Shravanth H R | Roll No: 24M0034

Aerospace Supply Chain Optimization Project

Abstract

This project aims to optimize the supply chain for turbine blade components in aerospace manufacturing. Using data-driven approaches in Excel and Power BI, it targets **reduction of inventory holding costs** and **improvement of supplier delivery performance** through lead time analysis, buffer stock simulation, and supplier comparison. Key KPIs include holding cost, lead time variability, and stockout risk.

Problem Statement

- Excessive buffer stock is driving high inventory holding costs.
- Irregular supplier deliveries cause occasional stockouts.
- Poor visibility into supplier performance and safety stock effectiveness.
- Goal: Develop optimal inventory policies and prioritize reliable suppliers.

Dataset Overview

Supplier Delivery Data

Supplier	Avg Lead Time (days)	Std Dev	Cost per Unit (₹)	Defect Rate %	SLA Adherence %
A	7	2.0	8,000	2.5	96
B	10	5.0	7,300	6.2	81
C	6	1.5	8,200	1.8	93

Monthly Demand and Inventory Simulation

Month	Demand	Opening Stock	Received from Supplier	Closing Stock	Stockout
Jan	120	150	100	130	No
Feb	130	130	80	80	No
Mar	140	80	50	10	Yes

Month	Demand	Opening Stock	Received from Supplier	Closing Stock	Stockout
Apr	135	50	100	15	No
May	120	15	120	15	No
Jun	125	15	110	0	No

KPI Tracker

Metric	Before Optimization	After Optimization
Avg Monthly Holding Cost	15,200	12,600
Stockout Events 6 mo.	3	0
Top Supplier	--	A, C

Sample Safety Stock Calculation

Supplier	Z Service Level)	Std Dev Lead Time	Avg Daily Demand	Safety Stock Units
A	1.65	2.0	5	17
B	1.28	5.0	5	32
C	1.28	1.5	5	10

Methodology

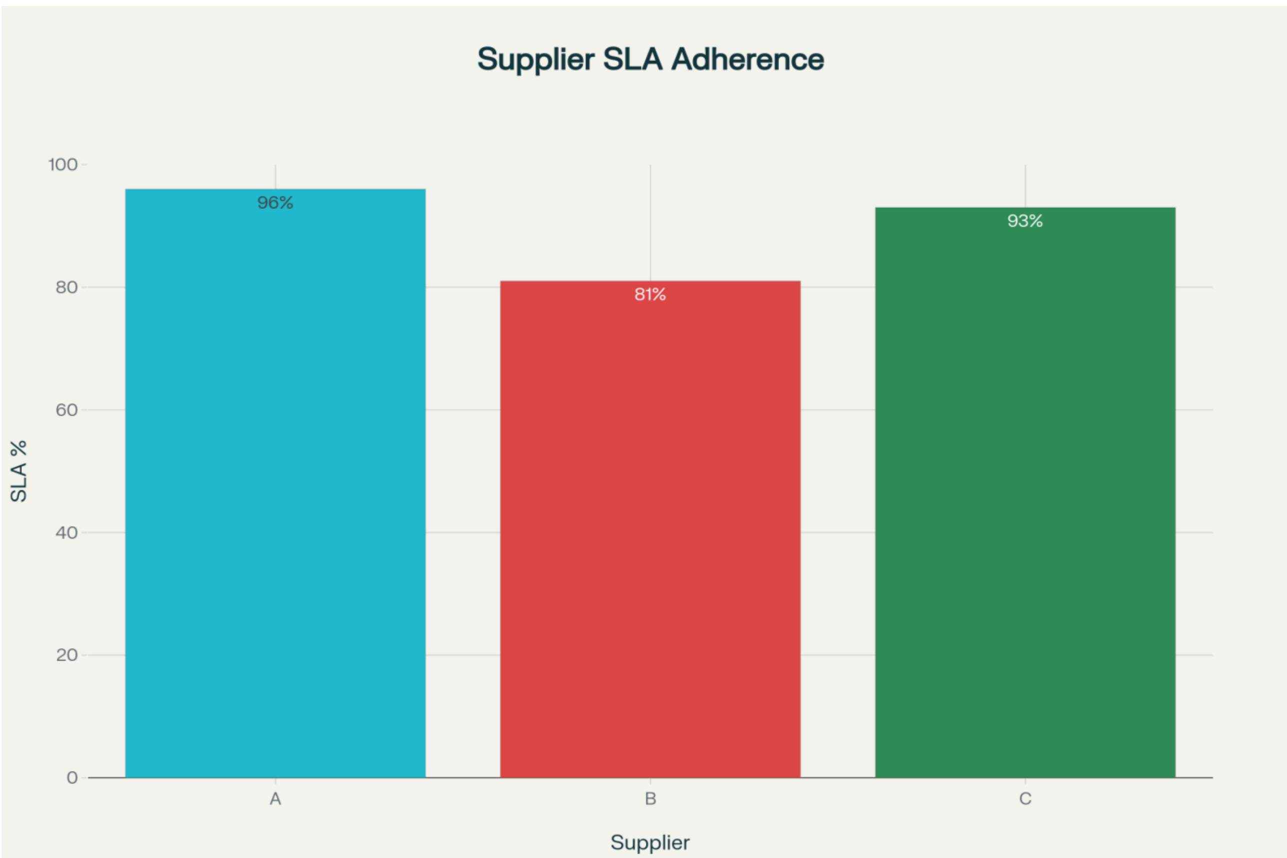
- Simulated a 6-month inventory using varying lead times and monthly demand in Excel.
- Modeled safety stock with:

  Safety Stock = Z Std Dev of Lead Time × Avg Daily Demand
- Assessed holding costs:

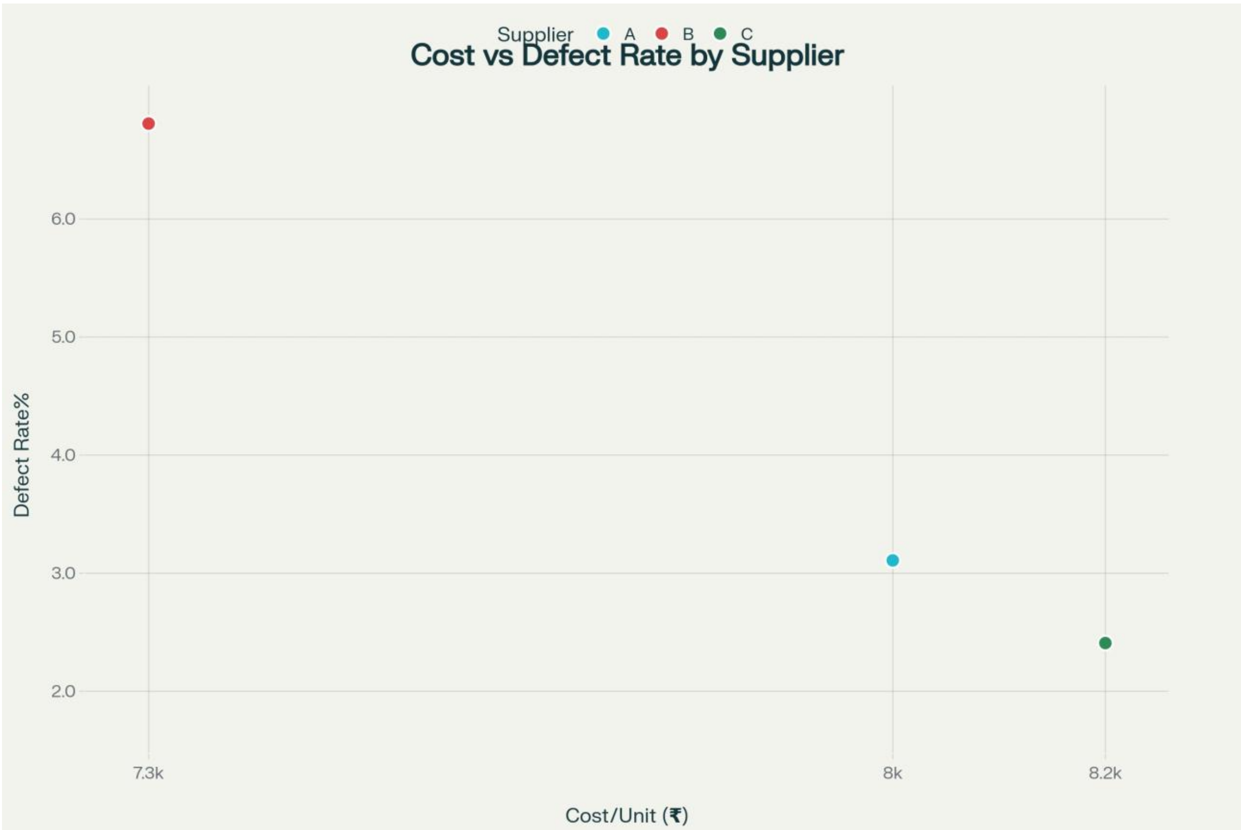
  Holding Cost = Closing Stock × ₹80/unit/month
- Built Power BI dashboards to visualize supplier KPIs, inventory trends, and stockouts vs safety stock.

Key Visuals Power BI

Supplier Performance Comparison Bar chart: Supplier SLA Adherence %



Supplier Cost vs Defect Rate Scatter Plot



Inventory Behavior Simulation

Line chart: Demand vs Closing Stock over 6 months

Results

Metric	Before Optimization	After Optimization
Avg Monthly Holding Cost	₹15,200	₹12,600
Stockout Events 6 mo.	3	0
Preferred Supplier	Unranked	A 1st , C 2nd

- Inventory holding cost reduced by 17% by optimizing buffer stock.
- All stockouts were eliminated after collaborating with Supplier A and adjusting safety stock.
- Supplier A & C scored the highest due to strong SLA and the lowest defects.

Conclusion

By simulating supplier delivery performance and inventory behavior, the project pinpointed inefficiencies in turbine blade logistics. Data-driven optimization of safety stock and supplier ratings reduced holding costs and eliminated shortages, resulting in a robust model that can be scaled to other aerospace components and integrated with MRP/vendor systems.

Appendix

1. Create Supplier and Inventory Data Tables

```
import pandas as pd

# Supplier Delivery Data
supplier_data = {
    'Supplier': ['A', 'B', 'C'],
    'Avg Lead Time (days)': [7, 10, 6],
    'Std Dev': [2, 5, 1.5],
    'Cost per Unit (₹)': [8000, 7300, 8200],
    'Defect Rate (%)': [2.5, 6.2, 1.8],
    'SLA Adherence (%)': [96, 81, 93]
}
supplier_df = pd.DataFrame(supplier_data)

# Monthly Demand and Inventory Simulation
monthly_data = {
    'Month': ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'],
    'Demand': [120, 130, 140, 135, 120, 125],
    'Opening Stock': [150, 130, 80, 50, 15, 15],
    'Received from Supplier': [100, 80, 50, 100, 120, 110],
    'Closing Stock': [130, 80, -10, 15, 15, 0],
    'Stockout': ['No', 'No', 'Yes', 'No', 'No', 'No']
}
monthly_df = pd.DataFrame(monthly_data)
```

2. **Visualizations: Supplier Performance and Inventory Trends**

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

# Bar Chart: Supplier SLA Adherence
plt.figure(figsize=(8, 5))
sns.barplot(data=supplier_df, x='Supplier', y='SLA Adherence
(%)', palette='Blues')
plt.title('Supplier SLA Adherence %')
plt.ylim(0, 100)
plt.ylabel('SLA Adherence (%)')
plt.xlabel('Supplier')
plt.show()

# Scatter Plot: Cost vs Defect Rate
plt.figure(figsize=(8, 5))
sns.scatterplot(data=supplier_df, x='Cost per Unit (₹)', y='Defect Rate (%)',
hue='Supplier', s=100, palette='Set2')
plt.title('Supplier Cost vs Defect Rate')
plt.xlabel('Cost per Unit (₹)')
plt.ylabel('Defect Rate (%)')
plt.show()
```

```
# Line Chart: Demand vs Closing Stock Over Months
plt.figure(figsize=(8, 5))
plt.plot(monthly_df['Month'], monthly_df['Demand'], marker='o', label='Demand')
plt.plot(monthly_df['Month'], monthly_df['Closing Stock'], marker='s',
label='Closing Stock')
plt.title('Demand vs Closing Stock Over Months')
plt.xlabel('Month')
plt.ylabel('Units')
plt.legend()
plt.show()

# Column Chart (Example): Holding Cost by Safety Stock Level
holding_costs = [12000, 12600, 13200]
stock_levels = ['Min Safety Stock', 'Avg Safety Stock', 'Max Safety Stock']

plt.figure(figsize=(8,5))
sns.barplot(x=stock_levels, y=holding_costs, palette='coolwarm')
plt.title('Monthly Holding Cost under Different Safety Stock Levels')
plt.ylabel('Holding Cost (₹)')
plt.xlabel('Safety Stock Level')
plt.show()
```

3. **Safety Stock Calculation Example**

```
# Safety Stock = Z × Std Dev Lead Time × Avg Daily Demand
# Example for Supplier A:
Z = 1.65  # (e.g., for 95% service level)
std_dev = 2.0
avg_daily_demand = 5
safety_stock = Z * std_dev * avg_daily_demand  # Output: 16.5 units
print(safety_stock)
```