Retail Demand Forecasting

Hamza Salah
DSC680 – Applied Data Science
Professor Amirfarrokh Iranitalab
Fall 2025

Table of Contents

Introduction

Inventory management is a critical component of retail operations, forming the foundation for ensuring that the right products are available at the right time and in the right place. Effective inventory practices not only support sales and customer satisfaction but also directly impact on operational efficiency and profitability.

A persistent challenge in inventory management is addressing capacity constraints. Companies are increasingly holding higher levels of inventory due to difficulties in demand forecasting, which raises costs and increases the risk of both overstocking and stockouts (MHI & Deloitte, 2025, p. 24).

The primary objective of this study is to explore predictive analytics to improve demand forecasting to ensure efficiency across the supply chain, with particular focus on the capacity challenges faced by both stores and distribution centers. By addressing these issues, the research highlights the importance of thoughtful inventory planning in achieving streamlined operations and improved business performance.

Research Problem

The problem this study addresses is the inefficiency in retail inventory management caused by inaccurate tracking, poor demand forecasting, supply chain disruptions, and limited data visibility, which result in overstocking, stockouts, wasted space, and tied-up capital.

Purpose of the Study

The purpose of this study is to improve demand forecast to optimize inventory levels to reduce excess stock, minimize tied-up capital, and improve product availability across retail operations.

Significance of the Research

This research is significant because it addresses one of the costliest challenges in retail: excess inventory that ties up millions of dollars in capital. By utilizing predictive analytics to improve demand forecasting, the study can help retailers reduce waste, improve cash flow, and enhance product availability for customers. Ultimately, the findings have the potential to increase operational efficiency, strengthen competitiveness, and support long-term business growth

Origin

This project originated from the need for retailers to adapt to demand fluctuations.

Stakeholders

The primary stakeholders in this research are inventory analysts and managers at retail organizations across the nation who are directly responsible for overseeing stock levels, demand forecasting, and allocation decisions.

Scope

The study focuses on inventory management practices at retail store locations, examining strategies to optimize stock levels, reducing tied-up capital, and improving product availability.

The analysis is limited to stores included in the available dataset and may not fully represent distribution center operations or the broader retail network.

Limitations also include reliance on historical data, which may not capture sudden demand fluctuations or supply chain disruptions.

Data Collection and Structure

Data Sources

- Retail Sales Dataset – Kaggle: realistic, synthetic simulation of daily sales for a retail store over a 3-year period.

Variables Description

Defines each variable and its role in the analysis.

- **Date**: The date of the sales record, covering 3 years of daily sales.
- **Product ID**: A unique SKU identifier for each of the **15 products**.
- **Product Name**: The descriptive name of the product associated with its SKU.
- **Category**: The product's high-level category (**Apparel**, **Electronics**, **Toys**).
- **Promotion Flag**: A boolean flag indicating if the product was on promotion on that specific date.
- **Sales Volume**: The number of units sold for the product on a given day.

Data Cleaning and Preparation

1. **Corrected Data Types**: Converted the Date column to a datetime object.
2. **Feature Engineering**: Created new time-based features (Year, Month, Year_Month) from the Date column.
3. **Data Resampling and Aggregation**: Grouped and summed daily data to a monthly frequency.
4. **Chronological Data Split**: Divided the dataset into training and testing sets based on time.

Data Limitations

The study relies on the Retail Store Dataset from Kaggle. Limitations include potential inconsistencies or inaccuracies in the Kaggle datasets, and the artificial nature of the synthetic data, which may not fully reflect real-world inventory patterns. Additionally, the datasets may be biased toward product categories, limiting the generalizability of the results.

Research Questions

Primary Research Question

Can retail product demand be accurately forecasted using time series models like ARIMA and SARIMAX, and do external factors, such as promotions, improve forecasting accuracy?

Secondary/Sub-Questions

- Which time series model, including those with exogenous variables like promotions, is most effective for forecasting retail demand?
- What are the limitations and challenges of using time series forecasting for a dataset with multiple product categories and seasonal trends?

Methodology

This research was conducted to forecast retail product sales volume demand using a time series analysis approach. The methodology involved several key phases: data collection, cleaning, preprocessing, exploratory data analysis (EDA), model development, and evaluation. The analysis focused on monthly sales data to identify trends, seasonality, and the impact of promotional activities.

Tools and Techniques

> Python:

- **Pandas** and **NumPy** were used for data manipulation, cleaning, and preprocessing.
- **Matplotlib** and **Seaborn** were utilized for data visualization and exploratory analysis.
- **Scikit-learn** was used for data scaling.
- **Statsmodels** and **Pmdarima** were used for developing and fitting the time series forecasting models.

Data Analysis Methods

The primary analytical method was time series forecasting. The following models were employed:

- **ARIMA (Autoregressive Integrated Moving Average):** A baseline model to forecast future values based on past values, without considering seasonality.
- **Auto ARIMA:** An automated version of ARIMA that automatically selects the best parameters (p, d, q) for the model.
- **SARIMAX (Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors):** A more advanced model that extends ARIMA by incorporating both seasonal components and external variables. This model was used both with and without the Promotion_Flag as an exogenous variable.

Justification of Methods

Time series analysis was the most appropriate methodology because the core of the problem was to forecast future values based on historical, time-sequenced data.

- **SARIMAX:** This was the most suitable model because the sales data exhibited clear seasonality (e.g., peak sales in the fourth quarter), and there was a need to evaluate the impact of an external factor (promotions). The strong performance of SARIMAX with an exogenous variable proved its relevance.
- **ARIMA (and Auto ARIMA):** These models served as essential baselines. By comparing the performance of the seasonal SARIMAX models to the non-seasonal ARIMA, it was possible to quantitatively demonstrate the critical importance of accounting for seasonality in this specific dataset.

Data Analysis

Exploratory Data Analysis (EDA)

The dataset contained 16,440 rows and 6 columns, with no missing values. The summary statistics showed that daily sales volume ranged from 1 to 114 units. A histogram of the Sales_Volume revealed a right-skewed distribution, indicating that most daily sales were at the lower end of the volume range.

A pivot table was used to analyze total sales volume by category and year. This revealed distinct trends: Toys sales showed steady growth, while Apparel and Electronics sales fluctuated year-over-year.

A more detailed pivot table by Product_ID further highlighted varied performance within each category, with some products experiencing growth while others declined.

The analysis also found a 40% correlation between Sales_Volume and Promotion_Flag, suggesting a positive relationship between promotions and sales.

Techniques

The primary technique was time series analysis. Data was transformed by aggregating daily sales to a monthly frequency. This allowed for the identification of trends and seasonality.

- **Correlation Analysis:** A correlation matrix was used to quantify the relationship between Sales_Volume and Promotion_Flag.
- **Pivot Tables:** These were used to summarize sales data by category and product, providing a clear, tabular view of performance over time.

Category / Year	2022	2023	2024
Apparel	24993	25511	24828
Electronics	24783	25099	24468
Toys	26298	26528	27002

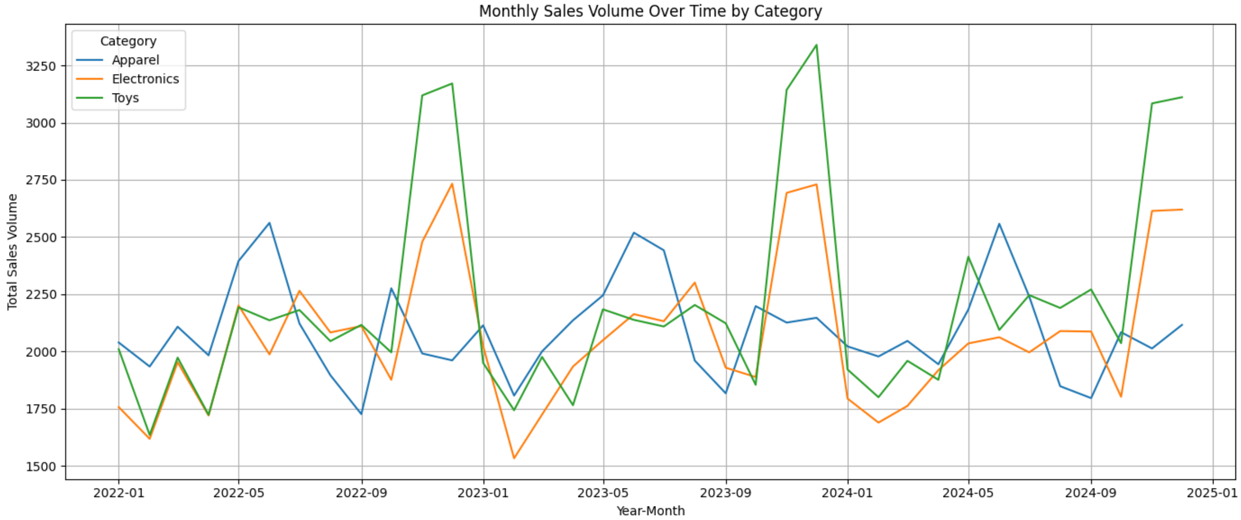
From the pivot table we can see that:

- Apparel sales volume was 24,993 in 2022, increased to 25,511 in 2023, and then decreased to 24,828 in 2024.
- Electronics sales volume was 24,783 in 2022, increased to 25,099 in 2023, and then decreased to 24,468 in 2024.
- Toys sales volume steadily increased from 26,298 in 2022 to 26,528 in 2023 and reached 27,002 in 2024.

Visualizations

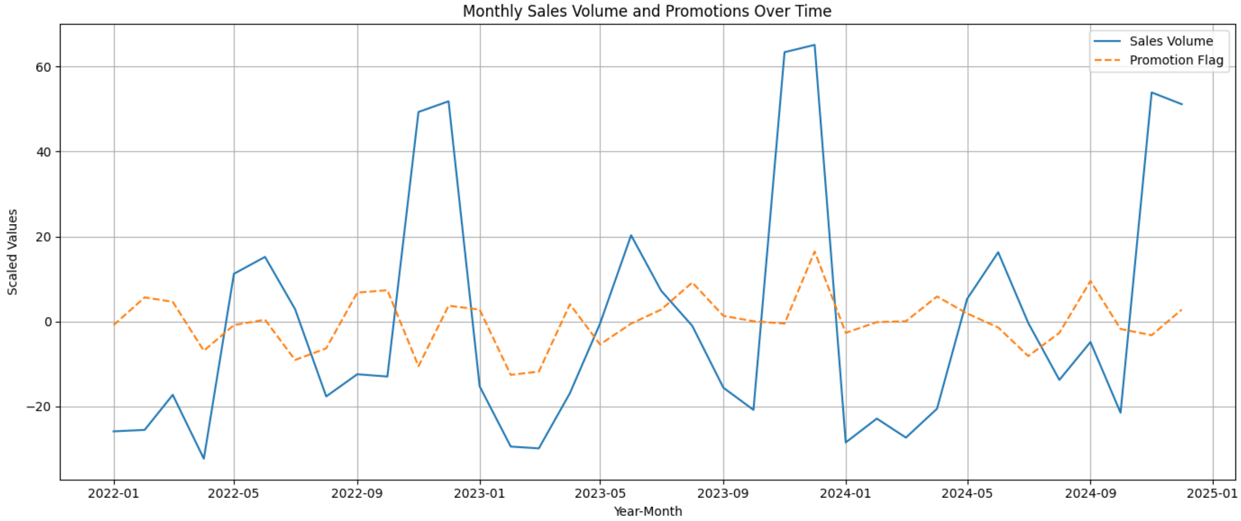
Visualizations were crucial for understanding the data and communicating findings. Analyzing sales data with a series of line plots revealed significant seasonal trends across different product categories.

A main plot of monthly sales by category showed that Electronics and Toys experienced a sales peak in the fourth quarter, coinciding with the holiday season, while Apparel sales were highest in the spring and summer.



To gain more specific insights, a second set of plots was created for individual products within their categories (See Appendix A: Graph 1-3). This deeper analysis confirmed the varied trends, for example, a winter jacket's sales peaked in the colder months, whereas a water blaster's sales soared in the summer, underscoring the necessity of product specific analysis.

A final plot combined and scaled the Sales_Volume and Promotion_Flag over time to provide a comprehensive view of how sales were influenced by promotions.



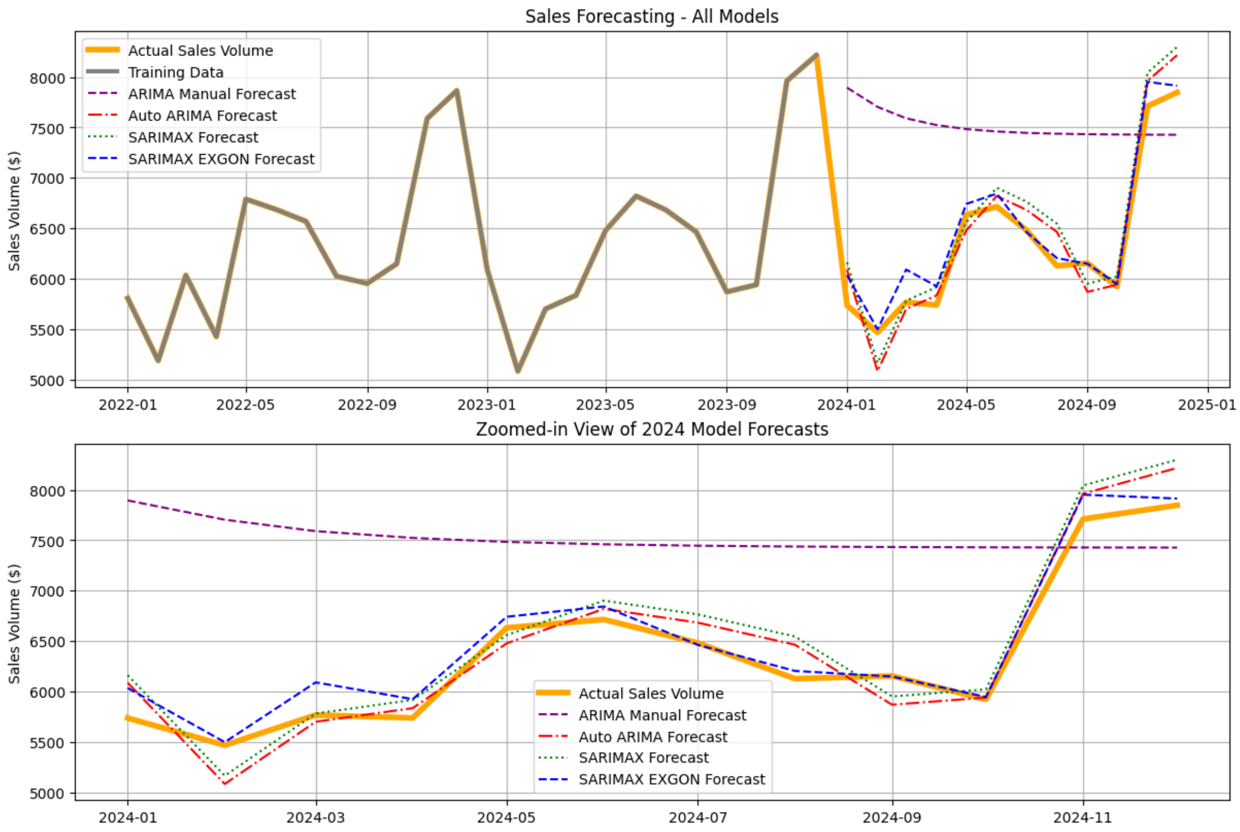
Next, I utilized time series modeling to forecast 2024's sales volume. Based on the graph below, the SARIMAX models are clearly the most accurate. Both the SARIMAX Forecast and SARIMAX EXGON Forecast lines closely track the actual sales data, successfully capturing the sharp seasonal fluctuations and overall trend. This strong performance indicates that these models correctly identified the seasonal and external factors influencing sales.

In contrast, the Auto ARIMA Forecast performs reasonably well but shows slightly larger deviations from the actual values. The least effective model is the ARIMA Manual Forecast, which fails to account for seasonality, resulting in a flat projection and significant errors throughout the forecast period.

The visual evidence strongly supports the conclusion that the inclusion of seasonal and exogenous variables is critical for accurate forecasting with this dataset.

The results of the forecast and evaluation metrics can be viewed in Appendix B Table (2-3). Based on the evaluation metrics, the SARIMAX_EXGON model is the most effective. It consistently outperformed all other models across every metric, most notably achieving the lowest RMSE of 165.10, a significant improvement over the standard SARIMAX model's RMSE of 283.64.

This strong performance demonstrates that incorporating an exogenous variable (Promotion_Flag) added substantial predictive power to the model, allowing it to make more accurate forecasts. The table's data clearly supports the conclusion that external factors played a crucial role in predicting the sales volume.



Key Findings

Summary of Results

- **Seasonality is a Critical Factor:** Models that did not account for seasonality, like the basic ARIMA, failed to produce accurate forecasts.
- **SARIMAX is the Superior Model:** The SARIMAX model significantly outperformed all other models, including the standard Auto ARIMA.
- **Promotions as a Key Predictor:** The SARIMAX with an exogenous variable (Promotion_Flag) was the most accurate model, achieving the lowest RMSE of 165.10. This indicates that incorporating external factors is crucial for improving predictive power.

Interpretation of Findings

The findings confirm the main research question: retail demand can be accurately forecasted using time series models, and external factors greatly improve accuracy. The poor performance of the manual ARIMA model highlights a key insight, that for this dataset, seasonality is a more influential factor than autoregressive or moving average components alone.

The superior performance of SARIMAX with the promotion variable demonstrates that promotional activity is a significant driver of sales volume. This result is not just a statistical finding but a critical business insight, suggesting that future inventory and marketing strategies should be closely aligned.

Recommendations

The analysis provides a strong foundation for a data-driven approach to retail management. By leveraging the forecasting capabilities of the SARIMAX model, stakeholders can significantly improve operational efficiency and product availability.

Practical Implications

The project's key finding: that promotions are a critical driver of sales and can be used to improve forecast accuracy has direct, real-world applications. Retail managers can use these insights to:

- **Optimize Inventory Management**: By using the SARIMAX with exogenous variables, inventory teams can generate more accurate demand forecasts. This allows them to order the right amount of stock, minimizing both **stockouts** and **overstocking**, thereby reducing holding costs.
- **Enhance Promotional Planning**: Marketing teams can use the model to simulate the impact of future promotions. By inputting planned promotional activities, they can get a more precise sales forecast, helping them to better plan marketing campaigns and allocate resources.
- **Improve Product-Specific Strategies**: The analysis showed that different product categories have unique seasonal trends. Retailers should abandon a one-size-fits-all approach and instead develop tailored strategies for each category. For example, they should launch promotions for summer apparel in late spring and for toys and electronics in late fall.

Suggested Actions or Improvements

- **Expand the Dataset**: To further improve model accuracy, include more exogenous variables beyond promotions. Potential variables could include local economic indicators, competitor activity, weather data, or holidays.
- **Develop a Hierarchical Forecasting Model**: Instead of a single model for all products, develop a hierarchical system that forecasts at the category, sub-category, and individual product levels. This would allow for more granular and precise predictions.

Risks and Mitigation Strategies

Identified Risks

There is a significant risk that the introduction of poor quality or missing data in the future could compromise the accuracy of sales forecasts. To mitigate this, more data should be gathered, and a robust data cleaning and validation process should be implemented. This includes setting up automated checks to identify flag inconsistencies or missing values before they are used in the forecasting models.

Ethical Considerations

Bias and Fairness

The models themselves do not create bias; they learn and reflect the patterns present in the data. The models' output should therefore be used as a tool to inform decisions, not as a replacement for human judgment.

Compliance with Ethical Standards

The project uses publicly available, non-sensitive synthetic data and focuses on a purely analytical goal without any intent to manipulate or unfairly target individuals. The findings are intended for the positive business application of improving efficiency and resource allocation.

Conclusion

Summary of Research

This research successfully demonstrated that retail demand can be accurately forecasted using time series analysis, with SARIMAX models proving to be superior by effectively capturing seasonality and the impact of promotions.

The project's primary contribution is its quantitative evidence that a simple, non-seasonal model is inadequate for this type of data, while the inclusion of exogenous variables like promotional activity significantly improves predictive accuracy. The findings provide a robust, data-driven framework that can be used to optimize inventory management and promotional strategies.

Future Research Directions

would like to extend this project by comparing the performance of the SARIMAX model with various machine learning models that are well-suited for time series forecasting. This comparison will help determine which approach offers the highest predictive accuracy and the most practical applicability.

My next step will be to optimize inventory levels based on the demand forecast to improve operational efficiency and reduce holding costs.

Questions & Answers from the Audience

1. **Will you consider ML in future analysis?**
    - Yes, I will consider using Machine Learning to forecast sales volume, but I will use more than one external variable like temperature and market data.
2. **What other factors can be considered as external variables?**
    - Temperature & Market Data.
3. **What are some of the recommendations for the company?**
    - I would ask that they include more promotions for products that performed low in unexpected months to improve sales.
4. **How will you further improve the model?**
    - By feeding more data so it can learn to generalize.
5. **What are some improvements in data collection that you suggest?**
    - Including more years of data collection and more columns like product pricing and temperature on that current date.
6. **How well will the model handle volatile changes in demand?**
    - If really unexpected then no model can predict that but what it can do is minimize that difference. As soon as the changes happen, feed it into the model so it can learn from it.
7. **Why did you choose time series over ML for this project.**
    - I did not choose rather it was the perfect first step for my project as ARIMA & SARIMAX are great series models and are used widely for these problems. Machine Learning is great to understand complex relationships with variables, and I will delve into that later.
8. **How will this improve inventory planning?**
    - When the company has a demand prediction they can better allocate resources and order more quantity to cover for increase

in demand and when there is a dip in demand more promotion can be utilized. This will help optimize inventory overall.

9. **Is the code ready for production?**

    - No, this is a project that can be used to understand data. For product more work needs to be done to the code, starting with improving reproducibility and functionality.

10. **Will retailers still have to carry buffer and safety stock with when forecasting demand? Why?**

    - Yes, because there will need to be safety stock to cover against forecast errors. Usually this is determined prior to ordering but I would suggest having buffer stock as well to cover anticipated increase in demand as well. They both work great together to make sure there is no stockouts.

References

Inventory Planning: What it is, importance, and challenges. (2025, May 16). Inbound Logistics. https://www.inboundlogistics.com/articles/inventory-planning-what-it-is-importance-and-challenges/

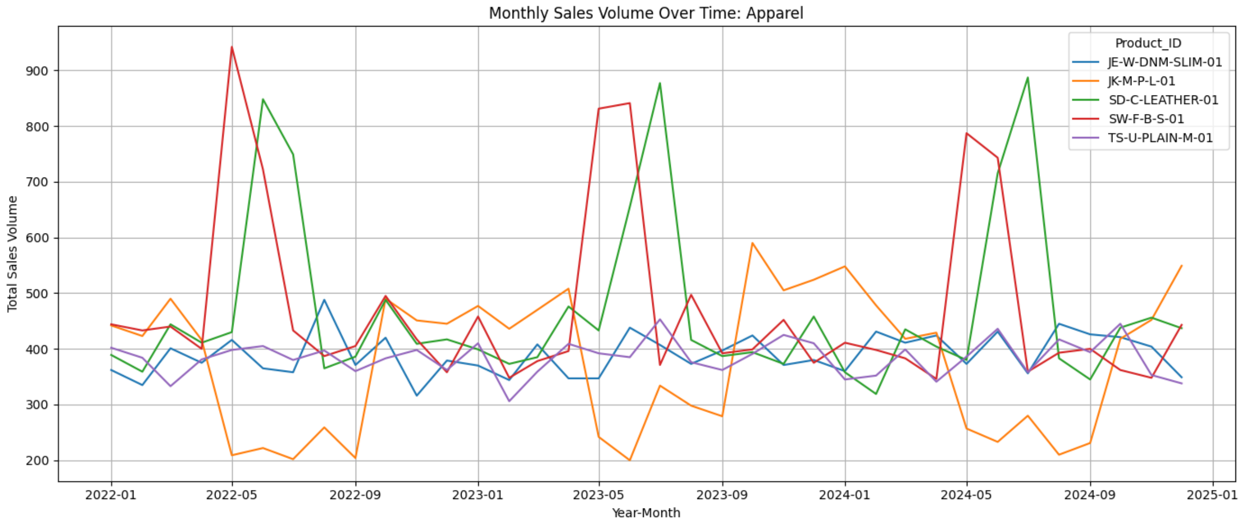
MHI & Deloitte. *The 2025 MHI Annual Industry Report: Transforming the Future of Supply Chains*. MHI & Deloitte, 2025. PDF. Available upon request from https://og.mhi.org/publications/report

Stevens, Courtenay. “Why Is Inventory Management Important?” *Business.org*, 2 Dec. 2019, www.business.org/finance/inventory-management/why-is-inventory-management-important/

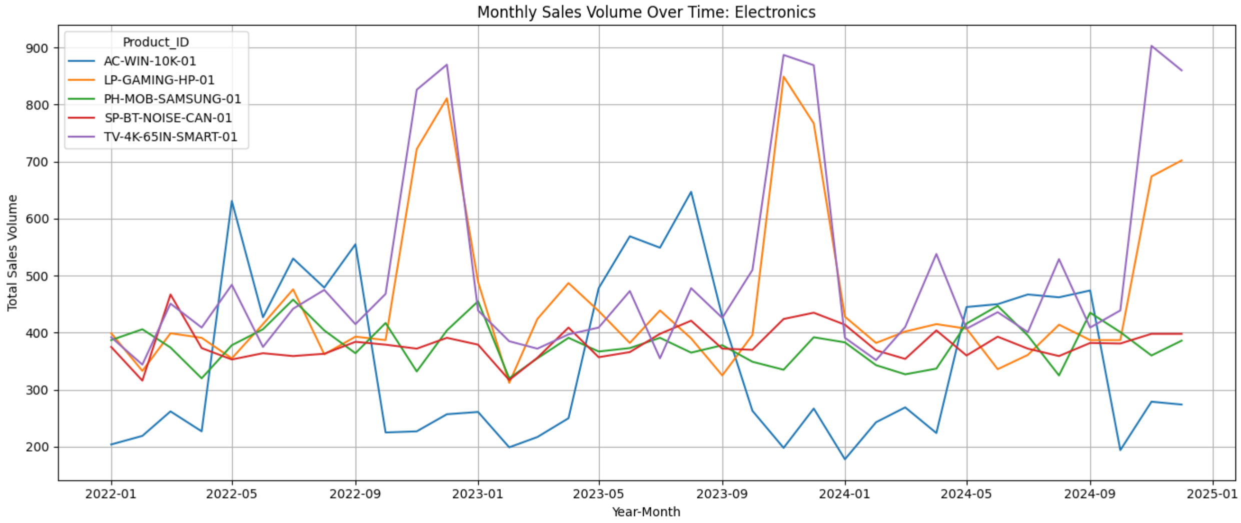
Subramaniam, PS. “The next Supply-Chain Challenge Isn’t a Shortage - It’s Inventory Glut.” *Harvard Business Review*, 29 Sept. 2023, www.hbr.org/2023/09/the-next-supply-chain-challenge-isnt-a-shortage-its-inventory-glut

Appendices

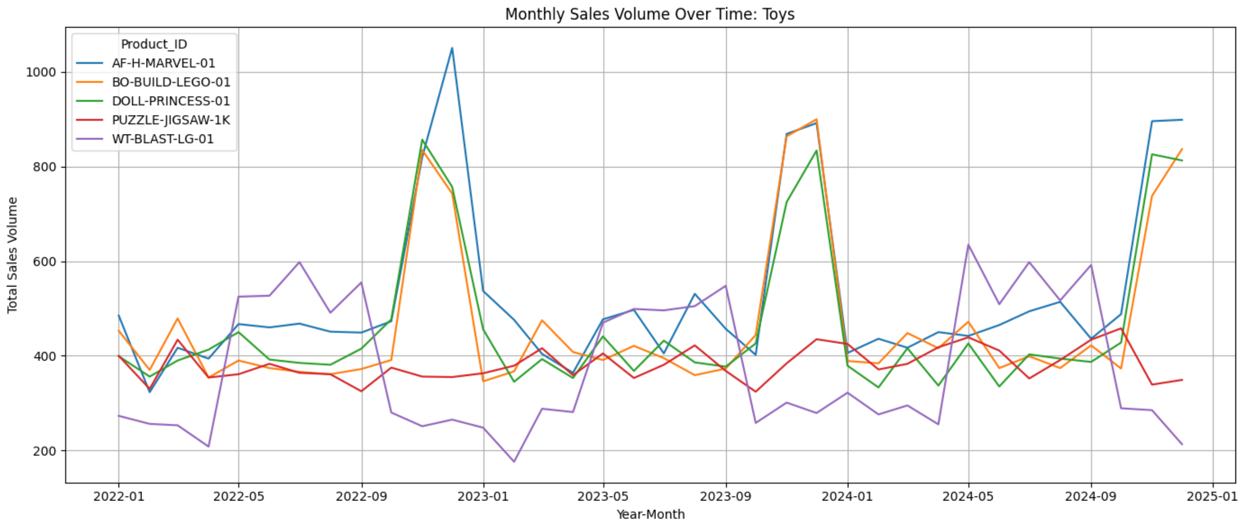
Appendix A: Graphs



Graph 1: Monthly Sales Volume Apparel



Graph 2: Monthly Sales Volume Electronics



Graph 3: Monthly Sales Volume Toys

Appendix B: Tables

Category	Product_ID	2022	2023	2024
Apparel	JE-W-DNM-SLIM-01	4586	4606	4831
	JK-M-P-L-01	4253	4863	4503
	SD-C-LEATHER-01	5695	5626	5557
	SW-F-B-S-01	5876	5738	5373
	TS-U-PLAIN-M-01	4583	4678	4564
Electronics	AC-WIN-10K-01	4243	4326	3959
	LP-GAMING-HP-01	5443	5698	5295
	PH-MOB-SAMSUNG-01	4650	4471	4555
	SP-BT-NOISE-CAN-01	4496	4604	4584
	TV-4K-65IN-SMART-01	5951	6000	6075
Toys	AF-H-MARVEL-01	6258	6311	6343
	BO-BUILD-LEGO-01	5488	5743	5626
	DOLL-PRINCESS-01	5672	5536	5477
	PUZZLE-JIGSAW-1K	4398	4589	4770
	WT-BLAST-LG-01	4482	4349	4786

Table 1: Pivot Table Categories and Product Sales Volume

Month	Actual Sales Volume	ARIMA Manual Forecast	Auto ARIMA Forecast	SARIMAX Forecast	SARIMAX EXGON Forecast	ARIMA Manual Forecast Error %	Auto ARIMA Forecast Error %	SARIMAX Forecast Error %	SARIMAX EXGON Forecast Error %
2024-01	5737	7894.568	6087	6161.737	6034.782	37.6	6.1	7.4	5.2
2024-02	5467	7703.513	5083	5164.85	5497.566	40.9	7	5.5	0.6
2024-03	5767	7590.305	5700	5781.667	6090.607	31.6	1.2	0.3	5.6
2024-04	5738	7523.224	5835	5916.672	5924.045	31.1	1.7	3.1	3.2
2024-05	6632	7483.476	6478	6559.672	6741.377	12.8	2.3	1.1	1.6
2024-06	6714	7459.924	6820	6901.672	6843.351	11.1	1.6	2.8	1.9
2024-07	6482	7445.968	6683	6764.672	6462.942	14.9	3.1	4.4	0.3
2024-08	6127	7437.699	6464	6545.672	6204.137	21.4	5.5	6.8	1.3
2024-09	6154	7432.799	5869	5950.672	6148.279	20.8	4.6	3.3	0.1
2024-10	5922	7429.895	5940	6021.672	5945.861	25.5	0.3	1.7	0.4
2024-11	7711	7428.175	7962	8043.672	7952.126	3.7	3.3	4.3	3.1
2024-12	7847	7427.156	8217	8298.672	7913.424	5.4	4.7	5.8	0.8

Table 2: Model Comparison Table

Metric	ARIMA Manual	Auto ARIMA	SARIMAX	SARIMAX_EXGON
MAE	1,280.34	218.33	247.41	125.84
RMSE	1,421.93	250.7	283.64	165.1

Table 3: Model Evaluation Metrics

Appendix C: Code

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from pmdarima import auto_arima
from sklearn.metrics import (
    mean_absolute_error,
    mean_absolute_percentage_error,
    root_mean_squared_error)

import warnings
warnings.filterwarnings("ignore")
```

Code 1: Import Libraries

```
data = r'retail_data.csv'
df = pd.read_csv(data)
```

Code 2: Load Data

```
df['Date'] = pd.to_datetime(df['Date'])
df['Year'] = df['Date'].dt.year.astype(str)
df['Month'] = df['Date'].dt.month.astype(str)
df['Year_Month'] = df['Year'].astype(str) + '-' + df['Month'].astype(str)
df['Year_Month'] = pd.to_datetime(df['Year_Month'], format='%Y-%m')
```

Code 3: Convert Date

```
by_month = df.groupby(['Category', 'Year_Month'])
sum_by_month = by_month['Sales_Volume'].sum().reset_index()
# Create the line plot
plt.figure(figsize=(14, 6)) # Optional: adjusts plot size

sns.lineplot(sum_by_month, x = sum_by_month['Year_Month'], y = sum_by_month['Sales_Volume'], hue='Category')

plt.title('Monthly Sales Volume Over Time by Category')
plt.xlabel('Year-Month')
plt.ylabel('Total Sales Volume')
plt.grid(True)
plt.tight_layout()
plt.show()
```

Code 4: Grouping by Category – Sales Over Time

```
categories = ["Apparel", 'Electronics', "Toys"]

for category in categories:
    df_cat = df[df['Category'] == category]
    by_month_category = df_cat.groupby(['Product_ID', 'Year_Month'])
    sum_by_month_category= by_month_category['Sales_Volume'].sum().reset_index()

    # Create the line plot
    plt.figure(figsize=(14, 6)) # Optional: adjusts plot size

    sns.lineplot(sum_by_month_category, x = sum_by_month_category['Year_Month'], y =
sum_by_month_category['Sales_Volume'], hue='Product_ID')

    plt.title(f'Monthly Sales Volume Over Time: {category}')
    plt.xlabel('Year-Month')
    plt.ylabel('Total Sales Volume')
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

Code 5: Product Level Charts

```
# instantiating an object from StandardScaler
scaler = StandardScaler()

df[['Sales_Volume_Scaled', 'Promotion_Flag_Scaled']] = scaler.fit_transform(df[['Sales_Volume', 'Promotion_Flag']])

by_month_scaled = df.groupby(['Category', 'Year_Month'])
sum_by_month_scaled = by_month_scaled['Sales_Volume_Scaled'].sum().reset_index()
sum_by_month_scaled2 = by_month_scaled['Promotion_Flag_Scaled'].sum().reset_index()
```

Code 6: Data Transformation

```
# Create the line plot
plt.figure(figsize=(14, 6))

sns.lineplot(sum_by_month_scaled, x='Year_Month', y='Sales_Volume_Scaled', label='Sales Volume', errorbar=None)
sns.lineplot(sum_by_month_scaled2, x='Year_Month', y='Promotion_Flag_Scaled', linestyle='--', label='Promotion Flag',
errorbar=None)

plt.title('Monthly Sales Volume and Promotions Over Time')
plt.xlabel('Year-Month')
plt.ylabel('Scaled Values')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

Code 7: Sales Chart vs. Promotion

```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

# Drop rows with invalid or missing dates
df = df.dropna(subset=['Date'])

# Set the index to the datetime for time series modeling
df.set_index("Date", inplace=True)
df.sort_index(inplace=True)

# Corrected code for daily sales
df_daily_sales = df.groupby(df.index)[['Sales_Volume', 'Promotion_Flag']].sum()

# Convert the grouped index back to datetime for resampling
df_daily_sales.index = pd.to_datetime(df_daily_sales.index)

# Resample the daily sums to a monthly frequency
df_monthly_sales = df_daily_sales.resample('MS').sum()

training_df = df_monthly_sales[df_monthly_sales.index < '2024-01-01']
testing_df = df_monthly_sales[df_monthly_sales.index >= '2024-01-01']

training_df.index = training_df.index.to_period('M')
testing_df.index = testing_df.index.to_period('M')

X_train = training_df['Sales_Volume']
exog_var_train = training_df['Promotion_Flag']

y_test = testing_df['Sales_Volume']
exog_var_test = testing_df['Promotion_Flag']
```

Code 8: Train Test Split

```
# --- 1. ARIMA model (seasonal) ---
model = ARIMA(X_train, order=(1,1,1))
model_fit = model.fit()

# Forecast on testing data using get_forecast()
arima_manual_forecast = model_fit.get_forecast(steps=len(y_test)).predicted_mean
arima_manual_forecast.index = y_test.index

# --- 2. auto_arima model (seasonal) ---
auto_arima_model = auto_arima(
    X_train,
    seasonal=True,
    m=12,
    D=1,  # Set the seasonal differencing parameter
    stepwise=True,
    suppress_warnings=True,
    error_action='ignore'
)
auto_arima_forecast = auto_arima_model.predict(n_periods=len(y_test))
auto_arima_forecast_series = pd.Series(auto_arima_forecast, index=y_test.index)

# --- 3. SARIMAX model (seasonal) ---
sarimax_model = SARIMAX(
    X_train,
    order=(1, 1, 1),
    seasonal_order=(1, 1, 1, 12),
    enforce_stationarity=False,
    enforce_invertibility=False
)
sarimax_results = sarimax_model.fit(disp=False)

# Forecast on testing data using get_forecast()
sarimax_forecast = sarimax_results.get_forecast(steps=len(y_test)).predicted_mean
sarimax_forecast.index = y_test.index

# --- 4. SARIMAX model (seasonal + exogenous variable) ---
sarimax_model_exgon = SARIMAX(
    X_train,
    exog=exog_var_train,
    order=(1, 1, 1),
    seasonal_order=(1, 1, 1, 12),
    enforce_stationarity=False,
    enforce_invertibility=False
)
sarimax_exgon_results = sarimax_model_exgon.fit(disp=False)

# Forecast on testing data using get_forecast() with exogenous variables
sarimax_exgon_forecast = sarimax_exgon_results.get_forecast(steps=len(y_test), exog=exog_var_test).predicted_mean
sarimax_exgon_forecast.index = y_test.index
```

Code 9: Time Series Forecast

```
print("Model Forecast Comparison")
# Create comparison DataFrame
comparison_df = pd.DataFrame({
    'Month': y_test.index,
    'Actual Sales Volume': y_test.values,
    'ARIMA Manual Forecast': arima_manual_forecast.values,
    'Auto ARIMA Forecast': auto_arima_forecast_series.values,
    'SARIMAX Forecast': sarimax_forecast.values,
    'SARIMAX EXGON Forecast': sarimax_exgon_forecast.values
})

# Calculate Errors
for col in ['ARIMA Manual Forecast', 'Auto ARIMA Forecast', 'SARIMAX Forecast', 'SARIMAX EXGON Forecast']:
    comparison_df[f'{col} Error %'] = (
        round((abs(comparison_df[col] - comparison_df['Actual Sales Volume']) / comparison_df['Actual Sales Volume'] * 100),
1)
    )

comparison_df
```

Code 10: Table of Model Forecast Results

```
print("Model Evaluation Metrics:")
metrics_data = {
    'Metric': ['MAE', 'RMSE', 'MAPE %'],
    'ARIMA Manual': [
        mean_absolute_error(y_test, arima_manual_forecast),
        root_mean_squared_error(y_test, arima_manual_forecast),
        mean_absolute_percentage_error(y_test, arima_manual_forecast) * 100
    ],
    'Auto ARIMA': [
        mean_absolute_error(y_test, auto_arima_forecast_series),
        root_mean_squared_error(y_test, auto_arima_forecast_series),
        mean_absolute_percentage_error(y_test, auto_arima_forecast_series) * 100
    ],
    'SARIMAX': [
        mean_absolute_error(y_test, sarimax_forecast),
        root_mean_squared_error(y_test, sarimax_forecast),
        mean_absolute_percentage_error(y_test, sarimax_forecast) * 100
    ],
    'SARIMAX_EXGON': [
        mean_absolute_error(y_test, sarimax_exgon_forecast),
        root_mean_squared_error(y_test, sarimax_exgon_forecast),
        mean_absolute_percentage_error(y_test, sarimax_exgon_forecast) * 100
    ]
}
metrics_df = pd.DataFrame(metrics_data)
for col in ['ARIMA Manual', 'Auto ARIMA', 'SARIMAX', 'SARIMAX_EXGON']:
    metrics_df[col] = metrics_df[col].map(lambda x: f"{x:,.2f}")

metrics_df
```

Code 11: Evaluation Metrics Table

```
# --- Plotting the results ---
fig, axs = plt.subplots(2, 1, figsize=(12, 8), sharey=True, constrained_layout=True)

# Create a single series for the actual data
full_data = pd.concat([X_train, y_test])

# --- Top chart: yearly x-axis ---
axs[0].plot(full_data.index.to_timestamp(), full_data, label='Actual Sales Volume', color='orange', linewidth=4)
axs[0].plot(X_train.index.to_timestamp(), X_train, label='Training Data', color='grey', linewidth=3)
axs[0].plot(arima_manual_forecast.index.to_timestamp(), arima_manual_forecast, label='ARIMA Manual Forecast',
color='purple', linestyle='--')
axs[0].plot(auto_arima_forecast_series.index.to_timestamp(), auto_arima_forecast_series, label='Auto ARIMA Forecast',
color='red', linestyle='-.')
axs[0].plot(sarimax_forecast.index.to_timestamp(), sarimax_forecast, label='SARIMAX Forecast', color='green', linestyle=':')
axs[0].plot(sarimax_exgon_forecast.index.to_timestamp(), sarimax_exgon_forecast, label='SARIMAX EXGON Forecast',
color='blue', linestyle='--')

axs[0].set_title('Sales Forecasting - All Models')
axs[0].set_ylabel('Sales Volume')
axs[0].legend()
axs[0].grid(True)

# --- Bottom chart: monthly x-axis (zoomed in) ---
axs[1].plot(y_test.index.to_timestamp(), y_test, label='Actual Sales Volume', color='orange', linewidth=4)
axs[1].plot(arima_manual_forecast.index.to_timestamp(), arima_manual_forecast, label='ARIMA Manual Forecast',
color='purple', linestyle='--')
axs[1].plot(auto_arima_forecast_series.index.to_timestamp(), auto_arima_forecast_series, label='Auto ARIMA Forecast',
color='red', linestyle='-.')
axs[1].plot(sarimax_forecast.index.to_timestamp(), sarimax_forecast, label='SARIMAX Forecast', color='green', linestyle=':')
axs[1].plot(sarimax_exgon_forecast.index.to_timestamp(), sarimax_exgon_forecast, label='SARIMAX EXGON Forecast',
color='blue', linestyle='--')

axs[1].set_title('Zoomed-in View of 2024 Model Forecasts')
axs[1].set_ylabel('Sales Volume')
axs[1].legend()
axs[1].grid(True)

plt.show()
```

Code 12: Time Series Forecast Chart