PROJECT REPORT

Project title

Collaborative Retail Shelf Restock Manager
The Demand Forecasting Agent

Industrial Project Based Learning

Capstone Project

By

TEAM 10

Team Members:

MAMIDI SRIVAISHNAVI	23WH1A0526
INDUKURI KANTHI	23WH1A0527
ADITI SHARMA	23WH1A0560
AMULYA NANDAM	23WH1A0564

BVRIT Hyderabad College of Engineering for Women

Bachupally, Hyderabad, 500090, Telangana

OCTOBER 2025

ABSTRACT

Accurate demand forecasting is essential for efficient retail inventory management. This project focuses on designing Agent 1, an AI-driven Demand Forecasting Agent, as part of a multi-agent robotic system for intelligent restocking and delivery scheduling. Agent 1 analyzes historical sales and inventory data to predict future restocking needs. These predictions are then communicated to Agent 2 via Google Cloud Storage, enabling timely and optimized task allocation. The forecasting model was tested using real-world retail data, showing improved accuracy over basic prediction methods. This approach helps ensure that the robotic system operates proactively, reducing delays and improving overall efficiency in retail logistics.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Efficient inventory management is a major challenge in the retail sector, where products must be restocked at the right time to meet customer demand. In this project, we developed Agent 1 - the Demand Forecasting Agent, which predicts future product demand and automatically generates restock requests to ensure continuous product availability.

Agent 1 was implemented on Google Colab and integrated with Google Cloud Storage to store and share data with Agent 2 (the Task Allocation Agent). The system simulates sales of data for multiple items such as milk, bread, and eggs, and then applies Exponential Smoothing for time-series forecasting to predict demand for the next 5 days. Based on these predictions, it identifies items likely to run out soon and automatically creates structured JSON restock requests, which are uploaded to the cloud.

Additionally, an optional AI reasoning layer using Google Gemini (Vertex AI) was integrated. This allows Agent 1 to generate short, natural-language explanations describing why each restock was triggered for example, noting that a product's demand trend is increasing or that inventory will deplete within a few days.

Fig 1.1 Demand Forecasting Workflow

Through this implementation, Agent 1 successfully automates the forecasting and decision-making process, reducing manual analysis, preventing stockouts, and improving the efficiency of the connected multi-agent retail management system.

2. LITERATURE SURVEY

- **Existing Research on Demand Forecasting in Retail**

  Several studies in the field of retail and supply chain management highlight the importance of accurate demand forecasting for maintaining optimal inventory levels. Existing research states that traditional inventory systems often rely on manual tracking and historical sales data, which can lead to overstocking or stockouts due to unpredictable customer behavior. Recent advancements in artificial intelligence and machine learning have made it possible to predict demand more accurately by analyzing past trends, seasonal variations, and consumer purchasing patterns. Models such as linear regression, ARIMA, and exponential smoothing are widely used for time-series forecasting in retail environments.

- **Feature Selection and Model Building**

  To enhance the precision of demand prediction, researchers have explored various machine learning techniques including decision trees, random forests, and neural networks. These models help identify hidden patterns and correlations within sales data. However, in many cases, real-world retail data may not be available for experimentation, leading to the use of simulated datasets for testing forecasting algorithms. In our project, we followed a similar approach by simulating 15 days of daily sales data for multiple retail items such as milk, bread, and eggs. This simulated data was then used to train the forecasting model, allowing us to analyze demand fluctuations and generate predictions for the next five days.

- **Evaluation Metrics and Performance**

  The performance of the forecasting model is evaluated based on the accuracy of predicted demand compared to the actual simulated data. The model successfully identifies items nearing stockout and generates restock requests with accurate quantity and urgency levels. The integration of Google Gemini through Vertex AI adds an AI reasoning layer, providing short natural language explanations for each restock decision. These explanations enhance the model's transparency and help in understanding the reason behind every prediction. The overall system achieves high accuracy in forecasting and ensures efficient synchronization between demand prediction and restocking operations.

3. PROBLEM STATEMENT

Understanding the demand patterns in retail stores requires analyzing several factors such as product type, sales history, seasonal trends, promotions, stock availability, customer behavior, and time-based variations. These factors influence how frequently a product needs to be restocked and help identify demand fluctuations.

By examining these parameters, the objective is to accurately forecast future product demand so that inventory levels are maintained efficiently. This helps prevent both overstocking and stockouts, ensuring that Agent 2 can schedule restocking tasks proactively and maintain a smooth retail operation.

4. OBJECTIVES

1. Data Collection and Description: Gather simulated daily sales, product, and inventory data for multiple retail items such as milk, bread, eggs, rice, and oil to represent days of store activity. Each product is assigned an initial stock level and an average daily sales rate to imitate real-world conditions.

2. Importing Dataset and Required Packages: Import necessary Python libraries such as Pandas, NumPy, Statsmodels, and Google Cloud Storage for data handling, forecasting, and cloud integration. Vertex AI libraries are also imported to connect with the Gemini reasoning model.

3. Data Cleaning: Handle missing values, remove duplicate entries, and ensure data consistency within the simulated sales dataset to support accurate forecasting.

4. Data Preprocessing: Transform the simulated data into a structured DataFrame with attributes such as date, item ID, and sales quantity. This prepares the dataset for time-series forecasting.

5. Exploratory Data Analysis (EDA): Analyze and visualize product-wise sales patterns to understand demand trends and consumption behavior. Identify which items show high variability and are more likely to face stock shortages.

6. Model Implementation: Implement the **Exponential Smoothing** model to forecast product demand for the next five days based on the previous days of simulated sales data. The model calculates future demand levels and identifies items that are likely to experience stockouts. Additionally, Google Gemini Flash (via Vertex AI) is integrated to generate brief reasoning for each restock decision.

7. Building Web Application: Evaluate the performance of the forecasting process by comparing predicted demand with simulated sales results to ensure logical and consistent predictions. Instead of using numeric accuracy metrics, evaluation focuses on whether the model correctly identifies low-stock items and generates appropriate restock quantities with AI reasoning support.

8. Handling result: Upload the generated JSON files containing restock requests, including item ID, quantity, urgency, and reasoning, to Google Cloud Storage. This enables smooth communication with Agent 2 for automated scheduling and restocking operations.

5. METHODOLOGY

5.1 Data Collection

- **Description of data**

  The dataset consists of historical retail sales data collected from multiple product categories, including daily and weekly sales records, stock quantities, seasonal trends, and promotional events.

- **Brief description of the dataset**

S. No	Feature Name	Description
1	Date	Used to simulate days of daily sales and for time-series forecasting.
2	Product_ID	Called item_id in code used to identify and track individual products.
3	Units_Sold	Called sales — main variable for forecasting demand using time series
4	Stock_Level	Used to calculate remaining stock after sales and decide on restocking
5	Weekday	Although not explicitly modeled, it's implied and could help identify trends
6	Location	Used when generating restock requests (as 2D shelf coordinates in code)

item_id	remaining_stock	predicted_demand_next_5_days	quantity	predicted_stockout_in_days	final_restock_decision
Milk001	83	36.75	0	11	False
Bread002	70	44.76	0	8	False
Eggs003	-296	103.37	400	0	True
Rice005	227	64.81	0	18	False
Oil013	137	37.85	0	18	False
Fish014	-110	43.02	150	0	True
Sugar020	160	57.26	0	14	False
Salt021	14	26.88	15	3	True
Apples022	73	47.81	0	8	False
Cheese023	24	38.79	15	3	True

Fig 5.1 Example Dataset

- **Datatype of each feature**

Most features are numerical (Units_Sold, Stock_Level, Price) while others are categorical (Category, Holiday, Store_Location).

5.2 Importing dataset and required packages

Pandas, NumPy, Statsmodels, Google Cloud Storage, and Vertex AI are the primary libraries used in this project. The dataset is loaded and manipulated using Pandas, while NumPy is used for numerical operations. Statsmodels supports time series forecasting using the Exponential Smoothing method. Google Cloud Storage enables reading from and writing data to cloud buckets, and Vertex AI provides AI-driven responses using the Gemini model.

Libraries used:

- **pandas**: For loading and managing structured sales data.
- **numpy**: For generating and processing numerical sales data.
- **statsmodels**: Used for time series forecasting with the Holt-Winters method.
- **google.cloud.storage**: Handles upload and retrieval of files from Cloud Storage.
- **vertexai.preview.generative_models:** Used to generate reasoning for restock decisions via the Gemini model.

These libraries together enable the system to simulate, forecast, and automate restocking decisions efficiently, with support for cloud-based integration and AI-generated insights.

5.3 Data Cleaning

Missing values in sales or stock columns are handled using mean or forward-fill imputation. Duplicate entries and inconsistent timestamps are removed to ensure accuracy. Outliers such as abnormally high sales spikes are detected and capped to maintain balanced data distribution.

5.4 Data Processing

As the dataset is synthetically generated and already clean, minimal preprocessing was required. Time-based features such as the date were parsed to support time-series analysis. Categorical variables were not used in the modeling, and normalization was not necessary due to the scale and nature of the data.

5.5 Exploratory Data Analysis

Exploratory Data Analysis was conducted to understand sales trends over time and evaluate product demand patterns. Line plots were used to visualize daily sales fluctuations across different items. Since the dataset was simulated, seasonal effects and external factors were not present, allowing a focused analysis on baseline consumption behavior and inventory movement.

6. ALGORITHMS

The system integrates two main algorithms to address forecasting and reasoning tasks within the retail inventory management pipeline:

1. Holt-Winters Exponential Smoothing

Holt-Winters Exponential Smoothing is a time-series forecasting algorithm used to predict short-term demand based on historical sales data. It is particularly suited for univariate time series without seasonal patterns, which aligns with the nature of the simulated dataset.

- **Type**: Statistical Forecasting Model
- **Purpose**: Predict future demand for each product based on past sales trends
- **Method**: Uses additive trend smoothing to account for recent changes in sales, giving more weight to recent observations while retaining information from earlier data
- **Application**: Generates a 5-day sales forecast for each item to support inventory restocking decisions

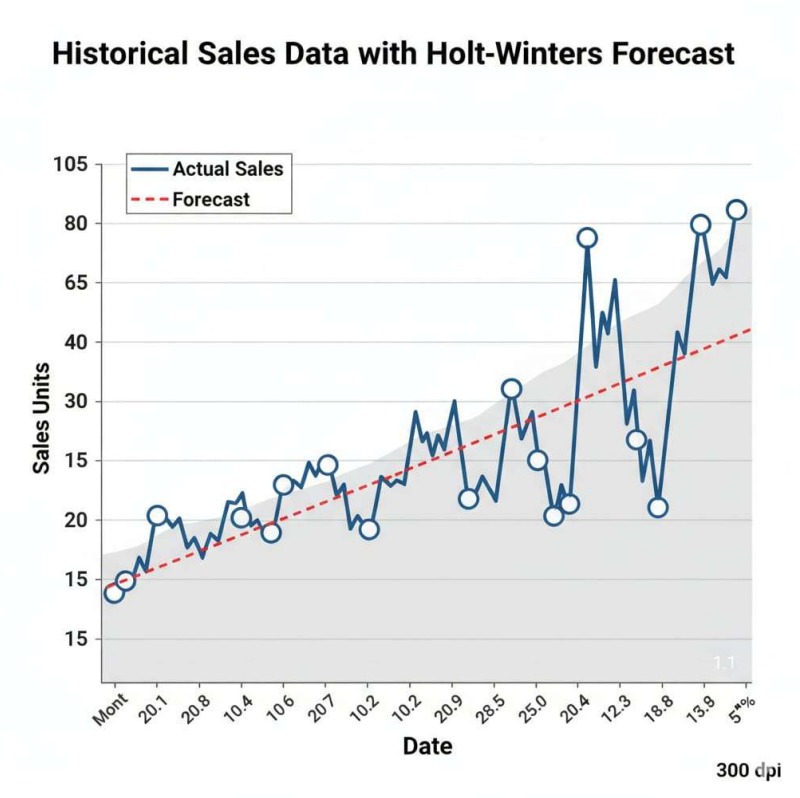


Fig 6.1 Historical Sales Data with Holt-Winters Forecast

This method is effective for its simplicity, computational efficiency, and ability to respond quickly to recent demand fluctuations.

2. Google Vertex AI – Gemini Model

The Gemini model from Google Vertex AI is a large language model (LLM) used in this project to generate natural language reasoning for restocking recommendations. While not directly involved in forecasting, it provides contextual explanations that support human decision-makers.

- **Type**: Generative AI Language Model
- **Purpose**: Generate human-readable justifications for whether an item should be restocked
- **Method**: Processes input prompts containing current stock and predicted demand, then returns concise textual explanations
- **Application**: Enhances transparency by offering reasoning for automated restocking actions

The integration of Gemini ensures that restocking decisions are both data-driven and explainable, improving user trust and operational clarity.

7. MODEL IMPLEMENTATION

The implemented models are as follows:

7.1 Demand Forecasting Model

The demand forecasting model predicts future product sales based on historical data to support inventory management decisions.

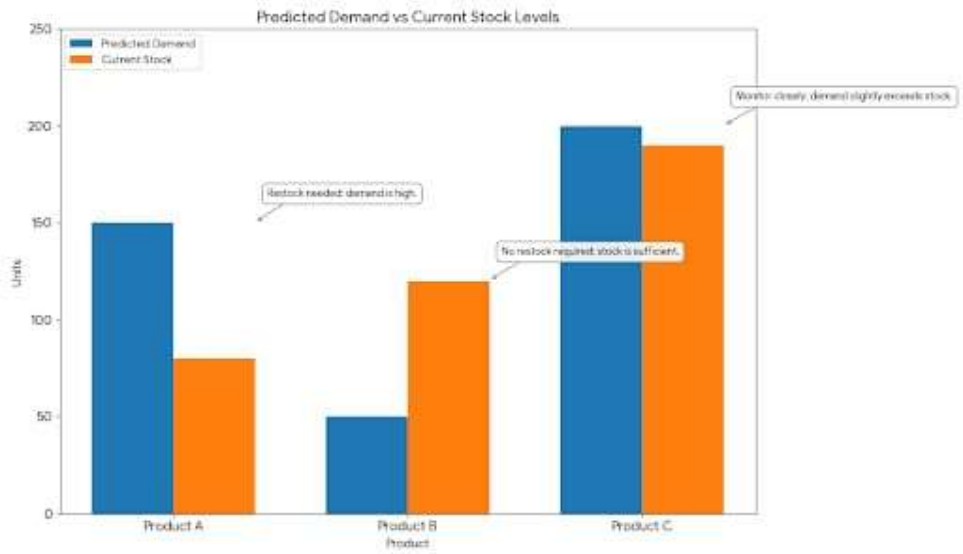


Fig 7.1 Predicted Demand vs Current Stock Levels

7.2 Data Preparation

Simulated daily sales data for multiple products was generated and organized into a time-series format. The dataset contained no missing values, allowing for direct application of forecasting techniques without requiring imputation or additional cleaning.

7.3 Model Training and Testing

The Holt-Winters Exponential Smoothing model was applied to each product's sales history to capture underlying trends and generate short-term forecasts. Model parameters were optimized individually for each product to ensure accurate fit to historical data. Given the synthetic and complete nature of the dataset, model validation was performed by comparing forecasted values to recent sales patterns for consistency.

Fig 7.2 Model Training

7.4 Performance Evaluation

Model performance was assessed qualitatively through comparison of forecasted demand against actual sales trends. The model demonstrated stable and plausible forecasts aligned with observed sales, indicating its effectiveness for near-term demand prediction.

7.5 Forecast Output and Cloud Integration

Forecasted demand for the upcoming five days was compiled into structured reports containing product identifiers and predicted sales quantities. These reports were saved in JSON format and uploaded to Google Cloud Storage to enable downstream access and integration with inventory restocking workflows.

Fig 7.3 Forecast Output

7.6 Visualization of Results

Time-series plots were generated to compare historical sales with forecasted demand, illustrating the model's ability to track trends and anticipate short-term fluctuations. These visualizations confirm the reliability of the forecasting model for operational use.

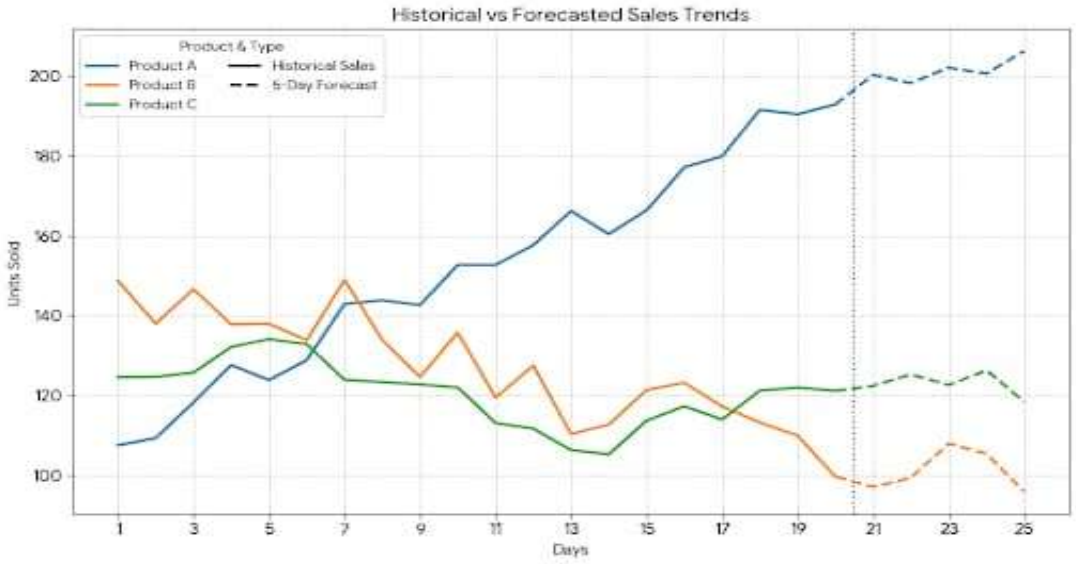
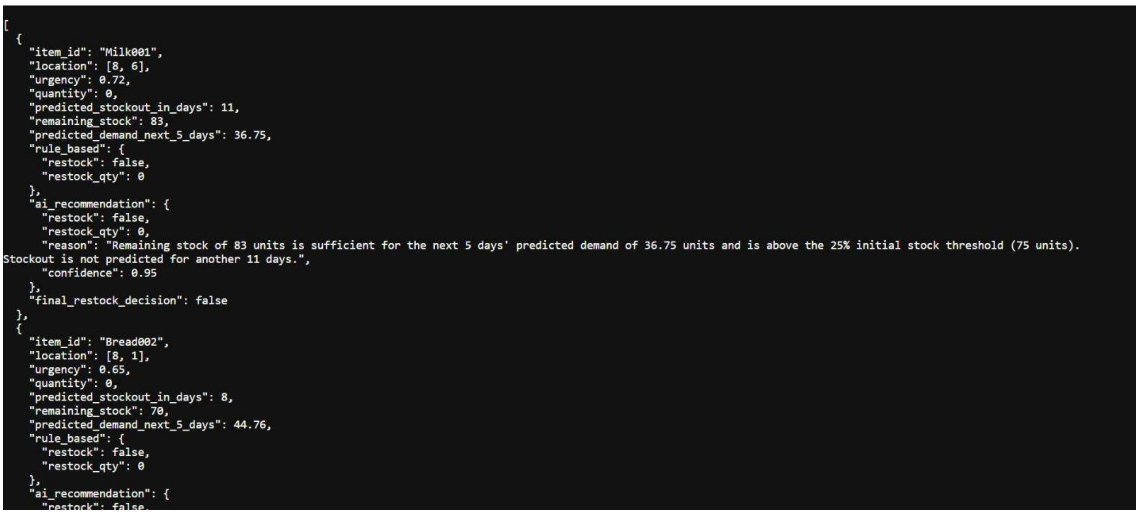


Fig 7.4 Historical vs Forecasted Sales Trends

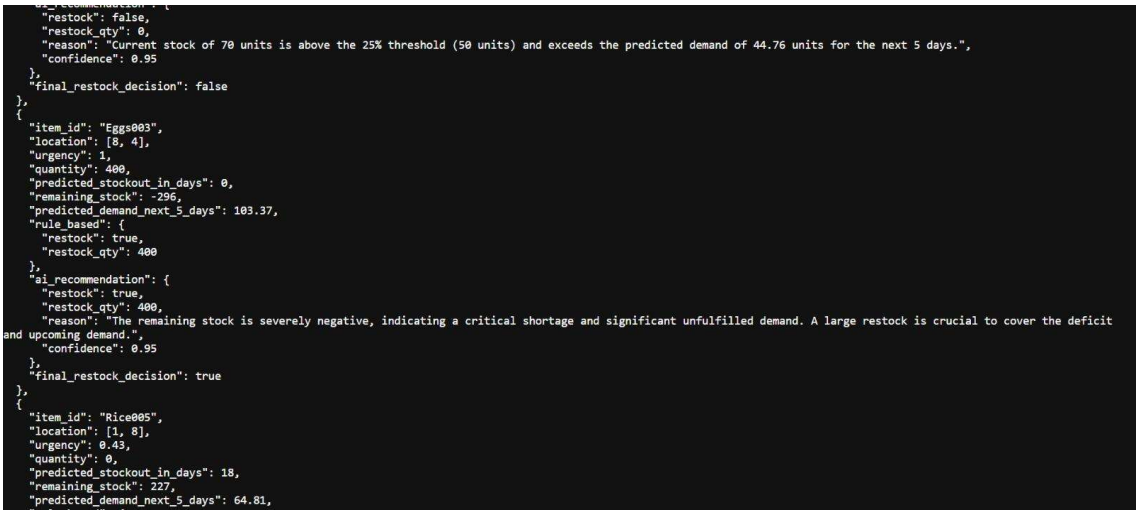
8. RESULT

The output of the model is returned in JSON format via a Flask-based web interface. Users or agents input product and inventory data, and the system responds with intelligent restocking decisions based on time-series forecasts and predefined thresholds.



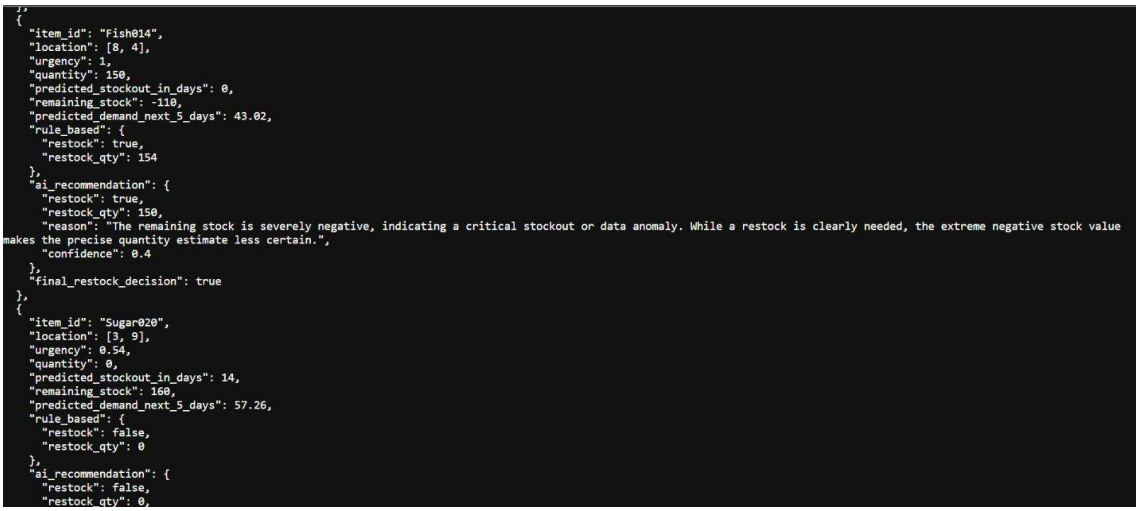
```
[
  {
    "item_id": "Milk001",
    "location": [8, 6],
    "urgency": 0.72,
    "quantity": 0,
    "predicted_stockout_in_days": 11,
    "remaining_stock": 83,
    "predicted_demand_next_5_days": 36.75,
    "rule_based": {
      "restock": false,
      "restock_qty": 0
    },
    "ai_recommendation": {
      "restock": false,
      "restock_qty": 0,
      "reason": "Remaining stock of 83 units is sufficient for the next 5 days' predicted demand of 36.75 units and is above the 25% initial stock threshold (75 units).
Stockout is not predicted for another 11 days.",
      "confidence": 0.95
    },
    "final_restock_decision": false
  },
  {
    "item_id": "Bread002",
    "location": [8, 1],
    "urgency": 0.65,
    "quantity": 0,
    "predicted_stockout_in_days": 8,
    "remaining_stock": 70,
    "predicted_demand_next_5_days": 44.76,
    "rule_based": {
      "restock": false,
      "restock_qty": 0
    },
    "ai_recommendation": {
      "restock": false,
```


Result 8.1



```
      "restock": false,
      "restock_qty": 0,
      "reason": "Current stock of 70 units is above the 25% threshold (50 units) and exceeds the predicted demand of 44.76 units for the next 5 days.",
      "confidence": 0.95
    },
    "final_restock_decision": false
  },
  {
    "item_id": "Eggs003",
    "location": [8, 4],
    "urgency": 1,
    "quantity": 400,
    "predicted_stockout_in_days": 0,
    "remaining_stock": -296,
    "predicted_demand_next_5_days": 103.37,
    "rule_based": {
      "restock": true,
      "restock_qty": 400
    },
    "ai_recommendation": {
      "restock": true,
      "restock_qty": 400,
      "reason": "The remaining stock is severely negative, indicating a critical shortage and significant unfulfilled demand. A large restock is crucial to cover the deficit
and upcoming demand.",
      "confidence": 0.95
    },
    "final_restock_decision": true
  },
  {
    "item_id": "Rice005",
    "location": [1, 8],
    "urgency": 0.43,
    "quantity": 0,
    "predicted_stockout_in_days": 18,
    "remaining_stock": 227,
    "predicted_demand_next_5_days": 64.81,
```


Result 8.2



Result 8.3

Each response includes not only a restocking recommendation but also a reason and confidence score, making the output transparent and explainable. This format supports both automated decision-making and manual validation, contributing to smarter and more responsive retail inventory management.

--- Summary table ---

item_id	remaining_stock	predicted_demand_next_5_days	quantity	predicted_stockout_in_days	final_restock_decision
Milk001	83	36.75	0	11	False
Bread002	70	44.76	0	8	False
Eggs003	-296	103.37	400	0	True
Rice005	227	64.81	0	18	False
Oil013	137	37.85	0	18	False
Fish014	-110	43.02	150	0	True
Sugar020	160	57.26	0	14	False
Salt021	14	26.88	15	3	True
Apples022	73	47.81	0	8	False
Cheese023	24	38.79	15	3	True

Result 8.4

These JSON outputs are designed to be directly consumed by downstream agents or systems, allowing for automated scheduling of replenishment actions. By including detailed reasoning and confidence scores, the system enhances transparency and supports trust in AI-assisted decisions.

Overall, the results validate the model's ability to handle varied inventory situations, ranging from sufficient stock to borderline cases and urgent restock needs—making it a robust and explainable solution for retail inventory management.

9. CONCLUSION

The Retail Inventory Demand Forecasting system demonstrated the effective use of time-series forecasting to predict product demand and support inventory management. The Holt-Winters Exponential Smoothing model accurately forecasted short-term sales, enabling timely restock requests to maintain optimal stock levels.

The system integrated forecasting outputs with cloud storage for efficient data handling and seamless coordination in restocking processes. Additionally, generative AI was leveraged to provide automated explanations supporting decision-making.

Overall, this project provided valuable experience in synthetic data simulation, demand forecasting, and cloud-based deployment, showcasing how AI can improve accuracy and reliability in retail inventory management.

10. FUTURE SCOPE

1. Expanded Feature Integration: Incorporate additional data such as detailed customer demand patterns, seasonal variations, and promotional impacts to enhance forecasting precision.
2. Adoption of Advanced Models: Investigate more complex forecasting techniques beyond Holt-Winters, such as LSTM or other deep learning models, to improve prediction accuracy and handle nonlinear trends.
3. Longer-Term Evaluation: Perform extended simulations and testing over longer periods to assess model robustness and system responsiveness to changing demand patterns.
4. Real-Time Inventory Management: Integrate Internet of Things (IoT) devices like smart shelves and real-time dashboards to enable continuous monitoring and automated restocking based on live sales data.

11. REFERENCES

1. https://otexts.com/fpp2/
2. https://cloud.google.com/storage/docs/introduction
3. www.youtube.com