



Project Phase II Report On

**Advanced Supply and Trade Resource Optimisation
(A.S.T.R.O.)**

*Submitted in partial fulfillment of the requirements for the
award of the degree of*

Bachelor of Technology

in

Computer Science and Business Systems

By

Amel Chandra (U2109009)

Bharath S. (U2109018)

Joepaul Vilsan (U2109033)

Shane George Salphie (U2109063)

Under the guidance of

Dr. Nikhila T. Bhuvan

**Department of Computer Science and Business Systems
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)**

Rajagiri Valley, Kakkanad, Kochi, 682039

April 2025

CERTIFICATE

*This is to certify that the project report entitled "**Advanced Supply and Trade Resource Optimisation**" is a bonafide record of the work done by **Amel Chandra (U2109009)**, **Bharath S (U2109018)**, **Joepaul Vilsan (U2109033)**, **Shane George Salphie (U2109063)**, submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in "Computer Science and Business Systems" during the academic year 2024-2025.*

Dr. Nikhila T. Bhuvan
Project Guide
Associate Professor
Dept. of CU
RSET

Mr. Mahesh K.M.
Project Co-ordinator
Assistant Professor
Dept. of CU
RSET

Dr. Divya James
Head Of Department
Associate Professor
Dept. of CU
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal**, Principal of RSET, and **Dr. Divya James**, Head of the Department of Computer Science and Business Systems, for providing us with the opportunity to undertake our project, "**Advanced Supply and Trade Resource Optimisation**".

We are highly indebted to our project coordinators, **Mr. Mahesh K. M.**, Assistant Professor, Department of Computer Science and Business Systems, **Dr. Nikhila T. Bhuvan**, Associate Professor, Department of Computer Science and Business Systems, **Ms. Ancy C. A.**, Assistant Professor, Department of Computer Science and Business Systems, for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Dr. Nikhila T. Bhuvan** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Amel Chandra

Bharath S.

Joepaul Vilsan

Shane George Salphie

Abstract

Small-scale vendors face significant challenges in competing with large retail chains due to limited purchasing power, inefficient logistics, and restricted access to financial services. These barriers often hinder their ability to manage demand fluctuations and make informed business decisions, threatening their long-term sustainability. This project aims to address these issues by developing a comprehensive platform that empowers small vendors through collaborative purchasing, optimized logistics, and data-driven decision-making. By enabling vendors to combine orders, the platform facilitates bulk purchasing, unlocking competitive discounts and cost savings. Advanced algorithms optimize logistics by streamlining delivery routes, thereby reducing transportation costs and enhancing operational efficiency. These features empower vendors to make strategic decisions that align with market demands and their business goals. The ultimate goal is to level the playing field for small vendors, enabling them to compete effectively with larger retail chains while fostering sustainable growth in local economies. The deliverables of this project include a fully functional platform supporting vendor collaboration and an optimized logistics module to reduce operational costs. By promoting economic resilience, enhancing competitiveness, and driving sustainable development, this project seeks to strengthen the local vendor ecosystem and contribute to the overall development of communities.

Contents

Acknowledgment	i
Abstract	ii
List of Abbreviations	v
List of Figures	vi
List of Tables	vii
1 Introduction	1
1.1 Background	2
1.2 Problem Definition	3
1.3 Scope and Motivation	4
1.4 Objectives	5
1.5 Relevance	6
2 System Architecture	10
2.1 Vendor Clustering Module	11
2.2 Route Optimisation Module	11
2.3 Demand Forecasting Module	12
3 Methodology	14
3.1 Vendor Clustering Module and Order Aggregation Module	14
3.1.1 Parameter Selection with Genetic Algorithm	15
3.1.2 DBSCAN Algorithm for Clustering Orders	16
3.2 Logistics Using Environment Adaptive Genetic Algorithm (EAGA)	17
3.2.1 Mechanics of Logistics Optimisation	17
3.2.2 Advantages of Using EAGA for Route Optimization	20
3.3 Demand Forecasting with Neural Prophet	21

3.3.1	Inputs for Neural Prophet	21
3.3.2	Neural Prophet Structure	22
4	Implementation	24
4.1	Backend Implementation	24
4.1.1	FastAPI Framework	24
4.1.2	Database Schema	25
4.1.3	API Endpoints	25
4.2	Frontend Implementation	26
4.2.1	React and TypeScript Application	26
4.2.2	Application Architecture	27
4.2.3	Key UI Components	28
4.3	Integration and Communication	28
4.3.1	API Integration	28
4.3.2	Data Models and API Integration	29
4.4	Summary	31
5	Results	32
5.1	Vendor Clustering	32
5.2	Route Optimization	34
5.3	Demand Forecasting	35
5.4	Cost Comparison Analysis	36
6	Conclusion and Future Enhancements	40
	References	44
	Appendix A: Presentation	46
	Appendix B: Vision, Mission, PO, PSO, and CO	57
	Appendix C: CO-PO-PSO Mapping	60

List of Abbreviations

ASTRO - Advanced Supply and Trade Resource Optimisation

SDG - Sustainable Development Goals

ACO - Ant Colony Optimisation

MACOA - Modified Ant Colony Optimisation Algorithm

IMFO - Improved Moth-Flame Optimization

GA - Genetic Algorithm

SME - Small to Medium-sized Enterprises

HGA - Hybrid Genetic Algorithm

API - Application Programming Interfaces

MOQ - Minimum Order Quantity

EAGA - Environment Adaptive Genetic Algorithm

List of Figures

2.1	System Architecture	10
5.1	Effect of learning rate on MAE and RMSE for NeuralProphet model	36
5.2	Effect of Epochs on MAE and RMSE for NeuralProphet model	36
5.3	NeuralProphet Demand Forecasting	38
5.4	NeuralProphet Long-term Trend Analysis	38
5.5	NeuralProphet Weekly Seasonality Analysis	38
5.6	NeuralProphet Yearly Seasonality Analysis	39

List of Tables

3.1	Example of Vendor Clustering in Order Aggregation	16
3.2	Key Inputs for Neural Prophet	22
3.3	Example of Forecasted Sales and Smoothed Trend	23
5.1	Comparative Analysis of Clustering Algorithms	33
5.2	Algorithm Performance Comparison for Route Optimization	35
5.3	Cost Comparison: Traditional vs. Collaborative Route Optimization	37

Chapter 1

Introduction

Small-scale vendors are essential contributors to local economies, offering unique products and personalized services that enrich communities. However, they face significant challenges competing against large retail chains, which benefit from greater purchasing power, efficient logistics, and easier access to financial resources. Due to limited resources, small vendors often struggle with high supply costs, fragmented logistics, and restricted access to credit, hindering their ability to grow and compete effectively. The ASTRO (Advanced Supply and Trade Resource Optimisation) platform addresses these challenges by providing a comprehensive solution that empowers small vendors through collaborative purchasing, logistics optimisation, and data-driven financial services. By allowing vendors to pool orders, ASTRO enables them to achieve bulk pricing similar to large retailers, reducing per-unit costs and enhancing profitability. The platform also incorporates advanced logistics optimisation algorithms, improving delivery efficiency and reducing operational expenses. Additionally, ASTRO offers financial services based on transaction data, enabling vendors to access credit and make informed decisions for business expansion.

ASTRO's functionality extends further with real-time demand forecasting tools, equipping vendors with predictive insights that help manage inventory and adapt to changing market demands. Through these features, ASTRO not only improves individual vendor competitiveness but also fosters economic resilience in local communities by supporting sustainable growth and operational efficiency. As a result, ASTRO is positioned as a vital resource for small vendors, helping them navigate a competitive retail landscape while contributing to the stability and diversity of local economies.

1.1 Background

Small-scale vendors are integral to the fabric of local economies, providing unique products, personalized services, and fostering vibrant community interactions that larger retail chains often cannot replicate. These vendors contribute significantly to economic diversity, cultural richness, and employment within their communities. However, despite their crucial role, small-scale vendors face persistent challenges that impede their ability to compete effectively with larger retail entities. These challenges include limited purchasing power, inefficient logistics, restricted access to financial services, and a lack of data-driven decision-making tools. In today's highly competitive retail landscape, large retail chains leverage economies of scale to negotiate bulk purchasing discounts, implement sophisticated logistics networks, and utilize advanced data analytics to optimize their operations and marketing strategies. In contrast, small-scale vendors typically operate with constrained resources, making it difficult for them to achieve similar efficiencies and cost-effectiveness. This disparity not only affects their profitability but also limits their capacity to respond to market fluctuations and evolving consumer demands. The ASTRO platform (Advanced Supply and Trade Resource Optimisation) is designed to address these disparities by providing small-scale vendors with tools and services that enhance their competitive edge. By enabling collaborative purchasing, optimizing logistics, and offering data-driven financial services, ASTRO empowers local vendors to achieve bulk pricing, streamline their operations, and make informed business decisions. This platform aims to bridge the gap between small vendors and large retail chains, fostering economic resilience and sustainable growth within local communities. Moreover, the advent of digital transformation has revolutionized various industries, including retail. The integration of technology into traditional business models is essential for small-scale vendors to remain relevant and competitive. ASTRO leverages modern technological advancements, such as machine learning algorithms for logistics optimisation and predictive analytics for demand forecasting, to deliver practical solutions tailored to the specific needs of small vendors. The significance of ASTRO extends beyond individual business success; it contributes to the overall economic health of communities by supporting the sustainability and growth of small-scale enterprises.

1.2 Problem Definition

Small-scale vendors face a significant challenge in today’s competitive retail landscape, where large retail chains dominate with bulk pricing, advanced logistics, and extensive resources. This disparity puts small vendors at a disadvantage, as they often lack the economies of scale and infrastructure necessary to compete effectively. The problem is that, despite being crucial to local economies, small vendors struggle with high operational costs, inefficient logistics, and limited access to affordable credit, all of which hinder their ability to grow sustainably. This issue is exacerbated by the lack of platforms that cater to the specific needs of small-scale businesses, especially when it comes to aggregating demand, optimizing supply chains, and improving cash flow.

Without access to bulk purchasing and efficient logistics, small vendors pay higher prices for goods, limiting their profit margins and making it difficult to reinvest in their businesses. Furthermore, the absence of accurate demand forecasting and inventory management tools leads to overstocking or stockouts, reducing customer satisfaction and revenue. These issues underscore the need for a system that enables small vendors to pool their purchasing power, streamline their supply chains, and make data-driven decisions to improve their competitiveness.

The Advanced Supply and Trade Resource Optimisation (ASTRO) platform seeks to address this gap by providing small-scale vendors with the tools to compete with larger retail chains. ASTRO enables collaborative purchasing, allowing vendors to aggregate their orders and access bulk discounts, thus reducing product costs. Additionally, the platform offers real-time demand forecasting, helping vendors optimize inventory levels to avoid both overstock and stockouts.

By facilitating these services, ASTRO aims to foster economic resilience and sustainable growth in local communities. Its focus on empowering small vendors with advanced tools levels the playing field, promoting a more equitable and competitive market environment. This approach not only ensures that small vendors can thrive but also contributes to the overall health and diversity of the retail ecosystem.

1.3 Scope and Motivation

The ASTRO platform is a multifaceted solution designed to support small-scale vendors by addressing key operational challenges through technological integration and collaborative strategies. The scope of ASTRO encompasses the development and implementation of several core functionalities:

1. **Collaborative Purchasing:** ASTRO facilitates group purchasing among small vendors, allowing them to pool their orders to achieve bulk pricing discounts. This feature enhances purchasing power, enabling vendors to reduce their cost of goods sold and improve profit margins.
2. **Logistics Optimisation:** Utilizing advanced algorithms, ASTRO optimizes delivery routes to reduce transportation costs and improve delivery efficiency. By streamlining logistics, the platform helps vendors achieve timely deliveries and minimize operational expenses.
3. **Real-Time Demand Forecasting:** Leveraging predictive analytics, ASTRO provides real-time demand forecasting tools that help vendors anticipate market demand and manage inventory levels effectively. Accurate demand forecasting reduces the risk of overstocking or stockouts, ensuring that vendors can meet customer needs efficiently.
4. **User-Friendly Interface:** ASTRO is designed with an intuitive user interface that ensures ease of use for vendors with varying levels of technical expertise. This feature maximizes user engagement and ensures that vendors can effectively utilize the platform's functionalities to enhance their business operations.

The motivation behind ASTRO stems from the urgent need to empower small-scale vendors in an increasingly competitive retail environment. As large retail chains continue to dominate the market through superior purchasing power and advanced logistics, small vendors risk marginalization and potential business failure. ASTRO addresses these challenges by providing a robust platform that enhances operational efficiency, reduces costs, and increases access to financial resources, thereby enabling small vendors to compete on a more equal footing.

Additionally, ASTRO is motivated by the broader goal of fostering economic resilience and sustainability within local communities. Small-scale vendors are pivotal to the economic diversity and vitality of local markets, contributing to job creation and community development. By supporting these vendors, ASTRO not only enhances individual business success but also strengthens the overall economic fabric of communities, promoting long-term sustainable growth and economic stability.

1.4 Objectives

The ASTRO project is guided by a set of clear and actionable objectives aimed at addressing the challenges faced by small-scale vendors. These objectives are designed to enhance operational efficiency, reduce costs, and provide financial support, thereby empowering vendors to compete effectively in the marketplace. The primary objectives of ASTRO include:

1. Enable Collaborative Purchasing:

- **Objective:** Develop a robust system that allows small vendors to combine their orders, thereby achieving bulk purchasing discounts.
- **Outcome:** Increased purchasing power, reduced cost of goods sold, and improved profit margins for vendors.

2. Optimize Logistics:

- **Objective:** Implement advanced route optimisation algorithms to streamline delivery processes, reduce transportation costs, and enhance operational efficiency.
- **Outcome:** Cost-effective logistics operations, timely deliveries, and reduced operational expenses.

3. Provide Real-Time Demand Forecasting:

- **Objective:** Integrate predictive analytics tools to offer real-time demand forecasting, enabling vendors to anticipate market demand and manage inventory levels effectively.

- **Outcome:** Minimization of overstocking and stockouts, improved inventory management, and enhanced ability to meet customer demands.

4. **Promote Economic Resilience and Sustainability:**

- **Objective:** Foster economic resilience by providing small vendors with the tools and resources needed to sustain and grow their businesses.
- **Outcome:** Long-term sustainability, increased competitiveness, and strengthened economic stability within local communities.

5. **Enhance User Experience and Accessibility:**

- **Objective:** Design a user-friendly interface that ensures easy access to the platform's features, regardless of vendors' technical expertise.
- **Outcome:** High user engagement, effective utilization of platform functionalities, and improved overall user satisfaction.

6. **Support Community Development:**

- **Objective:** Encourage collaboration and mutual support among vendors, fostering a sense of community and shared growth.
- **Outcome:** Collective bargaining power, knowledge sharing, and resource optimisation, leading to a stronger and more cohesive vendor network.

By achieving these objectives, ASTRO aims to create a comprehensive solution that empowers small-scale vendors to overcome their operational challenges, enhance their competitiveness, and contribute to the sustainable growth of local economies. Each objective is strategically aligned to address specific pain points, ensuring that the platform delivers tangible benefits and drives meaningful improvements in vendors' business operations.

1.5 **Relevance**

The relevance of the ASTRO project is multifaceted, addressing critical needs within the retail ecosystem and contributing to broader economic and social goals. Key aspects of ASTRO's relevance include:

1. Economic Empowerment of Small Vendors:

- **Impact:** ASTRO directly addresses the economic challenges faced by small-scale vendors, enhancing their purchasing power and optimizing logistics cost. This empowerment enables vendors to operate more efficiently, reduce costs, and increase profitability, thereby improving their economic standing and sustainability.

2. Enhancing Local Economies:

- **Impact:** By supporting small vendors, ASTRO contributes to the vitality and resilience of local economies. Small businesses are integral to economic diversification and job creation, promoting the long-term sustainability of local markets.

3. Technological Advancement and Digital Transformation:

- **Impact:** ASTRO exemplifies the role of technology in transforming traditional business models. By leveraging advanced algorithms for logistics optimisation, predictive analytics for demand forecasting, ASTRO introduces modern technological solutions to enhance the operational capabilities of small vendors. This digital transformation is crucial for small businesses to remain competitive in an increasingly digital and data-driven marketplace.

4. Sustainable Growth and Environmental Impact:

- **Impact:** ASTRO's logistics optimisation not only reduces operational costs but also minimizes the environmental footprint of delivery processes. By streamlining routes and reducing transportation distances, ASTRO contributes to more sustainable business practices, aligning with global efforts to reduce carbon emissions and promote environmental sustainability.

5. Competitive Parity:

- **Impact:** ASTRO helps small vendors achieve competitive parity with larger retail chains by providing tools that enhance their operational efficiency and cost-effectiveness. This parity is crucial for maintaining market diversity and

preventing the monopolistic dominance of large retailers, ensuring a healthy and competitive marketplace that benefits both vendors and consumers.

6. Adaptation to Market Trends:

- **Impact:** In a rapidly evolving retail environment, the ability to adapt to changing market trends and consumer behaviors is essential. ASTRO's real-time demand forecasting and data-driven insights enable small vendors to stay ahead of market trends, respond promptly to consumer needs, and adapt their inventory and marketing strategies accordingly.

7. Community Building and Collaboration:

- **Impact:** ASTRO fosters a collaborative ecosystem where small vendors can work together to achieve common goals. This sense of community and mutual support enhances collective bargaining power, knowledge sharing, and resource optimisation, contributing to the overall strength and cohesion of the local vendor network.

8. Scalability and Replicability:

- **Impact:** ASTRO's platform is designed to be scalable, allowing it to be replicated in various regions and adapted to different market conditions. This scalability ensures that the platform can benefit a wide range of small-scale vendors across different industries, promoting widespread economic empowerment and community resilience.

9. Support for Economic Diversity and Stability:

- **Impact:** By enabling small vendors to thrive, ASTRO supports economic diversity, which is essential for the stability and resilience of local economies. Diverse economic activities reduce dependence on a single sector or large corporations, making communities more adaptable to economic shocks and changes.

10. Promoting Entrepreneurship and Innovation:

- **Impact:** ASTRO encourages entrepreneurship by lowering the barriers to entry for small vendors. By providing the necessary tools and resources, ASTRO

enables aspiring entrepreneurs to start and grow their businesses, fostering a culture of innovation and economic dynamism within local communities.

11. Enhanced Customer Experience:

- **Impact:** By improving operational efficiency and reducing costs, ASTRO enables small vendors to offer better pricing and more reliable services to their customers. Enhanced customer experiences lead to increased customer loyalty and satisfaction, driving repeat business and positive word-of-mouth referrals.

12. Economic Resilience and Crisis Management:

- **Impact:** ASTRO equips small vendors with the tools to better manage economic fluctuations and crises. By optimizing inventory levels, forecasting demand accurately, and accessing financial support, vendors are better prepared to navigate economic downturns, supply chain disruptions, and other unforeseen challenges.

13. Alignment with Sustainable Development Goals (SDGs):

- **Impact:** ASTRO aligns with key United Nations Sustainable Development Goals, particularly Decent Work and Economic Growth (SDG 8) and Responsible Consumption and Production (SDG 12). By empowering small vendors, ASTRO promotes inclusive economic growth and decent work opportunities in local communities. Through optimized logistics and collaborative purchasing, the platform reduces resource waste and transportation emissions, supporting more sustainable consumption and production patterns in retail supply chains.

14. Enhancing Vendor Autonomy and Empowerment:

- **Impact:** ASTRO empowers small vendors by giving them greater control over their purchasing, logistics, and financial decisions. This autonomy fosters a sense of ownership and confidence among vendors, encouraging them to take proactive steps towards business improvement and growth.

Chapter 2

System Architecture

Figure 2.1 illustrates the system architecture for the ASTRO Platform, designed to support small-scale vendors by facilitating bulk order discounts, demand forecasting, and logistics optimisation. This architecture provides a clear overview of the platform's components and how they work together to streamline the entire process from order initiation to optimized delivery.

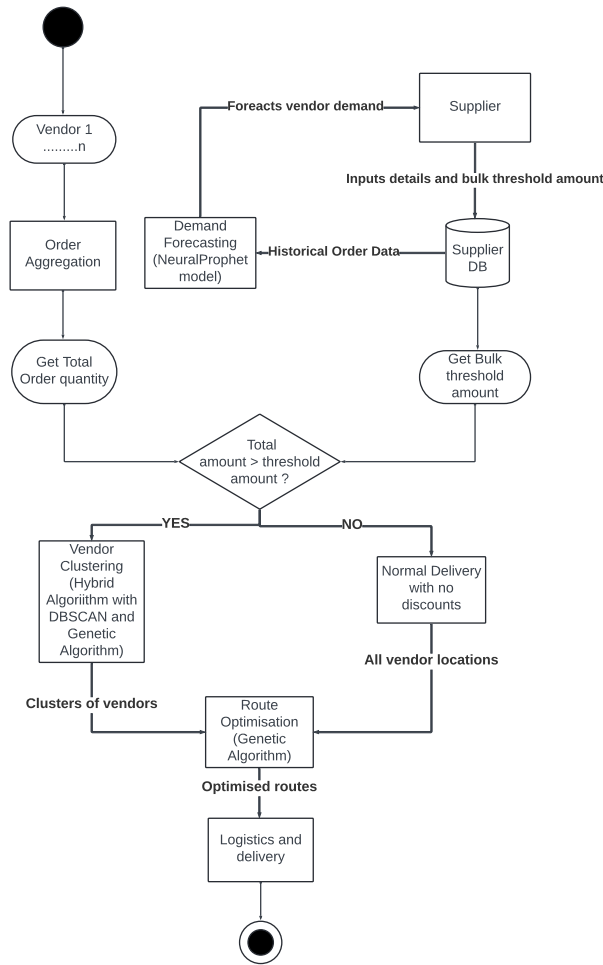


Figure 2.1: System Architecture

The system as shown in Fig. 2.1 is composed of multiple modules, each designed to enhance specific functions of the supply chain process. The Order Aggregation Module is responsible for grouping similar orders from multiple vendors to facilitate bulk purchasing. The Vendor Clustering Module applies machine learning techniques to identify optimal clusters of vendors based on their order patterns and geographic locations. The Route Optimization Module determines the most efficient delivery paths by selecting from multiple algorithms based on the number of vendors in a cluster. The Logistics and Delivery Module ensures the smooth execution of deliveries by incorporating real-time tracking and optimized scheduling. Lastly, the Demand Forecasting Module analyzes historical order data to predict future demand trends, enabling better decision-making for suppliers and reducing the risks of overstocking or understocking. The system also includes a Supplier Management Module, which allows suppliers to define bulk discount thresholds and manage product availability.

2.1 Vendor Clustering Module

The Order Aggregation Module builds on the insights provided by the demand forecasting module. Once multiple vendors agree to participate in a joint order, this module combines their requirements into a single bulk order. The process ensures that the MOQ thresholds are met, unlocking discounts from suppliers that would be unattainable for individual vendors.

The aggregation process also categorizes and organizes the joint order to ensure that products are distributed efficiently. Each vendor's share of the bulk order is clearly delineated, and the aggregated data serves as input for the subsequent logistics operations.

Notifications play a key role in this module: vendors are informed about the opportunity to join an order and, once confirmed, receive updates about the order status and expected delivery timelines. The module ensures transparency and coordination among vendors, simplifying the process of collective purchasing.

2.2 Route Optimisation Module

The Route Optimisation Module is responsible for ensuring efficient delivery of the aggregated orders. After the order is finalized and fulfilled by the supplier, this module uses

Green Route Optimisation algorithms to design an optimal delivery route.

Since the joint order often involves multiple vendors located in different areas, the Route optimisation module focuses on clustering delivery locations and minimizing logistics costs. The Route Flexibility and Service Time Window feature ensures that deliveries are scheduled efficiently while adhering to time constraints.

The module optimizes the route for a single vehicle tasked with delivering products to multiple vendors. Factors such as load capacity, distance between vendors, and delivery time windows are considered to achieve the most cost-effective and eco-friendly route. By reducing unnecessary travel and fuel consumption, this module contributes to both financial savings and environmental sustainability.

2.3 Demand Forecasting Module

The Demand Forecasting Module lies at the heart of the platform, enabling vendors to predict future sales and order cycles. By analysing historical sales data, the module leverages the Neural Prophet model to identify trends, seasonal patterns, and demand fluctuations. The predictions provide vendors with actionable insights into when and how much to order, ensuring they maintain optimal inventory levels while avoiding overstocking or stockouts.

Additionally, this module facilitates order cycle prediction, which is critical for synchronizing vendor orders. When a vendor places an order with a supplier, the platform uses the demand forecasting output to identify similar demand cycles for other vendors. This allows the platform to notify these vendors about the opportunity to join the order, achieving the Minimum Order Quantity (MOQ) threshold required for bulk discounts. By doing so, the demand forecasting module not only supports individual vendors but also sets the foundation for collective collaboration.

The seamless integration of demand forecasting, order aggregation, and route optimisation modules creates a powerful, collaborative platform that addresses the challenges faced by small-scale vendors. The demand forecasting module drives proactive order management, the order aggregation module facilitates cost-effective purchasing, and the route optimisation module ensures efficient delivery. Together, these modules enable vendors to compete with large retail chains by reducing costs, improving operations, and fostering a

cooperative ecosystem.

Chapter 3

Methodology

3.1 Vendor Clustering Module and Order Aggregation Module

Order aggregation is pivotal in supply chain management because it enables the consolidation of multiple small orders into a larger, bulk order, thereby reducing per unit costs, lowering shipping expenses, and streamlining procurement processes. Clustering algorithms have emerged as a powerful data-driven solution to group orders based on attributes such as timing, location, and product similarity; among these, Density-Based Spatial Clustering[1] of Applications with Noise (DBSCAN) is particularly attractive because it automatically detects clusters based on data density and effectively identifies outliers without forcing every order into a cluster, ensuring that only naturally cohesive orders are consolidated for bulk purchasing.

Despite these advantages, DBSCAN's[2] performance relies heavily on the correct setting of two parameters: ε (the neighborhood radius) and `min_samples` (the minimum number of points necessary to form a dense region). An improperly chosen ε can lead to overly fragmented micro-clusters or one excessively large cluster, while a poorly chosen `min_samples` might incorrectly label potential clusters or force groupings that do not reflect actual order density. These challenges become even more pronounced in a dynamic environment such as daily order aggregation, where patterns vary significantly. Hence, a Genetic Algorithm (GA)[3] is employed to automatically find the most effective DBSCAN parameters leading to the hybrid DBSCAN approach. Moreover, appropriately tuned ε and `min_samples` serve as critical thresholds: a well-chosen ε ensures that orders located significantly far from dense regions are not aggregated but are instead classified as noise, while the `min_samples` parameter guarantees that only sufficiently dense groups of orders form clusters, thereby preventing the inclusion of isolated orders that should remain unaggregated.

The hybrid methodology operates in two broad phases: (1) Parameter selection for DBSCAN using genetic algorithm, and (2) DBSCAN algorithm for clustering orders.

3.1.1 Parameter Selection with Genetic Algorithm

The GA begins by generating a population of parameter pairs $(\varepsilon, \text{min_samples})$ within predefined ranges say, ε in $[0.001, 0.1]$ and min_samples in $[2, 10]$. Each candidate solution is tested by running DBSCAN on the current dataset of orders. A simple fitness function, defined as:

$$\text{fitness}(\varepsilon, m) = (|\{\ell \in L \mid \ell \neq -1\}|) - \left(\sum_{i=1}^N \mathbf{1}\{L_i = -1\} \right) \quad (3.1)$$

- ε : Neighborhood radius in DBSCAN (eps).
- m : Minimum number of points required to form a cluster (min_samples).
- L : Set (or list) of labels assigned by DBSCAN to each data point.
- ℓ : An individual label from L .
- -1 : Label indicating noise/outliers (points not assigned to any cluster).
- $\{\ell \in L \mid \ell \neq -1\}$: Subset of labels corresponding to all clustered points (excluding noise).
- $|\{\ell \in L \mid \ell \neq -1\}|$: Number of clustered points (i.e., total points with labels $\neq -1$).
- $\sum_{i=1}^N \mathbf{1}\{L_i = -1\}$: Number of outliers/noise points, where $\mathbf{1}\{\cdot\}$ is the indicator function.
- N : Total number of data points under consideration.
- $\text{fitness}(\varepsilon, m)$: Objective to maximize: number of clustered points minus number of outliers.

Equation 3.1 rewards configurations that produce more dense groupings and fewer leftover or “noise” points. Following the canonical GA steps (selection, crossover, and mutation), the population evolves over multiple generations, gradually converging on an optimal parameter pair. These final $(\varepsilon, \text{min_samples})$ values yield the best clustering outcome for that day’s distribution of orders.

3.1.2 DBSCAN Algorithm for Clustering Orders

Once the system obtains the GA-tuned DBSCAN parameters, these parameters are applied to the orders if the dataset contains 100 or more orders. For datasets with fewer than 100 orders, the method defaults to a simpler DBSCAN approach that uses a median-based ε and a fixed $\text{min_samples} = 2$. This tiered design minimizes computational overhead for small datasets while ensuring that larger datasets benefit from thorough optimization. After DBSCAN completes its clustering, a subsequent function aggregates the orders within each cluster, but only if the total order quantity for that cluster falls within the acceptable range:

$$\text{MOQ} \leq (\text{Cluster Quantity}) \leq \text{MOQ} \times \left(1 + \frac{\text{Buffer \%}}{100}\right) \quad (3.2)$$

The output of the hybrid DBSCAN approach is the optimal clustering of vendors to participate in aggregated orders. This grouping ensures that:

- Orders within the same cluster have natural cohesion based on their attributes.
- Each cluster’s total order meets the MOQ requirements without excessive overordering.
- Outlier orders that don’t naturally fit into clusters remain unaggregated.

Vendor	Order Quantity	Cluster ID
Vendor A	50	1
Vendor B	30	1
Vendor C	10	-1 (Noise)
Vendor D	20	1

Table 3.1: Example of Vendor Clustering in Order Aggregation

In this example, Vendors A, B, and D are clustered together (Cluster ID 1), contributing a total order of 100 units, meeting the MOQ. Vendor C is identified as an outlier (noise) and is not included in the aggregated order.

Once the optimal vendor clustering is determined, the platform finalizes the bulk order for each viable cluster and notifies all participating vendors. The order details are then

passed to the logistics module for route optimisation and delivery scheduling, ensuring seamless fulfilment. By leveraging this hybrid approach, the platform provides an efficient, scalable, and automated solution for order aggregation, empowering vendors to compete effectively with larger retailers while ensuring only naturally cohesive orders are combined.

3.2 Logistics Using Environment Adaptive Genetic Algorithm (EAGA)

Logistics optimisation is a key component of the ASTRO platform, ensuring efficient delivery of aggregated orders to vendors while minimizing transportation costs and environmental impact. In the project, we employ the Environment Adaptive Genetic Algorithm (EAGA), a specialized variant of the Genetic Algorithm[4] tailored to optimize routes for delivery vehicles. This algorithm determines the most efficient Route to deliver products to different vendors included in an aggregated order, focusing on reducing fuel consumption, delivery time, and carbon emissions while adapting to environmental conditions.

3.2.1 Mechanics of Logistics Optimisation

Once the order aggregation process is complete, the platform identifies the vendors participating in the bulk order and their respective delivery locations. The challenge is to determine the optimal route for a delivery vehicle to service all vendors while adapting to environmental conditions and adhering to constraints such as delivery time windows, vehicle capacity, and geographic proximity.

The EAGA is designed to achieve this by incorporating concepts of environmental adaptability, Route flexibility, and comprehensive cost evaluation. Here's how it works:

Input Data: The algorithm takes the following inputs:

- Vendor locations (latitude and longitude).
- Delivery quantities for each vendor (ensuring vehicle capacity constraints are met).
- Service time windows for each vendor (if applicable).
- Vehicle starting point (typically the supplier's location).
- Environmental data including weather conditions, temperature, and elevation changes.

Route Representation: The delivery route is represented as a sequence of vendor locations, starting and ending at the supplier's location. For example:

Route: Supplier \rightarrow Vendor A \rightarrow Vendor B \rightarrow Vendor C \rightarrow Supplier

Comprehensive Cost Function: The EAGA employs a sophisticated cost function that evaluates route quality based on multiple factors. The total cost is calculated as:

$$TC = w_{\text{dist}} \times D + w_{\text{weather}} \times W + w_{\text{elev}} \times E + w_{\text{fuel}} \times F \quad (3.3)$$

where:

- TC is the total cost
- w_{dist} is the weight for distance component
- w_{weather} is the weight for weather component
- w_{elev} is the weight for elevation component
- w_{fuel} is the weight for fuel component
- D is the total distance
- W is the weather cost
- E is the elevation cost
- F is the fuel cost

Weather Cost Calculation: Weather cost incorporates delays and additional fuel consumption caused by adverse weather conditions:

$$W = W_t + W_p + W_w \quad (3.4)$$

where W is the total weather cost, W_t is the temperature cost, W_p is the precipitation cost, and W_w is the wind cost.

- Temperature cost: If the average temperature (avg_temp) exceeds 30°C, the penalty is calculated as:

$$W_t = 0.05 \times (\text{avg_temp} - 25)^2 \quad (3.5)$$

- **Precipitation cost:** If there is any precipitation, the penalty is calculated as:

$$W_p = 0.2 \times \text{avg_precip} \quad (3.6)$$

- **Wind cost:** If the average wind speed (avg_wind) exceeds 10 m/s, the penalty is calculated as:

$$W_w = 0.1 \times (\text{avg_wind} - 5)^2 \quad (3.7)$$

Elevation Cost Calculation: Elevation cost reflects the impact of terrain on vehicle performance:

$$E = 0.005 \times \left(\frac{|\text{elevation_change}|}{100} \right)^2 \quad (3.8)$$

Fuel Cost Calculation: Fuel cost is normalized to account for variations in consumption due to multiple factors:

$$F = F_b + F_t + F_w + F_e \quad (3.9)$$

- **Temperature component (F_t):** For temperatures above 30°C:

$$F_t = 0.01 \times (\text{avg_temp} - 30) \quad (3.10)$$

- **Wind component (F_w):** For wind speeds above 5 m/s:

$$F_w = 0.02 \times (\text{avg_wind} - 5) \quad (3.11)$$

- **Elevation component (F_e):**

$$F_e = \begin{cases} 0.0005 \times \text{elevation_change} & \text{if uphill} \\ -0.0003 \times |\text{elevation_change}| & \text{if downhill} \end{cases} \quad (3.12)$$

Genetic Algorithm Operations:

- **Selection:** Tournament selection is employed, choosing low-cost routes for reproduction.
- **Crossover:** The ordered crossover operator ensures valid offspring routes while maintaining vendor order.

- **Mutation:** Swap mutation introduces diversity by randomly swapping vendors with a probability that dynamically decays over generations.
- **Parameter Adaptation:** The algorithm dynamically adjusts parameters based on the number of vendors:

- Population Size: $P = P_0 \times \frac{v}{100}$
- Number of generations: $G = G_0 \times \frac{v}{100}$
- Mutation rate: $\mu = \mu_0 \times \left(1 + \frac{v-100}{1000}\right)$

where $P_0 = 100$, $G_0 = 300$, $\mu_0 = 0.1$, and v is the number of vendors.

- **Elitism:** The best solutions from each generation are preserved, accelerating convergence toward an optimal solution.

Output: The algorithm provides the optimized delivery route, detailing the order in which vendors should be serviced.

3.2.2 Advantages of Using EAGA for Route Optimization

The Environment Adaptive Genetic Algorithm is specifically chosen for the project because it aligns with the platform's goals of cost efficiency, environmental sustainability, and adaptability. Key advantages include:

- **Environmental Adaptability:** By factoring in weather conditions, temperature, and elevation changes, the algorithm adapts routes to real-world conditions.
- **Comprehensive Cost Evaluation:** The multi-factor cost function ensures that all relevant aspects of route efficiency are considered.
- **Scalability:** Parameter adaptation allows the algorithm to efficiently handle varying numbers of vendors.
- **Flexibility with Constraints:** It handles dynamic vendor locations, varying delivery quantities, and service time windows, ensuring practical applicability.
- **Sustainability Focus:** By minimizing fuel consumption and considering environmental factors, the algorithm promotes eco-friendly logistics.

After the aggregated order is finalized, the vendor delivery details are fed into the logistics module. The EAGA computes the optimal delivery route and provides the route to the delivery driver. The platform ensures that real-time environmental data, such as weather conditions, traffic updates, or vendor availability changes, are seamlessly integrated into the routing process.

This integration ensures timely and efficient delivery of products, enhancing vendor satisfaction while supporting the platform’s commitment to sustainability and cost-effective operations.

This chapter has outlined the system’s architecture and described each component’s role in creating a cohesive platform for demand forecasting, order aggregation, and logistics management. Through effective system integration and real-time notifications, the platform provides a unified experience that supports vendors in optimizing their operations, managing inventory, and coordinating logistics. The collaborative approach facilitated by this platform empowers vendors to maintain competitive advantages, meeting demand efficiently and enhancing overall supply chain resilience.

3.3 Demand Forecasting with Neural Prophet

ASTRO platform, demand forecasting[5] is essential for streamlining inventory management, optimizing order cycles, and enabling vendor collaboration. Using Neural Prophet[6], we forecast future sales for each store-item combination. This helps vendors make timely decisions about inventory and orders while facilitating joint purchasing to achieve bulk discounts. The model’s capability to handle trends and seasonality makes it ideal for predicting sales patterns critical to the platform’s success.

3.3.1 Inputs for Neural Prophet

The forecasting process begins with preparing the data. The key inputs used in the project include:

The `ds` column represents the timeline for sales data, while `y` contains the actual number of items sold at a given store for a particular item. Both columns are preprocessed to ensure consistency. Missing sales values (`y`) are interpolated to maintain data continuity, and the `ds` column is checked for valid date formatting.

Column Name	Description	Format
ds	Date of sales	Datetime (YYYY-MM-DD)
y	Sales value for the given date	Numeric

Table 3.2: Key Inputs for Neural Prophet

3.3.2 Neural Prophet Structure

Neural Prophet decomposes the sales data into the following components to identify patterns: The NeuralProphet model consists of multiple modules, each contributing an additive component to the forecast. The forecasted value \hat{y}_t is the sum of the following components:

$$\hat{y}_t = T(t) + S(t) + E(t) + F(t) + A(t) + L(t) \quad (3.13)$$

Where:

- $T(t)$ = Trend at time t : This captures long-term growth or decline in the sales data.
- $S(t)$ = Seasonal effects at time t : This models recurring patterns in sales, such as seasonal peaks.
- $E(t)$ = Event and holiday effects at time t : This component adjusts for external events or holidays that affect demand.
- $F(t)$ = Regression effects for future-known exogenous variables at time t : This accounts for external factors, such as planned promotions or events, that are known in advance.
- $A(t)$ = Auto-regression effects based on past observations at time t : This models the relationship between past sales data and current demand.
- $L(t)$ = Regression effects for lagged observations of exogenous variables at time t : This captures the delayed impact of external variables on sales.

Each component is modeled independently, and their outputs are summed to generate the final forecast \hat{y}_t . The NeuralProphet framework is flexible, allowing each of these

components to be switched on or off depending on the data and requirements of the project.

The model generates forecasts over a predefined time horizon, typically ranging from 30 to 60 days based on vendor needs. The outputs include:

- **Predicted Sales (yhat1):** Projected daily sales for the forecasted period.
- **Smoothed Sales Trend (yhat1_trend):** Rolling averages (e.g., 30-day mean) applied to predicted values to highlight trends and reduce noise.

Date (ds)	Predicted Sales (yhat1)	Smoothed Sales (yhat1_trend)
2024-12-01	100	98
2024-12-02	105	99
2024-12-03	110	101

Table 3.3: Example of Forecasted Sales and Smoothed Trend

The demand forecasting module integrates seamlessly with other parts of the platform, creating a cohesive system that enhances vendor operations:

- **Inventory Management:** Forecasted sales help vendors prepare for upcoming demand, ensuring stock levels are optimized.
- **Order Aggregation:** Predictions trigger notifications to vendors nearing their replenishment cycle. These notifications encourage vendors to place joint orders, maximizing cost savings.
- **Logistics Optimisation:** Predicted delivery schedules align with logistics planning, enabling route optimisation and efficient load distribution.

By integrating Neural Prophet’s forecasts into the platform, vendors can proactively manage inventory, coordinate joint purchases, and reduce costs. The model’s ability to deliver actionable insights ensures the platform’s operational efficiency and supports the growth of small-scale vendors.

Chapter 4

Implementation

The ASTRO platform has been implemented using a modern technology stack that ensures scalability, performance, and cross-platform compatibility. The implementation follows a client-server architecture with clear separation of concerns:

- **Backend:** Developed using FastAPI, a modern, high-performance Python web framework
- **Frontend:** Built with Flutter to provide a seamless cross-platform experience for both mobile and web clients
- **Database:** Utilizes Firebase Firestore for flexible, scalable NoSQL document storage with real-time capabilities
- **Machine Learning Pipeline:** Implemented in Python using libraries such as NeuralProphet, scikit-learn, and pandas

4.1 Backend Implementation

4.1.1 FastAPI Framework

The backend system is implemented using FastAPI, a modern Python web framework designed for building high-performance APIs. FastAPI was chosen for several key advantages:

- **Performance:** Built on Starlette and Pydantic, FastAPI offers near-native performance comparable to Node.js and Go
- **Type Checking:** Utilizes Python type hints for automatic validation and documentation

- **Asynchronous Support:** Native support for `async/await` patterns to handle concurrent requests efficiently
- **API Documentation:** Automatic generation of OpenAPI and Swagger UI documentation
- **Dependency Injection:** Built-in system for managing dependencies and services

4.1.2 Database Schema

The database schema is implemented using Firebase Firestore, a NoSQL document database that provides real-time synchronization and offline capabilities. The system uses Pydantic models for data validation and serialization:

Firestore's flexible document structure enables easy storage and retrieval of complex nested data, which is particularly beneficial for order items and shipping addresses. The system leverages Firestore's querying capabilities for efficient data filtering and real-time updates.

4.1.3 API Endpoints

The API is organized into several functional modules, each handling specific aspects of the platform:

Key API endpoints include:

1. Core Endpoints:

- `GET /`: Root endpoint
- `GET /info`: System information

2. Vendor Operations:

- `GET /vendor/suppliers`: Get list of suppliers
- `GET /vendor/products/{supplier_id}`: Get products from a supplier
- `POST /vendor/orders`: Place a new order
- `POST /vendor/instant_order`: Place an order for immediate fulfillment

3. Product Management:

- `POST /supplier/{supplier_id}/add_product`: Add a new product
- `PUT /supplier/update_product/{product_id}`: Update product details
- `DELETE /supplier/delete_product/{product_id}`: Remove a product
- `GET /supplier/get_products/{supplier_id}`: Get supplier's products

4. Order Aggregation:

- `POST /supplier/aggregate_orders`: Process orders for aggregation
- `POST /supplier/update_instant_order_status`: Update status of instant orders

5. Route Optimization:

- `GET /route-optimization/{aggregation_id}`: Generate optimal delivery route

6. Demand Forecasting:

- `GET /demand-forecast/{product_id}`: Get demand forecast for a product

4.2 Frontend Implementation

4.2.1 React and TypeScript Application

The ASTRO platform frontend is implemented using React with TypeScript and Tailwind CSS, featuring a comprehensive routing structure and modern UI components. Our implementation features:

- **Type-Safe Development:** TypeScript provides static typing throughout the codebase, enabling early error detection during development and improving code quality
- **Component Architecture:** A modular system of UI components organized by business domain (vendors, suppliers, orders) to maximize code reusability
- **Tailwind Integration:** Utility-first CSS approach through Tailwind enables rapid styling without context-switching between files

- **Responsive Layout:** Adaptive interface designs that work across desktop and mobile devices using Tailwind's responsive utility classes
- **Modern UI Components:** Custom component library including toasts, tooltips, and interactive elements for a consistent user experience

4.2.2 Application Architecture

Our React application follows a structured architecture organized around the following principles:

- **React Query Integration:** Centralized data fetching through QueryClient-Provider for efficient server state management
- **Context-Based Authentication:** Custom AuthProvider context manages user authentication state and permissions
- **Route-Based Code Structure:** Features organized by routes (e.g., /dashboard, /products, /orders) for intuitive navigation
- **Separation of Concerns:** Clear distinction between vendor and supplier interfaces with dedicated components and routes
- **Centralized Notifications:** Global toast notifications through Toaster components for system alerts and user feedback

Our routing structure demonstrates the application's organization:

```
1 <Routes>
2   <Route path="/" element={<Index />} />
3   <Route path="/login" element={<Login />} />
4   <Route path="/register" element={<Register />} />
5   <Route path="/complete-profile" element={<CompleteProfile />} />
6   <Route path="/dashboard" element={<Dashboard />} />
7   <Route path="/products" element={<Products />} />
8   <Route path="/orders" element={<Orders />} />
```

```

9   <Route path="/suppliers" element={<Suppliers />} />
10  <Route path="/settings" element={<Settings />} />
11  <Route path="/forecast" element={<DemandForecast />} />
12  <Route path="/route-optimization" element={<OrderAggregation />} />
13  <Route path="*" element={<NotFound />} />
14 </Routes>

```

4.2.3 Key UI Components

The application is structured around several core screens that support the platform’s functionality:

1. **Dashboard (/dashboard):** Central hub displaying key metrics, recent orders, and aggregation opportunities
2. **Products (/products):** Interface for vendors to browse and select products from suppliers
3. **Orders (/orders):** Comprehensive order management with status tracking and history
4. **Suppliers (/suppliers):** Directory of available suppliers with filtering capabilities
5. **Demand Forecast (/forecast):** Interactive visualization of product demand predictions
6. **Order Aggregation (/order-aggregation):** Interface for suppliers to manage aggregated orders and optimize delivery routes

Each screen is designed with role-based access control, ensuring users only see functionality relevant to their role as either vendor or supplier.

4.3 Integration and Communication

4.3.1 API Integration

The frontend communicates with our FastAPI backend through React Query, providing efficient data fetching and caching:

- **Centralized Query Client:** A shared `QueryClientProvider` wraps the application to manage all API requests
- **Typed API Responses:** TypeScript interfaces ensure type safety between frontend components and API data
- **Authentication Headers:** JWT tokens are automatically included in requests through Axios interceptors
- **Custom API Hooks:** Domain-specific React hooks encapsulate data fetching logic (e.g., `useOrders`, `useProducts`)

4.3.2 Data Models and API Integration

The ASTRO platform implements a comprehensive set of TypeScript interfaces to ensure type safety and data consistency across the application. These interfaces define the contract between frontend components and backend API services.

Core Business Entity Interfaces

We structured our data models around key business entities to ensure proper type checking and validation:

- **User Models:** The system distinguishes between different user types using specialized interfaces:
 - `UserBase` interface provides common user attributes (ID, name, contact information)
 - `Vendor` and `Supplier` interfaces extend the base with role-specific properties
- **Product Interface:** Represents product data including:
 - Core product details (name, description, category)
 - Inventory management fields (price, stock level, MOQ)
 - Tracking metadata (supplier references, timestamps)
- **Order Model:** Comprehensive order representation with:

- Order status tracking using a strongly-typed `OrderStatus` enum
- Nested `OrderItem` objects with product references and quantities
- Shipping details and payment information

Algorithm Response Interfaces

The platform employs specialized interfaces for machine learning components:

- **Demand Forecasting:** The `DemandForecast` interface structures time-series forecasting data:
 - Sequential `ForecastPoint` objects with quantities and confidence intervals
 - Trend analysis and direction indicators
 - Model configuration parameters and performance metrics
- **Order Aggregation:** The `OrderAggregationResponse` interface captures clustering results:
 - `OrderAggregationCluster` objects with consolidated order information
 - Geographic coordinates for delivery planning
 - Algorithm performance metrics and configuration parameters
- **Route Optimization:** The `RouteOptimizationResult` interface structures delivery routes:
 - Sequenced waypoints with geographic coordinates
 - Distance and time calculations between points
 - Environmental data integration for adaptive routing

Dashboard Data Interfaces

For the dashboard, specialized interfaces aggregate data for presentation:

- **DashboardData:** Combines order statistics, supplier information, and financial metrics

- `OrderStatusCount`: Provides a breakdown of orders by current status
- `MonthlyOrderData`: Tracks order volume trends for visualization
- `StatCardProps` and other component-specific interfaces for UI rendering

These strongly-typed interfaces ensure data consistency throughout the application while providing robust documentation and autocomplete support during development. The interface-driven approach facilitates seamless integration between frontend components and backend services by establishing clear contracts for data exchange. The notification system is particularly important for the order aggregation feature, as it alerts vendors about potential bulk ordering opportunities with nearby businesses, enabling real-time collaboration that would otherwise be difficult to coordinate.

4.4 Summary

The implementation of the ASTRO platform leverages modern technologies and architecture patterns to deliver a robust, scalable, and performant solution. By utilizing FastAPI for the backend and Flutter for the frontend, the system achieves a balance between development efficiency and runtime performance. The modular design ensures that each component order aggregation, logistics optimization and demand forecasting can evolve independently while maintaining integration with the broader system.

The implementation demonstrates how complex algorithms and advanced machine learning techniques can be effectively deployed in a production environment, providing tangible benefits to vendors through improved demand forecasting, efficient order aggregation, and optimized logistics.

Chapter 5

Results

5.1 Vendor Clustering

Agglomerative Clustering[7], K-means[8], DBSCAN, and a Genetic Algorithm-enhanced DBSCAN (GA-DBSCAN) were evaluated to group orders based on geographic locations. Agglomerative Clustering employs a hierarchical merging approach, K-means minimizes intra-cluster variance by partitioning the data into pre-specified clusters, and DBSCAN utilizes a density-based criterion, automatically identifying outliers and clusters of arbitrary shapes without the need to predefine the number of clusters. GA-DBSCAN improves upon standard DBSCAN by using a Genetic Algorithm (GA) to dynamically optimize DBSCAN's parameters: ε (the neighborhood radius) and `min_samples` (the minimum points required to define a cluster).

Synthetic datasets were generated to realistically simulate order distributions, with locations uniformly randomized between latitudes 8° to 18° and longitudes 72° to 82°, representing diverse geographical conditions. Each synthetic order was assigned a random quantity between 50% and 150% of the minimum order quantity (MOQ). For the experiments, the MOQ was set to 100 units, and a buffer percentage of 20% was adopted, resulting in an acceptable cluster quantity range from 100 to 120 units. Clustering performance was assessed on datasets comprising 25, 50, 100, 150, 200, 300, and 350 orders. Evaluation metrics included execution time, number of valid clusters (those meeting MOQ constraints), leftover orders (orders not aggregated due to either quantity constraints or geographic isolation), and silhouette scores a measure of clustering quality ranging from -1 (poor) to +1 (excellent), calculated based on intra cluster compactness and inter cluster separation.

As shown in Table 5.1, GA-DBSCAN consistently provided the highest clustering quality, evidenced by superior silhouette scores (0.629–0.780) in datasets of 100 orders

Table 5.1: Comparative Analysis of Clustering Algorithms

Algorithm	Orders	Time (s)	Clusters	Outliers	Silh. Score
DBSCAN	25	0.002	7	16	0.350
	50	0.002	13	32	0.330
	100	0.003	28	68	0.392
	150	0.004	39	99	0.380
	200	0.005	63	118	0.375
	300	0.009	85	187	0.361
	350	0.010	98	219	0.355
GA-DBSCAN	25	0.020	0	25	N/A
	50	0.025	0	50	N/A
	100	0.025	11	89	0.735
	150	0.030	12	138	0.780
	200	0.036	39	156	0.726
	300	0.048	35	262	0.715
	350	0.055	45	301	0.629
K-means	25	0.030	6	18	0.358
	50	0.004	12	34	0.324
	100	0.004	26	72	0.392
	150	0.005	37	103	0.390
	200	0.032	61	122	0.385
	300	0.007	84	189	0.363
	350	0.008	94	227	0.350
Agglomerative	25	0.002	6	18	0.341
	50	0.002	12	34	0.337
	100	0.002	27	70	0.393
	150	0.004	38	101	0.358
	200	0.005	61	122	0.359
	300	0.007	84	189	0.359
	350	0.009	95	225	0.349

and above. However, GA-DBSCAN yielded no valid clusters for smaller datasets (25 and 50 orders), primarily because the genetic algorithm optimization was not beneficial for datasets with limited points. In these smaller scenarios, DBSCAN, K-means, and Agglomerative clustering achieved more effective results, with DBSCAN offering the fastest performance and minimal computational overhead.

For larger datasets (100 orders or more), GA-DBSCAN demonstrated significant advantages due to its adaptive tuning of ε and `min_samples`, optimizing clustering parameters dynamically for each dataset. Conversely, K-means and Agglomerative clustering consistently produced higher numbers of clusters and leftover orders due to their static parameterization, often resulting in over-segmentation or the inability to flexibly handle noise points. Additionally, standard DBSCAN, although computationally efficient, exhibited limitations due to its static parameter selection, leading to less coherent clusters and increased leftover orders as dataset sizes grew.

Leftover orders represent those that could not be aggregated due to either insufficient cumulative quantity to meet MOQ constraints or geographic isolation making clustering impractical. The silhouette score quantifies clustering quality, computed as the mean intra-cluster distance compared to the mean nearest-cluster distance, thus indicating how compact and well-separated the clusters are.

This evaluation clearly supports adopting a hybrid clustering strategy. Specifically, DBSCAN is computationally efficient and effective for smaller datasets (fewer than 100 orders), where dataset complexity does not justify extensive parameter tuning. For larger and more complex order distributions (100 or more orders), GA-DBSCAN is strongly recommended, given its superior clustering accuracy and effectiveness in handling noise, ensuring clusters align optimally with real-world operational constraints.

5.2 Route Optimization

As shown in Table 5.2, the EAGA outperformed ACO[9], which was modified to incorporate the same environmental constraints as EAGA, in both travel distance and execution time. For 20 vendors, EAGA generated routes covering 4703.90 km in 20.53 seconds, while ACO required 23.26 seconds to produce routes spanning 5197.54 km. With 50 vendors, EAGA optimized the route to 6993.30 km in 60.75 seconds, whereas ACO produced routes

Table 5.2: Algorithm Performance Comparison for Route Optimization

Algorithm	Vendors	Dist. (km)	Cost (Rs.)	Time (s)
ACO	20	5197.54	29811.51	23.26
	50	8333.63	47618.80	50.43
	100	9648.29	54444.64	148.69
EAGA	20	4703.90	22320.78	20.53
	50	6993.30	33770.55	60.75
	100	8581.06	40956.56	142.21

covering 8333.63 km in 50.43 seconds. For 100 vendors, EAGA achieved routes of 8581.06 km in 142.21 seconds, while ACO took 148.69 seconds to generate routes of 9648.29 km. EAGA performed better due to its faster convergence and effective balance between exploration and exploitation, whereas ACO required more iterations to refine paths, leading to higher travel distances and execution times despite using identical environmental data inputs.

5.3 Demand Forecasting

The NeuralProphet model was evaluated using the Store Item Demand Forecasting Challenge dataset, which provides historical daily sales data for multiple store-item combinations. For our analysis, we specifically focused on item 1 from the dataset to demonstrate the model’s performance on a representative product time series. Performance was assessed using standard error metrics to evaluate forecasting accuracy across different learning rate and training epoch.

Fig. 5.1 illustrates the impact of learning rate on model accuracy. Both error metrics exhibit a characteristic U-shaped pattern, with optimal performance achieved at learning rates between 0.005 and 0.01. At extremely low rates (0.0001), the model converges slowly, resulting in higher errors (MAE=32.14, RMSE=39.12). Interestingly, at 0.0005, performance significantly deteriorates (MAE=85.04, RMSE=92.52), likely due to the optimizer becoming trapped in a poor local minimum. As learning rates increase beyond 0.01, error metrics stabilize, indicating the model’s robustness across a range of higher

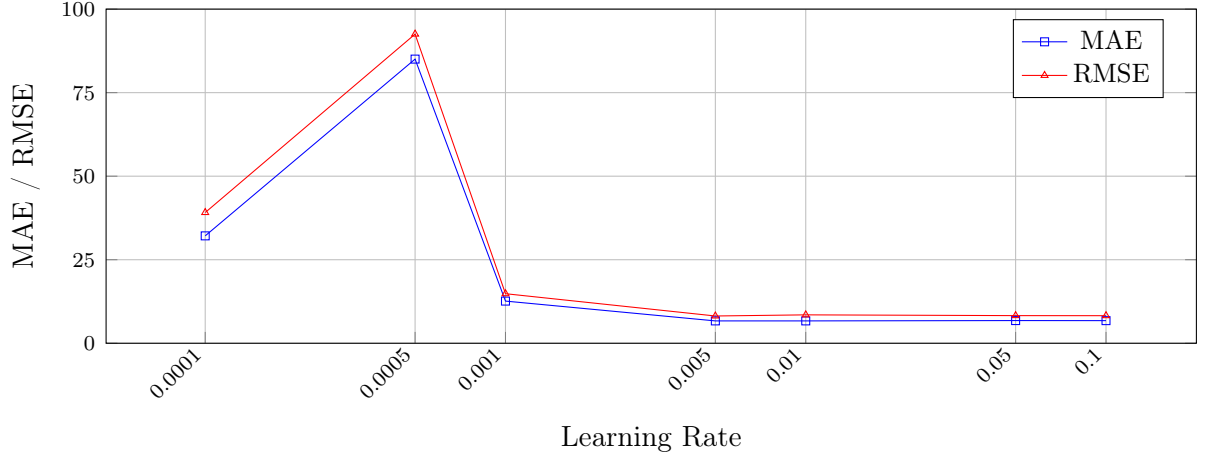


Figure 5.1: Effect of learning rate on MAE and RMSE for NeuralProphet model

learning rates. The optimal learning rate of 0.005 achieves the lowest MAE (6.69) and RMSE (8.16).

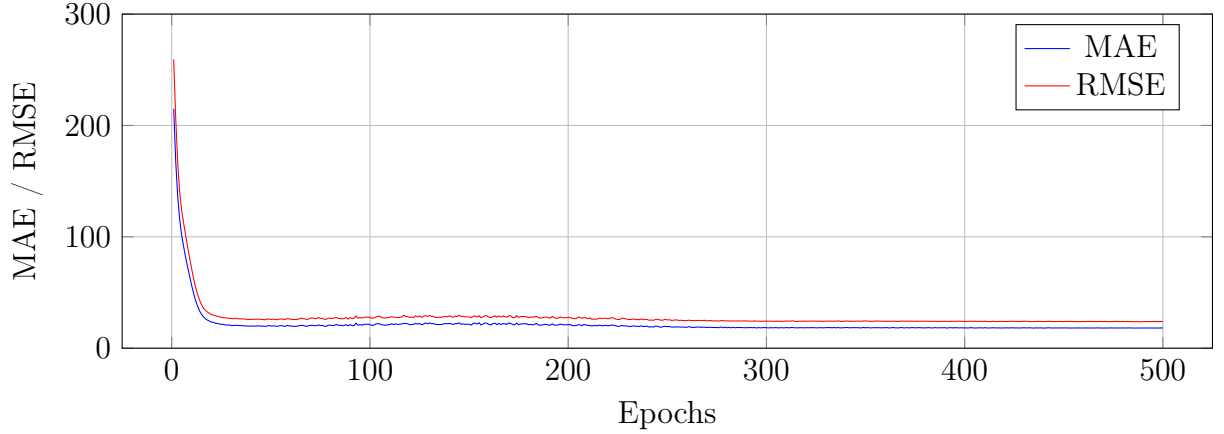


Figure 5.2: Effect of Epochs on MAE and RMSE for NeuralProphet model

5.4 Cost Comparison Analysis

To evaluate the effectiveness of the proposed collaborative ordering and shared delivery model, we analyzed route optimization performance across 50 vendors, comparing traditional individual delivery routes against our collaborative approach. In the 50-vendor scenario presented in Table 5.3, our system processed five staggered orders of varying sizes, analyzing both traditional and collaborative routing methods. The traditional approach required five separate delivery routes, often with significant overlap. In contrast, our

collaborative model consolidated these into a single optimized route, yielding substantial savings in distance, cost, and fuel consumption.

- Scenario: 50 vendors requiring deliveries from a central supplier
- Traditional approach: Individual routes planned for each vendor or small group
- Collaborative approach: Single optimized route serving all vendors
- Distance analyzed: Regional delivery network covering approximately 7,000 km

Table 5.3: Cost Comparison: Traditional vs. Collaborative Route Optimization

Metric	Trad. Approach	Collab. Approach	Savings
Distance (km)	35,606	6,993	28,613 (80.4%)
Route Cost (Rs.)	36,929	7,255	29,675 (80.4%)
Fuel (L)	1,811	357	1,454 (80.3%)
Fuel Cost (Rs.)	171,135	33,770	137,366 (80.3%)

Fig. 5.2 demonstrates how error metrics evolve over training epochs. Both MAE and RMSE follow similar trajectories, decreasing rapidly during the initial 100 epochs, followed by gradual refinement and eventual convergence around 400 epochs. This pattern reflects the typical learning dynamics of neural network-based forecasting models—quick initial gains followed by diminishing returns as training progresses. The relatively smooth convergence curve indicates stable training dynamics without significant oscillations, suggesting appropriate hyperparameter settings.

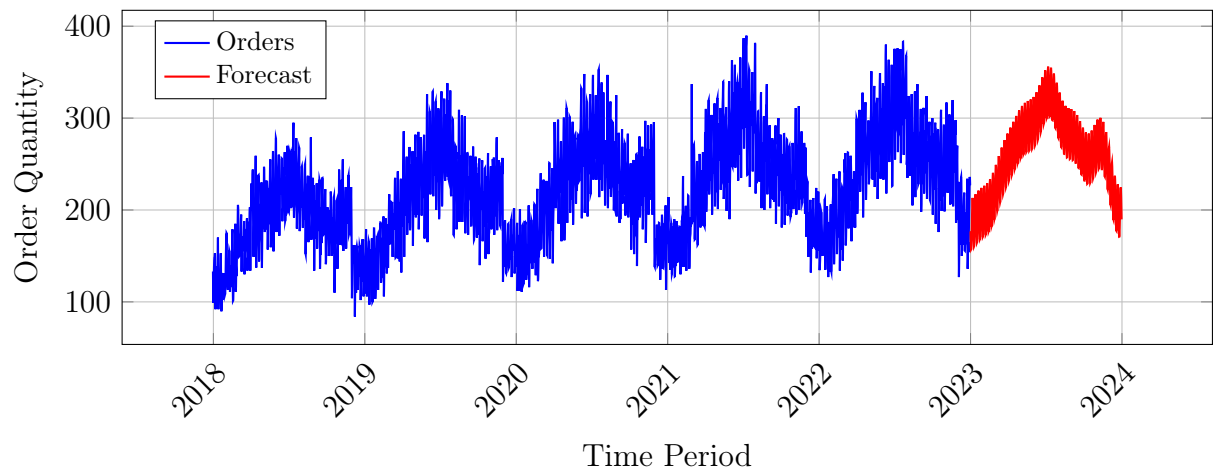


Figure 5.3: NeuralProphet Demand Forecasting

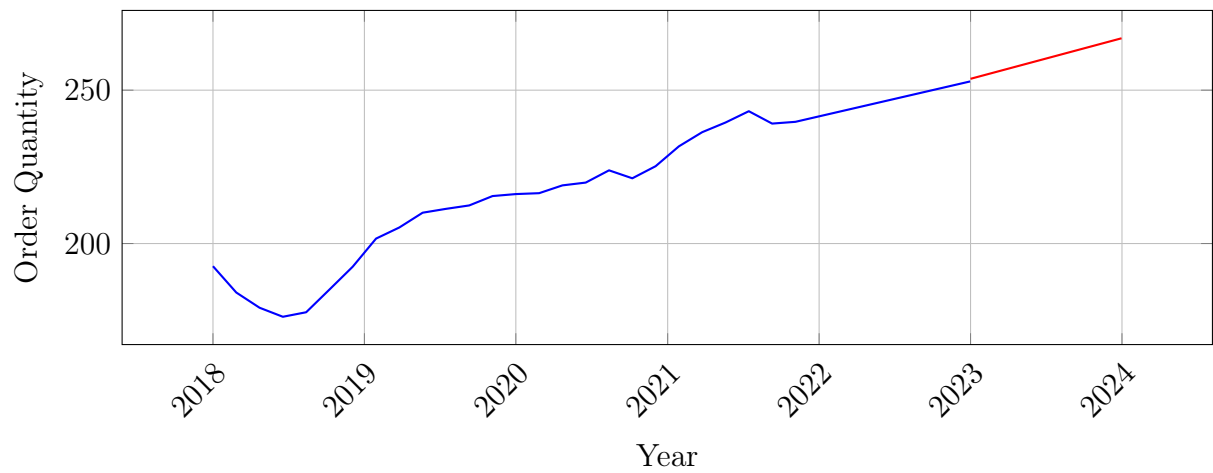


Figure 5.4: NeuralProphet Long-term Trend Analysis

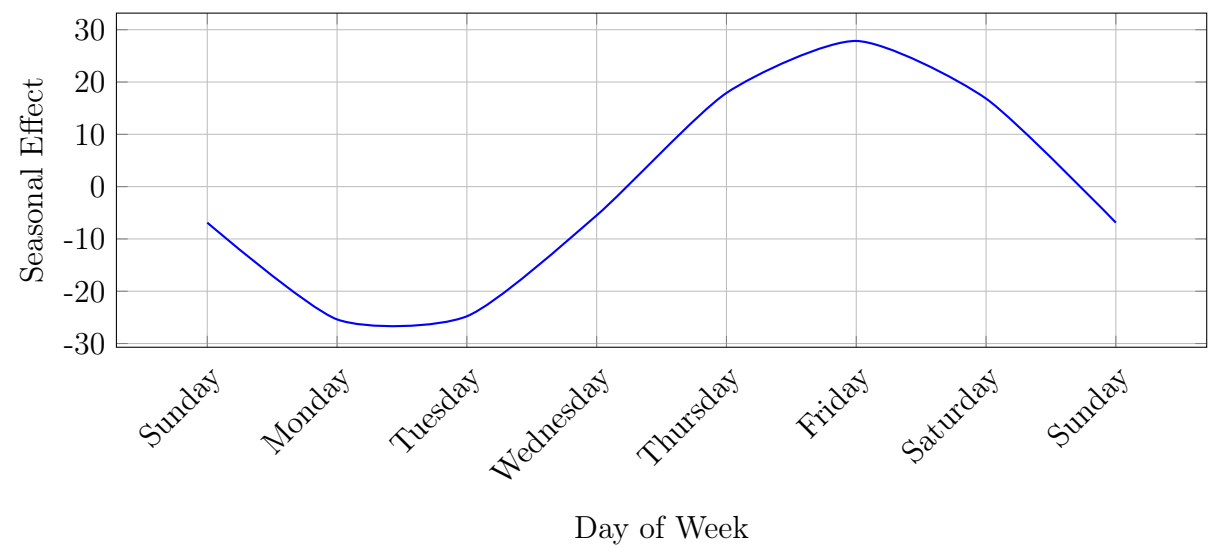


Figure 5.5: NeuralProphet Weekly Seasonality Analysis

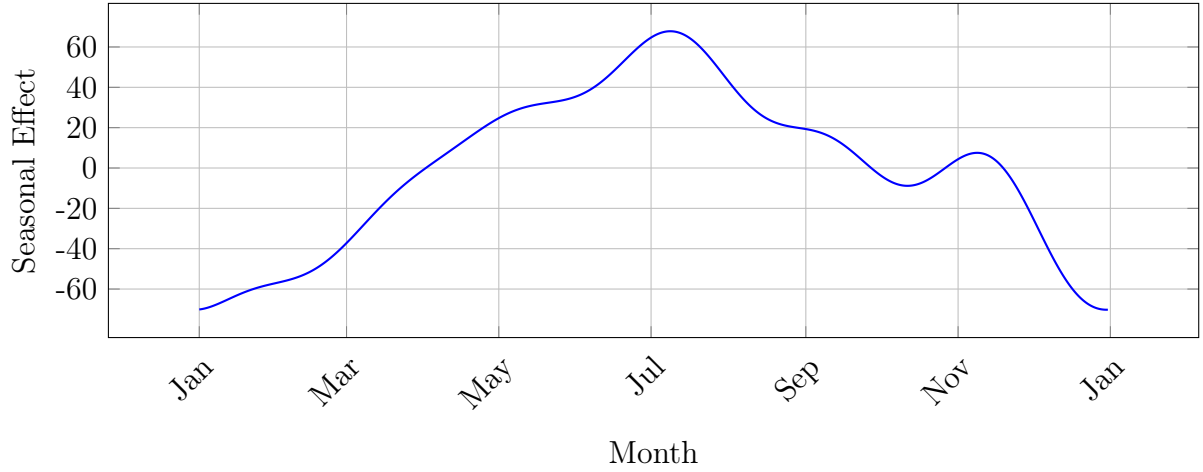


Figure 5.6: NeuralProphet Yearly Seasonality Analysis

Fig. 5.3 illustrates NeuralProphet’s ability to predict future order quantities based on historical data, providing vendors with reliable projections for inventory planning. Supporting this predictive capability, Fig. 5.4 reveals the underlying long-term growth trajectory that informs strategic business decisions and expansion planning. Further enhancing these insights, Fig. 5.5 and Fig. 5.6 demonstrate how our model accurately captures both weekly variations and annual seasonal patterns in vendor ordering behavior. This comprehensive forecasting capability is crucial for vendors and suppliers to anticipate demand fluctuations, optimize inventory management, and plan logistics operations efficiently throughout the year.

Chapter 6

Conclusion and Future Enhancements

The ASTRO platform is a comprehensive solution designed to empower small-scale vendors by addressing their core operational challenges. By facilitating collaborative purchasing, optimizing logistics, and offering data-driven financial services, ASTRO significantly enhances the competitive edge of small vendors, enabling them to compete more effectively with large retail chains. Its multifaceted approach not only reduces operational costs and improves efficiency but also equips vendors with financial resources and data-driven insights essential for informed decision making and strategic growth.

Through collaborative purchasing, ASTRO enables vendors to access bulk pricing discounts, reducing the cost of goods sold and increasing profitability. The platform's logistics optimization feature streamlines delivery routes, minimizing transportation costs and ensuring timely deliveries. This not only enhances operational reliability but also improves customer satisfaction. Additionally, ASTRO integrates data driven financial services to address the critical challenge of limited access to capital. By leveraging transaction data, the platform offers tailored credit assessments and loan options, empowering vendors to expand their businesses, manage inventory effectively, and invest in technology.

A key component of ASTRO is its real-time demand forecasting tool, powered by NeuralProphet. This predictive capability enables vendors to anticipate market demand with high accuracy, optimizing inventory levels and reducing the risks of overstocking or stockouts. By ensuring vendors can meet customer needs promptly and efficiently, ASTRO enhances overall operational efficiency. The platform is also designed with a user-friendly interface, making its features accessible to vendors with varying levels of technical expertise.

Overall, ASTRO fosters economic resilience and sustainability within local communities by empowering small-scale vendors to thrive in a competitive marketplace. By addressing critical challenges such as limited purchasing power, inefficient logistics, re-

stricted financial access, and a lack of data driven decision making, ASTRO contributes to the long term success and stability of small vendors. In doing so, it promotes a more diverse and robust local economy.

Our implementation demonstrates significant improvements in supply chain efficiency and cost reduction through order aggregation and route optimization. The advanced forecasting capabilities powered by NeuralProphet provide vendors with unprecedented insights into future demand patterns, as evidenced by our comprehensive analysis results.

Traditional Delivery Approach In the conventional model, vendors receive deliveries through separate routes or small grouped deliveries, resulting in significant route overlap and inefficiency. This approach required a total travel distance of 35,606.06 km with associated route costs of Rs. 36,929.20. The excessive distance resulted in high fuel consumption (1,811.34 L) and fuel costs (Rs. 171,135.40).

Collaborative Delivery Approach With our platform, deliveries are consolidated into a single optimized route serving all vendors. Our Environmental-Adaptive Genetic Algorithm (EAGA) produced a route of just 6,993.30 km with total costs of Rs. 7,254.56. This dramatically reduced fuel consumption to 357.43 L with corresponding fuel costs of Rs. 33,769.57.

Impact Assessment As shown in Table 5.3, the collaborative delivery model demonstrates an 80.4% reduction in total distance traveled and route costs, with fuel savings of 80.3%. These substantial efficiencies are achieved through intelligent route optimization that eliminates redundant travel paths. The model provides small-scale vendors with logistics cost advantages traditionally reserved for large enterprises, enabling them to compete more effectively while maintaining their operational independence.

The results of our implementation validate the core thesis that technology-driven collaboration can substantially improve the competitive position of small-scale vendors in today's market. By combining advanced machine learning techniques with practical business solutions, ASTRO demonstrates that sophisticated technologies can be made accessible and valuable to small businesses that traditionally lack access to such tools. The platform's ability to decompose complex demand patterns into interpretable compo-

nents—trend, weekly cycles, and annual seasonality—empowers vendors with actionable intelligence previously available only to large retail corporations with dedicated analytics teams.

While ASTRO has made significant strides in empowering small-scale vendors, there are numerous opportunities for future enhancements and expansions to further augment its impact and effectiveness. The following areas outline potential future developments:

1. **Advanced Forecasting and Predictive Analytics:** Integrate more sophisticated machine learning algorithms to enhance the accuracy and granularity of demand forecasting, allowing vendors to anticipate market trends more accurately and optimize inventory management.
2. **AI-Enhanced Credit Assessment:** Develop AI-driven credit scoring models that utilize a broader range of data points, including transaction history and purchasing patterns, to provide personalized and accurate financial services.
3. **Integration with Broader Supply Chains:** Establish partnerships with larger suppliers and logistics providers to expand ASTRO's logistics optimisation features, enabling vendors to access a wider range of products at competitive prices.
4. **Sustainability Initiatives:** Incorporate environmentally sustainable practices[10], such as carbon footprint tracking and eco-friendly route planning, to align vendors with growing consumer demand for eco-friendly products.
5. **Customizable Vendor Portals and Data Insights:** Develop customizable dashboards for tailored analytics based on vendor-specific needs, enabling strategic business decisions.
6. **Expansion of Financial Services:** Introduce more financial products, such as insurance and investment options, to provide vendors with comprehensive financial support.
7. **Enhanced Collaboration Features:** Implement tools for communication and resource sharing among vendors, fostering a sense of community and mutual support.
8. **Localization and Customization:** Adapt ASTRO to different regions' specific needs and regulatory requirements, ensuring relevance across diverse contexts.

9. **Integration with E-Commerce Platforms:** Integrate ASTRO with popular e-commerce platforms to enable seamless online ordering and sales management, expanding vendors' market reach.
10. **User Training and Support Programs:** Provide training programs to help vendors maximize ASTRO's benefits, enhancing adoption and success.
11. **Data Privacy and Security Enhancements:** Implement advanced measures for data protection, building user trust and ensuring data confidentiality.
12. **Feedback and Continuous Improvement Mechanism:** Establish a feedback system to gather user input and improve ASTRO's features and functionalities over time.
13. **Scalability and Performance Optimisation:** Ensure the platform's scalability and performance can handle increased users and data as ASTRO grows.
14. **Exploration of New Markets and Industries:** Expand ASTRO's framework to apply to other markets and industries beyond small-scale vendors, opening new avenues for growth and impact.

References

- [1] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [2] J. M. John, O. Shobayo, and B. Ogunleye, “An exploration of clustering algorithms for customer segmentation in the uk retail market,” *Analytics*, vol. 2, no. 4, pp. 809–823, 2023.
- [3] J. C. Perafan-Lopez, V. L. Ferrer-Gregory, C. Nieto-Londoño, and J. Sierra-Pérez, “Performance analysis and architecture of a clustering hybrid algorithm called fa+ga-dbscan using artificial datasets,” *Entropy*, vol. 24, no. 7, p. 875, 2022.
- [4] B. M. Baker and M. Ayechev, “A genetic algorithm for the vehicle routing problem,” *Computers and Operations Research*, vol. 30, no. 5, pp. 787–800, 2003. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054802000515>
- [5] E. S. Gardner Jr and D. G. Dannenbring, “Forecasting with exponential smoothing: Some guidelines for model selection,” *Decision sciences*, vol. 11, no. 2, pp. 370–383, 1980.
- [6] O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, “Neuralprophet: Explainable forecasting at scale,” 2021. [Online]. Available: <https://arxiv.org/abs/2111.15397>
- [7] J. H. Ward Jr, “Hierarchical grouping to optimize an objective function,” *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.
- [8] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, vol. 5. University of California press, 1967, pp. 281–298.

- [9] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [10] X. Wang and L. Wang, “Green routing optimization for logistics distribution with path flexibility and service time window,” in *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*, 2018, pp. 1–5.

Appendix A: Presentation

A.S.T.R.O. (Advanced Supply and Trade Resource Optimisation)

Project Group 03

Amel Chandra (U2109009)
Bharath S (U2109018)
Jeepaul Vilsan (U2109033)
Shane George Salphie (U2109063)

Under the Guidance of

Dr. Nikhila T. Bhuvan,
Associate Professor,
Department of CSBS



Dept. of Computer Science and Business Systems,
RSET

INTRODUCTION

- **Introduction**
 - ASTRO helps small vendors with purchasing and delivery challenges.
 - Vendors struggle with high costs, poor supply chains, and low bargaining power.
 - Key features:
 - Order aggregation for bulk discounts.
 - Demand forecasting for better planning.
 - Optimized supplier selection for cost savings.
 - Uses Genetic Algorithm for efficient deliveries.
 - Compares supplier selection methods for best deals.
 - Provides a scalable, cost-effective supply chain solution.
 - Boosts competitiveness and sustainability for vendors.

2

PROBLEM STATEMENT

- Design a collaborative platform which
 - Enables multiple vendors to aggregate orders, facilitating bulk purchasing and cost savings.
 - Demand forecasting to predict product needs accurately
 - Implement route optimization techniques to establish an efficient delivery network.
- This approach aims to enhance supply chain efficiency, reduce operational costs, and strengthen the market position of small-scale vendors.

3

OBJECTIVES

- **Objectives of the Project**
 - Order Aggregation : Allow vendors to combine purchases for bulk discounts.
 - Demand Forecasting : Predict demand to optimize purchasing.
 - Logistics Optimization : Reduce delivery costs using Green Routing Optimization.
 - Supplier Selection : Identify cost-effective and reliable suppliers.
 - Vendor Competitiveness : Help small vendors compete with large retail chains.
- **Importance and relevance of the project**

ASTRO helps small-scale vendors reduce costs, improve supply chain efficiency, and compete with large retail chains. By enabling order aggregation, demand forecasting, and optimized logistics, the platform ensures better pricing, faster deliveries, and smarter purchasing decisions. It promotes cost-effectiveness, scalability, and sustainability, making it highly relevant in today's competitive market.

4

PROJECT SCOPE

- Project Scope: Boundaries and Focus
 - ASTRO is designed to enhance procurement and logistics for small-scale vendors by leveraging order aggregation, demand forecasting, and supplier selection optimization. The platform aims to reduce procurement costs, streamline deliveries, and improve vendor-supplier collaboration, making supply chain management more efficient, data-driven, and cost-effective.
- What is Covered
 - Order Aggregation** : Enables vendors to combine orders for bulk purchase discounts.
 - Demand Forecasting** : Utilizes data-driven analytics to predict demand and optimize procurement.
 - Logistics Optimization** : Implements Green Routing Optimization for cost-efficient and timely deliveries.
 - Supplier Coordination** : Streamlines vendor-supplier interactions for efficient order processing
- What is Excluded
 - Inventory Management** : Does not handle warehouse stock tracking or inventory control.
 - Retail Consumer Sales** : Exclusively focuses on vendor-supplier transactions, not direct customer sales.
 - Financial Transactions** : Does not process payments, financing, or lending services.

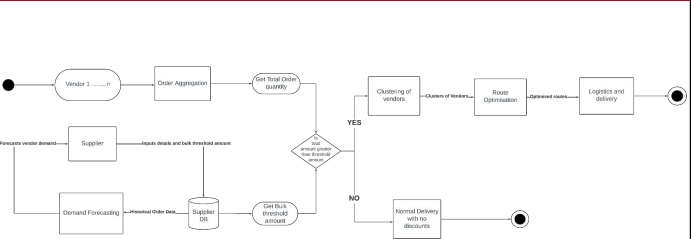
5

METHODOLOGY

- Tools & Technologies
 - Frontend** : Flutter
 - Backend** : FastAPI Python
 - Programming Languages** : Python and Dart
 - Cloud** : Firebase
 - APIs & Integrations** : Google Maps API
- Dataset
 - Store Item Demand Forecasting Challenge - Kaggle Dataset ([Kaggle Link](#))
 - Random Google Maps Business Locations - South India
- Algorithms Used
 - Route Optimization** : Genetic Algorithm
 - Order Aggregation** : DBSCAN alongwith Genetic Algorithm
 - Demand Forecasting** : NeuralProphet ([Link](#))

6

SYSTEM DESIGN



7

IMPLEMENTATION

- Data Collection & Storage
 - Orders are stored in Google Firestore with details like supplier ID, vendor location, order quantity, and status.
 - Suppliers set a Minimum Order Quantity (MOQ) for aggregation.
- Order Aggregation using DBSCAN alongwith GENETIC ALGORITHM
 - Location-Based Clustering:Orders from vendors are clustered using DBSCAN based on real-world (Haversine) distances.
 - Dynamic Parameter Tuning:A Genetic Algorithm optimizes DBSCAN's parameters (eps, min_samples) for better clustering accuracy..
 - MOQ Validation & Fallback:Clusters are accepted only if they meet Minimum Order Quantity (MOQ); otherwise, orders remain pending. Default DBSCAN is used if order volume is low.

8

IMPLEMENTATION

- Order Aggregation using DBSCAN alongwith GENETIC ALGORITHM
 - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
 - Unsupervised clustering algorithm.
 - Groups dense points; marks low-density points as outliers.
 - No need to predefine the number of clusters (unlike k-means).
 - Why Use DBSCAN Instead of Other Algorithms?
 - Better for Geospatial Clustering: Excels at location-based clustering.
 - Automatic Cluster Discovery: No need to predefine the number of clusters.
 - Handles Noise & Outliers: Avoids forcing outliers into clusters.
 - Flexible Shapes: Detects irregular cluster shapes (unlike k-means).
 - Why Use GENETIC ALGORITHM
 - Traditional DBSCAN requires fixed eps and min_samples, which don't adapt to changing vendor order patterns.
 - To overcome this, we combine DBSCAN with a Genetic Algorithm (GA) that automatically optimizes these parameters for each dataset.

9

IMPLEMENTATION

- Order Aggregation using DBSCAN Clustering
 - How DBSCAN alongwith GENETIC ALGORITHM Works?
 - Input: Order coordinates (latitude, longitude), and quantities.
 - Distance Calculation: Compute Haversine distance matrix between all vendor locations.
 - Optimization: Run Genetic Algorithm to find optimal eps and min_samples for DBSCAN.
 - DBSCAN Clustering:
 - Run DBSCAN using the GA-tuned parameters.
 - Identify clusters of geographically close orders.
 - MOQ Validation:
 - Each cluster is accepted only if it meets MOQ (plus buffer).
 - Outliers or low-quantity clusters are kept pending.
 - Advantages Over Traditional DBSCAN
 - Dynamic Parameter Tuning: Automatically finds the best eps and min_samples using Genetic Algorithm.
 - MOQ-Aware Clustering: Ensures clusters are only accepted if they meet Minimum Order Quantity.
 - Adaptive to Varying Data Volumes: Switches between optimized and default DBSCAN based on order count.
 - Real-World Distance Support: Uses Haversine distance for accurate location-based clustering.
 - Robust Outlier Handling: Effectively filters noise and avoids misgrouping of outliers.

10

IMPLEMENTATION

- Route Optimization using Ant Colony Optimization (ACO) :
 - Aggregated orders require efficient delivery routes.
 - ACO algorithm is used to optimize the shortest and most cost-effective route from supplier to vendors.
 - This minimizes fuel costs and delivery time.
- Demand Forecasting using NeuralProphet :
 - Historical order data is analyzed using NeuralProphet (an advanced time-series forecasting model).
 - Forecasts future demand patterns, helping suppliers plan inventory and pricing.

11

IMPLEMENTATION

- Route Optimization using Ant Colony Optimization (ACO) :
 - Ant Colony Optimization (ACO) for Route Optimization
 - A metaheuristic optimization algorithm inspired by the foraging behavior of ants.
 - It is used to find the optimal route for logistics and transportation problems.
 - Why Use ACO?
 - Efficient Routing: Finds cost-effective and time-efficient paths.
 - Dynamic Adaptation: Adjusts to real-time changes in routes.
 - Scalable: Works with large datasets of locations.
 - Energy-Efficient: Minimizes fuel costs and distance traveled.
 - How ACO Works?
 - Define Parameters:
 - Pheromone Influence (α): Importance of past solutions.
 - Heuristic Influence (β): Importance of distance.
 - Evaporation Rate (ρ): Prevents unlimited pheromone accumulation.

12

IMPLEMENTATION

- **Route Optimization using Ant Colony Optimization (ACO) :**
 - **How ACO Works?**
 - Route Construction:
 - Ants are placed at random locations.
 - Each ant builds a route probabilistically based on:
 - Pheromone trails (past successful paths).
 - Heuristic value (shorter distances, lower fuel cost).
 - **ACO in this Project**
 - Extracts Order Locations (latitude & longitude).
 - Constructs Distance Matrix using Google Maps API.
 - Optimizes Fuel Cost by considering cost per liter.
 - Builds Routes using probability-based selection.
 - Applies 2-Opt Local Search for further route optimization.
 - Ensures Eco-Friendly Routing by minimizing unnecessary travel.
 - Stops Early if no improvement is detected to save computation time.

13

IMPLEMENTATION

- **Route Optimization using Genetic Algorithm (GA):**
 - **Genetic Algorithm (GA) for Route Optimization**
 - Inspired by natural selection, GA evolves multiple solutions to find the best delivery routes.
 - More efficient than traditional routing by exploring multiple possibilities in parallel.
 - **Why Use Genetic Algorithms?**
 - Efficient Optimization: Faster than brute-force search.
 - Handles Constraints: Adapts to time windows & traffic.
 - Scalable: Works with large datasets.
 - Avoids Local Minima: Prevents getting stuck in suboptimal solutions.
 - **How It Works:**
 - Key Components:
 - Population: Set of possible routes.
 - Fitness Function: Evaluates route efficiency.
 - Selection: Picks best routes for reproduction.
 - Crossover: Combines parent routes.
 - Mutation: Introduces random changes for diversity.

14

IMPLEMENTATION

- **Route Optimization using Genetic Algorithm (GA):**
 - **How It Works:**
 - Initialize: Generate routes (Greedy Nearest Neighbor, 2-Opt).
 - Evaluate: Compute distance, favor shorter routes.
 - Select: Pick best routes (Tournament Selection).
 - Crossover & Mutation: Mix routes (OX, ERX) + random changes.
 - Refine: Apply 2-Opt for optimization.
 - Converge: Stop when no further improvement.
 - **GA in this Project**
 - Extracts order locations (latitude & longitude).
 - Computes Distance Matrix using Google Maps API & Haversine Formula.
 - Initializes diverse route solutions for efficiency.
 - Runs GA with adaptive mutation rates for optimization.
 - Applies 2-Opt Local Search for final refinement.
 - Selects the best route with minimal delivery time & cost.

15

CHALLENGES

- **Order Aggregation Challenges**
 - **Issue:**
 - Clustering vendor orders using DBSCAN required careful tuning of parameters like eps (neighborhood radius) and min_samples.
 - Additionally, clustering had to satisfy **Minimum Order Quantity (MOQ)** constraints while reducing outliers.
 - **Solution:**
 - Replaced manual tuning with **Genetic Algorithm (GA)** to dynamically optimize eps and min_samples for DBSCAN.
 - GA uses a fitness function that maximizes valid clusters while minimizing noise.
 - Integrated MOQ checks post-clustering to ensure only qualifying clusters are confirmed for aggregation.
 - Enhanced outlier handling by isolating orders that don't meet MOQ, marking them as pending for future rounds.
- **Route Optimization Challenges**
 - **Issue:**
 - Genetic Algorithm required careful tuning of parameters such as population size, mutation rate, and crossover strategy to effectively handle real-world constraints like traffic patterns, delivery time windows, and vendor/supplier availability.

16

CHALLENGES

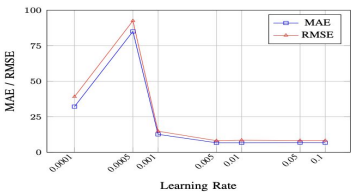
- **Solution:**
 - Optimized GA parameters to balance exploration and exploitation in the route search process.
 - Optimized GA parameters to balance exploration and exploitation in the route search process.
 - Integrated domain-specific heuristics into the fitness function to account for real-time delivery constraints such as time windows, capacity limits, and location clustering.
 - Ensured diversity in the population using mutation and elitism to avoid premature convergence.

Demand Forecasting Issue

- **Issue:**
 - Forecasting demand using NeuralProphet required high-quality historical data.
- **Solution:**
 - Applied data cleaning techniques (removing inconsistencies, handling missing values).
 - Used hyperparameter tuning to improve forecast accuracy.

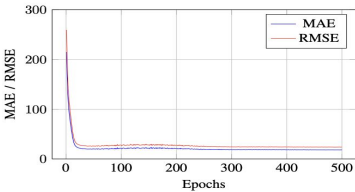
RESULTS AND EVALUATION

- We used Prophet and NeuralProphet for comparison, but we chose NeuralProphet for our work due to its active development and performance benefits.
- Prophet is discontinued, while NeuralProphet is actively maintained and improved. Bug fixes and feature enhancements are still happening in NeuralProphet.
- NeuralProphet is built on PyTorch, it can leverage GPUs, making it faster for large datasets compared to Prophet as it uses Stan which does not support GPU computation.



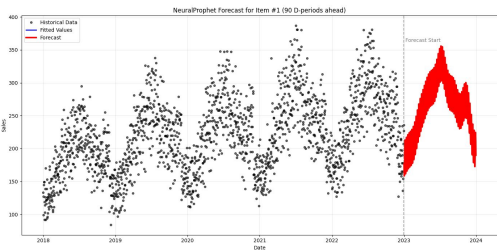
RESULTS AND EVALUATION

Effect of Epochs on MAE and RMSE

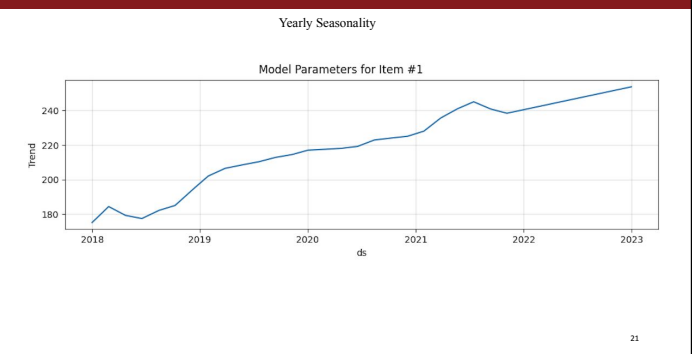


RESULTS AND EVALUATION

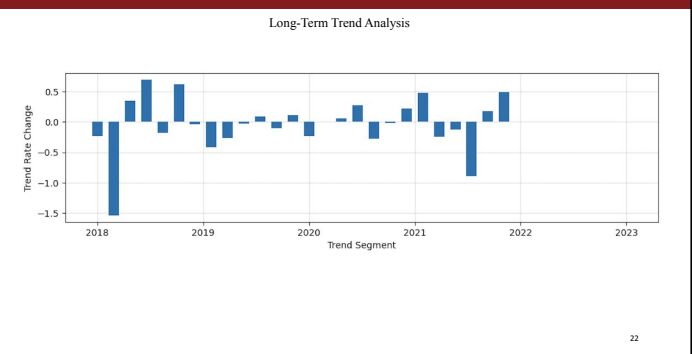
NeuralProphet Demand Forecasting



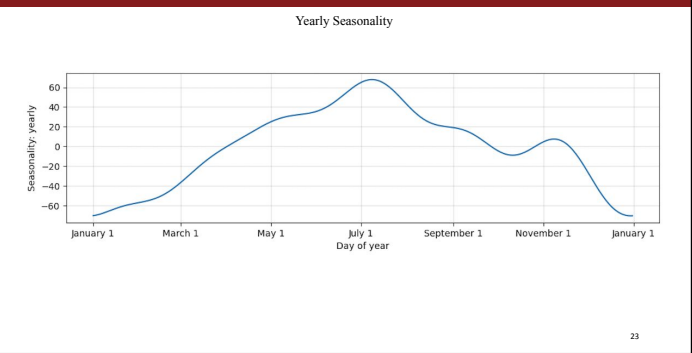
RESULTS AND EVALUATION



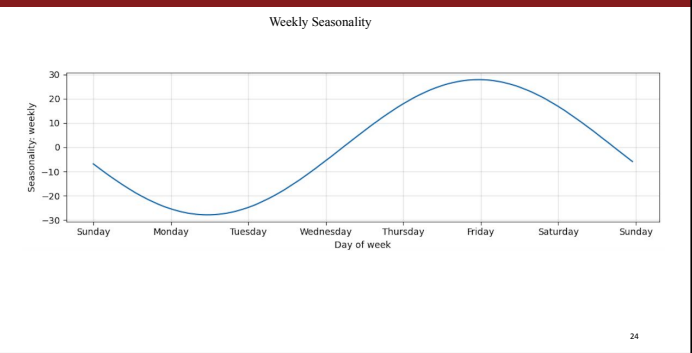
RESULTS AND EVALUATION



RESULTS AND EVALUATION



RESULTS AND EVALUATION



RESULTS AND EVALUATION

We compared Genetic Algorithm (GA), Ant Colony Optimization (ACO), and the Christofides Algorithm for solving the route optimization problem in our system.

Genetic Algorithm (GA) uses evolutionary techniques such as crossover, mutation, and selection to iteratively evolve better routes. It is highly adaptable, capable of handling complex constraints like delivery windows, capacity limits, and varying cluster sizes.

ACO is designed specifically for routing problems and uses pheromone trails to reinforce promising paths. While it can dynamically adapt to optimal routes, we found it prone to premature convergence, especially in scenarios with fewer iterations or complex constraints.

In our evaluation, GA consistently outperformed ACO, finding shorter and more feasible delivery routes. This may be due to GA's ability to explore the solution space more effectively, avoiding local optima through genetic variation.

The Christofides Algorithm, while useful for theoretical approximation of the Travelling Salesman Problem (TSP), lacked flexibility. Its static nature made it less suitable for real-world delivery constraints where adaptability is crucial.

Based on these comparisons, Genetic Algorithm was chosen for route optimization due to its superior adaptability, scalability, and ability to deliver high-quality solutions under real-world conditions.

RESULTS AND EVALUATION

Algorithm	Vendors	Dist. (km)	Cost (Rs.)	Time (s)
ACO	20	5197.54	29811.51	23.26
	50	8333.63	47618.80	50.43
	100	9648.29	54444.64	148.69
GA	20	4703.90	22320.78	20.53
	50	6993.30	33770.55	60.75
	100	8581.06	40956.56	142.21

RESULTS AND EVALUATION

We evaluated Agglomerative Clustering, K-means, DBSCAN, and a Genetic Algorithm-enhanced DBSCAN (GA-DBSCAN) to group orders geographically. GA-DBSCAN outperformed all others in silhouette scores and noise reduction for datasets of 100 orders and above.

- **DBSCAN** works well for smaller datasets due to low overhead.
- **GA-DBSCAN** adaptively selects parameters (ϵ and MinPts), ensuring optimal clustering even under dynamic order distributions.
- **K-means** and **Agglomerative** produced higher cluster counts but lacked flexibility in handling noise.

RESULTS AND EVALUATION

Algorithm	Orders	Time (s)	Clusters	Outliers	Silh. Score
DBSCAN	25	0.002	7	16	0.350
	50	0.002	13	32	0.330
	100	0.003	28	68	0.392
	150	0.004	39	99	0.380
	200	0.005	63	118	0.375
	300	0.009	85	187	0.361
GA-DBSCAN	350	0.010	98	219	0.355
	25	0.020	0	25	N/A
	50	0.025	0	50	N/A
	100	0.025	11	89	0.735
	150	0.030	12	138	0.780
	200	0.036	39	156	0.726
K-means	300	0.048	35	262	0.715
	350	0.055	45	301	0.629
	25	0.030	6	18	0.358
	50	0.004	12	34	0.324
	100	0.004	26	72	0.392
	150	0.005	37	103	0.390
Agglomerative	200	0.032	61	122	0.385
	300	0.007	84	189	0.363
	350	0.008	94	227	0.350
	25	0.002	6	18	0.341
	50	0.002	12	34	0.337
	100	0.002	27	70	0.393
	150	0.004	38	101	0.358
	200	0.005	61	122	0.359
	300	0.007	84	189	0.359
	350	0.009	95	225	0.349

RESULTS AND EVALUATION

Cost-Benefit Analysis

In a 50-vendor scenario, our collaborative route model showed:

- Distance savings: 80.4%
- Route cost savings: 80.4%
- Fuel cost savings: 80.3%

Metric	Trad.	Collab.	Savings
Distance (km)	35,606	6,993	28,613 (80.4%)
Route Cost (Rs.)	36,929	7,255	29,675 (80.4%)
Fuel (L)	1,811	357	1,454 (80.3%)
Fuel Cost (Rs.)	171,135	33,770	137,366 (80.3%)

CONCLUSION

- Summary of the Work Done
 - ASTRO was developed to enhance procurement and logistics for small-scale vendors.
 - Implemented DBSCAN with GENETIC algorithm for order aggregation, Genetic Algorithm for route planning, and NeuralProphet for demand forecasting.
- Limitations of the Project
 - Accuracy of demand forecasting depends on data availability.
 - Real-world implementation may require additional infrastructure and funding.
 - Inventory management is not covered in the current scope.
- Suggestions for Future Work
 - Improve supplier selection with advanced ML models.
 - Integrate real-time tracking for better delivery monitoring.
 - Expand dataset coverage to improve forecasting accuracy.

REFERENCES

• X. Wang and L. Wang, “Green routing optimization for logistics distribution with path flexibility and service time window,” in 15th International Conference on Service Systems and Service Management (ICSSSM), 2018, pp. 1–5.

• E. M. U. S. B. Ekanayake, S. P. C. Perera, W. B. Daundasekara, and Z. A. M. S. Juman, “A modified ant colony optimization algorithm for solving a transportation problem,” J. Adv. Math. Comput. Sci., vol. 35, no. 5, pp. 83–101, Aug. 2020.

• J. Zhou, “Research on multi-objective green vehicle path optimization based on whale algorithm,” in International Conference on Networking, Informatics and Computing (ICNETIC), 2023, pp. 385–389.

• K. A. Mamoun, L. Hammadi, A. E. Ballout, A. G. Novaes, and E. S. D. Cursi, “Vehicle routing optimization algorithms for pharmaceutical supply chain: A systematic comparison,” Transp. Telecommun. J., vol. 25, no. 2, pp. 161–173, 2024.

• A. Maroof, B. Ayvaz, and K. Naeem, “Logistics optimization using hybrid genetic algorithm (HGA): A solution to the vehicle routing problem with time windows (VRPTW),” IEEE Access, vol. 12, pp. 36974–36989, 2024.

REFERENCES

• S. J. Taylor and B. Letham, “Forecasting at scale,” PeerJ Preprints, vol. 5, p. e3190v2, Sep. 2017.

• O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, “NeuralProphet: Explainable forecasting at scale,” 2021.

• X. Ai, Y. Yue, H. Xu, and X. Deng, “Optimizing multi-supplier multi-item joint replenishment problem for non-instantaneous deteriorating items with quantity discounts,” PLOS ONE, vol. 16, no. 2, pp. 1–22, Feb. 2021.

• S. B. Jouda and S. Krichen, “A genetic algorithm for supplier selection problem under collaboration opportunities,” J. Exp. Theor. Artif. Intell., vol. 34, no. 1, pp. 53–79, 2022.

• M. Seifbarghy, M. Shoeib, and D. Pishva, “Coordination of a single-supplier multi-retailer supply chain via joint ordering policy considering incentives for retailers and utilizing economies of scale,” Scientia Iranica, 2022.

SCREENSHOTS

ASTRO Supply Chain HubFeaturesHelpAbout

Log In

Get Started

Revolutionize Your Supply Chain with ASTRO

Connect suppliers and vendors seamlessly on our cutting-edge supply chain management platform

Register Now

Learn More

ASTRO

Supply Chain Management Platform

Create an Account

Register as a Supplier

Register as a Vendor

Company Name

Your company name

Email

your@email.com

Password

password

Confirm Password

password

Register

Already have an account? Log In Here

33

SCREENSHOTS

Powerful Features

Our platform offers comprehensive tools for both suppliers and vendors

Product Management

Suppliers can easily add, update, and manage products with detailed information and stock levels

Streamlined Ordering

Vendors can browse products from multiple suppliers and place orders with just a few clicks

Order Tracking

Track how order status updates with complete visibility throughout the fulfillment process

ASTRO Summary

Dashboard

Products

Orders

Settings

Log Out

Log In

Supplier Dashboard

View your basic, core dashboard. Here's what's happening with your products today.

Total Orders

7

Completed

Pending Orders

7

Pending

Total Revenue

\$66,232

Completed

Low Stock Products

0

Low Stock

Order Status

Order status of order resources

100%

Monthly Orders

Number of orders per month

Star

Monthly Orders

34

SCREENSHOTS

ASTRO Summary

Dashboard

Products

Orders

Settings

Log Out

Log In

Products

Manage your product inventory

+ Add Product

Product Inventory

You have 10 products in your inventory

Search products...

Name	Category	Price	Inventory	Min. Stock Qty	Status	Action
Soap	Household Items	\$10.00	100	50	In Stock	
Handwash	Hygiene Products	\$12.00	85	100	In Stock	
Toiletries	Cleaning Supplies	\$18.00	50	100	In Stock	
Shampoos	Hygiene Products	\$15.00	90	100	In Stock	
Conditioner	Laundry Care	\$16.00	94	100	In Stock	
Detergent Powder	Hygiene Products	\$18.00	88	100	In Stock	
Softener	Cleaning Supplies	\$10.00	100	100	In Stock	

ASTRO Summary

Dashboard

Products

Orders

Settings

Log Out

Log In

Orders

Manage orders from your vendors

Order Management

You have 10 orders in your

Search orders...

Order ID

Product

Quantity

Total

Status

Date

Action

ASTRO-00123456789	Soap	50	\$450.00	Out of Stock	11/14/2023	
ASTRO-00123456789	Soap	50	\$450.00	Out of Stock	11/14/2023	
ASTRO-00123456789	Soap	50	\$450.00	Out of Stock	11/14/2023	
ASTRO-00123456789	Soap	50	\$450.00	Out of Stock	11/14/2023	
ASTRO-00123456789	Soap	50	\$450.00	Out of Stock	11/14/2023	
ASTRO-00123456789	Soap	50	\$450.00	Out of Stock	11/14/2023	
ASTRO-00123456789	Soap	50	\$450.00	Out of Stock	11/14/2023	
ASTRO-00123456789	Soap	50	\$450.00	Out of Stock	11/14/2023	
ASTRO-00123456789	Soap	50	\$450.00	Out of Stock	11/14/2023	
ASTRO-00123456789	Soap	50	\$450.00	Out of Stock	11/14/2023	

35

SCREENSHOTS

ASTRO Summary

Dashboard

Products

Orders

Settings

Log Out

Log In

Products

Manage your product inventory

+ Add Product

Product Inventory

You have 10 products in your inventory

Search products...

Name	Category	Price	Inventory	Min. Stock Qty	Status	Action
Soap	Household Items	\$10.00	100	50	In Stock	
Handwash	Hygiene Products	\$12.00	85	100	In Stock	
Toiletries	Cleaning Supplies	\$18.00	50	100	In Stock	
Shampoos	Hygiene Products	\$15.00	90	100	In Stock	
Conditioner	Laundry Care	\$16.00	94	100	In Stock	
Detergent Powder	Hygiene Products	\$18.00	88	100	In Stock	
Softener	Cleaning Supplies	\$10.00	100	100	In Stock	

ASTRO Summary

Dashboard

Products

Orders

Settings

Log Out

Log In

Settings

Manage your account and preferences

Account

Notifications

Security

Account Information

Update your account details and preferences

First Name

Last Name

Username

Phone

Email

200154712

Shop Name

Star

Company Email

supplier

Address

Building/Room No.

Post/Zip

Region/Country/State

Company/Department

City

State

Zip

36

THANK YOU

Appendix A: Vision, Mission, PO, PSO, and CO

RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)

Vision

To evolve into a premier technological and research institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

Vision

To evolve into a department of excellence in information technology by the creation and exchange of knowledge through leading-edge research, innovation and services, which will in turn contribute towards solving complex societal problems and thus building a peaceful and prosperous mankind.

Mission

To impart high-quality technical education, research training, professionalism and strong ethical values in the young minds for ensuring their productive careers in industry and academia so as to work with a commitment to the betterment of mankind.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex Problems:** Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Team Work:** Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project Management and Finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.
12. Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Business Systems Programme will:

- **PSO 1: Programming and Software Development Skills**

Demonstrate ability to analyse, design, and implement software solutions incorporating various programming concepts.

- **PSO 2: Engineering Management and Collaboration**

Comprehend professional, managerial, and financial aspects of business and collaborate on the design, implementation, and integration of engineering solutions.

- **PSO 3: Decision-Making and Analytical Techniques in Engineering and Business**

Create, select, and apply appropriate techniques and business tools, including prediction and data analytics, for complex engineering activities and business solutions.

Course Outcomes (CO)

After successful completion of the course, the students will be able to:

- CO1: Model and solve real-world problems by applying knowledge across domains (Cognitive knowledge level: Apply).
- CO2: Develop products, processes, or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).
- CO3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).
- CO4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).
- CO5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyse).
- CO6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

Appendix B: CO-PO-PSO Mapping

Mapping of Course Outcomes (CO) with Programme Outcomes (PO)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	2	2	1	2	2	2	1	1	1	1	2
CO 2	2	2	2		1	3	3	1	1		1	1
CO 3									3	2	2	1
CO 4					2			3	2	2	3	2
CO 5	2	3	3	1	2							1
CO 6					2			2	2	3	1	1

Mapping of Course Outcomes (CO) with Programme Specific Outcomes (PSO)

	PSO 1	PSO 2	PSO 3
CO 1	3	1	2
CO 2	3	3	2
CO 3		3	
CO 4		1	1
CO 5	1	1	1
CO 6		2	