Supply Chain Optimization

Project Report

By: Siddhant Prakash Satote

Date: 18–01-2025

Table of Contents

Summary

This project focuses on Supply Chain Optimization, analyzing various aspects of inventory management, delivery time optimization,
cost reduction, and bottleneck identification. The goal is to develop actionable insights to enhance efficiency, reduce costs,
and improve overall supply chain performance.

Key findings include:
- High shipping costs relative to profits in certain regions
- Potential delivery bottlenecks that need attention
- Opportunities for optimizing product category inventory based on sales and profit performance.

Introduction

Supply chain management is a critical aspect of business operations. Companies are constantly seeking ways to optimize their
supply chains, reduce costs, and improve delivery efficiency. This project analyzes the Sample Superstore and Delivery Information
datasets to identify inefficiencies and propose strategies for optimization.

The key objectives of this project are to:
1. Optimize inventory management by analyzing product categories and profitability.
2. Identify potential cost reduction opportunities by analyzing shipping costs and their correlation with profits.
3. Improve delivery times by identifying delays and bottlenecks in the delivery process.
4. Provide actionable insights that can be used to improve supply chain efficiency.

Methodology

Data Sources

The data used for this analysis consists of the following datasets:

1. Sample Superstore - Contains sales, profit, and shipping information.
2. Delivery Information - Contains data on delivery time, shipping modes, and bottlenecks.

Data Preprocessing

The datasets were cleaned and merged to ensure they were aligned by common features such as order IDs and delivery times. Missing
values were handled, and columns were transformed for ease of analysis.

Tools Used

- Python: For data manipulation and analysis
- Libraries: Pandas, NumPy, Matplotlib, Seaborn, Plotly
- Jupyter Notebook: For interactive analysis and report generation

Data Preprocessing

The data was cleaned and merged for analysis. The following steps were performed:
1. Loading the datasets
2. Handling missing values
3. Merging the datasets

Code Snippet: Loading and merging data

```
# Code to load and merge datasets
import pandas as pd

superstore_df = pd.read_csv("Sample_Superstore.csv")
deliveries_df = pd.read_csv("deliveries.csv")

# Merge datasets (assuming 'Order ID' is the common column)
merged_df = pd.merge(superstore_df, deliveries_df, on='Order ID')

# Display the first few rows
merged_df.head()
```

Analysis and Findings

1. Product Category Sales and Profit

Analysis:
We analyzed the total sales and profit by Product Category to identify the most profitable categories and optimize inventory.

Code Snippet: Sales and Profit by Product Category

```
# Code for analysis of Sales and Profit by Category
category_sales_profit = superstore_df.groupby('Product
Category').agg({'Sales': 'sum', 'Profit': 'sum'}).reset_index()

category_sales_profit.plot(kind='bar', x='Product Category',
y=['Sales', 'Profit'], stacked=True, figsize=(10, 6))
```

Output: Bar chart showing total sales and profit by product category.

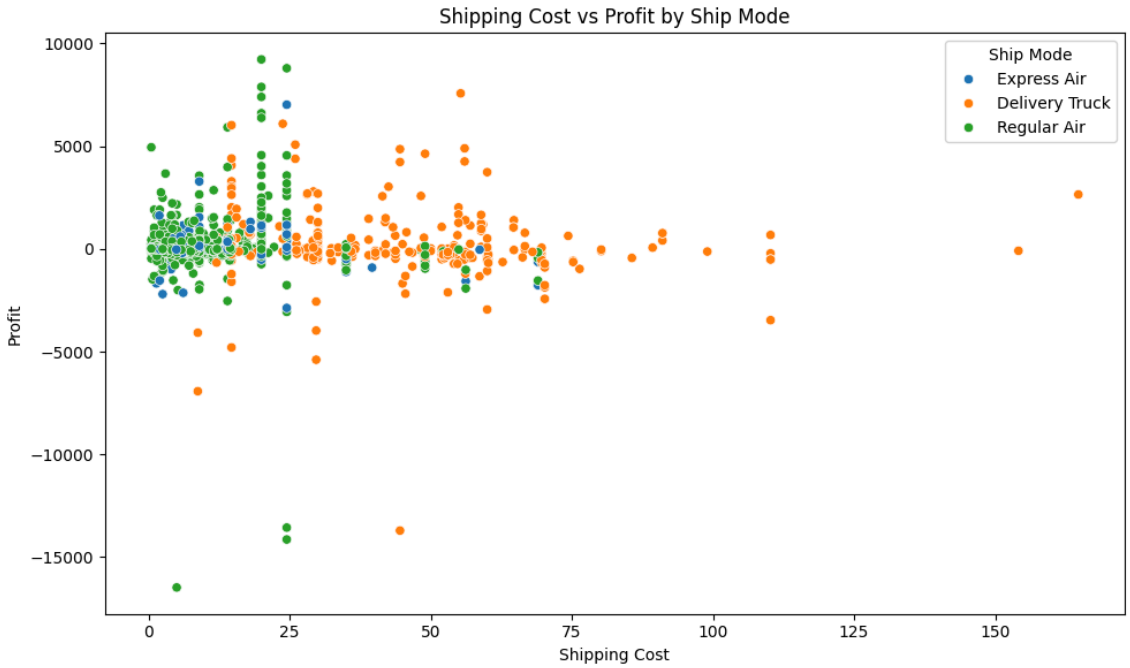
2. Shipping Cost vs Profit

Analysis:
This scatter plot visualizes the relationship between Shipping Cost and Profit to identify areas with high shipping costs and low profitability.

Code Snippet: Shipping Cost vs Profit scatter plot

```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=superstore_df, x='Shipping Cost', y='Profit',
hue='Ship Mode')
plt.title('Shipping Cost vs Profit by Ship Mode')
plt.xlabel('Shipping Cost')
plt.ylabel('Profit')
plt.tight_layout()
plt.show()
```

Output: A scatter plot showing Shipping Cost vs Profit by Ship Mode.



3. Delivery Time Distribution

Analysis:
We analyzed the **Delivery Time** distribution to identify any patterns or inefficiencies in the delivery process that could lead to delays.

Code Snippet: Delivery Time Distribution

```
# Code for plotting the Delivery Time distribution
plt.figure(figsize=(10, 6))
sns.histplot(merged_df['Delivery Time'], kde=True)
plt.title('Delivery Time Distribution')
plt.xlabel('Delivery Time (Days)')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

Output: A histogram showing the Delivery Time distribution.

4. Delivery Bottleneck Analysis

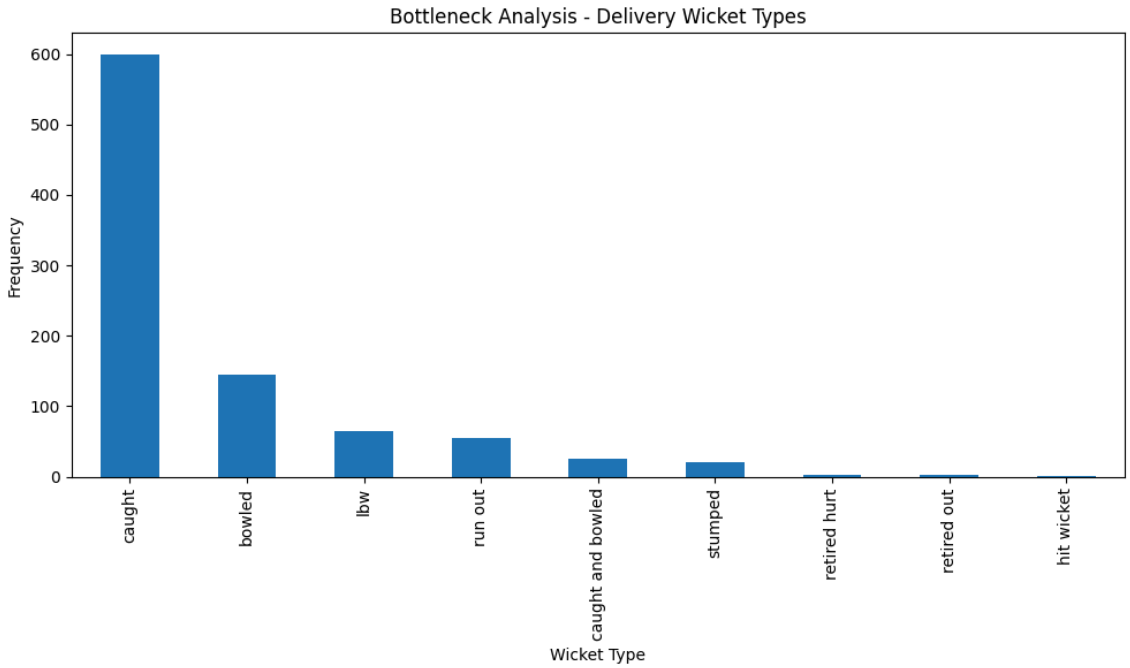
Analysis:
We examined the frequency of **delivery issues** (wicket types) to identify potential bottlenecks in the delivery process.

Code Snippet: Delivery Bottleneck Analysis

```
python
CopyEdit
# Code for bottleneck analysis based on Wicket Type
bottleneck_analysis = deliveries_df['wicket_type'].value_counts()

# Plotting Bottleneck types
bottleneck_analysis.plot(kind='bar', figsize=(10, 6))
plt.title('Bottleneck Analysis - Delivery Wicket Types')
plt.xlabel('Wicket Type')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

Output: A bar chart showing the frequency of delivery issues (wicket types).



Conclusion

This analysis provides actionable insights into inventory optimization, cost reduction, delivery time improvement, and
bottleneck identification within the supply chain. Key findings include:
1. Certain Product Categories show higher profit margins and can be prioritized for better inventory management.
2. Shipping costs in some regions are disproportionately high compared to profits, suggesting areas for cost optimization.
3. Delivery bottlenecks are impacting the efficiency of the delivery process and need attention for smoother operations.

By implementing these findings, businesses can improve supply chain efficiency, reduce costs, and enhance customer satisfaction.

References

1. Data source: Sample Superstore Dataset
2. Python libraries: Pandas, Seaborn, Matplotlib, Plotly