Data-Driven Warehouse Inventory Optimization & Stock Movement Analysis

Mr. Hemant

Table of contents

1 Project Overview

Efficient inventory management is a critical factor in maintaining smooth warehouse operations while reducing costs and improving service. This project focuses on analyzing and optimizing inventory performance using real-world style datasets, advanced analytics, and business intelligence tools.

The dataset used in this project simulates a complete warehouse ecosystem, including inventory records, sales transactions, purchase orders, supplier performance metrics, warehouse zones, bin locations, batch numbers, and expiry information. With this data, the project aims to uncover inefficiencies, identify risks, and propose data-driven solutions to improve inventory health.

This project replicates real industry challenges, aligns with supply chain and operations analytics roles, and demonstrates the ability to connect domain expertise with data-driven decision-making.

---

2 Business Problem

Warehouse operations often struggle with balancing stock availability against cost efficiency. Excess inventory blocks working capital, increases holding costs, and leads to waste, especially for perishable and slow-moving items. At the same time, inadequate stock levels create stockouts, order delays, and dissatisfied customers.

Unreliable supplier performance, inconsistent lead times, poor demand visibility, and lack of real-time insights further complicate decision making. As a result, organizations face higher operational costs, reduced service levels, and inefficiencies across the supply chain.

This project aims to address these challenges by identifying inventory risks, analyzing stock behavior, evaluating supplier reliability, and optimizing reorder strategies-turning operational data into clear, actionable insights for better warehouse management.

---

3 Project Objectives

The objective of this project is to enhance warehouse decision-making by leveraging data analytics to uncover inefficiencies, reduce operational risks, and optimize inventory performance. The project focuses on delivering clear, actionable insights that support strategic and day-to-day operations. Key objectives include:

- Identify stock risks such as slow-moving, dead, and near-expiry items.

- Reduce stockouts by analyzing demand patterns and monitoring reorder levels.
- Optimize inventory levels using data-driven reorder strategies.
- Improve supplier performance through lead-time and reliability analysis.
- Lower operational and holding costs by eliminating excess stock and improving inventory rotation.
- Strengthen warehouse efficiency by evaluating zone utilization and storage distribution.
- Enable informed decision-making through interactive dashboards and analytical reporting.

These objectives collectively demonstrate the impact of analytics in improving warehouse operations and ensuring better control over inventory health.

4 Dataset Description

This project uses a realistic, multi-table dataset that represents core warehouse operations. It includes:

1. **Inventory Data:** SKU details such as category, stock levels, reorder thresholds, bin locations, batch numbers, and expiry dates.
2. **Sales Data:** Daily item-level sales quantities used for demand analysis and inventory turnover.
3. **Purchase Orders Data:** Replenishment records showing order quantities, dates, and supplier references—supporting lead-time and procurement analysis.
4. **Supplier Data:** Supplier details with lead times and delivery performance metrics.

This dataset provides all essential elements required to evaluate stock health, forecast demand, assess supplier reliability, and optimize inventory decisions.

5 Data Analysis

Q1. Which items are at risk of stockout based on current stock and demand trends?

```sql
WITH
    last_month_qty_sold AS (
        SELECT
            item_id,
            SUM(sold_qty) AS qty_sold_last_month
        FROM
            sales
        WHERE
            sale_date BETWEEN '2025-06-01' AND '2025-06-30'
        GROUP BY
            item_id
    ),
    current_stock_demand AS (
        SELECT
            inv.item_id,
            inv.item_name,
            inv.current_stock,
            COALESCE(lmqs.qty_sold_last_month, 0) AS qty_sold_last_month,
            COALESCE(lmqs.qty_sold_last_month / 30, 0) AS avg_daily_demand
        FROM
            inventory inv
            LEFT JOIN last_month_qty_sold lmqs ON lmqs.item_id = inv.item_id
    )
SELECT
    item_id,
    item_name,
    current_stock,
    qty_sold_last_month,
    avg_daily_demand,
    CASE
        WHEN avg_daily_demand > 0 THEN current_stock / avg_daily_demand
        ELSE NULL
    END AS days_of_stock_left,
    CASE
        WHEN avg_daily_demand = 0 THEN 'No Demand'
        WHEN current_stock / avg_daily_demand <= 20 THEN 'High Risk'
        WHEN current_stock / avg_daily_demand <= 50 THEN 'Medium Risk'
        ELSE 'Low Risk'
    END AS stockout_risk
FROM
    current_stock_demand
ORDER BY
    days_of_stock_left;
```

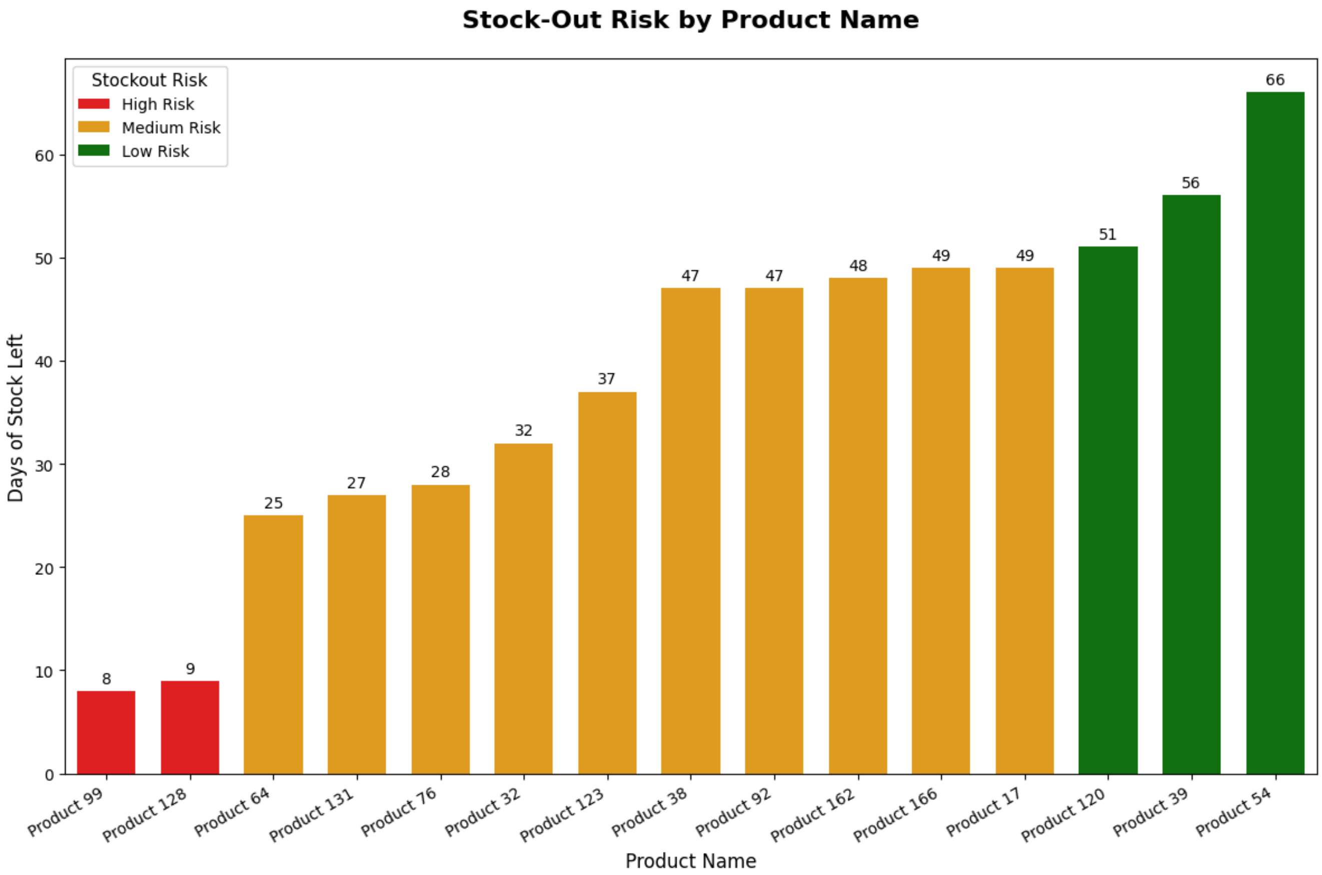

Figure 1: Stock-Out Risk by Product

Q2. How much inventory value is tied up in slow-moving or dead stock?

```
WITH
    inventory_value AS (
        SELECT
            inv.item_id,
            SUM(inv.current_stock * inv.unit_cost) AS stock_value,
            MAX(sls.sale_date) AS sale_date
        FROM
            sales sls
            JOIN inventory inv ON inv.item_id = sls.item_id
        GROUP BY
            inv.item_id
    ),
    inventory_value_status AS (
        SELECT
            item_id,
            stock_value,
            sale_date,
            CASE
                WHEN sale_date <= '2025-03-31' THEN 'slow-stock'
                ELSE 'fast-stock'
            END AS stock_status
        FROM
            inventory_value
    )
SELECT
    stock_status,
    ROUND(SUM(stock_value) / 1000000, 2) AS stock_value_mn
FROM
    inventory_value_status
GROUP BY
    stock_status;
```

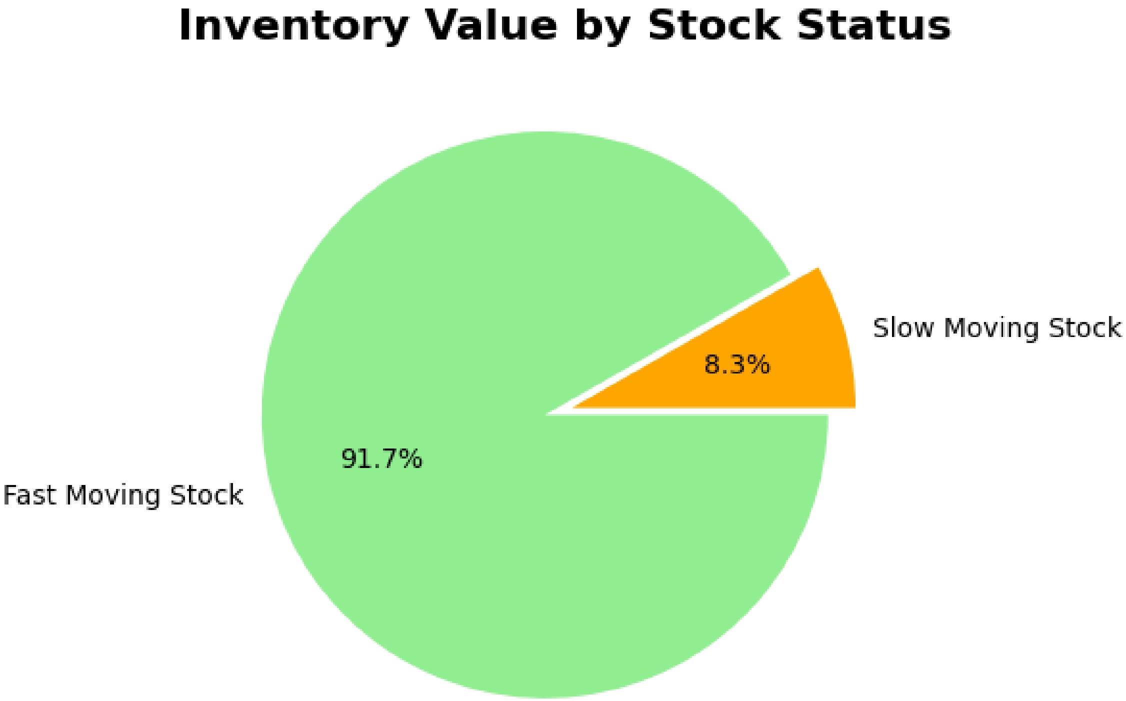

Figure 2: Inventory Value by Stock Status

Insights :

1. Over 91% of stock which is fast moving, could be due to low price, essential products which is used in day to day life or popularity of the brands among the customers.
2. 8.3% stock is identified as slow moving, In this category the products are more likely to be luxury goods or seasonal.
3. In slow moving stock we have to identify why these products stuck at the warehouse and plan inventory accordingly.

Q3. Which SKUs are approaching expiry and require priority action?

```sql
WITH
    product_expiry AS (
        SELECT
            item_id,
            item_name,
            (current_stock * unit_cost) AS inventory_value,
            date_expiry,
            (date_expiry - '2025-07-16') AS days_in_expiry
        FROM
            inventory
    )
SELECT
    item_id,
    item_name,
    inventory_value,
    date_expiry,
    days_in_expiry,
    CASE
        WHEN days_in_expiry < 0 THEN 'Expired'
        WHEN days_in_expiry <= 30 THEN 'High Risk'
        WHEN days_in_expiry <= 90 THEN 'Medium Risk'
        WHEN days_in_expiry <= 180 THEN 'Low Risk'
        ELSE 'Safe'
    END AS expiry_status
FROM
    product_expiry
ORDER BY
    days_in_expiry;
```

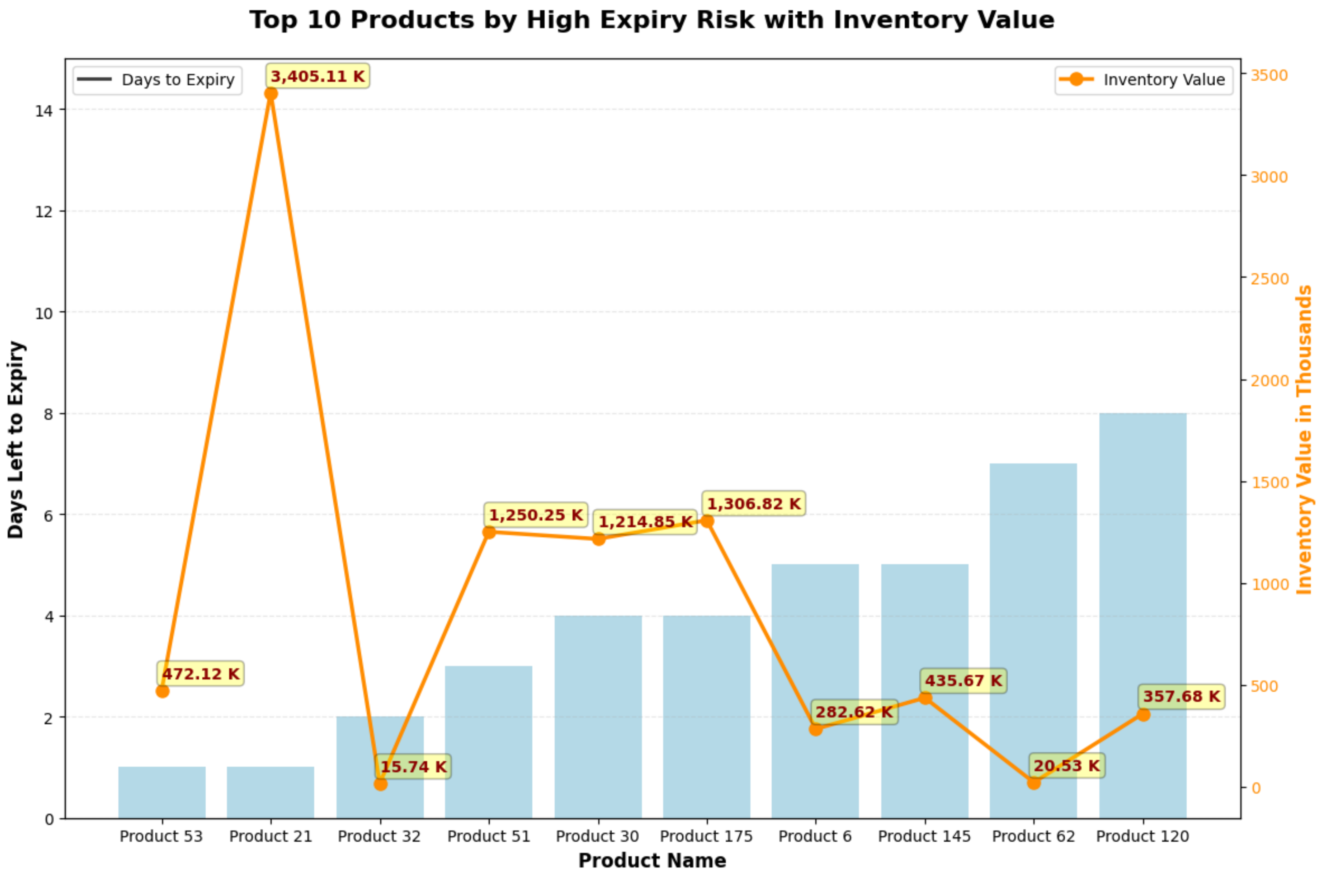

Figure 3: Product Expiry Status

Insights:

1. Some products are near to expiry and those products need immediate attention.
2. Product 21 is near to it's expiry date, and has the highest inventory value 3405.11 K.
3. For stock clearence we can introduce new discount offers and promotions strategy to clear the stocks.

Q4. What is the current inventory turnover rate, and which categories perform the best or worst?

```
WITH
    sales_cogs AS (
        SELECT
            inv.item_id,
            inv.category,
            SUM(inv.current_stock * inv.unit_cost) AS inventory_value,
            SUM(sls.sold_qty) AS total_sold_qty,
            MIN(inv.unit_cost) AS unit_cost,
            SUM(sls.sold_qty * inv.unit_cost) AS cogs
        FROM
```

```
            inventory inv
            JOIN sales sls ON (sls.item_id = inv.item_id)
        GROUP BY
            inv.item_id,
            inv.category
    ),
    inventory_turnover_per_item AS (
        SELECT
            item_id,
            category,
            inventory_value,
            cogs,
            ROUND(
                CASE
                    WHEN inventory_value > 0 THEN cogs / (
                        SELECT
                            AVG(inventory_value)
                        FROM
                            sales_cogs
                    )
                    ELSE 0
                END,
                2
            ) AS inventory_turnover_rate
        FROM
            sales_cogs
    )
SELECT
    category,
    ROUND(AVG(inventory_turnover_rate), 2) AS inventory_turnover_rate
FROM
    inventory_turnover_per_item
GROUP BY
    category;
```

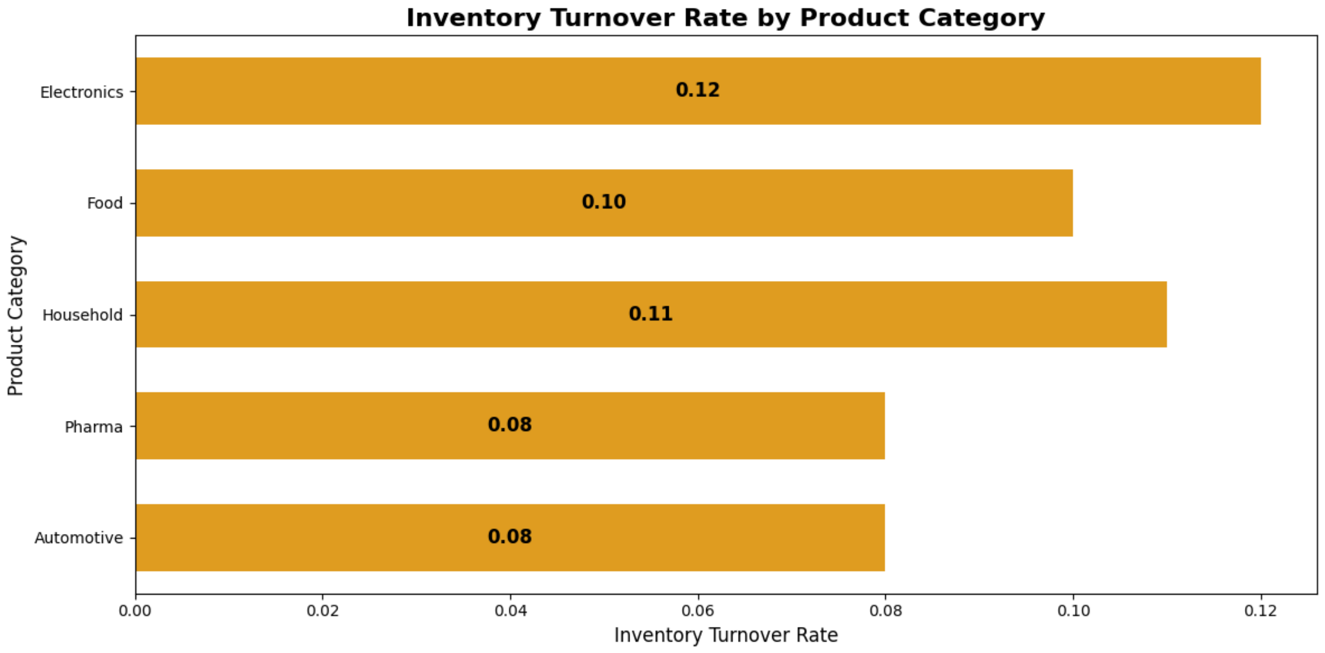

Figure 4: Inventory Turnover Rate by Product Category

Insights :

1. Electronics performing best when compare to rest of the other categories. It's high turnover rate 0.12 indicate that electronics products stock movement is fast.
2. Worst Performing categories are Pharma and Automotive in all the categories with turnover rate 0.8 for each.
3. Low turnover rate suggest that these product more likely to sit in warehouse compared to other categories.
4. We should plan our inventory purchase accordingly.

---

6 Key Insights

---

7 Recommendations

---

8 Conclusion