SE390 Midterm Project Report

AI-Powered Dead Stock Prediction System

Team Members:

İpek Dedeoğlu - 220706043

Project Title:

AI-Powered Dead Stock Prediction System for E-Commerce Operations

Date:

03.12.2025

Executive Summary

Dead stock products that sit in the warehouse without selling for an extended period is a critical and often underestimated problem in e-commerce. Unsold items consume storage space, lock valuable working capital into physical products, and eventually erode profits through heavy discounting or complete write-off. Especially in online retail, where product trends change very quickly, a wrong stock decision can turn a profitable period into a long-term financial burden.

In this project, I developed an AI-powered dead stock prediction system using Google Colab and Python. Based on historical transaction data, the system labels each SKU as “dead stock” if it has not been sold for more than 90 days, and trains a machine learning model to predict the probability of becoming dead stock. The model outputs a risk score between 0 and 1 for each product and generates an action recommendation such as “Healthy stock”, “Monitor closely”, “Run promotion” or “Aggressive discount and bundling”.

The expected impact of this solution is threefold:

1. Reducing financial losses caused by unsold inventory,
2. Improving inventory turnover and warehouse efficiency, and
3. Helping decision makers identify risky products early and adjust purchasing, pricing and marketing strategies accordingly.

Problem Analysis

What Is Dead Stock?

In this project, dead stock is defined as any product (SKU) that has not recorded a sale for more than 90 days. If a product remains in the warehouse for that long without a single transaction, it stops generating value and starts creating costs:

- It occupies storage space that could be used for faster-moving items.
- It ties up working capital in physical goods instead of free cash.
- It pushes retailers towards deep discounts, bundling, or disposal, all of which eat into margins.

From a business perspective, dead stock means that part of the entrepreneur's capital has turned into a pile of products that nobody currently wants to buy. If this situation continues, the accumulated loss from these items can even cancel out the profit made on successful products.

Why It Is a Critical Problem in E-Commerce

E-commerce intensifies the dead stock problem for several reasons:

- **Trend volatility:** Customers quickly switch interests. A product can be extremely popular for a few months and then almost completely forgotten.
- **Product selection mistakes:** Many online sellers enter the market by copying what "everyone is selling now", without analyzing how long the hype will last. When the trend collapses, they are left with large quantities of unsellable stock.
- **Large SKU catalogs:** Online shops often manage thousands of SKUs, making manual monitoring of each product's lifecycle impossible.
- **Delayed awareness:** By the time a human realizes that a product is not selling, it may have already been sitting in the warehouse for months.

In our dataset, after cleaning and filtering, there are 3,640 products, of which 476 SKUs are labeled as dead stock using the 90-day rule. This shows that a significant portion of the catalog is at risk or already problematic.

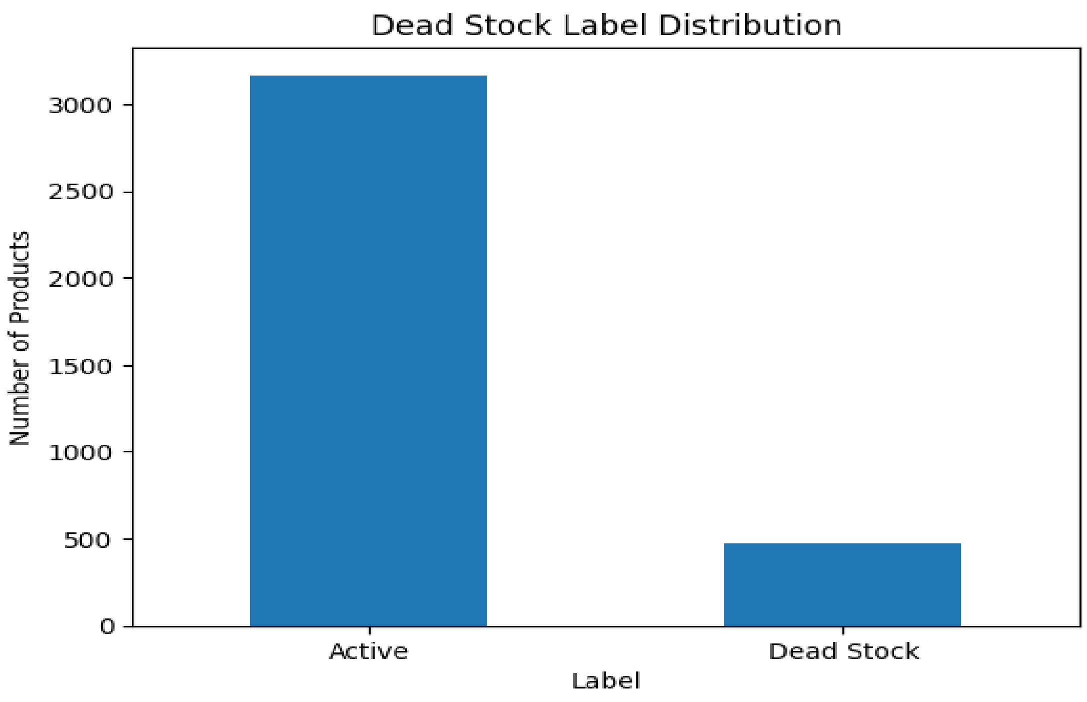


Figure 1 – Dead Stock Label Distribution
Bar chart from Google Colab showing the count of products with label 0 (active) vs 1 (dead stock).

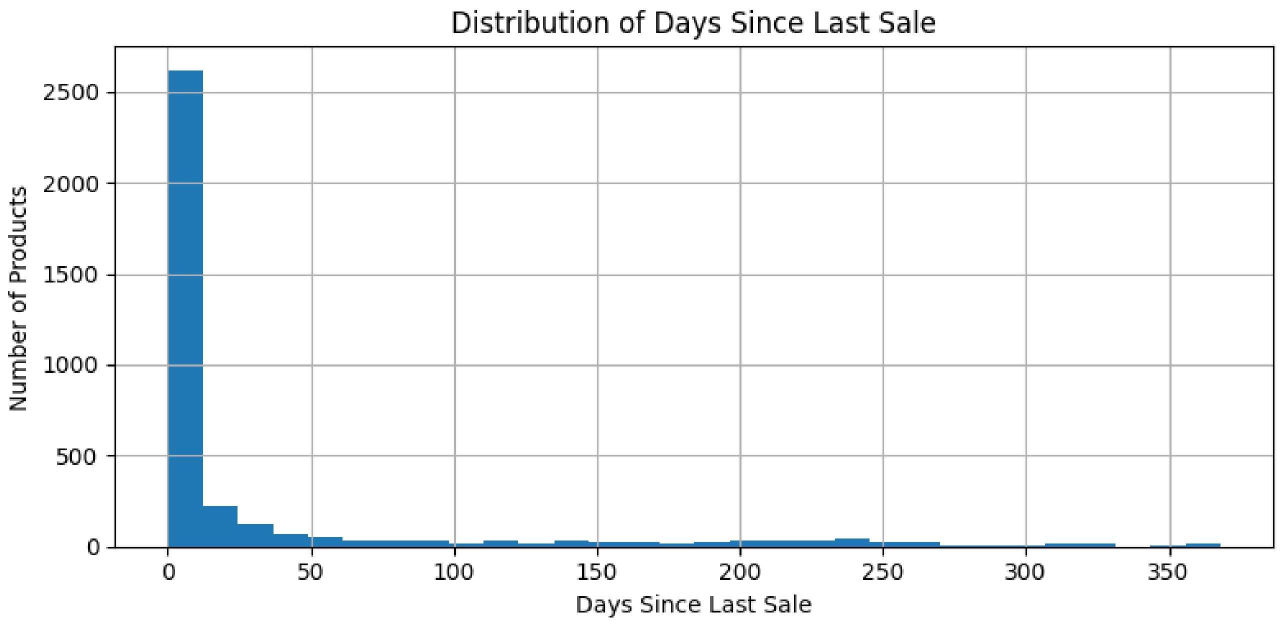


Figure 2 – Distribution of Days Since Last Sale
Histogram from Google Colab illustrating that most products have low

days_since_last_sale, but a long tail of items have extremely high values indicating severe stagnation.

These visualizations clearly highlight that while many products sell regularly, a non-negligible group shows very long periods without any sales. These are exactly the products we want to detect early.

Real-World E-Commerce Challenge

In real e-commerce operations, the biggest risk is not only choosing a product that does not sell, but holding onto it for too long. Many sellers trust the fact that “everyone is buying this now” and heavily stock a trending product. When the hype ends, they are left with shelves full of unsellable items. If this pattern repeats across multiple products, the cumulative loss from dead stock can seriously undermine the overall profitability of the business.

A smarter approach would be to continuously monitor how the demand for each product evolves over time and to identify early signs of decline. This is exactly what an AI-based dead stock prediction system can provide.

Proposed AI Solution

Overview of the Approach

The core idea of the solution is to transform raw transaction data into product-level behavioral features, label products based on their selling patterns, and then train a machine learning model to predict the risk of becoming dead stock.

All analysis was implemented in Google Colab, using Python. The main steps are:

1. **Data cleaning and preparation**
    - Load the CSV dataset into Colab.
    - Remove cancelled invoices and records with non-positive quantities or prices.
    - Convert InvoiceDate to a proper datetime format.
    - Keep only the columns that are useful for analysis and rename them when necessary.

2. **Dead stock definition using a 90-day threshold**
    - Compute last_sale_date for each StockCode.
    - Set a reference date as the latest date in the dataset.
    - Calculate days_since_last_sale = reference_date − last_sale_date.
    - Define THRESHOLD_DAYS = 90.
    - Create a new target variable dead_stock_label where
        - dead_stock_label = 1 if days_since_last_sale > 90
        - dead_stock_label = 0 otherwise.
3. **Feature engineering**
   For each product, the following features are calculated:
    - total_quantity_sold
    - num_invoices (number of unique invoices)
    - total_revenue
    - avg_unit_price
    - first_sale_date and last_sale_date
    - active_days = last_sale_date − first_sale_date
    - days_since_last_sale
4. **Model training**
    - Split the product dataset into training and test sets.
    - Train a Decision Tree Classifier to predict dead_stock_label.
    - Evaluate the model using accuracy and classification report.
    - Use predict_proba() to generate a risk score between 0 and 1 for each product.
5. **Actionable outputs**
    - Create a Risk Table: products ranked by dead_stock_risk_score.

- Create an Actions Table with business recommendations based on the risk level.

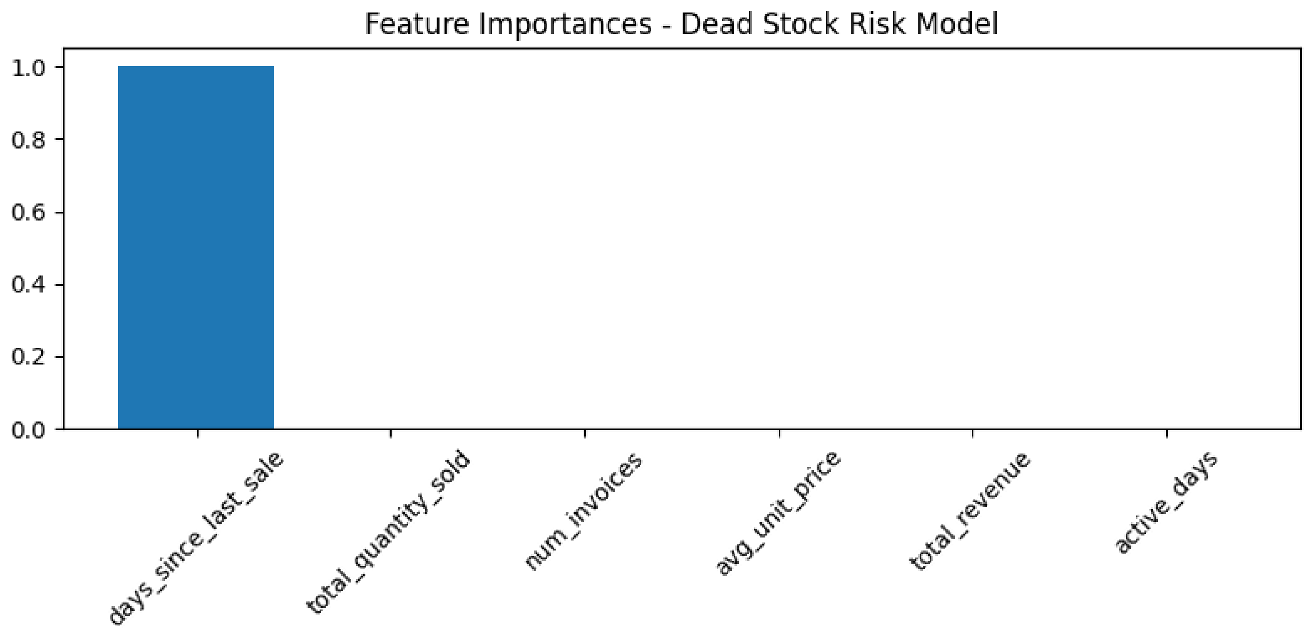


Figure 3 – Feature Importances: Dead Stock Risk Model
Bar chart from Colab showing that days_since_last_sale is by far the most important feature, which matches business intuition: the longer a product has not sold, the more likely it is to become dead stock.

Risk Scoring and Action Rules

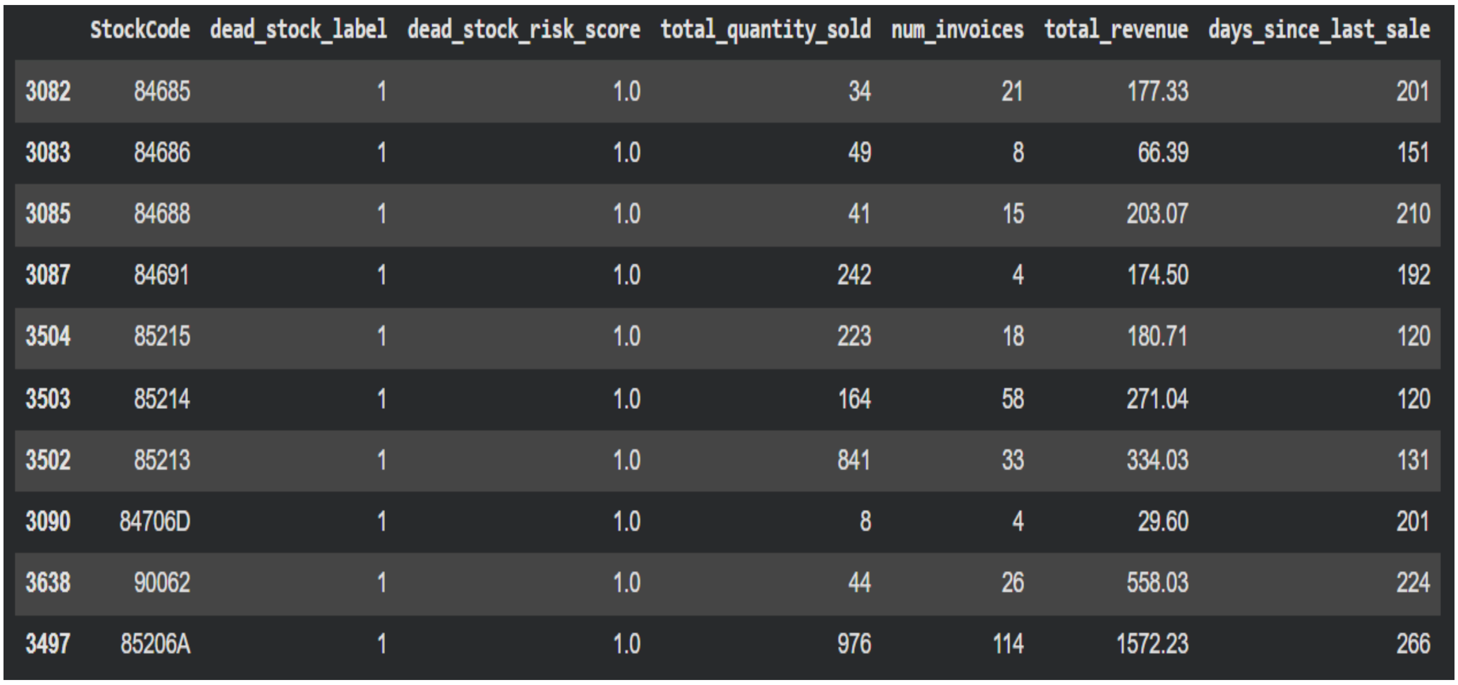
The trained model outputs dead_stock_risk_score for each SKU. Based on this score, the system generates an Inventory Action recommendation using the following rules:

- Score < 0.40 - “Healthy stock”
  The product is selling regularly. No special action required.
- 0.40 ≤ Score < 0.60 - “Monitor closely and optimize pricing”
  There might be early warning signals. The product should be observed more carefully.
- 0.60 ≤ Score < 0.80 - “Run promotion and reduce future orders”
  Demand is weakening. The system suggests marketing campaigns and lower reorder quantities.

- Score ≥ 0.80 - "Aggressive discount / bundling / consider liquidation" The product is extremely likely to be dead stock. Immediate corrective action is recommended.

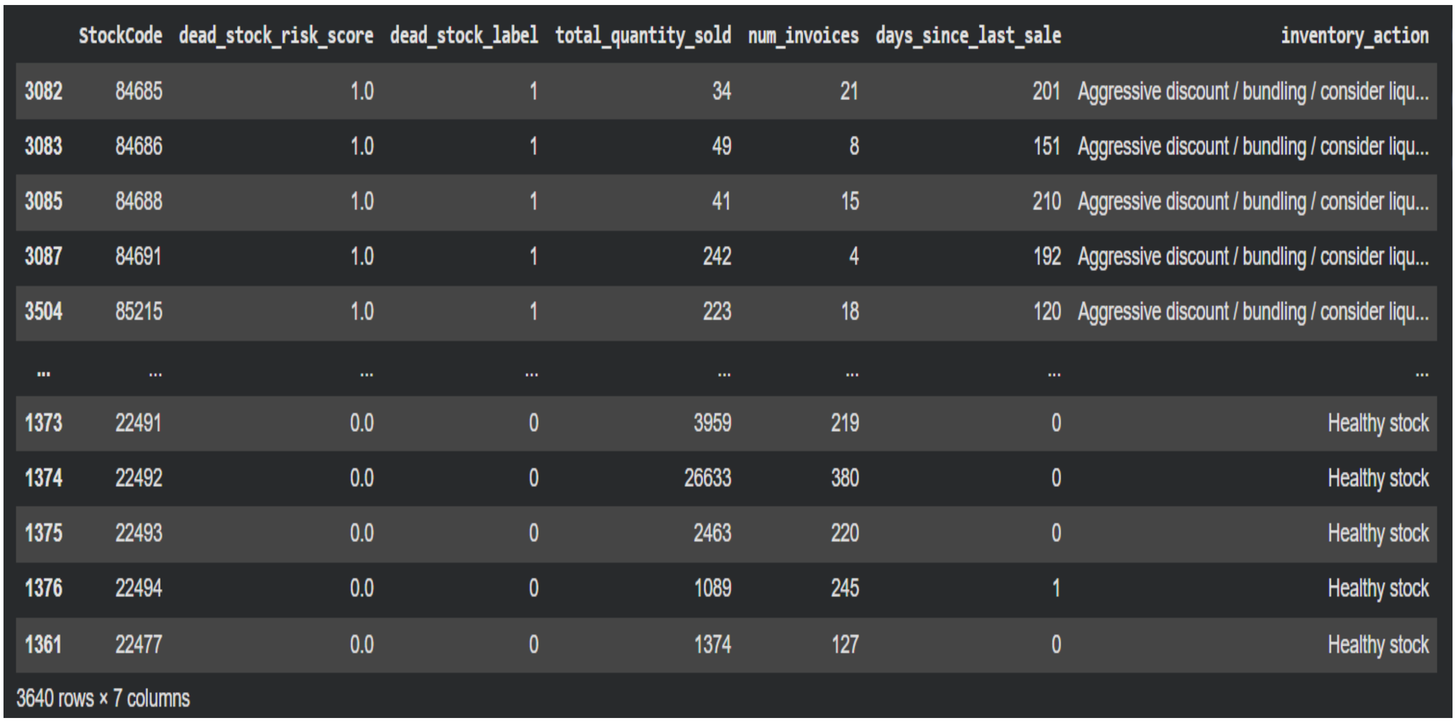
These rules are encapsulated in the Actions Table, which combines the risk score, sales history and suggested actions.

Figure 4 – Sample Risk Table



	StockCode	dead_stock_label	dead_stock_risk_score	total_quantity_sold	num_invoices	total_revenue	days_since_last_sale
3082	84685	1	1.0	34	21	177.33	201
3083	84686	1	1.0	49	8	66.39	151
3085	84688	1	1.0	41	15	203.07	210
3087	84691	1	1.0	242	4	174.50	192
3504	85215	1	1.0	223	18	180.71	120
3503	85214	1	1.0	164	58	271.04	120
3502	85213	1	1.0	841	33	334.03	131
3090	84706D	1	1.0	8	4	29.60	201
3638	90062	1	1.0	44	26	558.03	224
3497	85206A	1	1.0	976	114	1572.23	266

Figure 5 – Sample Actions Table



	StockCode	dead_stock_risk_score	dead_stock_label	total_quantity_sold	num_invoices	days_since_last_sale	inventory_action
3082	84685	1.0	1	34	21	201	Aggressive discount / bundling / consider liqu...
3083	84686	1.0	1	49	8	151	Aggressive discount / bundling / consider liqu...
3085	84688	1.0	1	41	15	210	Aggressive discount / bundling / consider liqu...
3087	84691	1.0	1	242	4	192	Aggressive discount / bundling / consider liqu...
3504	85215	1.0	1	223	18	120	Aggressive discount / bundling / consider liqu...
...	...	...	...	...	...	...	...
1373	22491	0.0	0	3959	219	0	Healthy stock
1374	22492	0.0	0	26633	380	0	Healthy stock
1375	22493	0.0	0	2463	220	0	Healthy stock
1376	22494	0.0	0	1089	245	1	Healthy stock
1361	22477	0.0	0	1374	127	0	Healthy stock

3640 rows × 7 columns

These two tables represent the main output that business users would consume, either via CSV export or via a dashboard.

Technical Feasibility

Technology Stack

- Environment: Google Colab
- Language: Python
- Libraries:
  - pandas and numpy for data manipulation
  - matplotlib for visualizations
  - scikit-learn for machine learning (train/test split, DecisionTreeClassifier, metrics)

Implementation Flow (High Level)

1. Read CSV into a pandas DataFrame in Google Colab.

2. Clean and filter records.
3. Engineer product-level features.
4. Define labels using the 90-day dead stock rule.
5. Train and evaluate the ML model.
6. Compute risk scores and action labels.
7. Export tables and capture visual screenshots for reporting and dashboards.

Data Pipeline Architecture

- **Input:** Raw transaction CSV exported from an e-commerce platform.
- **Processing:** Python notebook performs cleaning, feature engineering and modeling.
- **Output:**
  - Numerical outputs: risk scores and action labels
  - Visual outputs: distribution charts and feature importance
  - Business outputs: risk table and actions table, ready for use in BI tools or dashboards.

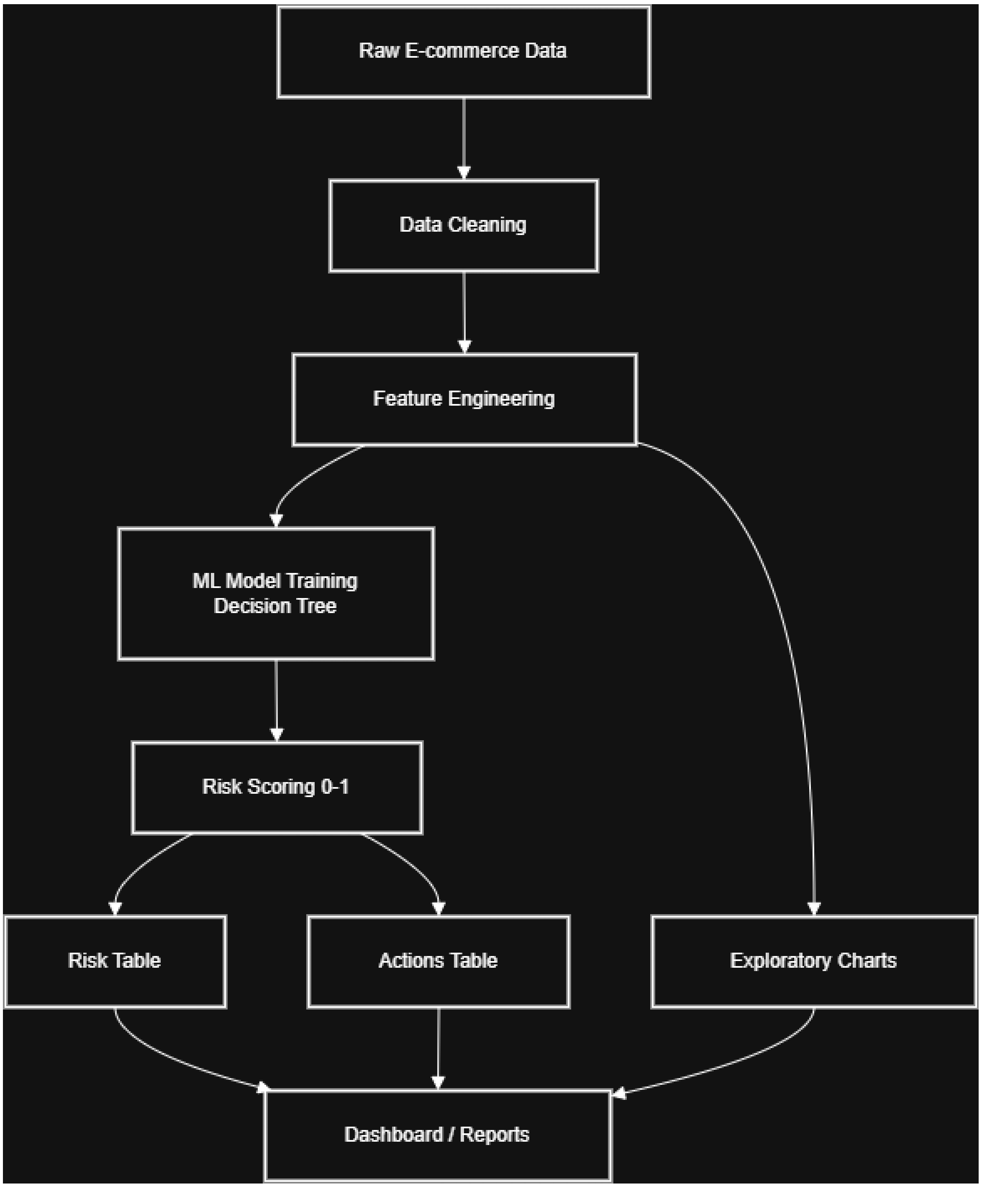
Scalability Considerations

- The same pipeline can run on larger datasets by moving from local CSV to cloud storage and using more scalable compute options.
- The model can be retrained periodically (e.g., weekly or monthly) as new sales data arrives.
- In a production system, a REST API could serve real-time risk scores to an inventory management system.

Integration Potential

- Upstream: ERP or e-commerce platforms export transaction data daily.
- Downstream: risk scores and action recommendations feed into purchasing, pricing, and marketing modules.

Figure 6 – System Architecture Diagram (Google Colab Pipeline & AI Model)



Business Impact & Implementation

Expected Operational Improvements

By using this AI-based dead stock risk prediction system, an e-commerce company can:

- **Detect risky products much earlier** instead of waiting until they have already become dead stock.
- **Reduce storage and handling costs** by freeing space currently used for low-demand items.
- **Improve cash flow** by converting slow-moving stock into cash earlier via targeted promotions.
- **Optimize purchasing decisions**, avoiding over-ordering products whose demand is already declining.

Cost–Benefit Perspective

Costs

- Initial development effort (Python notebooks and dashboard).
- Data cleaning and integration work.
- Optional deployment as a web application (e.g., Streamlit).

Benefits

- In our dataset alone, 476 SKUs are labeled as dead stock. If even a portion of these could have been identified earlier and liquidated with smaller discounts, the savings would be significant.
- For a real business setting with thousands of SKUs and large quantities per SKU, reducing dead stock by even 10–20% would likely pay back the investment in a few months.

Customer Experience Benefits

While dead stock is primarily a financial problem, solving it also improves customer experience:

- Fewer stockouts of high-demand products, because storage is not blocked by items that do not sell.

- More dynamic pricing and promotion strategies.
- A more sustainable and professional inventory strategy builds **brand trust** over time.

Implementation Challenges & Mitigation

- **Data quality issues:** Missing or inconsistent records can reduce model accuracy.
  Solution: implement robust preprocessing in the pipeline (filtering invalid rows, handling negative quantities, etc.).
- **Changing trends and demand patterns:** Fashion, electronics or niche products may have very short life cycles.
  Solution: retrain the model frequently and incorporate more features such as seasonality or marketing campaigns.
- **Trust in AI decisions:** Business users may hesitate to rely on model outputs.
  Solution: use interpretable models and explainability tools; provide clear visualizations and simple action labels.

Conclusion & Future Work

Conclusion

This project shows that dead stock risk prediction can be effectively approached using relatively simple but well-designed AI techniques. By defining a clear business rule engineering meaningful features in Google Colab, and training a Decision Tree Classifier, it is possible to create a practical tool that:

- Quantifies dead stock risk as a probability,
- Ranks products by urgency, and
- Provides concrete, human-readable recommendations for each SKU.

From a personal perspective, working on this project alone allowed me to understand both the technical side and the business side.

Future Work

Future improvements could focus on expanding the system beyond historical sales data.
By incorporating store-specific demand trends, the model could detect early

declines that differ across locations or product categories. Additionally, integrating competitor pricing and market trend data would allow the system to recognize industry-wide drops in product interest and adjust risk scores more accurately.

A more advanced version could also connect directly to procurement and warehouse systems, automatically reducing purchase orders or production levels when dead stock risk increases. Finally, building a real-time dashboard with dynamic pricing or promotion suggestions would transform the solution from a predictive tool into a fully automated inventory optimization system.

References

[1] Kaggle – E-Commerce Data Source
Kaggle. *E-Commerce Transactions Dataset*. Retrieved from:
https://www.kaggle.com/
(Accessed: 2025)

[2] Google Colab – Development Environment
Google. *Google Colaboratory*. Retrieved from:
https://colab.research.google.com/
(Accessed: 2025)

[3] draw.io – System Architecture Diagram Tool
diagrams.net. *draw.io Diagramming Application*. Retrieved from:
https://www.draw.io/
(Accessed: 2025)

[4] Canva – Dashboard & Mockup Design Tool
Canva. *Visual Dashboard and UI Mockup Design Platform*. Retrieved from:
https://www.canva.com/
(Accessed: 2025)