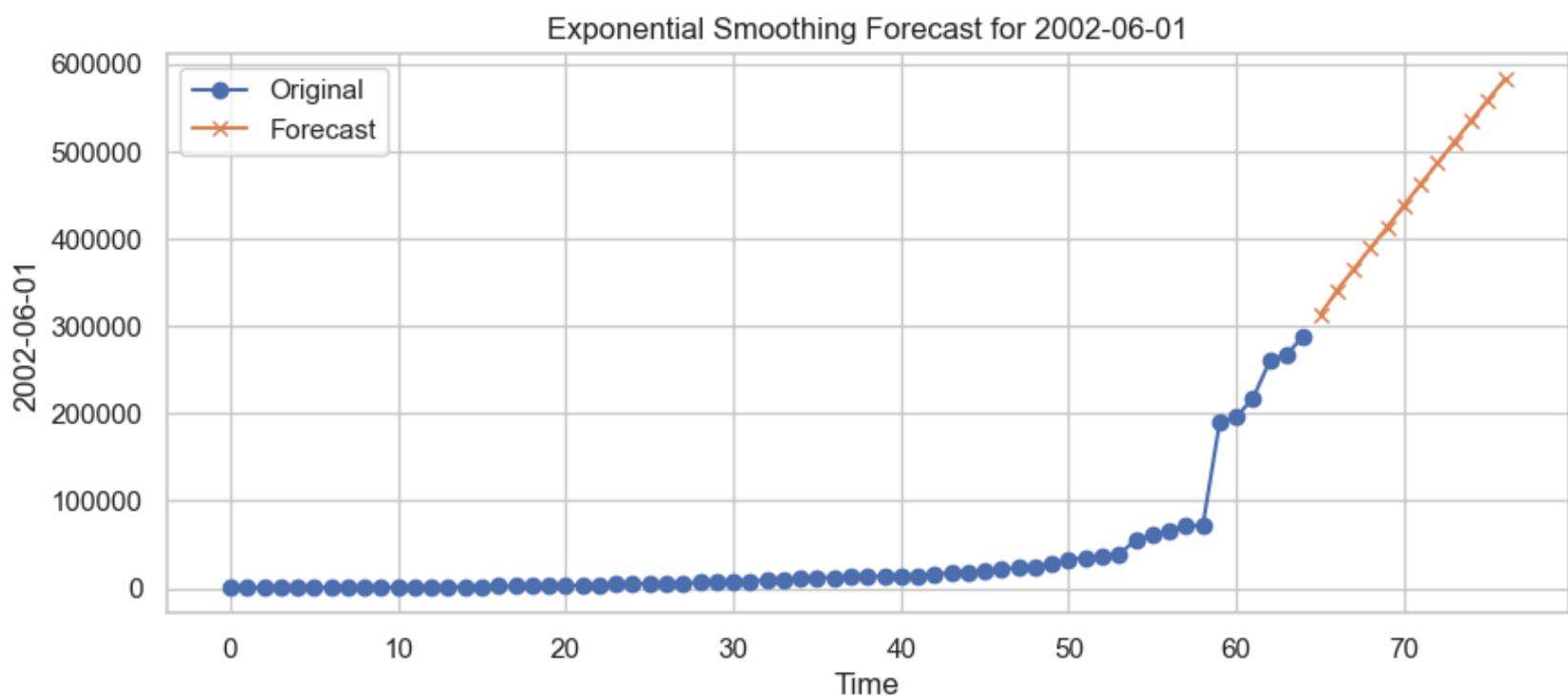


```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2002-05-01

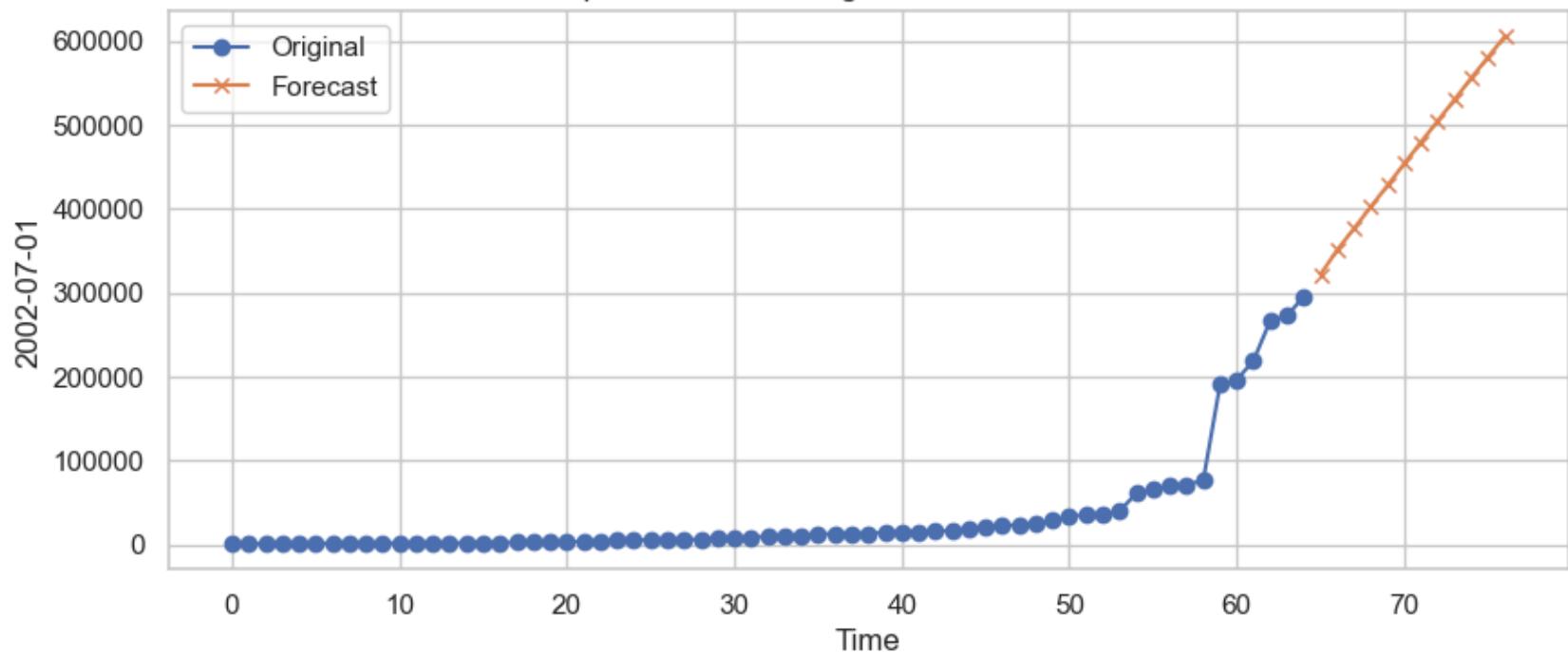


```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```



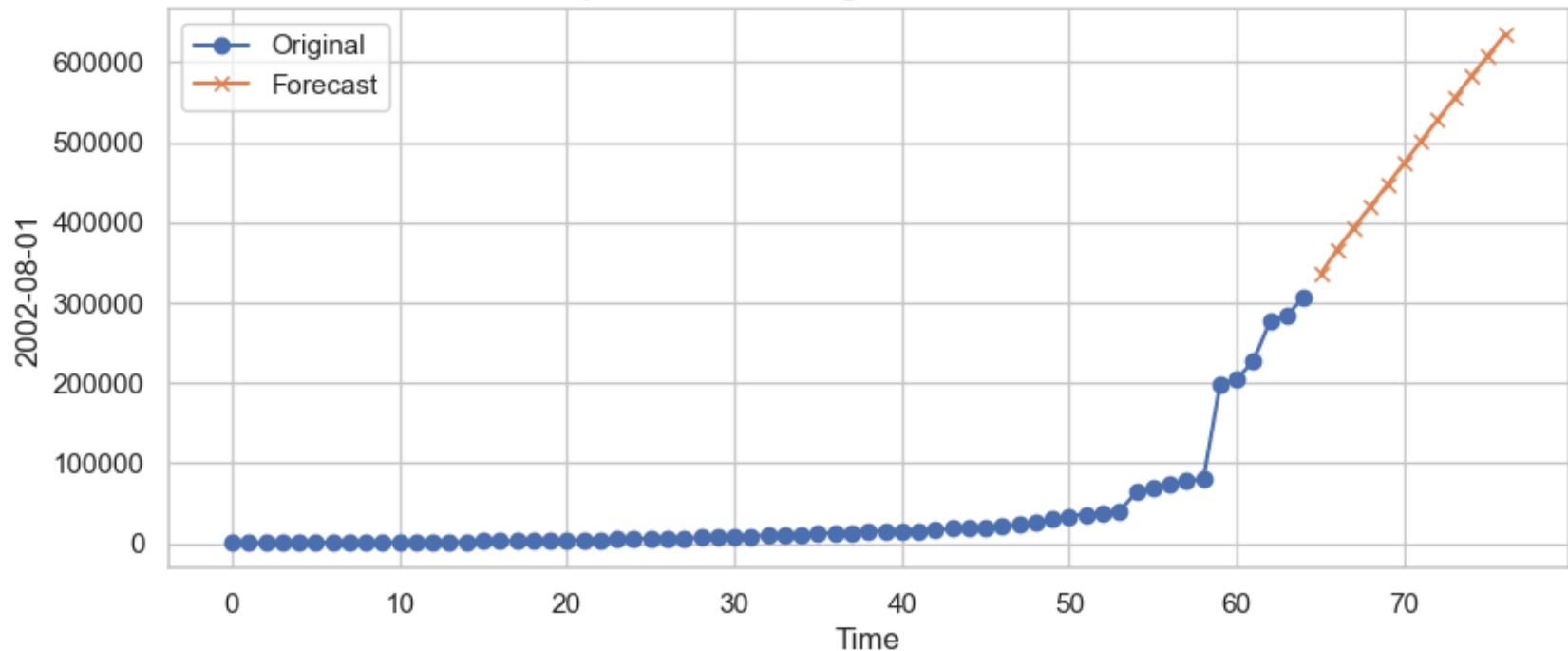
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2002-07-01



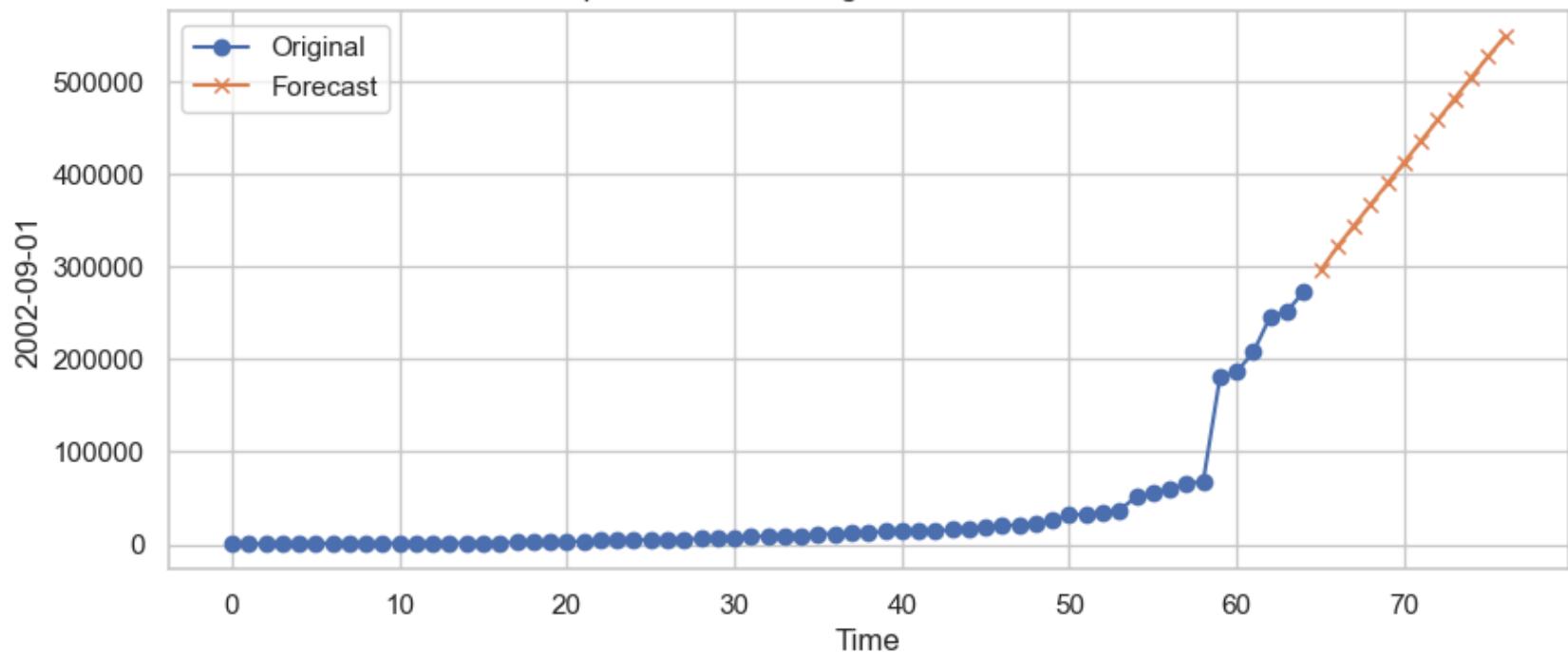
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2002-08-01



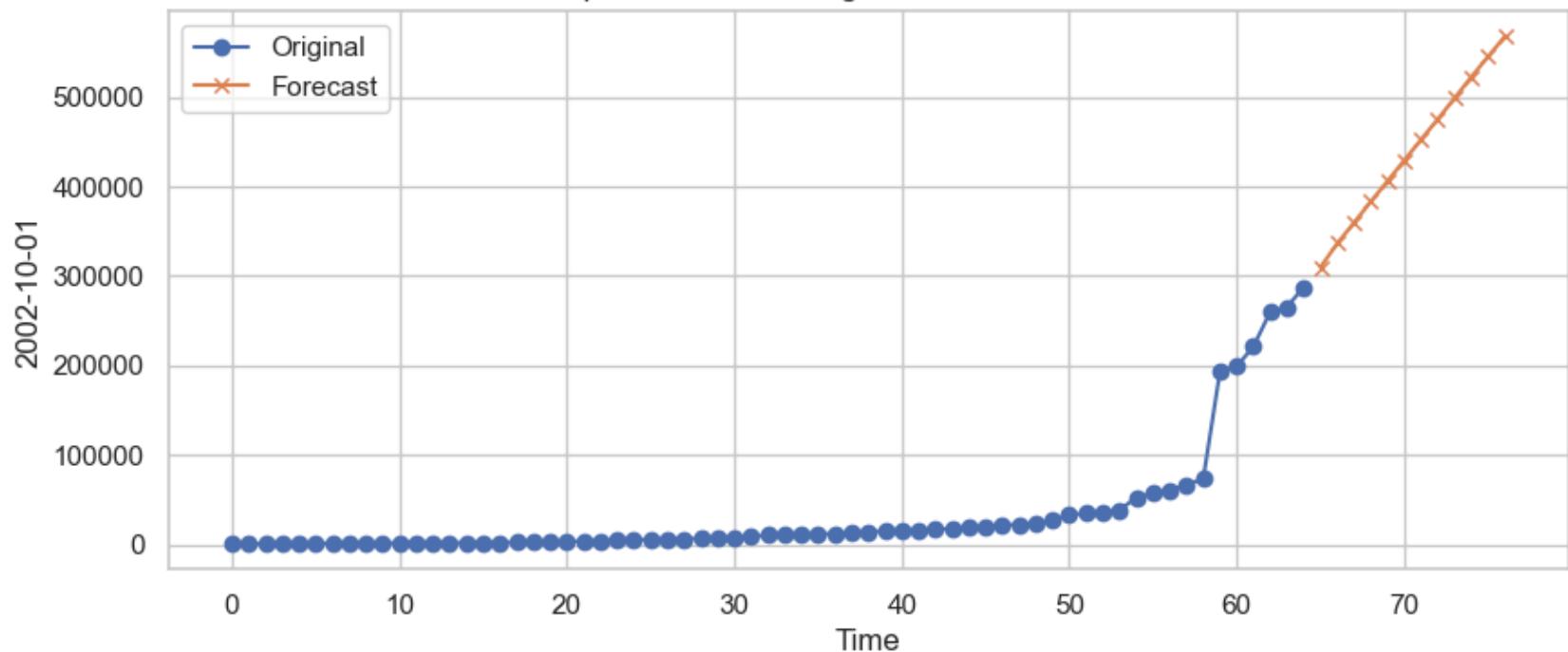
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2002-09-01



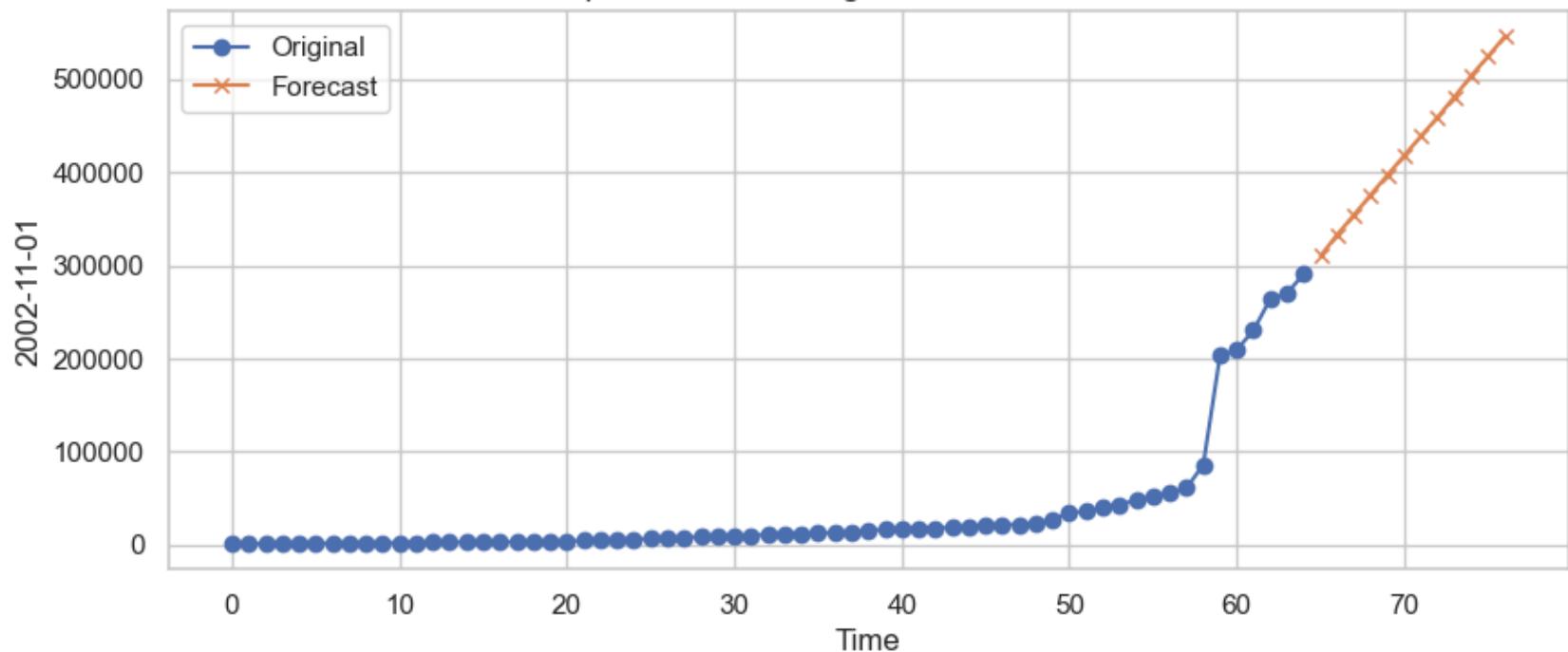
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2002-10-01



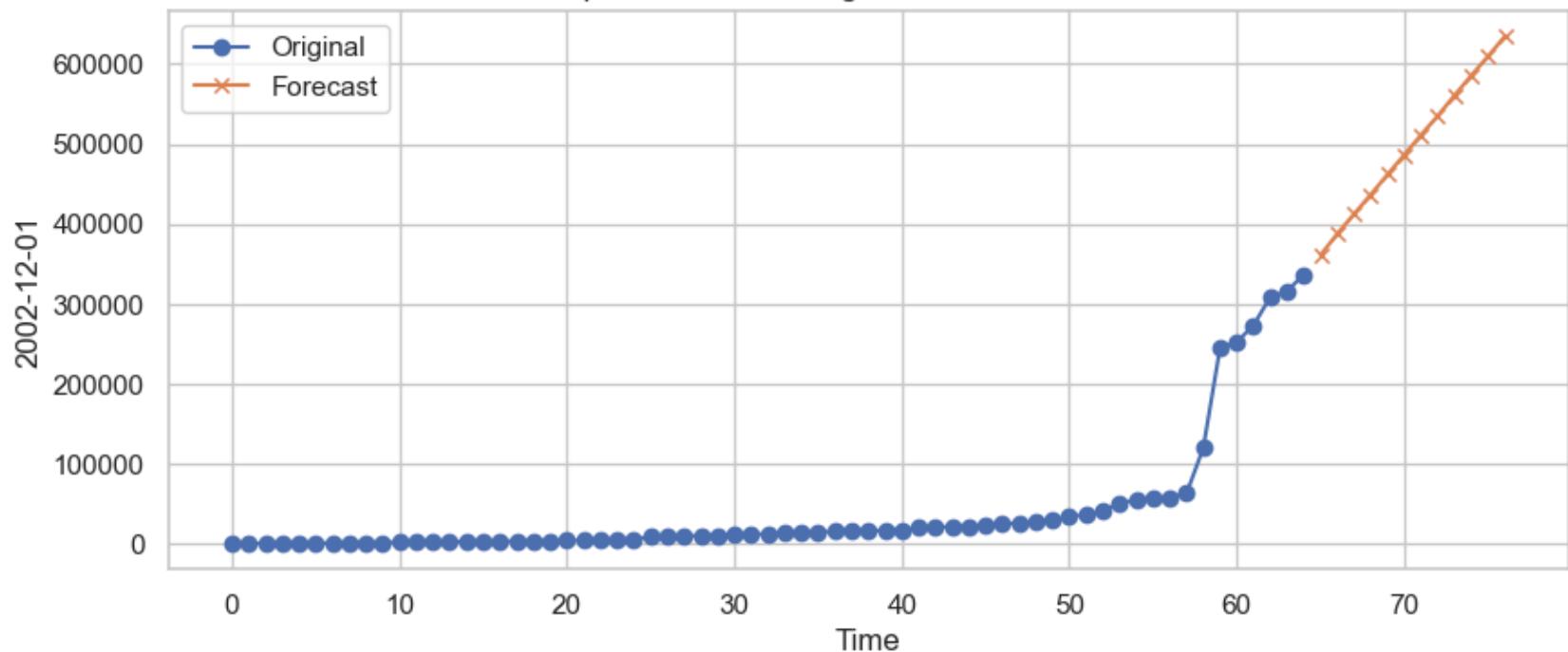
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2002-11-01



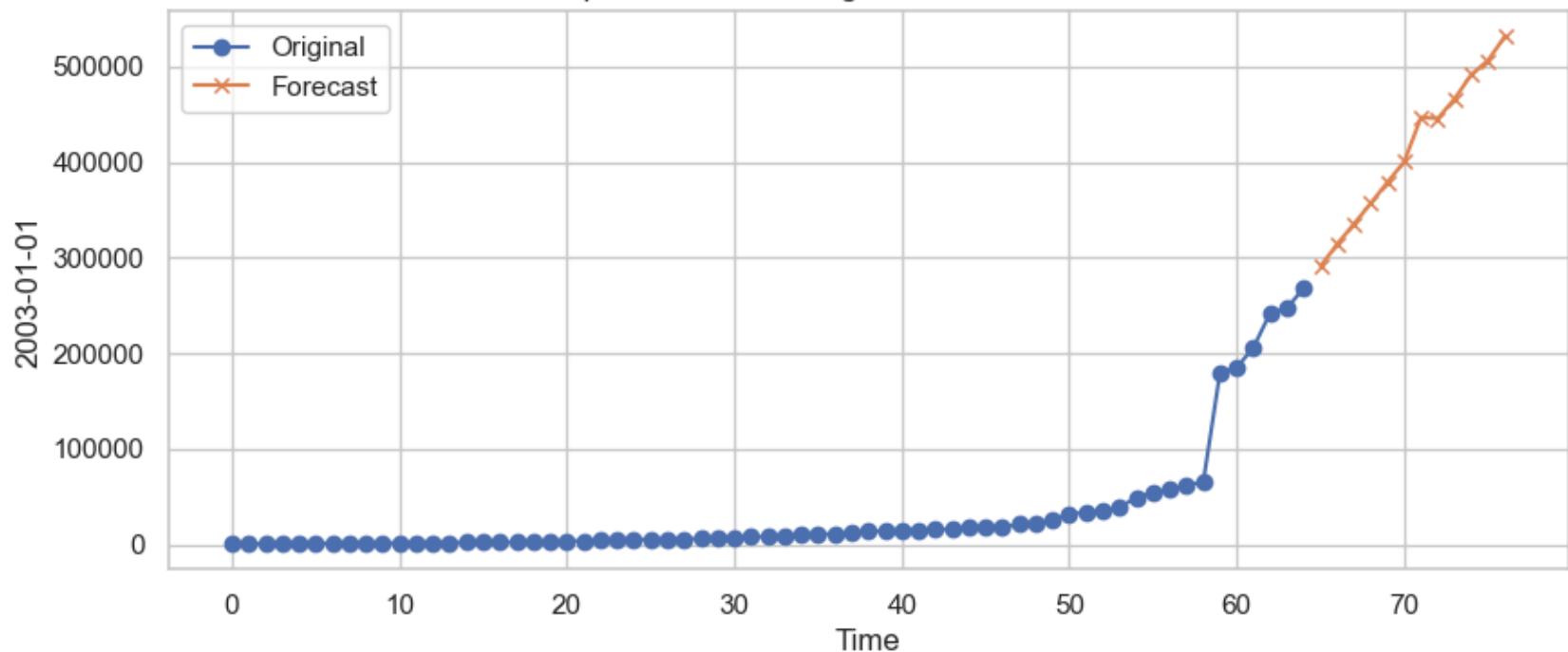
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2002-12-01



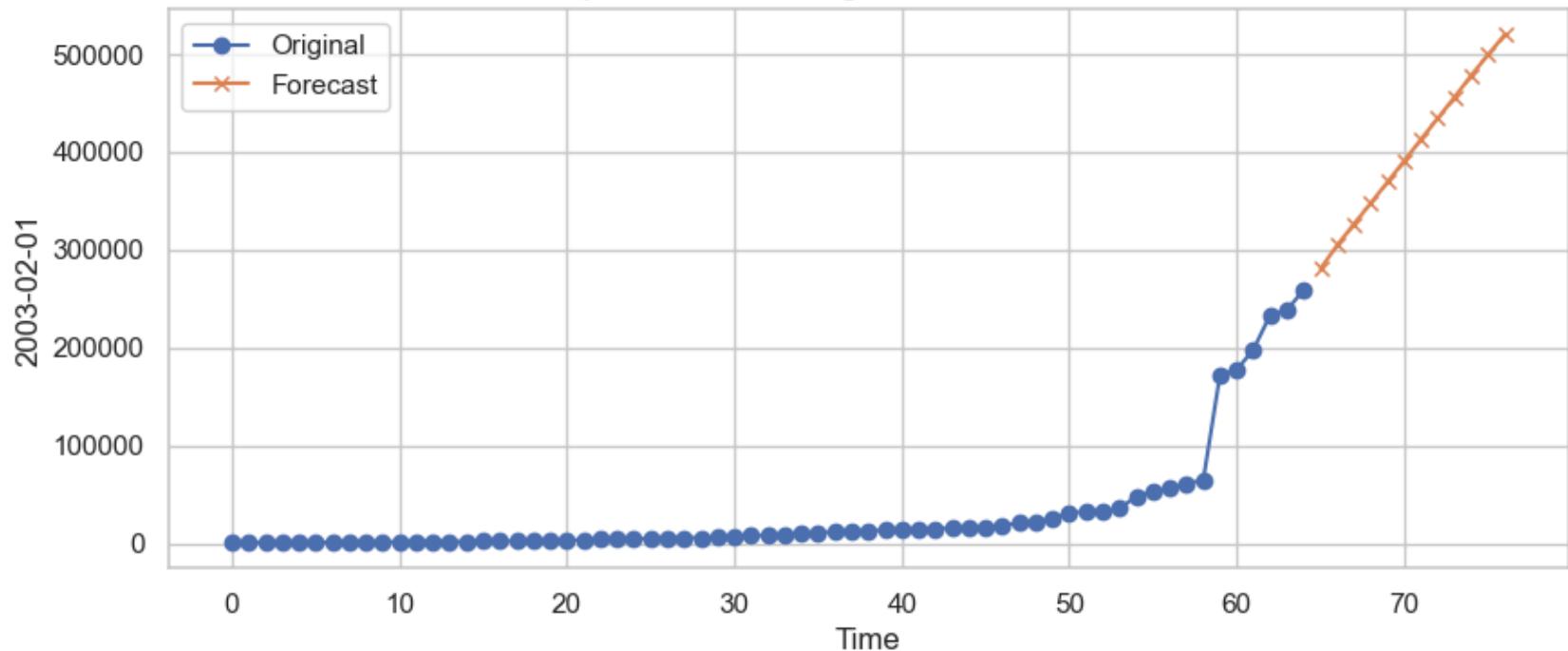
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2003-01-01



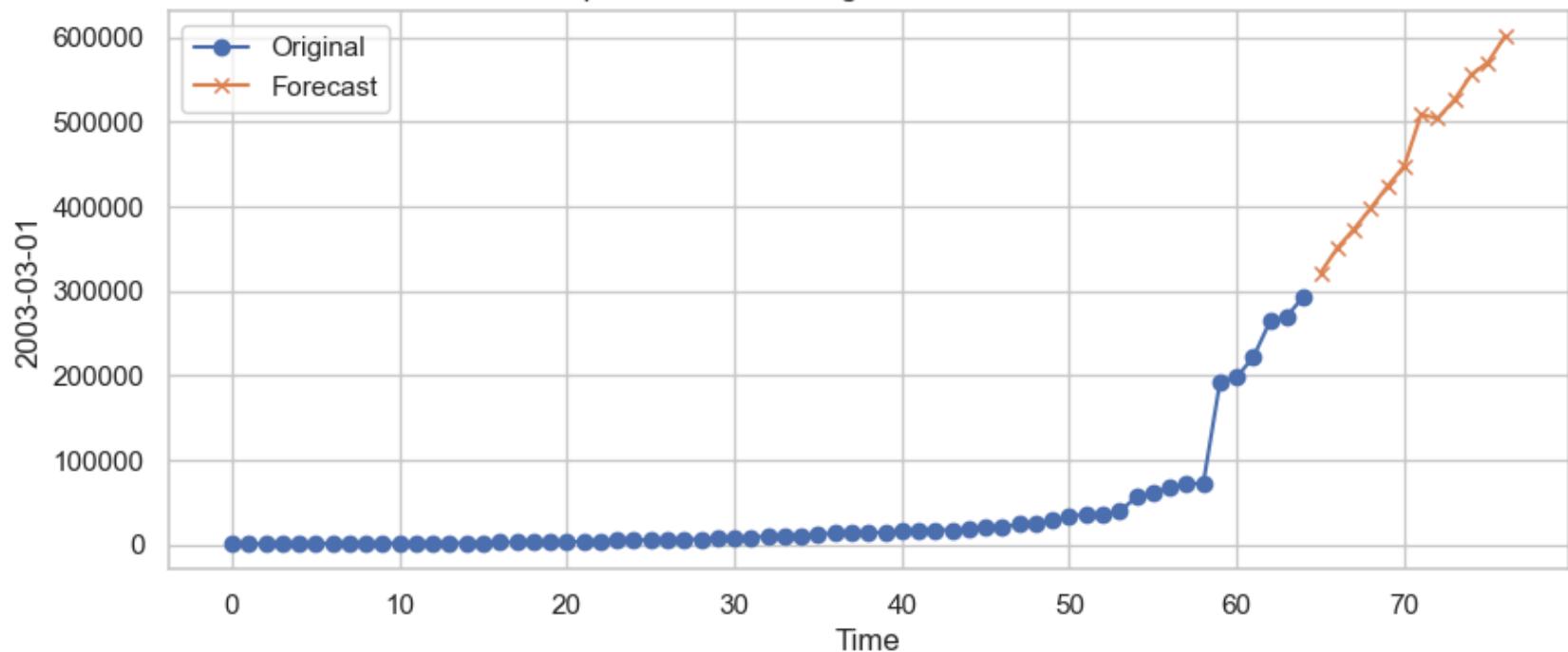
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2003-02-01



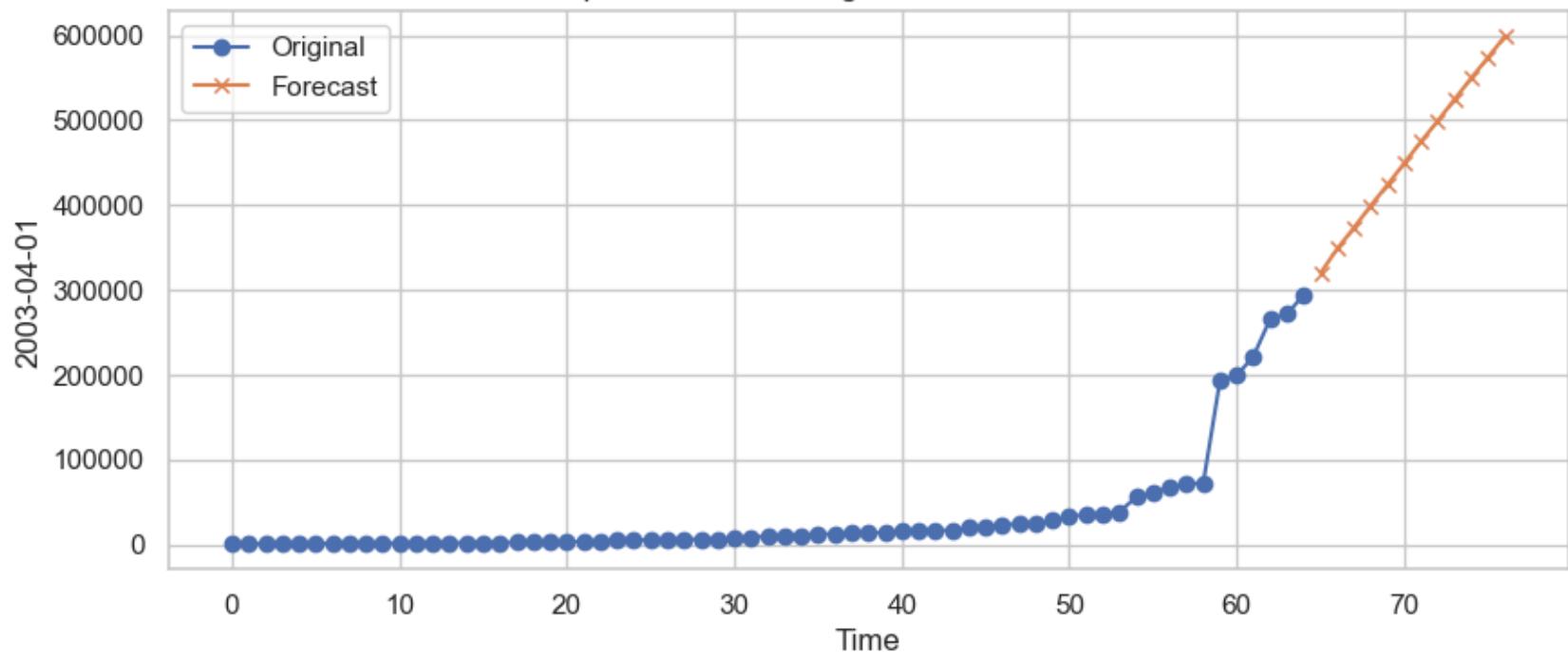
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2003-03-01



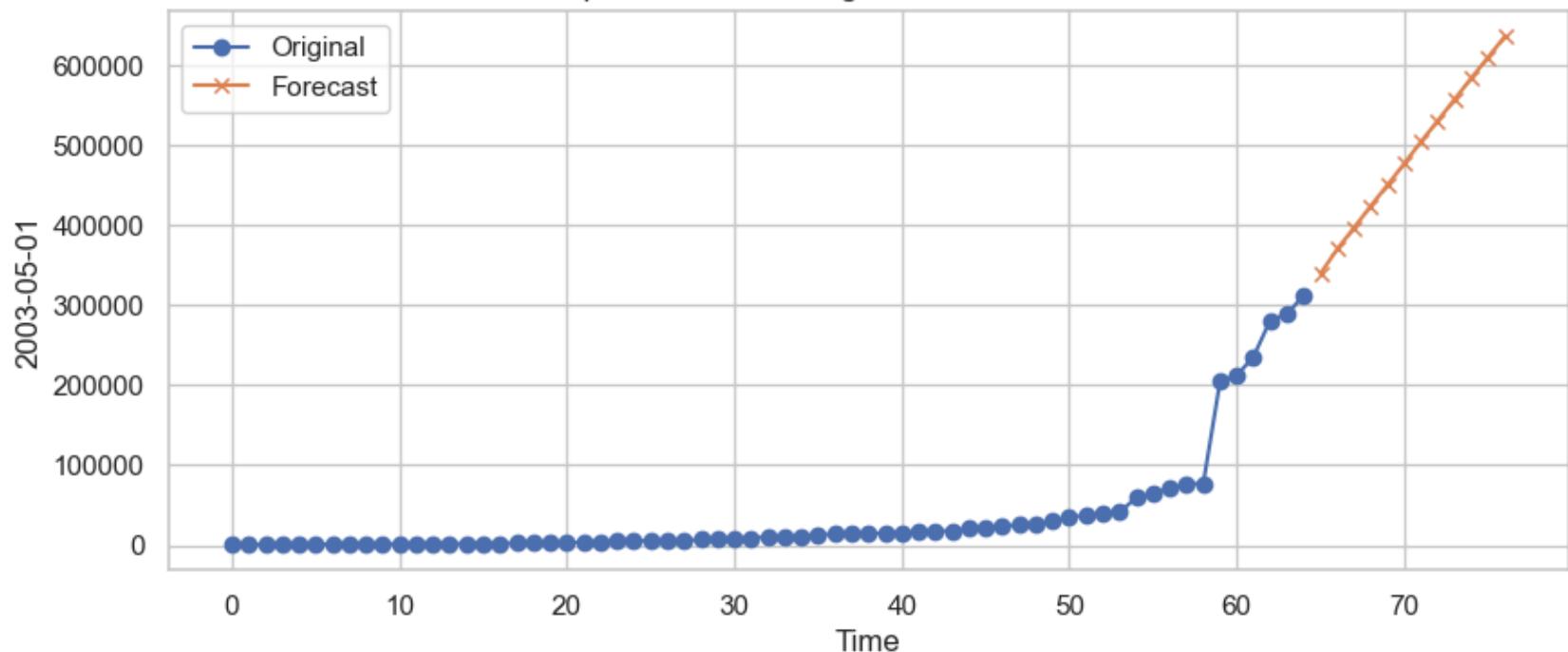
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2003-04-01



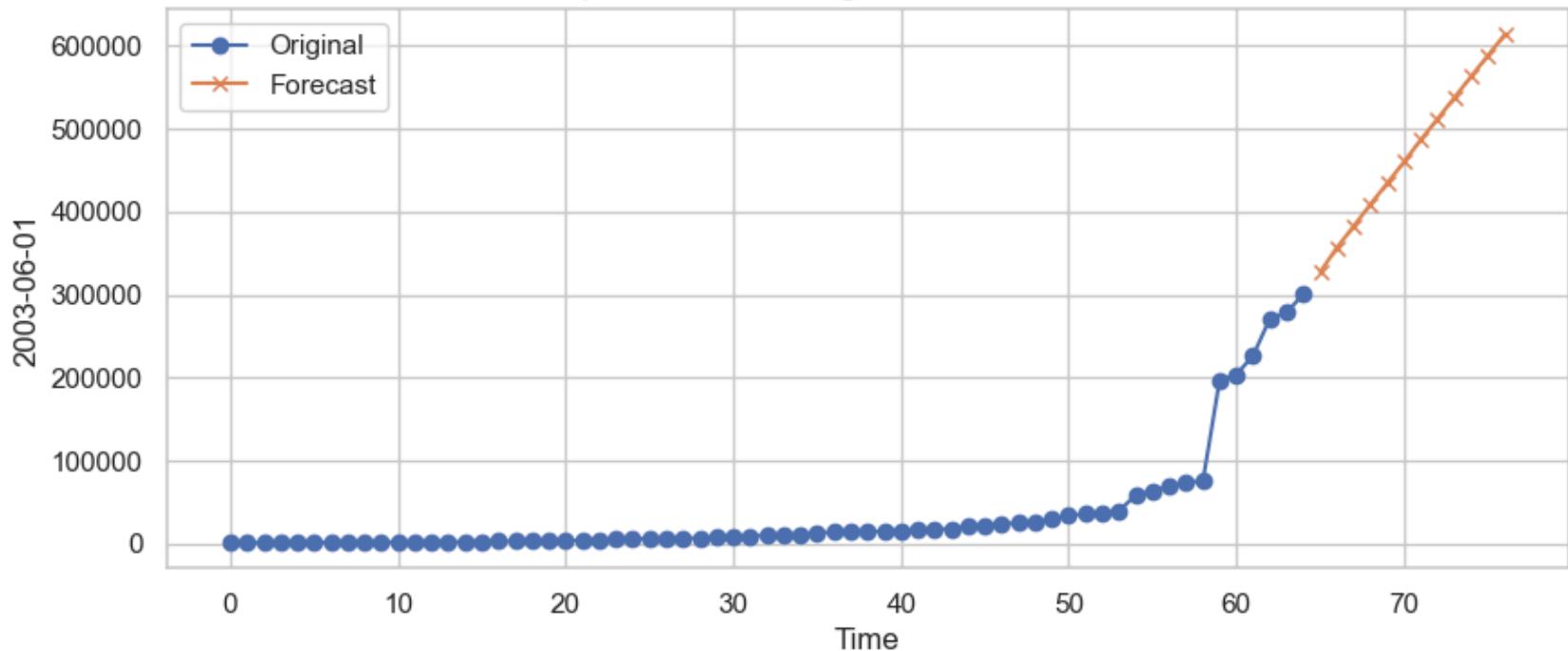
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2003-05-01



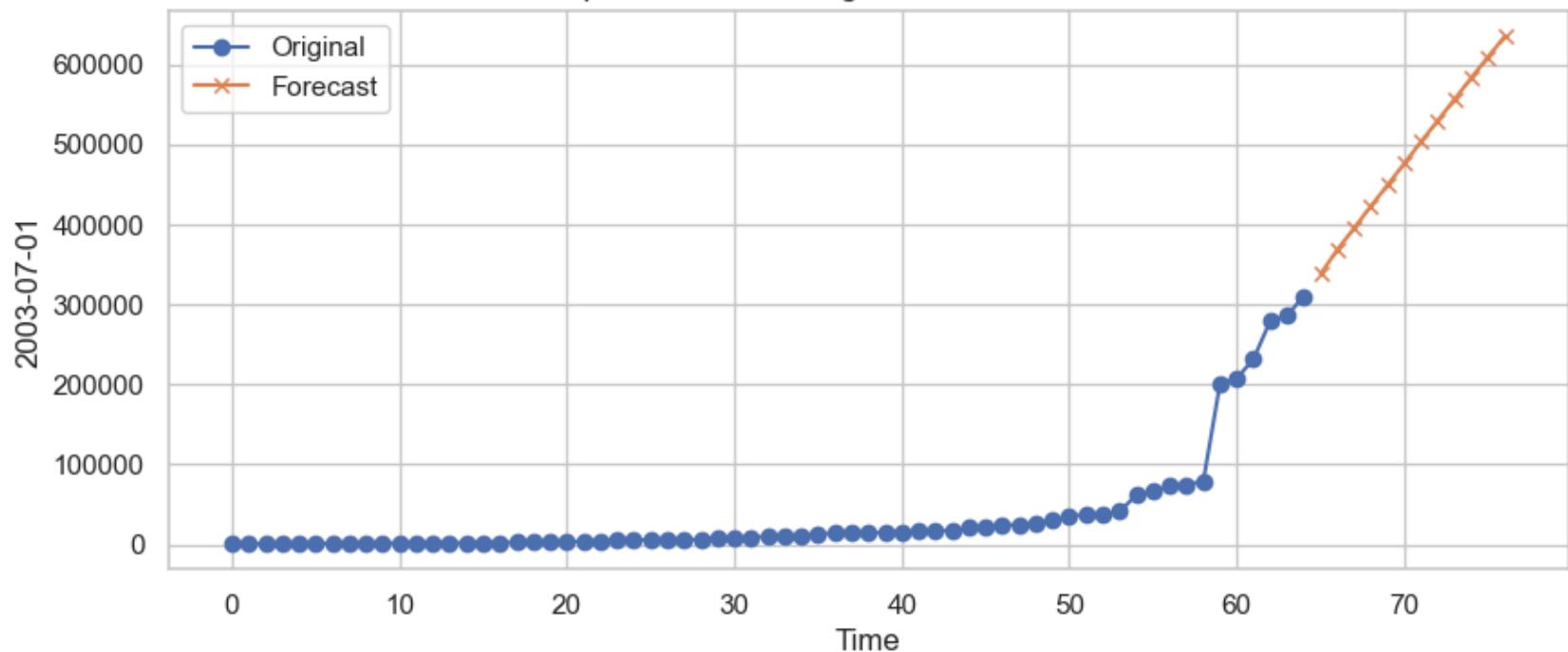
```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2003-06-01



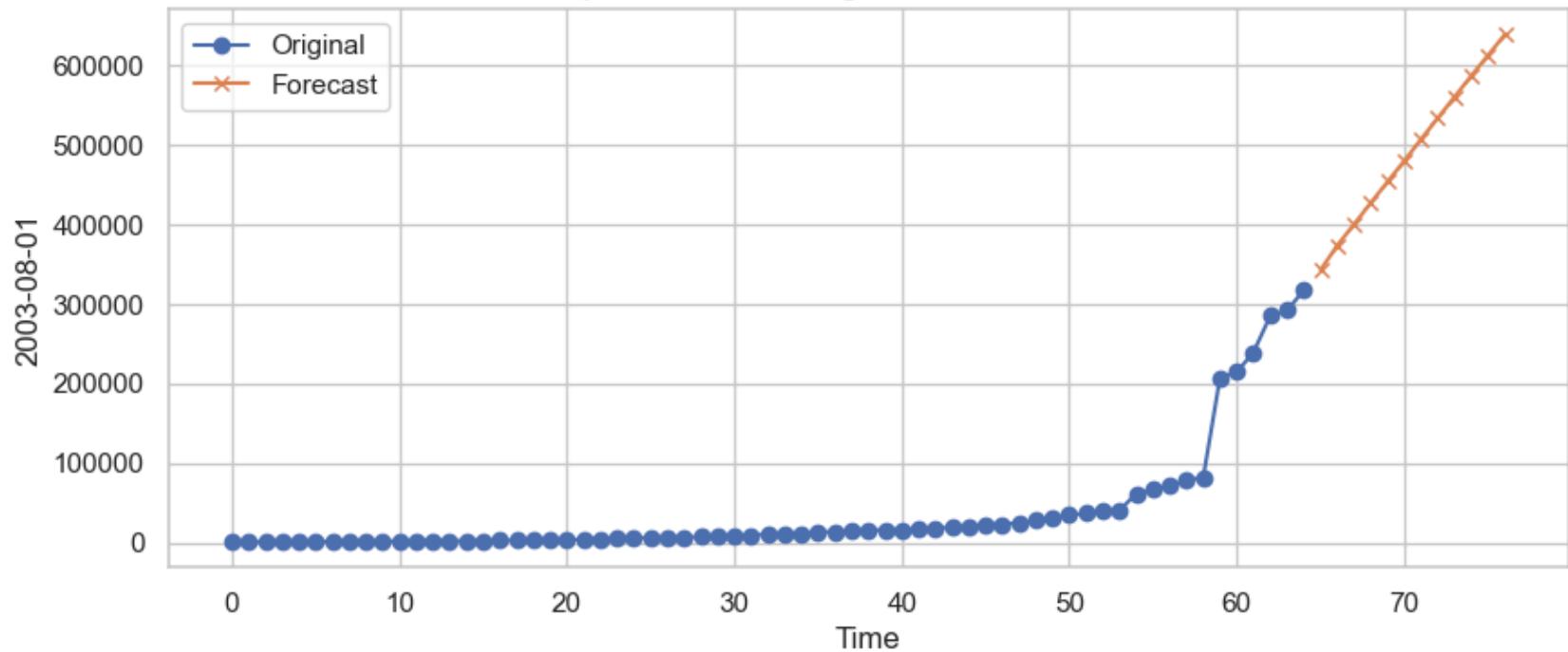
```
C:\Users\ADMIN\anaconda3\Lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2003-07-01



```
C:\Users\ADMIN\anaconda3\lib\site-packages\statsmodels\tsa\holtwinters\model.py:918: ConvergenceWarning: Optimization failed to converge. Check mle_retvals.  
warnings.warn(
```

Exponential Smoothing Forecast for 2003-08-01



In [110]:

```
# Display the forecasted values for each series
for series_name, forecast_values in forecasts.items():
    print(f"Last forecast value for {series_name}: {forecast_values.iloc[-1]}\n")
```

Last forecast value for 1992-01-01: 273698.12249524274
Last forecast value for 1992-02-01: 276219.6206854252
Last forecast value for 1992-03-01: 305547.50501670036
Last forecast value for 1992-04-01: 313575.44077605003
Last forecast value for 1992-05-01: 319254.8485850678
Last forecast value for 1992-06-01: 325142.35116741335
Last forecast value for 1992-07-01: 326676.67772844783
Last forecast value for 1992-08-01: 319410.8674353424
Last forecast value for 1992-09-01: 309362.2584738358
Last forecast value for 1992-10-01: 326073.83624780393
Last forecast value for 1992-11-01: 320467.1960005789
Last forecast value for 1992-12-01: 383787.8945675336
Last forecast value for 1993-01-01: 287559.84680253116
Last forecast value for 1993-02-01: 281235.07360351324
Last forecast value for 1993-03-01: 329082.9404834636
Last forecast value for 1993-04-01: 341041.19046670187
Last forecast value for 1993-05-01: 351929.52368284034
Last forecast value for 1993-06-01: 356941.1408940108
Last forecast value for 1993-07-01: 355172.6818465982
Last forecast value for 1993-08-01: 353135.14758469135
Last forecast value for 1993-09-01: 341295.5734696398
Last forecast value for 1993-10-01: 351015.8169762881
Last forecast value for 1993-11-01: 351078.23011708964
Last forecast value for 1993-12-01: 413347.2606229037

Last forecast value for 1994-01-01: 309994.47765074775
Last forecast value for 1994-02-01: 320008.6039686968
Last forecast value for 1994-03-01: 379691.88839473
Last forecast value for 1994-04-01: 374612.2474175341
Last forecast value for 1994-05-01: 384062.22140066023
Last forecast value for 1994-06-01: 396333.9081413575
Last forecast value for 1994-07-01: 380353.5475499046
Last forecast value for 1994-08-01: 398299.41768105497
Last forecast value for 1994-09-01: 381388.0806909341
Last forecast value for 1994-10-01: 383006.37627925945
Last forecast value for 1994-11-01: 380366.64982698974
Last forecast value for 1994-12-01: 443961.0626034539
Last forecast value for 1995-01-01: 338085.24862050853
Last forecast value for 1995-02-01: 338648.53105016355
Last forecast value for 1995-03-01: 403629.91696871247
Last forecast value for 1995-04-01: 391772.2461915472
Last forecast value for 1995-05-01: 422029.96919612243
Last forecast value for 1995-06-01: 426455.42556328065
Last forecast value for 1995-07-01: 407023.59355536674
Last forecast value for 1995-08-01: 429372.25391724124
Last forecast value for 1995-09-01: 398175.4724586946
Last forecast value for 1995-10-01: 401392.3480046245
Last forecast value for 1995-11-01: 398404.2718355498
Last forecast value for 1995-12-01: 455383.55726463906

Last forecast value for 1996-01-01: 365225.76080531825
Last forecast value for 1996-02-01: 380729.3229227324
Last forecast value for 1996-03-01: 430569.64525387547
Last forecast value for 1996-04-01: 425300.98613232264
Last forecast value for 1996-05-01: 456229.13496317726
Last forecast value for 1996-06-01: 440841.03465083695
Last forecast value for 1996-07-01: 440280.12913265225
Last forecast value for 1996-08-01: 452141.36126196175
Last forecast value for 1996-09-01: 418185.8802802276
Last forecast value for 1996-10-01: 444831.63501897576
Last forecast value for 1996-11-01: 420202.4114302768
Last forecast value for 1996-12-01: 475793.8828130498
Last forecast value for 1997-01-01: 396038.0855597379
Last forecast value for 1997-02-01: 396741.17808631464
Last forecast value for 1997-03-01: 451944.1800811868
Last forecast value for 1997-04-01: 443578.4525414155
Last forecast value for 1997-05-01: 469743.66743603494
Last forecast value for 1997-06-01: 457504.3607694299
Last forecast value for 1997-07-01: 465500.40683742263
Last forecast value for 1997-08-01: 471978.63426889706
Last forecast value for 1997-09-01: 446748.53216024186
Last forecast value for 1997-10-01: 458024.14613832865
Last forecast value for 1997-11-01: 428421.6571432381
Last forecast value for 1997-12-01: 505735.6206585023

Last forecast value for 1998-01-01: 408886.2535185072
Last forecast value for 1998-02-01: 408832.76566781837
Last forecast value for 1998-03-01: 467949.9872737967
Last forecast value for 1998-04-01: 472145.3894421436
Last forecast value for 1998-05-01: 493357.18048051186
Last forecast value for 1998-06-01: 494172.65500549873
Last forecast value for 1998-07-01: 484665.9767209156
Last forecast value for 1998-08-01: 473833.55682250916
Last forecast value for 1998-09-01: 456972.23822803685
Last forecast value for 1998-10-01: 482743.56526566093
Last forecast value for 1998-11-01: 451298.93438244186
Last forecast value for 1998-12-01: 539323.2500966701
Last forecast value for 1999-01-01: 429210.56012750446
Last forecast value for 1999-02-01: 444003.8424199567
Last forecast value for 1999-03-01: 512047.32269676024
Last forecast value for 1999-04-01: 506148.3618803459
Last forecast value for 1999-05-01: 533586.0929015212
Last forecast value for 1999-06-01: 533077.2328531145
Last forecast value for 1999-07-01: 529189.8073948028
Last forecast value for 1999-08-01: 540388.2989822367
Last forecast value for 1999-09-01: 514856.3696475319
Last forecast value for 1999-10-01: 510315.3831090405
Last forecast value for 1999-11-01: 506373.865759773
Last forecast value for 1999-12-01: 587772.9602033688

Last forecast value for 2000-01-01: 474950.22988791944
Last forecast value for 2000-02-01: 512367.74987007905
Last forecast value for 2000-03-01: 572476.3111083598
Last forecast value for 2000-04-01: 536815.4113961849
Last forecast value for 2000-05-01: 578274.9963032675
Last forecast value for 2000-06-01: 574564.7734586622
Last forecast value for 2000-07-01: 550639.5662744241
Last forecast value for 2000-08-01: 577169.7820266299
Last forecast value for 2000-09-01: 544881.1428969501
Last forecast value for 2000-10-01: 535360.836276309
Last forecast value for 2000-11-01: 518206.05263324996
Last forecast value for 2000-12-01: 590781.6607883581
Last forecast value for 2001-01-01: 496668.5852444328
Last forecast value for 2001-02-01: 504580.8752874039
Last forecast value for 2001-03-01: 571778.5867521936
Last forecast value for 2001-04-01: 558632.8979402769
Last forecast value for 2001-05-01: 603430.1424073138
Last forecast value for 2001-06-01: 587684.1314726683
Last forecast value for 2001-07-01: 568599.9675193076
Last forecast value for 2001-08-01: 599388.3401850966
Last forecast value for 2001-09-01: 517206.4985768501
Last forecast value for 2001-10-01: 598619.9616881742
Last forecast value for 2001-11-01: 539028.4692069908
Last forecast value for 2001-12-01: 610133.2171144994

Last forecast value for 2002-01-01: 503707.1489345507
Last forecast value for 2002-02-01: 509269.8699426428
Last forecast value for 2002-03-01: 573200.2720059569
Last forecast value for 2002-04-01: 580298.2597273708
Last forecast value for 2002-05-01: 603481.2289552942
Last forecast value for 2002-06-01: 582351.7758787796
Last forecast value for 2002-07-01: 605353.5642652849
Last forecast value for 2002-08-01: 634166.2318349325
Last forecast value for 2002-09-01: 548882.5492340481
Last forecast value for 2002-10-01: 567343.6594238809
Last forecast value for 2002-11-01: 545859.8418194836
Last forecast value for 2002-12-01: 634315.7445656108
Last forecast value for 2003-01-01: 531429.9953093034
Last forecast value for 2003-02-01: 519658.9228820297
Last forecast value for 2003-03-01: 600827.3043108938
Last forecast value for 2003-04-01: 598550.8408831777
Last forecast value for 2003-05-01: 636069.4806288353
Last forecast value for 2003-06-01: 613635.1796163266
Last forecast value for 2003-07-01: 634531.7207243263
Last forecast value for 2003-08-01: 638875.8339975673
Last forecast value for 2003-09-01: 591232.168996958
Last forecast value for 2003-10-01: 591079.4261544104
Last forecast value for 2003-11-01: 566618.3299285182
Last forecast value for 2003-12-01: 673243.9050991967

Last forecast value for 2004-01-01: 545130.873175787
Last forecast value for 2004-02-01: 558623.9617658069
Last forecast value for 2004-03-01: 650781.5508739797
Last forecast value for 2004-04-01: 633251.0087621053
Last forecast value for 2004-05-01: 660530.4752502566
Last forecast value for 2004-06-01: 646365.8943302744
Last forecast value for 2004-07-01: 668491.1597476676
Last forecast value for 2004-08-01: 655422.7551545803
Last forecast value for 2004-09-01: 630799.5639985696
Last forecast value for 2004-10-01: 620932.7002810042
Last forecast value for 2004-11-01: 609978.7340268935
Last forecast value for 2004-12-01: 722153.6653423655
Last forecast value for 2005-01-01: 568285.3500404395
Last forecast value for 2005-02-01: 583232.5598111616
Last forecast value for 2005-03-01: 688638.0256032693
Last forecast value for 2005-04-01: 676159.3687422415
Last forecast value for 2005-05-01: 693345.589073387
Last forecast value for 2005-06-01: 724105.9729970314
Last forecast value for 2005-07-01: 740538.8894240558
Last forecast value for 2005-08-01: 715751.1068482591
Last forecast value for 2005-09-01: 650396.9476970428
Last forecast value for 2005-10-01: 638550.0772969832
Last forecast value for 2005-11-01: 646680.9170576647
Last forecast value for 2005-12-01: 756581.565933395

Last forecast value for 2006-01-01: 613611.0576411035
Last forecast value for 2006-02-01: 610869.5320822474
Last forecast value for 2006-03-01: 734115.2174405761
Last forecast value for 2006-04-01: 697470.9079117699
Last forecast value for 2006-05-01: 742272.8334357875
Last forecast value for 2006-06-01: 728728.6942546006
Last forecast value for 2006-07-01: 731227.7204659245
Last forecast value for 2006-08-01: 747114.0598246659
Last forecast value for 2006-09-01: 673773.5437510313
Last forecast value for 2006-10-01: 676063.4512736715
Last forecast value for 2006-11-01: 652657.6631483311
Last forecast value for 2006-12-01: 780903.2412510299
Last forecast value for 2007-01-01: 636865.2994604149
Last forecast value for 2007-02-01: 628890.398117393
Last forecast value for 2007-03-01: 752410.2927038776
Last forecast value for 2007-04-01: 709589.7522996004
Last forecast value for 2007-05-01: 778043.6146089914
Last forecast value for 2007-06-01: 745800.1748474155
Last forecast value for 2007-07-01: 739202.9438428261
Last forecast value for 2007-08-01: 769435.4938348517
Last forecast value for 2007-09-01: 690557.7491644671
Last forecast value for 2007-10-01: 717734.6217570727
Last forecast value for 2007-11-01: 716828.5346434483
Last forecast value for 2007-12-01: 797409.5447300837

Last forecast value for 2008-01-01: 652686.1746058367
Last forecast value for 2008-02-01: 668848.2676649217
Last forecast value for 2008-03-01: 728267.4416005545
Last forecast value for 2008-04-01: 720362.8256936194
Last forecast value for 2008-05-01: 760767.6798176916
Last forecast value for 2008-06-01: 723716.6096925124
Last forecast value for 2008-07-01: 734424.2571049846
Last forecast value for 2008-08-01: 730988.1151408528
Last forecast value for 2008-09-01: 648461.2405886013
Last forecast value for 2008-10-01: 644713.8030096889
Last forecast value for 2008-11-01: 610630.628641585
Last forecast value for 2008-12-01: 720205.3746811063
Last forecast value for 2009-01-01: 566874.4653242172
Last forecast value for 2009-02-01: 567382.9258953922
Last forecast value for 2009-03-01: 603678.0677442087
Last forecast value for 2009-04-01: 605726.979643093
Last forecast value for 2009-05-01: 639279.8308796014
Last forecast value for 2009-06-01: 630019.5924535133
Last forecast value for 2009-07-01: 636457.4028222184
Last forecast value for 2009-08-01: 669997.8176722567
Last forecast value for 2009-09-01: 596253.8920866308
Last forecast value for 2009-10-01: 622067.0539287614
Last forecast value for 2009-11-01: 620295.1683262916
Last forecast value for 2009-12-01: 746568.0580994575

Last forecast value for 2010-01-01: 578049.1625200529
Last forecast value for 2010-02-01: 572096.3452939641
Last forecast value for 2010-03-01: 684549.9895357123
Last forecast value for 2010-04-01: 676119.5977129888
Last forecast value for 2010-05-01: 693541.6142724562
Last forecast value for 2010-06-01: 671301.8255266157
Last forecast value for 2010-07-01: 687685.7924464123
Last forecast value for 2010-08-01: 687874.4363610058
Last forecast value for 2010-09-01: 656275.6874590305
Last forecast value for 2010-10-01: 649711.0175762363
Last forecast value for 2010-11-01: 670742.3802723031
Last forecast value for 2010-12-01: 775051.2672807656
Last forecast value for 2011-01-01: 614166.5990577111
Last forecast value for 2011-02-01: 641186.4090033981
Last forecast value for 2011-03-01: 753446.4804973851
Last forecast value for 2011-04-01: 725407.063908836
Last forecast value for 2011-05-01: 746076.1626628485
Last forecast value for 2011-06-01: 740242.3207542162
Last forecast value for 2011-07-01: 733694.6556664502
Last forecast value for 2011-08-01: 752350.9985280298
Last forecast value for 2011-09-01: 714488.8955469927
Last forecast value for 2011-10-01: 694970.756058994
Last forecast value for 2011-11-01: 718255.8695030495
Last forecast value for 2011-12-01: 822428.8673479214

Last forecast value for 2012-01-01: 676219.6042856288
Last forecast value for 2012-02-01: 709632.7393336282
Last forecast value for 2012-03-01: 808394.8916067039
Last forecast value for 2012-04-01: 745764.2541962791
Last forecast value for 2012-05-01: 794076.1559777235
Last forecast value for 2012-06-01: 759834.2735644724
Last forecast value for 2012-07-01: 751411.8259768898
Last forecast value for 2012-08-01: 814124.806243897
Last forecast value for 2012-09-01: 736909.7913308266
Last forecast value for 2012-10-01: 761519.9713029054
Last forecast value for 2012-11-01: 754697.3527781776
Last forecast value for 2012-12-01: 842144.3086002904
Last forecast value for 2013-01-01: 711175.1136588911
Last forecast value for 2013-02-01: 728368.9065845138
Last forecast value for 2013-03-01: 822500.6498405184
Last forecast value for 2013-04-01: 798341.2891208793
Last forecast value for 2013-05-01: 852992.4535809964
Last forecast value for 2013-06-01: 809877.0493450694
Last forecast value for 2013-07-01: 826735.6367950547
Last forecast value for 2013-08-01: 849168.3119091409
Last forecast value for 2013-09-01: 752089.7499004876
Last forecast value for 2013-10-01: 787455.026857568
Last forecast value for 2013-11-01: 802380.7548226309
Last forecast value for 2013-12-01: 871140.6147939321

Last forecast value for 2014-01-01: 726055.5054461852
Last forecast value for 2014-02-01: 741136.9712580125
Last forecast value for 2014-03-01: 859777.2780079965
Last forecast value for 2014-04-01: 854795.9215719278
Last forecast value for 2014-05-01: 910062.9996168491
Last forecast value for 2014-06-01: 857887.9429210448
Last forecast value for 2014-07-01: 880114.9202828459
Last forecast value for 2014-08-01: 892960.634599621
Last forecast value for 2014-09-01: 822877.1304364932
Last forecast value for 2014-10-01: 831886.7952838559
Last forecast value for 2014-11-01: 806274.20415613
Last forecast value for 2014-12-01: 946183.6649432455
Last forecast value for 2015-01-01: 759654.167540809
Last forecast value for 2015-02-01: 742804.1340010538
Last forecast value for 2015-03-01: 875788.8976111654
Last forecast value for 2015-04-01: 862944.1506120898
Last forecast value for 2015-05-01: 911092.9212157594
Last forecast value for 2015-06-01: 882115.4585910271
Last forecast value for 2015-07-01: 915352.8190604125
Last forecast value for 2015-08-01: 897344.2194539192
Last forecast value for 2015-09-01: 847900.8011582958
Last forecast value for 2015-10-01: 853246.9309167725
Last forecast value for 2015-11-01: 847876.0328380684
Last forecast value for 2015-12-01: 977959.7856685898

Last forecast value for 2016-01-01: 767142.8743864909
Last forecast value for 2016-02-01: 812154.7236399733
Last forecast value for 2016-03-01: 902602.9948347782
Last forecast value for 2016-04-01: 885275.2319988967
Last forecast value for 2016-05-01: 919671.7072103657
Last forecast value for 2016-06-01: 910820.5299970227
Last forecast value for 2016-07-01: 907202.3402162489
Last forecast value for 2016-08-01: 925727.6080247564
Last forecast value for 2016-09-01: 880312.4844247659
Last forecast value for 2016-10-01: 886139.5090148483
Last forecast value for 2016-11-01: 859054.1207049105
Last forecast value for 2016-12-01: 1021099.8196124717
Last forecast value for 2017-01-01: 807935.4221521814
Last forecast value for 2017-02-01: 820558.2035767832
Last forecast value for 2017-03-01: 948248.4265622041
Last forecast value for 2017-04-01: 912255.4149350943
Last forecast value for 2017-05-01: 985636.5547391432
Last forecast value for 2017-06-01: 945961.9493445491
Last forecast value for 2017-07-01: 933937.1460582818
Last forecast value for 2017-08-01: 962305.1528805492
Last forecast value for 2017-09-01: 922498.4428706672
Last forecast value for 2017-10-01: 931164.98272986
Last forecast value for 2017-11-01: 944850.2218598643
Last forecast value for 2017-12-01: 1053888.9478812623

Last forecast value for 2018-01-01: 846134.937602201
Last forecast value for 2018-02-01: 851951.1619544261
Last forecast value for 2018-03-01: 996420.5038350325
Last forecast value for 2018-04-01: 942272.7024476255
Last forecast value for 2018-05-01: 1036065.025069447
Last forecast value for 2018-06-01: 995099.196108166
Last forecast value for 2018-07-01: 989935.4207759688
Last forecast value for 2018-08-01: 1020836.1513120775
Last forecast value for 2018-09-01: 933862.7451913825
Last forecast value for 2018-10-01: 958188.5039775706
Last forecast value for 2018-11-01: 953731.6497127399
Last forecast value for 2018-12-01: 1051289.415320788
Last forecast value for 2019-01-01: 870835.5552892713
Last forecast value for 2019-02-01: 844972.1397769483
Last forecast value for 2019-03-01: 1005697.6547028574
Last forecast value for 2019-04-01: 991946.4861791861
Last forecast value for 2019-05-01: 1066311.7287743383
Last forecast value for 2019-06-01: 980145.3734416144
Last forecast value for 2019-07-01: 1036036.3348252006
Last forecast value for 2019-08-01: 1063944.5101434975
Last forecast value for 2019-09-01: 938441.5613234625
Last forecast value for 2019-10-01: 998736.9490804807
Last forecast value for 2019-11-01: 980678.8520407971
Last forecast value for 2019-12-01: 1070930.6403705026

Last forecast value for 2020-01-01: 883390.8756969847
Last forecast value for 2020-02-01: 919124.6575631207
Last forecast value for 2020-03-01: 812756.4028791086
Last forecast value for 2020-04-01: 692171.7747656669
Last forecast value for 2020-05-01: 918029.5040386787
Last forecast value for 2020-06-01: 1006766.0713428129
Last forecast value for 2020-07-01: 1037943.4525722781
Last forecast value for 2020-08-01: 1032687.0391637676
Last forecast value for 2020-09-01: 970563.2691926771
Last forecast value for 2020-10-01: 1009932.5168493176
Last forecast value for 2020-11-01: 936312.931073092
Last forecast value for 2020-12-01: 1047759.559993472
Last forecast value for 2021-01-01: 940521.9576981263
Last forecast value for 2021-02-01: 894140.1701592549
Last forecast value for 2021-03-01: 1223242.8652118419
Last forecast value for 2021-04-01: 1190191.9906781132
Last forecast value for 2021-05-01: 1223899.1415919624
Last forecast value for 2021-06-01: 1207608.085514578
Last forecast value for 2021-07-01: 1207541.0008195678
Last forecast value for 2021-08-01: 1188140.7857189158
Last forecast value for 2021-09-01: 1141713.119597401
Last forecast value for 2021-10-01: 1151975.4721792375
Last forecast value for 2021-11-01: 1174462.2265096891
Last forecast value for 2021-12-01: 1277899.0074502616

Last forecast value for 2022-01-01: 1062140.7190226752
Last forecast value for 2022-02-01: 1095974.1171343697
Last forecast value for 2022-03-01: 1285797.058131585
Last forecast value for 2022-04-01: 1285204.8031600823
Last forecast value for 2022-05-01: 1326207.6632974776
Last forecast value for 2022-06-01: 1310649.2371694292
Last forecast value for 2022-07-01: 1293926.2703410606
Last forecast value for 2022-08-01: 1317416.338439232
Last forecast value for 2022-09-01: 1202463.5255062492
Last forecast value for 2022-10-01: 1242149.4950021366
Last forecast value for 2022-11-01: 1239704.8834223289
Last forecast value for 2022-12-01: 1345934.9613410882
Last forecast value for 2023-01-01: 1147800.585940025
Last forecast value for 2023-02-01: 1110999.7095459874
Last forecast value for 2023-03-01: 1278744.9527143151
Last forecast value for 2023-04-01: 1235924.183174276
Last forecast value for 2023-05-01: 1328231.8295987004
Last forecast value for 2023-06-01: 1295129.0130816854
Last forecast value for 2023-07-01: 1275857.3723895159
Last forecast value for 2023-08-01: 1316120.4363008055
Last forecast value for 2023-09-01: 1250363.3815727106
Last forecast value for 2023-10-01: 1277117.8512866977
Last forecast value for 2023-11-01: 1290076.740214513
Last forecast value for 2023-12-01: 1397810.408796227

Last forecast value for 2024-01-01: 1164611.014874099
Last forecast value for 2024-02-01: 1186694.4456207561
Last forecast value for 2024-03-01: 1303553.0655072248
Last forecast value for 2024-04-01: 1289172.6180214935
Last forecast value for 2024-05-01: 1370442.6809597316
Last forecast value for 2024-06-01: 1285482.1144445122
Last forecast value for 2024-07-01: 1337137.9363552583
Last forecast value for 2024-08-01: 1351451.2726528547
Last forecast value for 2024-09-01: 1258361.1605538884
Last forecast value for 2024-10-01: 1334729.5440015045
Last forecast value for 2024-11-01: 1272377.8561664205
Last forecast value for 2024-12-01: 1456295.3216437474

What I understand from the required table:

The table is comparing the forecast performance of different time series (TS_1, TS_3, etc.) after applying the TS Exponential Smoothing model. For the total sum of squares metric: TS_3 is deemed the "best" because it has the lowest error (1.635e12), meaning its forecasts align more closely with the observed data. TS_1 is considered the "worst" with a much higher error (7.076e13).

*Total sum of squares:

Best Fit: The table shows that for this metric, the best fit is series _TS_3 with a value of 1.635e12. This suggests that _TS_3, when forecasted using TS Exponential Smoothing, has the lowest overall variability or error as measured by the residual sum of squares. Essentially, the forecasts for _TS_3 are relatively close to the actual data.

Worst Fit: The worst fit is series _TS_1 with a value of 7.076e13. This much larger value indicates that the model's predictions for _TS_1 deviate significantly from the observed values, leading to a poorer fit

Group work

We, Jamaica Vee Buduan, Jenelle Guerrero Martinez, Joshua Kevin Jonathan, Julie Pham, declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. We have not copied any part of this assignment, manually or electronically, from any other source including web sites, unless specified as references. We have not distributed our work to other students

1. Jamaica: Part I, II, III
2. Jenelle: Part IV (question 6,7)
3. Jonathan: Part IV
4. Julie: Part IV (question 8), combine everyone works

In []: