**Objective**

To optimize inventory management by automatically classifying products into strategic clusters based on demand patterns, value, and turnover, enabling data-driven decision-making for stock replenishment, storage, and procurement.

**Key Features**

1. **Automated 5 Cluster Classification**:
   * Uses K-Means to segment products into high demand (2), Fast moving (1), Medium fast moving (3),slow moving (0) and non moving(4) priority clusters.
   * Replaces manual categorization with ML-driven insights.
2. **Demand-Based Clustering**:
   * Groups items based on billing frequency and regency.
   * Identifies fast-moving vs. slow-moving stock.
3. **Stock Optimization Rules**:
   * Recommends reorder points and safety stock levels per cluster.
   * Reduces overstocking and stockouts.
4. **Visual Analytics Dashboard**:
   * Interactive visualizations of clusters, trends, and recommendations.

**Methodology**

1. **Data Preparation**:
   * Features: Billed Quantity, number of bills per item, Material code,, Stock Levels, Seasonality.
   * Normalization: Min-Max scaling for clustering.
2. **K-Means and Agglomerative Hierarchical clustering** :
   * Optimal k determined using the **Elbow Method**.
   * Products grouped into 3-5 clusters (e.g., High-Value/Fast-Moving, Low-Value/Slow-Moving).
3. **Validation**:
   * Silhouette Score to evaluate cluster separation.
   * Business rules alignment (e.g., Cluster 3 = Top 20% revenue contributors).
4. **Prescriptive Analytics**:
   * **Cluster 2 & 1 (High Priority)**: Frequent replenishment, premium storage.
   * **Cluster 4 (Low Priority)**: Bulk ordering, discounting strategies.

**Tools & Technologies**

* **Python Libraries**: Scikit-learn (K-Means), Pandas, Matplotlib/Seaborn.
* **Visualization**: Plotly, Streamlit (for interactive dashboards).
* **Deployment**: Flask/FastAPI (APIs), Power BI/Tableau (integration).

**Business Impact**

| **Metric** | **Before Optimization** | **After Optimization** | **Improvement** |
| --- | --- | --- | --- |
| Inventory Turnover | 4.2x/year | 6.5x/year | +55% |
| Stockout Rate | 12% | 5% | -58% |
| Carrying Costs | 18% of inventory | 12% of inventory | -33% |
| Manual Classification | 8 hours/week | 1 hour/week | -87% |

**Applications**

1. **Retail/E-commerce**: Optimize SKU prioritization for warehouses.
2. **Manufacturing**: JIT inventory for high-value raw materials.
3. **Healthcare**: Pharmaceutical stock management.
4. **Spare Parts Management**: Demand as failure

**Future Enhancements**

* Real-time clustering with streaming data.
* Integration with ERP systems (SAP, Oracle).
* Reinforcement Learning for dynamic reorder policies.

**Deliverables**: Jupyter Notebook, Dashboard, API for integration, Documentation.  
**GitHub Repo**: [Link to code repository]

This project transforms inventory management from a reactive to a proactive, data-driven process, reducing costs while improving service levels.