ISSN: 2473-4901
v12 n6676

SmartCart AI: A Decision-Centric Agentic System for Inventory Waste Reduction and Pricing Optimization

Megha Simha
narendrasimhamegha@cityuniversity.edu

Sivakumar Visweswaran
visweswaransivakuma@cityu.edu

Sam Chung
chungsam@cityu.edu

City University of Seattle
Seattle, WA, 98121, USA

Abstract

Food waste in the e-Commerce supply chain is a major efficiency problem that requires immediate attention to prevent waste and economic losses. While current AI approaches in inventory management primarily rely on forecasting, they fall short of providing prescriptive strategies for application within the inventory. Human interaction within the inventory management system adds to the inefficiencies. This paper introduces SmartCart AI, an agent-based inventory intelligence framework powered by open-source Large Language Models (LLMs) such as Mistral, for inventory management, capable of automatically identifying waste-inventory items without human data entry to support prescriptive strategies. SmartCart AI is based on the implementation of the two-agent model, named Inventory Agent, which analyzes inventory status for constant scoring of waste, based on the transactional and temporal attributes of the inventory, while the other is the Decision Orchestrator with sub agents, which treats the entire inventory management scenario like the unfolding of a decision sequence, where the prescriptive strategies include dynamic price adjustments, discounting, bundling, rerouting, or liquidating the inventory. SmartCart AI combines rules and decision optimization with context, leveraging LangChain to enable immediate, waste-conscious interventions within the inventory without waiting for the forecast.

Keywords: Agentic AI, Prescriptive Inventory Management, Sequential Decision Making, Food Waste Reduction, Sustainable E-Commerce, Price Optimization

ISSN: 2473-4901
v12 n6676

SmartCart AI: A Decision-Centric Agentic System for Inventory Waste Reduction and Pricing Optimization

Megha Simha, Sivakumar Visweswaran and Sam Chung

1. INTRODUCTION

Food and product waste has become a critical challenge in e-commerce and retail supply chains, particularly for perishable and high-turnover products. Expiration and spoilage directly contribute to economic and environmental losses.

Although artificial intelligence (AI) has found broad use to enhance inventory efficiency, available solutions mostly focus on predictive goals such as demand forecasting, inventory balancing, and overstock reduction. The vast majority of inventory optimization methods based on AI concentrate on forecasting and not on how to transform the forecast into an action that will prevent the reduction of profit from product spoilage. Furthermore, such systems do not normally incorporate any pricing mechanism that will enable the avoidance of possible waste through adjusting the price based on the approaching expiry of the inventory. Further technical surveys on AI use in the retail industry show that, although machine learning and historical data analysis have increased forecasting accuracy and planning efficiency, their impact on waste elimination is limited because they are not proactive and decision-centered (Bharti, 2025; Ankam, 2025). This gap could be closed by integrating dynamic pricing into inventory management, so systems cannot only predict risk but also act now and make revenue-focused decisions that minimize waste. These restrictions highlight the significance of inventory systems that are no longer predicted, but are directly used to facilitate timely, waste-conscious operational action. Recent reviews also affirm that there is a wide gap that has existed in the translation of predictive insights into autonomous, sustainability-focused decisions in the systems of the supply chain (Samuels, 2025)

The SmartCart AI is inspired by both practice and industry problems. During the development of an inventory management system in a healthcare environment, it became clear that delays in decision-making, or manual decision-making, result in substantial resource waste. The same can be directly applied to retail and e-commerce supply chains, where existing AI systems primarily focus on demand forecasting but do not prescribe real-time operational activities. Additionally, the current systems hardly use dynamic pricing as an approach to shield the revenue from risk inventory; the price must be set based on the life span of the product, the demand trends, and the amount of inventory. This aligns with recent work by Zhou et al. (2024), which demonstrates that multi-agent reinforcement learning can jointly optimize pricing and inventory decisions in dynamic environments, outperforming traditional forecast-driven approaches. Since global food waste is estimated to reach roughly 540 billion by 2026 (Avery Dennison, n.d.), there is an obvious need to shift to predictive analytics and autonomous, decision-centric AI-based systems. The gap SmartCart AI aims to address is applying AI knowledge to real-time, waste-conscious interventions to improve operational efficiency and sustainability.

AI-based inventory forecasting has advanced, but e-commerce and retail systems still face a major challenge: they lack autonomous tools to recommend timely, waste-reducing actions for at-risk inventory. While current systems can predict potential risks using past data and trends, they do not decide which operational actions—such as dynamic discounting, bundling, price optimization, rerouting, or liquidation—should be taken to minimize waste. As a result, most intervention decisions are made manually, slowly, or rely on inflexible rules that do not reflect real-time inventory conditions. This leads to unnecessary waste and higher operational costs. Research shows that predictive analytics can improve inventory accuracy, but without decision-oriented intelligence, these tools are not effective at reducing waste (Bhavikatta, 2025; Tiwari & Kushwah, 2025; Ankam, 2025). The core challenge is to develop an autonomous system that continuously assesses inventory risk and prescribes timely, revenue-focused, and sustainable actions—directly turning predictions into outcomes that lower waste.

 ISSN: 2473-4901
v12 n6676

SmartCart AI addresses the challenge by adopting an agent-based, action-oriented approach to inventory intelligence, shifting inventory management from predictive analytics to prescriptive, real-time decision-making (Aylak, 2025). The system reimagines inventory control as a sequence of automated, low-friction actions that operate continuously, minimizing both cognitive and operational burdens on human managers. By integrating intelligent decision logic directly into daily workflows, SmartCart AI empowers organizations to intervene promptly and efficiently, reducing reliance on manual oversight and enabling data-driven actions precisely when needed. The system features a multi-agent architecture designed for intelligent, responsive inventory management. \This research demonstrates that SmartCart AI can effectively reduce food and product waste in e-commerce inventory systems by shifting from prediction-based analytics to action-oriented, prescriptive analytics. The system leverages lightweight, interpretable agent-based design to efficiently deliver substantial sustainability benefits, separating inventory monitoring from decision-making. Ultimately, SmartCart AI provides a scalable, autonomous, and practical solution for minimizing waste in modern e-commerce environments.

2. BACKGROUND

The retail industry is undergoing a paradigm shift, where the traditional and rigid concepts of inventory, namely Economic Order Quantity (EOQ) and Just-in-Time (JIT), are being replaced by the proactive and AI-based models that function on the principles of machine learning, deep learning (LSTM), and reinforcement learning to process large multi-variable datasets. More recently, these AI-driven systems are increasingly designed to be model-agnostic, consuming forward-looking demand and risk signals rather than relying on a single fixed predictive model. Though traditional methods often involve errors of 30-40 percent, owing to the use of limited past data and manual processing, AI-based models involve 50-100 external variables, such as weather patterns, social media sentiments, and competitive pricing, to reduce the predictive error to at most 10-15 percent and predictive accuracy to 98 percent. These technology platforms have offered tangible benefits to the supply chain, such as reducing inventory holding costs by 18-27 percent, increasing revenue by up to 19 percent, and reducing safety stock by 40 percent without impairing service levels, which remain over 95 percent. However, the successful implementation of this technology remains an arduous task, often taking 6-18 months and involving major hurdles such as data quality management, integration with legacy systems, and organizational change management.

The SmartCart AI model is based on the principle of bounded rationality (Simon, 1955) according to which individuals, making decisions in conditions of limited time and lack of information, satisfice—choose a satisfactory course of action—instead of trying to optimize their actions over all potential outcomes. In the case of inventory managers, who make real-time interventions to prevent waste (discount, bundle, donate, or dispose of excess inventory) lack of complete information on downstream demand, price elasticity, and spoilage timing occurs at the very moment of decision making. Instead of viewing it as a risk to overcome, the SmartCart AI project looks at the situation as a bounded rationality issue, incorporating the satisficing decision rules into the pipeline of the agent, such that a timely satisfactory decision is made by the system instead of a globally optimal one that comes too late and allows spoilage to happen.

3. RELATED WORK

There are a lot of academic studies on artificial intelligence in inventory management, and most of them are about prediction. For instance, Ankam (2025), Bharti (2025), Bhavikatta (2025), Tiwari and Kushwah (2025), and Samuels (2025) all found that machine learning and deep learning techniques perform better than conventional predictive approaches in estimating demand, predicting stockouts, and detecting overstock. Nevertheless, they all consider inventory intelligence to be an information production task: the system produces a prediction or a score, and it is the decision maker who needs to do something with it.

A different but growing literature focuses on prescriptive analytics, where the analytics not only predict but decide or automate the action. Prescriptive analytics was defined by Bertsimas & Kallus (2020) as shifting from predictive analytics (data to forecast to decision) to a new data-driven decision making methodology (data to decision). Regarding the supply chain, Smyth et al. (2024) did a systematic review of 76 studies concerning AI-driven prescriptive analytics for supply chain resilience and showed that although the opportunity has been extensively demonstrated, the actual prescriptive capabilities have

 ISSN: 2473-4901
v12 n6676

remained scattered and are mostly in the domains of disruption response and risk management and not in the domain of regular inventory operations. In perishable inventory, Bu et al. (2023) found optimal base-stock replenishment policies for perishable inventory, while Atan et al. (2025) analyzed the impact of discounts and display policies on profit and waste of retailers; both are examples of prescriptive analytics (producing optimal policies) in the classical operations research context but do not involve decision embedded within an automated system.

Criteria	**(Bharti, 2025)**	**(Ankam, 2025)**	**(Bhavikatta, 2025)**	**SmartCart AI**
Primary Focus	AI-driven retail inventory optimization	AI-based demand forecasting	Survey of AI methods for stockout and overstock mitigation	Waste-aware autonomous inventory intervention
Core AI Paradigm	Predictive analytics and optimization models	Forecasting models using historical data	Predictive and analytical review	Agentic, decision-centric AI
Decision-Making Capability	Rule-based or manual post-analysis	Not addressed	Identifies lack of prescriptive actions	Lightweight demand forecasting (simple moving average/naive baseline) for short-term estimation, combined with model-agnostic LLM-based agentic decision orchestration and explanation via Model Context Protocol (MCP)
AI Agents	No	No	No	Yes
Manual Intervention Required	Yes	Yes	NA	No
Sustainability Emphasis	Secondary	Minimal	Conceptual	Primary
Model / Algorithm Used	Traditional Machine Learning (ML) (regression, boosting), optimization	Rule-based + classical forecasting (Autoregressive Integrated Moving Average (ARIMA), moving averages)	Statistical & ML methods (regression, ensembles)	Model-agnostic forecasting via MCP + LLM for decision orchestration and explanation
Price Optimization Capability	Limited / Not addressed	Minimal / Not addressed	Not addressed	Autonomous price and promotion recommendations integrated with inventory decisions
Donation included?	No	No	No	Yes, select nearby Food donation centers

Table 1. Comparison of Existing AI Methods and SmartCart AI

ISSN: 2473-4901
v12 n6676

On a path closer to our proposal, Aylak (2025) puts forward a framework of agentic AI for sustainable and cost-effective automation of the supply chain processes, while Zhou et al. (2024) propose deep multi-agent reinforcement learning for joint pricing and inventory control. They are state-of-the-art examples of automated decision making in this field but do not cover the issue of optimizing the route based on the waste generated and do not incorporate lightweight forecasting combined with LLM-powered agentic decision-making as our solution does.

Table 1 shows how our proposal positions within the literature review results. In general, the current state of the research indicates that prescriptive analytics of the inventory is actively developed area and that the prescriptive solutions can be divided into (a) classical optimization that produces a static policy or (b) reinforcement learning/agentic automation that focuses on pricing and costs. It is what our proposed solution, SmartCart AI, aims to address.

Outside the academic discussion, it may be beneficial to analyze how SmartCart AI can be positioned relative to commercial practice. Inventory and warehouse management solutions commonly employed in mid-sized grocery retail business are mostly based on forecasting, reorder point reminders, and reporting; the decision regarding what action should be taken with regard to a certain at-risk product (to reduce its price, bundle it, make a donation of the item, or discard it) is left for a human manager to take in a manual fashion, outside of the inventory system. In the case of donation, managers commonly use an additional process of identifying products that are close to expiration date and contacting the food bank or entering data about such items into a dedicated tool for donation tracking, without a direct connection of an inventory record and its destination via automation.

4. SMARTCART AI

SmartCart AI represents a major advance in inventory management by shifting from prediction-only approaches to decision-based processes. Instead of just improving forecast accuracy, SmartCart AI focuses on making timely, waste-conscious decisions when inventory is at risk. This approach enables the system to choose the best actions based on operational and cost constraints, closing the intervention gap identified in earlier research. By embedding decision logic directly into the platform, SmartCart AI removes the need for human interpretation and manual responses.

SmartCart AI is an agent-based platform for inventory management that focuses on waste reduction, not just demand forecasting (see Figure 1). It uses autonomous agents that constantly track inventory, spot risks, and recommend actions with little need for human input. The main goal is to keep inventory running smoothly while cutting waste, solving problems early, and reducing costs.

SmartCart AI works with detailed retail data, such as inventory levels, past usage, thresholds, product details, and costs. It also adds information such as perishability, expiry dates, and waste factors to better reflect real-world food and grocery inventory. This helps the system assess inventory by both quantity and the likelihood it will spoil or go to waste, creating more realistic scenarios.

SmartCart AI is an agent-based inventory intelligence system, shown in Figure 2. It includes Python agents (Chat, Inventory Monitoring, Decision Orchestrator with three sub-agents), MCP layer, Java backend microservices, a PostgreSQL database, a web manager dashboard, and a Mistral language model for recommendations and explanations. This autonomous system uses RAG to support inventory decisions.

Agents Architecture

The system follows hybrid hierarchical multi-agent architecture (Figure 3) designed for modular, scalable, and explainable inventory interventions.

The Chat Agent serves as the user-facing entry point and handles user requests, intent parsing, and retrieval of relevant item data (via the Inventory Agent’s query interface). In the current implementation, the Java backend proxies’ requests to the Chat Agent and Decision Orchestrator primarily through MCP endpoints (Chat MCP on 9106, Orchestrator MCP on 9100), while other supporting services remain standard HTTP microservices.

ISSN: 2473-4901
v12 n6676

The Decision Orchestrator serves as the central coordinator that combines outputs from specialized subagents—Risk, Feasibility/Cost Impact, Food Bank, and Explanation—each responsible for a specific decision dimension. These subagents run as separate Flask REST services (e.g., Risk 9004, Feasibility+Cost 9002, Explanation 9003, Food Bank 9007), enabling separation of concerns and allowing individual components to be updated or scaled without redesigning the entire system.

Orchestration Framework
Orchestration is implemented using a LangGraph StateGraph pipeline to coordinate decision stages in a deterministic, traceable workflow. The orchestrator executes the sequence Risk → Feasibility/Cost → (optional) Food Bank → Synthesis → Explanation, passing a shared state object through each stage and updating it with intermediate outputs. Conditional execution is supported; for example, the Food Bank agent is only called when the selected action is donation.

Operationally, the orchestrator itself is invoked via the MCP tool interface (from the backend), while it communicates with the Flask sub-agents using HTTP POST calls with timeouts and fallbacks. This hybrid orchestration approach maintains robustness in a distributed setting while keeping the decision trail reproducible and explainable.

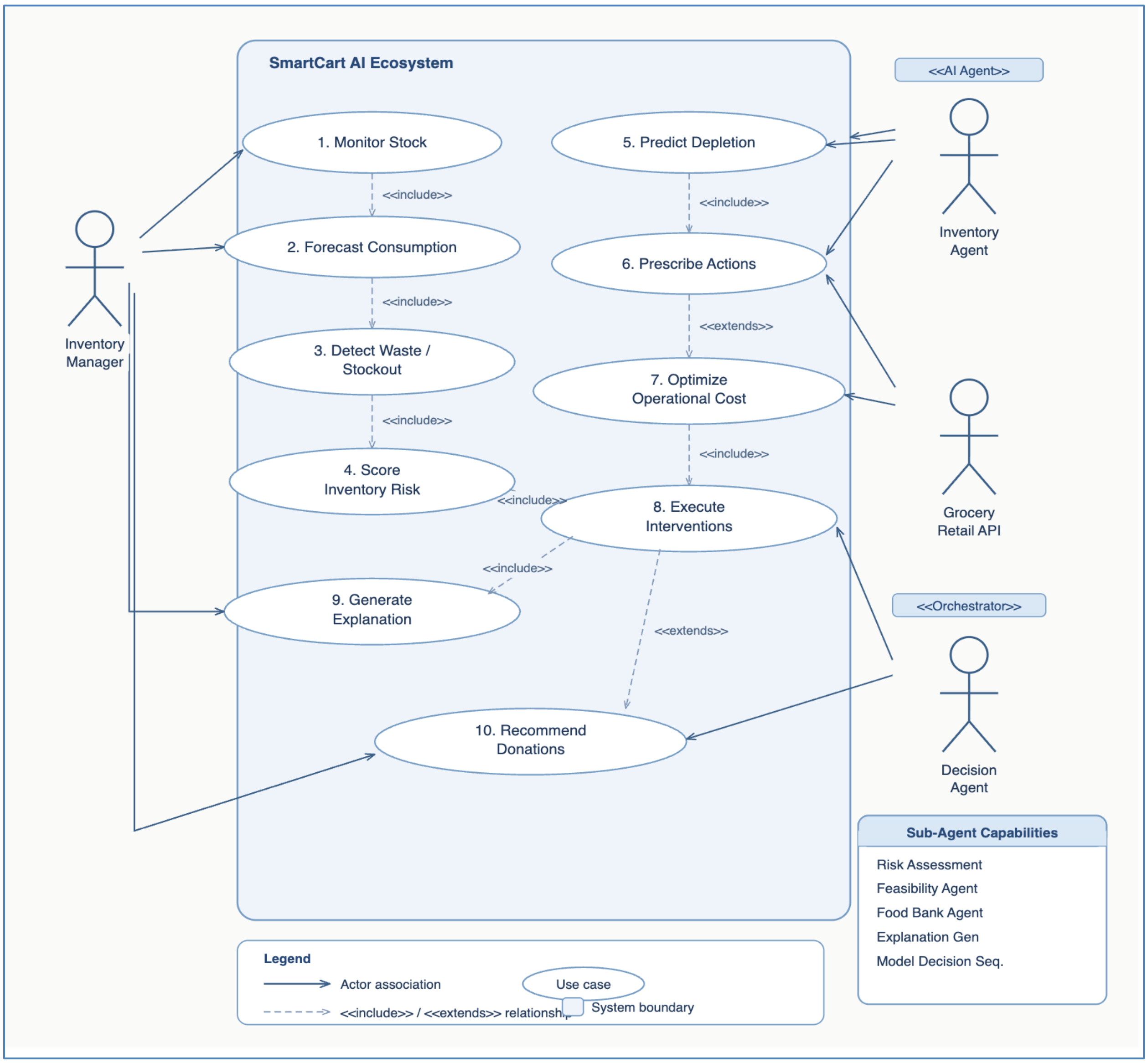

Figure 1. SmartCart AI Use Case Diagram

 ISSN: 2473-4901
v12 n6676

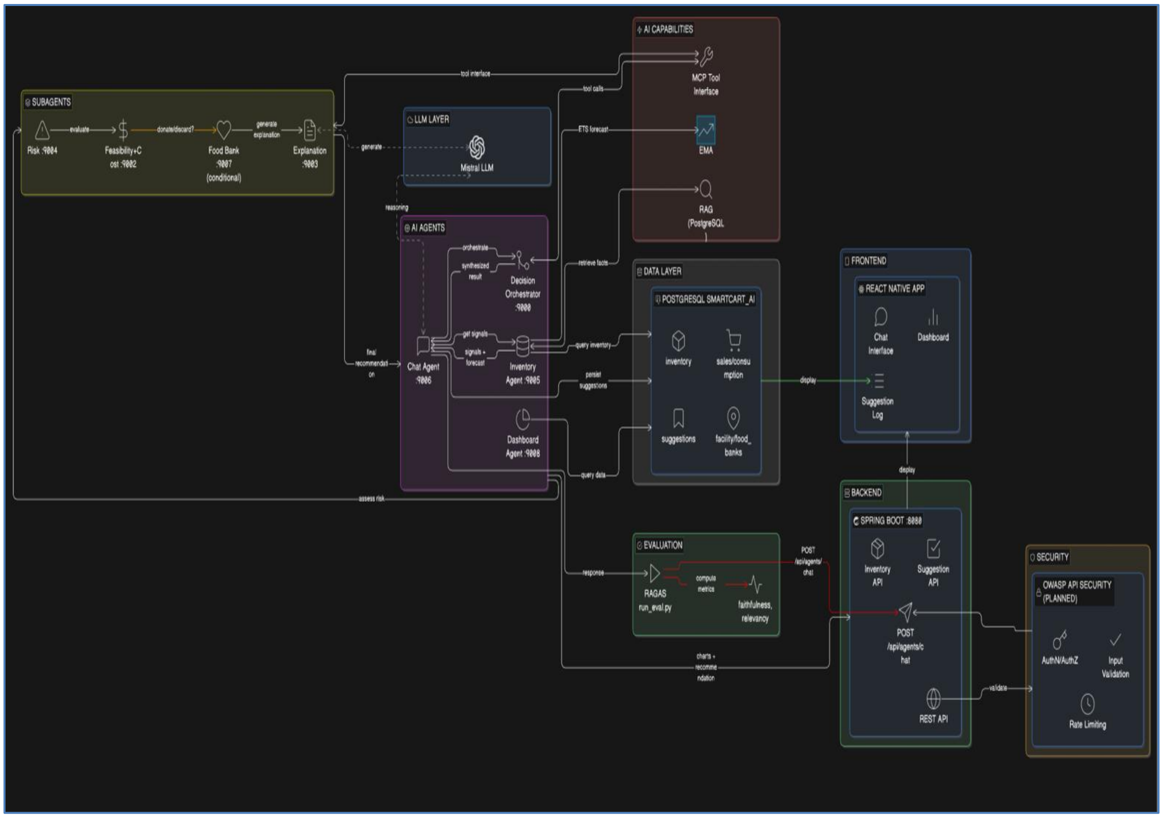

Figure 2. System Architecture for the SmartCart AI System

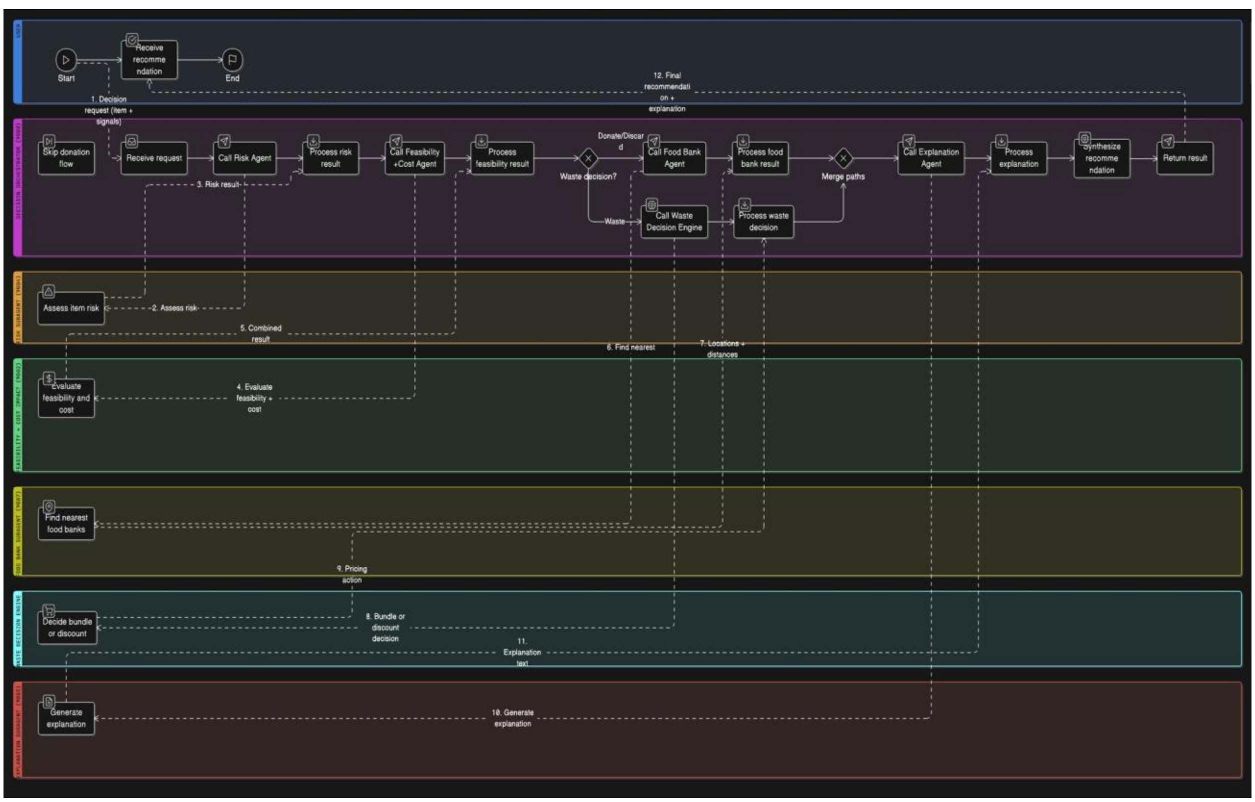

Figure 3. Multi Agent Architecture

ISSN: 2473-4901
v12 n6676

Prompt Engineering

The system uses structured, context-aware prompt templates to guide the Mistral LLM toward evidence-based recommendations. Prompts are assembled dynamically from item attributes, stock thresholds, baseline demand signals derived from consumption history, and outputs from the risk/feasibility/cost stages. Rather than a single generic prompt, each stage uses role-specific prompts with explicit constraints and output formats (e.g., risk assessment, synthesis, explanation).

This prompt structure reduces hallucination, improves consistency, and ensures the final recommendation remains aligned with measurable signals and policy constraints. Deterministic rules (e.g., expiry overrides, min-stock thresholds) are used alongside LLM reasoning to keep output stable and auditable.

Context Engineering and RAG

The system applies RAG to ground LLM responses in real database evidence. Instead of relying only on the immediate agent state, the pipeline retrieves relevant historical records (inventory history, recent consumption, and sales signals) to provide richer context for decision-making and explanations. Where applicable, similarity-based retrieval (via embeddings) is used to surface comparable past situations and related items.

By injecting this retrieved context into LLM prompts, the system improves answer relevance, reduces hallucinations, and produces more defensible recommendations. This design allows the system to combine structured numeric signals (stock/expiry/demand baselines) with language-model reasoning in a single evidence-driven decision flow.

Tool-Based Agent Integration (MCP Framework)

The platform uses MCP to standardize how core agent capabilities are invoked from the backend in a model-agnostic way. In the current implementation, only the Chat Agent and the Decision Orchestrator are exposed as FastMCP tools over Streamable HTTP, enabling the Java backend to call them through a consistent tool interface. Other agents and subagents (Inventory, Risk, Feasibility+Cost, Explanation, Food Bank, Dashboard) remain Flask REST services and are invoked via conventional HTTP endpoints.

This mixed integration approach provides the benefits of MCP (clean tool contracts and backend-controlled routing) while retaining simple REST microservices where appropriate. The result is a modular architecture in which MCP handles the top-level "tooling" boundary, while internal specialized services remain lightweight and independently deployable.

Demand Forecasting

The demand forecast for SmartCart AI is obtained via a shared forecasting utility, ensuring that all agents (Inventory Agent, Chat Agent, Decision Orchestrator) use the same demand estimate. Demand forecasting uses lightweight baseline techniques: by default, an Exponential Moving Average (EMA) that balances smoothness with recency, with an optional Naive last-value baseline for comparison against last week's PostgreSQL consumption history to estimate the day's demand rate and project demand for the following week.

These forecasts are injected into the agent state and prompt context for decision-making, which can be measurable, such as the urgency for reordering ("stock," "min_stock," "forecast"), stock-out risk, and action prioritization. This approach ensures the overall decision-making process is consistent and explainable, leveraging the strengths of rule/agent-based decision-making while providing the accuracy and reliability of statistical decision-making.

Security

SmartCart AI sets up the server boundary to secure traffic before it reaches the agents. The backend adds strong security headers to all responses—including Content Security Policy, same-origin, frame-denial, restricted permissions, and no-referrer-leakage—to prevent browsers from running or embedding malicious content. Client-side controls also limit the Application Programming Interface (API) requests, block spam and brute-force attacks, and allow only approved, rate-limited traffic through to the agents.

ISSN: 2473-4901
v12 n6676

Authentication uses JSON Web Tokens (JWTs) with an expiration time and a customizable secret. Each secured request must include a valid bearer token before reaching the MCP or Flask service. Health and token endpoints can be monitored, but all other operations block requests with missing, invalid, or expired tokens. These controls—secure headers, rate limiting, and token validation—ensure that only authorized, rate-limited clients can access the agents.

5. DATA COLLECTION

The process began by finding high-quality retail and supply chain datasets on Kaggle (*Kaggle Grocery Dataset, n.d.)*, including the Retail Store Inventory Forecasting Dataset. WE conducted an initial exploration analysis on the raw CSV files, focusing on Grocery Inventory, Sales, and Retail Store Inventory. These files contained key information, including product IDs, categories, and inventory levels. However, the spreadsheets were denormalized and included sales data that changed over time.

To improve the SmartCart AI system, we created extra data to expand its capabilities. We added more records to reflect higher usage and included new columns for advanced analysis, such as item forms ("powdered" or "liquid"), use cases, and payment status. We also introduced new tables, like consumption and demand, to separate internal system use and compare it with predictive model results and standard sales transactions.

We moved the cleaned data into a PostgreSQL database and created a relational schema from the flat CSV files. Next, we defined how the tables are connected to ensure smooth data flow. We used "inventory_id" to link the master inventory table with the sales, consumption, and demand tables in a One-to-Many relationship. This structure means that any update to product information automatically appears in all related sales and prediction records, supporting real-time inventory tracking and demand forecasting.

We also used a public government dataset, Food Resources in California (LA City Controller), to help with donation workflows. This dataset included food bank names, addresses, and geographic coordinates, which filled the "food_banks" table. First, "vendor_id" was a bare code in the inventory and sales tables without a lookup entity and has now been supported by a vendor reference table through the foreign key. Second, the nearest food bank matches for donation suggestions had been stored in a JSON array format in one TEXT field ("donation_info") that could not be queried, joined, nor indexed and repeated "food_banks.name/food_banks.address" as a denormalized snapshot - an anti-pattern.

To account for the fact that the sell-by, use-by, and best-by have different meaning, the schema contains the attribute "expiry_date_type" which separates the date type into food safety ("use_by"), stock ("sell_by"), and quality ("best_by") only, and is displayed together with the "expiry_date" by the chatbot and dashboard. Figure 4 shows the Entity-Relationship (ER) diagram for SmartCart AI Database structure, including how tables are related and how primary keys are assigned.

6. DATA ANALYSIS

6.1 Source Datasets

We start with two public Kaggle datasets and the official Open Data Portal for the Office of the Los Angeles City Controller (Control Panel L.A.), shown in Table 2. These datasets define the domain and structure. The data is then synthesized and aligned to a common schema.

6.2 Synthesis

Synthesis here means using the Kaggle data as a reference to generate a coherent, schema-aligned dataset (e.g., via scripts like Dataset/createdataset.py) that matches SmartCart AI's database and agents, as seen in Table 3.

ISSN: 2473-4901
v12 n6676

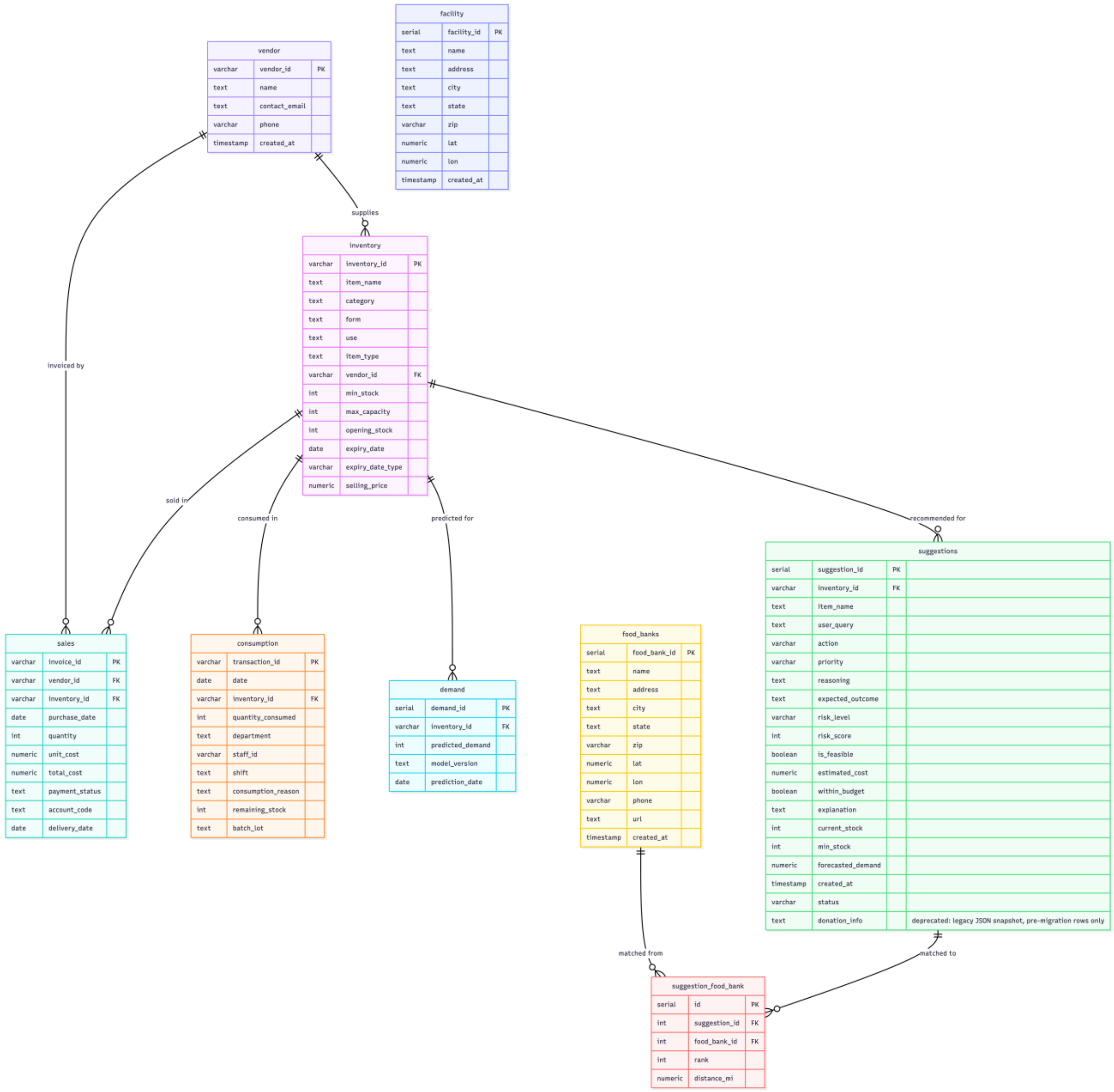

Figure 4. The Entity Relationship Diagram for SmartCart AI Database

ISSN: 2473-4901
v12 n6676

Dataset Name	Author / Publisher	URL	Typical Content	Role in Project
Grocery Inventory and Sales Dataset	Salahuddin Ahmed Shuvo (Kaggle)	Grocery Inventory and Sales Dataset	Grocery-focused inventory and sales records, including product names, categories, stock levels, sales transactions, vendors, and dates.	Serves as the basis for the product catalog, product categories, vendor relationships, and sales transaction data such as quantities, dates, and identifiers.
Retail Store Inventory Forecasting Dataset	Anirudh Chauhan (Kaggle)	Retail Store Inventory Forecasting Dataset	Retail inventory and time-series data suitable for forecasting, including stock levels, demand metrics, and dates.	Provides inventory levels, reorder logic, demand patterns, consumption trends, and forecasting data.
Food Resources in California Dataset	LA City Controller (Control Panel L.A.)	Food Resources in California Dataset	Information on food banks and food resources, including organization names, addresses, geographic coordinates (latitude/longitude), and resource types.	Used to populate the "food_banks" table and enable nearest-food-bank logic for donation decisions via geographic distance calculations.

Table 2. Summary of Datasets Utilized for Data Generation and Analysis

Component	Description
Scope	Data is reduced and standardized to a fixed data set (e.g., 50 product categories) and a defined time range (e.g., one year) to simplify pipeline execution and agent testing. Each category has unit stock.
Inventory	Includes Product IDs, names, categories (Bakery, Dairy, Produce, Beverage, Snack, Frozen), form (Solid/Liquid/Pack), usage (Food/Non-Food), item type (Perishable/Non-Perishable), vendor IDs, and min/max/opening stock levels.
Sales	Contains Invoice IDs, vendor and inventory IDs, purchase/delivery dates, quantity, unit cost, total cost, payment status, account codes, and cost-impact fields aligned with a sales/purchase accounting view.
Consumption	Tracks transaction IDs, dates, inventory IDs, quantity consumed, department, staff, shift, reason (Routine/Spoilage/Sample), remaining stock, batch/lot information for waste and usage analysis.
Generation	Data generation scripts (Python with Pandas/NumPy) use Kaggle schemas and seeded randomness to create daily sales, consumption, restocking events, and cost fields, producing CSVs ready for PostgreSQL ingestion.
Geospatial Alignment	Store and food bank latitude/longitude are aligned to enable nearest-food-bank selection using Haversine distance calculations.
Demand Table Generation	Predicted demand values are generated or derived from consumption trends and stored in a dedicated demand table for forecasting and risk analysis.
Embedding Preparation	Inventory records are enriched with text fields (item_name, category, form, usage, item_type) to support embedding generation for similarity-based AI reasoning.
Schema Support for AI Outputs	A structured suggestions table stores AI agent outputs including action, priority, risk level, cost impact, explanation, and donation recommendations.

Table 3. Structured Data Components and Generation Pipeline Design

6.3 Load

 ISSN: 2473-4901
v12 n6676

Table 4 summarizes the process of loading inventory, sales, and consumption data into PostgreSQL and how the loaded data is utilized by the backend services and AI agents for analytics and decision-making.

Component	Description
Loading Mechanism	Data is loaded into PostgreSQL using either the server-side COPY command or the client-side \copy command (e.g., from psql).
Source Files	Data is imported from files such as Dataset/inventory_master_50_unique.csv, Dataset/sales_50.csv, and Dataset/consumption_50.csv.
Reference Documentation	Example commands are provided in database/schema.sql (commented sections), Dataset/Copydata.txt, and AGENTS_SETUP.md, which include \copy ... CSV HEADER examples for inventory, sales, and consumption datasets.
Loading Order	Inventory data is loaded first since it contains no foreign key dependencies. Sales and consumption tables are loaded afterward because they reference inventory.inventory_id.
Database Relationships	Sales and consumption records maintain referential integrity through foreign key relationships with the inventory table.
Backend Integration	After data loading, the Java backend accesses the inventory, sales, and consumption tables for application operations and API services.
AI Agent Integration	Python-based AI agents use the loaded data for analytics and decision-making. Examples include the cost-impact agent functions get_average_unit_cost() and assess_cost_impact().

Table 4. Overview of Data Loading and Integration

Figure 5 shows a sample snapshot of the Inventory Master dataset, highlighting key fields such as inventory ID, product name, category, item type, vendor information, and stock levels used by the inventory management system.

inventory_master_50_unique

1	inventory_id	item_name	category	form	usage	item_type	vendor_id	min_stock	max_capacity	opening_stock
2	INV0001	Milk 1L	Beverage	Pack	Food	Non-Perishable	V009	39	201	77
3	INV0002	Cheddar Cheese	Snack	Pack	Food	Perishable	V004	21	453	160
4	INV0003	Yogurt	Bakery	Solid	Food	Perishable	V009	25	339	241
5	INV0004	Banana	Snack	Pack	Food	Non-Perishable	V003	41	236	274
6	INV0005	Apple	Snack	Pack	Non-Food	Perishable	V007	30	359	246
7	INV0006	Bread Loaf	Produce	Liquid	Food	Non-Perishable	V006	35	208	110
8	INV0007	Bagel	Bakery	Solid	Non-Food	Non-Perishable	V008	35	432	97
9	INV0008	Orange Juice	Bakery	Liquid	Food	Non-Perishable	V009	20	298	196
10	INV0009	Soda Can	Bakery	Liquid	Non-Food	Perishable	V005	28	346	53
11	INV0010	Chocolate Bar	Snack	Liquid	Food	Perishable	V001	47	407	84
12	INV0011	Eggs Dozen	Beverage	Liquid	Food	Perishable	V003	46	330	241

Figure 5. A Snapshot of Inventory Master Dataset

6.4 Visual Analysis

The dashboard in Figure 6 displays 50 items in total. Of these, 46 are in stock, which means about 92% are available. Two items are low in stock and two are out of stock, so around 8% require immediate attention. Inventory is evenly divided between perishable and non-perishable categories. This split highlights different risks: perishables are prone to expiry and waste, while non-perishables carry risks of overstock and tying up working capital.

ISSN: 2473-4901
v12 n6676

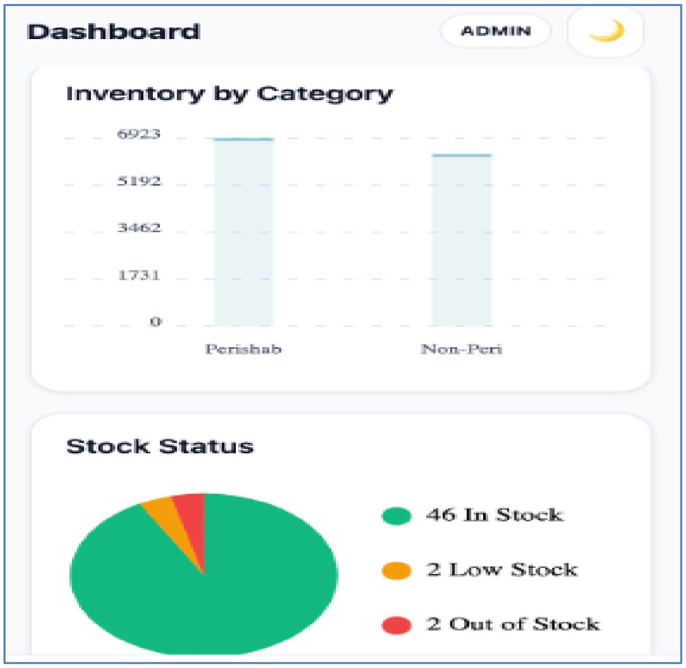

Figure 6. Inventory Distribution by Category and Stock Status

Figure 7 shows an upward 7-day sales trend, indicating rising demand. If reorder habits do not change, low-stock items are at higher risk of running out. These signals point to several clear actions: prioritize reordering low and out-of-stock items, review and increase minimum or safety stock for fast-moving products and adjust perishable orders to keep items fresh while reducing waste.

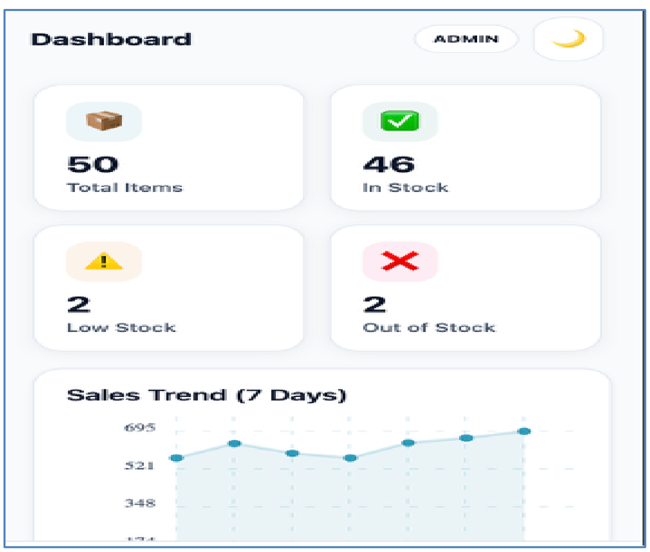

Figure 7. Inventory Status and Sales Analytics

ISSN: 2473-4901
v12 n6676

7. FINDINGS

7.1 Descriptive Statistics

In our application, descriptive statistics, as shown in Table 5, are used to summarize historical inventory and demand data. The use of these statistical summaries forms the basis of forecasting, risk estimation, and decision-making logic for backend agents.

Category	Metric / Calculation	Purpose
Average Unit Cost	AVG(unit_cost) from the latest 10 sales records of an inventory item	Represents recent pricing behavior and serves as the basis for cost-related calculations.
Reorder Cost Estimation	Average Unit Cost × Reorder Quantity	Estimates the expected cost of replenishing inventory.
Profit Margin Percentage	(Selling Price − Avg Unit Cost) / Selling Price × 100	Measures the profitability of an inventory item.
Total Consumption	SUM (quantity_consumed) over the last 7 days	Calculates total product demand during the recent period.
Mean Daily Demand	Total Consumption / 7	Provides a baseline demand estimate for short-term planning.
Total Item Count	COUNT (*)	Determines the total number of inventory items.
Total Stock Quantity	SUM (opening_stock)	Calculates the overall inventory quantity available.
Low-Stock Item Count	COUNT (CASE WHEN opening_stock ≤ min_stock THEN 1 END)	Identifies the number of items below the minimum stock threshold.
Stock Utilization Ratio	Current Stock / Maximum Capacity	Measures inventory utilization relative to storage capacity.
Days to Stockout	Remaining Stock / Average Daily Consumption	Estimates the number of days before stock depletion.
Low-Stock Classification	Stock ≤ Minimum Threshold	Flags inventory items at risk of shortage.
Optimal Stock Classification	Minimum Threshold < Stock < Capacity Threshold	Identifies inventory within the desired operating range.
Overcapacity Classification	Stock ≥ Capacity Threshold	Detects excess inventory levels.
Geographic Distance Analysis	Distance calculated using latitude and longitude coordinates.	Measures the distance between facilities and food banks.
Nearest Donation Center Selection	Minimum calculated distance	Identifies the closest food bank for donation recommendations.

Table 5. Descriptive Statistical Measures and Decision Metrics

7.2 Evaluation Metrics

The Retrieval-Augmented Generation Assessment (RAGAS) measures the quality of SmartCart AI's results using four main metrics:

- Faithfulness: Answers rely on retrieved evidence.
- Response Relevancy: Answers directly address the user's query.
- Context Precision: The retrieved context is useful and relevant.
- Context Recall: Retrieved context covers all necessary evidence.

The RAGAS metrics are chosen for SmartCart AI because the system is RAG-based and requires clear, reliable explanations for its decisions. As shown in Figure 8, Faithfulness and Response Relevancy are especially important for evaluating answer quality and building trust in inventory recommendations.

ISSN: 2473-4901
v12 n6676

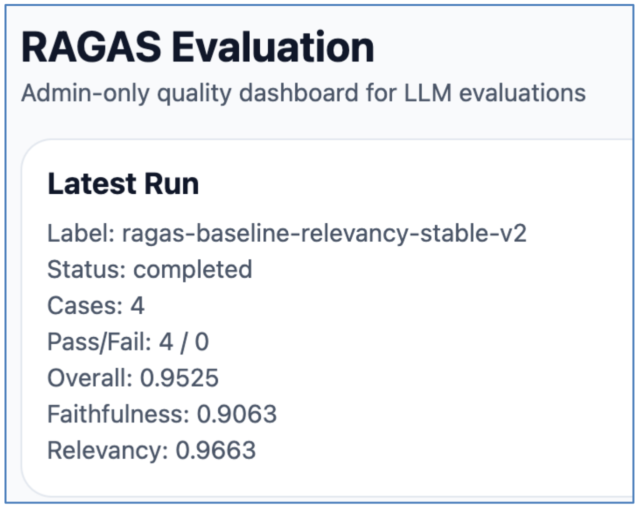

Figure 8. RAGAS Evaluation

The demand forecast evaluation procedure involves the use of basic, short-term models for the baseline: Simple Moving Average (default), and optionally, the Naïve (last value) one. The evaluation is done in terms of error size using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), as well as in relative terms with respect to scale-independent Symmetric Mean Absolute Percentage Error (sMAPE) and Weighted Absolute Percentage Error (WAPE) to make meaningful comparison per item possible. We also check whether the forecast brings important insights regarding operational needs like reorder urgency by evaluating classification results using F1 score, Precision, Recall, and Accuracy illustrated in Figure 9.

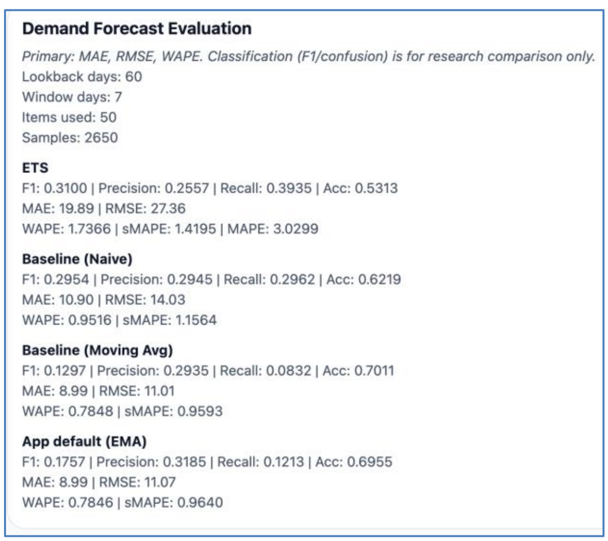

Figure 9. Demand Forecast Evaluation Metrics

Such models were selected since they are quick, robust against sparsity and noise, do not need much tuning, and provide reliable signals that are easy to analyze when making decisions based on the forecast results. It ensures stability of the decision-making process and makes it straightforward unlike models such as ETS (Exponential Smoothing: error, trend, and seasonal components), which is fast and useful for decision-making, but introduces high variance due to its complexity.

ISSN: 2473-4901
v12 n6676

7.3 Results

The following analysis is confined to those items marked with item_type = 'Perishable' (25 out of the 50 catalogued items, 4,062 units), following the principle put forward in the paper that only perishable stock can be prone to any form of waste owing to spoilage. The donation value of the donations is the sum of the selling price × units donated of each of the donated items, and not an assumed rate.

Total 4,062 perishable inventory units have been taken into consideration. On August 10, 2026, there was identification of 1,220 units of potential waste inventory that would spoil after seven days (Figure 10). After using the SmartCart pipeline, 742 units have been identified (60.8% success) while 478 units could not be identified (unresolved waste inventory) (Figure 11).

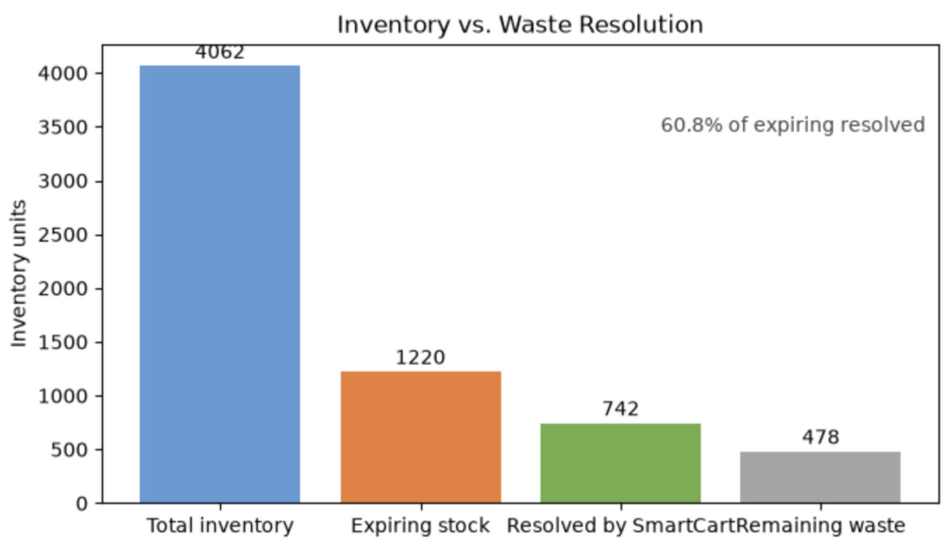

Figure 10. Waste Inventory Identified by SmartCart AI

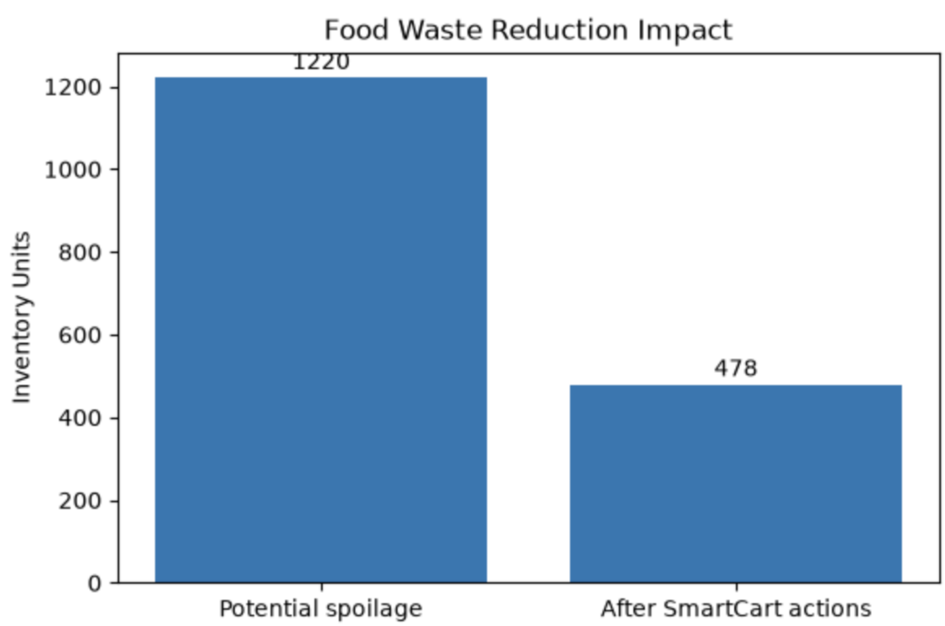

Figure 11. Inventory Resolved After Intervention

ISSN: 2473-4901
v12 n6676

The largest intervention unit was of bundling type, with total recovery of 372 units (30.8% of action share). The discounting of items recovered 335 units (27.7%), which has been identified through the process of predicting the demand for stock in Figure 12.

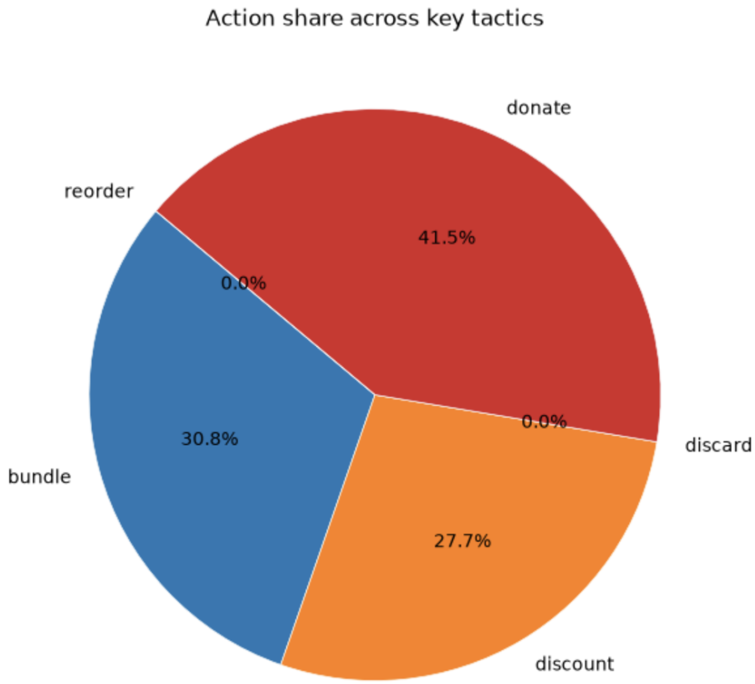

Figure 12. Intervention Effectiveness and Resolution Rate

Social benefit from donations was determined by redistribution of 502 units (41.5% of action share, Figure 13). The dollar benefit is estimated at $1,726.48 (average $3.44/unit, using each item's own selling price), plus $362.56 in tax savings (21% corporate rate), for a combined corporate benefit of $2,089.04. No units required emergency discard in this evaluation window all identified at-risk stock was addressed through discount, bundle, or donation before reaching an unsafe state.

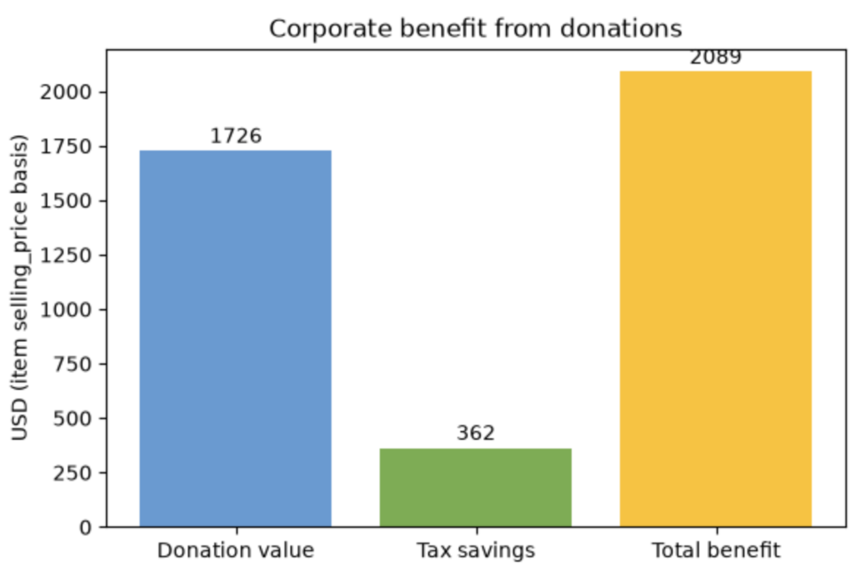

Figure 13. Donation Impact and Redistribution Results

 ISSN: 2473-4901
v12 n6676

As the basis of comparison without the SmartCart solution, the entire 1,220 units marked for expiration in seven days would have been wasted, as there is no other process that intervenes to stop spoilage in this setting. Through the SmartCart AI process, 742 units, representing 60.8%, were handled using the discounting, bundling, and donations methods, resulting in only 478 units being vulnerable to spoilage, which is an improvement by 60.8 percentage points.

8. CONCLUSION

The proposed research confirms that AI-driven agent-based systems can successfully translate demand predictions into real-time inventory actions for mid-size grocery stores. The system enables automated decision-making, including dynamic discounting, bundling, donation, and safe disposal, thereby minimizing potential food waste without incurring economic or social losses.

The findings show that the prioritized execution pipeline can clear most of the expiring inventory, demonstrating how AI-based operational interventions may drive inventory efficiency and waste management outcomes. Besides reducing the amount of spoilage, the system also balances the revenue collection, social contribution through donations, and the protection of the health of the population due to safe disposal.

All in all, the research indicates that predictive analytics combined with an action-oriented AI agent offers a feasible approach through which retailers could handle perishable goods in a more sustainable and effective manner, directly addressing the problem of minimizing food waste without compromising operational and financial results.

Lastly, our backend system will move up the maturity curve with MCP-specific health endpoints, aggregating telemetry and traces, as well as implementing backoff and circuit breakers to handle failure gracefully.

9. FUTURE WORK

The next phase of SmartCart AI development will shift from a reactive tool to a proactive, resilient, and economically intelligent ecosystem.

- Inventory decisions—such as reorder thresholds and food bank transfers versus disposal—should strategically balance profitability with waste reduction objectives.
- Decision sub-agents will be trained using multi-agent reinforcement learning, allowing pricing and intervention strategies to co-evolve in response to dynamic demand and margin environments. This approach will foster coordinated responses, replacing the siloed decisions of individual sub-agents.
- Additional sub-agents will be introduced to manage redistribution, liquidation routing, and multi-channel order fulfillment, unifying our protocol stack and streamlining retries and health checks across the distributed system.
- Future research may explore integrating bulk-breaking strategies, transportation logistics, and dual-source supply chain models to further strengthen the agentic AI decision-making framework.

10. REFERENCES

Ankam, S. (2025). AI-Driven Demand Forecasting in Enterprise Retail Systems: Leveraging Predictive Analytics for Enhanced Supply Chain. *International Journal on Science and Technology*, *16*(1). https://doi.org/10.71097/ijsat.v16.i1.2644

Atan, Z., Honhon, D., & Pan, X. A. (2025). Displaying and discounting perishables: Impact on retail profits and waste. *Management Science* (forthcoming).

Avery Dennison. (n.d.). $540 billion global food waste bill exposed for 2026 | https://www.averydennison.com/en/home/news/press-releases/540-billion-global-food-waste-bill-exposed-for-2026.html

ISSN: 2473-4901
v12 n6676

Aylak, B. L. (2025). SustAI-SCM: Intelligent Supply Chain Process Automation with Agentic AI for Sustainability and Cost Efficiency. Sustainability, 17(6), 2453. https://doi.org/10.3390/su17062453

Bertsimas, D., & Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science*, 66(3), 1025–1044.

Bharti, M. (2025). AI-Driven Retail Optimization: A technical analysis of modern inventory management. *International Journal of Advances in Engineering and Management*, *7*(2), 31–38. https://doi.org/10.35629/5252-07023138

Bhavikatta, N. B. (2025). AI-Driven Inventory Optimization in Supply Chains: A comprehensive review on reducing stockouts and mitigating overstock risks. *Journal of Computer Science and Technology Studies*, *7*(7), 01–13. https://doi.org/10.32996/jcsts.2025.7.7.1

Bu, J., Gong, X., & Chao, X. (2023). Asymptotic optimality of base-stock policies for perishable inventory systems. *Management Science*, 69(2), 846–864.

Food Resources in California | Control Panel LA. (2020, August 10). https://controllerdata.lacity.org/dataset/Food-Resources-in-California/v2mg-qsxf/data_preview

Kaggle Grocery Dataset. (n.d.). https://www.kaggle.com/datasets/salahuddinahmedshuvo/grocery-inventory-and-sales-dataset?resource=download

Retail Store Inventory Forecasting Dataset. (n.d.). https://www.kaggle.com/datasets/anirudhchauhan/retail-store-inventory-forecasting-dataset

Samuels, A. (2025). Examining the integration of artificial intelligence in supply chain management from Industry 4.0 to 6.0: a systematic literature review. *Frontiers in Artificial Intelligence*, *7*, 1477044. https://doi.org/10.3389/frai.2024.1477044

Simon, H. A. (1955). A behavioral model of rational choice. *The Quarterly Journal of Economics*, *69*(1), 99–118. https://doi.org/10.2307/1884852

Smyth, C., Dennehy, D., Fosso Wamba, S., Scott, M., & Harfouche, A. (2024). Artificial intelligence and prescriptive analytics for supply chain resilience: A systematic literature review and research agenda. *International Journal of Production Research*, 62(23), 8537–8561. https://doi.org/10.1080/00207543.2024.2341415

Tiwari, A., & Kushwah, R. (2025). Impact of Artificial Intelligence on Forecasting and Inventory Management. *International Journal for Multidisciplinary Research (IJFMR)*, *7*(3).

Zhou, Q., Dong, C., Yang, Y., & Ngai, W. T. E. (2024). Intelligent Pricing and Inventory Control: A Deep Multi-Agent Reinforcement Learning Algorithm. Available at SSRN 4889719. https://doi.org/10.2139/ssrn.4889717