Integrated Forecasting System for Inventory Management in Oil Depots for PGS

Data Science Project Report

Elaborated by:

Koussay Bedhief

Mohamed Fedi Ben Aoun

Mohamed Amine Tlili

Mohamed Laatar

Mohamed Aziz Loukil

Supervised by:

Mrs. Wiem Trabelsi

Mrs. Hanene Ennine

Mrs. Mabrouka Dchicha

Mrs. Amal Benchikh

14/05/2024

Table of content

Table of Figures

Acknowledgments

We extend our sincere gratitude to **Mrs. Wiem Trabelsi** and **Mrs. Hanene Ennine**, our esteemed project supervisors, **Mrs. Mabrouka Dchicha**, our English for Engineers, and **Mrs. Amal Benchikh**, our innovation module teacher, whose unwavering support, expertise, and encouragement have played a pivotal role in shaping our journey. Their dedication to fostering our growth and sharing invaluable insights has been indispensable throughout this project.

Furthermore, we express our profound appreciation to PGS for entrusting us with the opportunity to undertake a project that exceeded our customary scope of work. Their confidence in our abilities has inspired us to deliver innovative solutions and tackle complex challenges with determination.

Lastly, we acknowledge and value the collaborative spirit and dedication of each member of our team. Together, we have navigated challenges, leveraged diverse perspectives, and celebrated milestones. We extend our heartfelt thanks to each team member for their unwavering commitment and camaraderie, which have been instrumental in our collective success.

Introduction

In the current era of data science, industries are increasingly turning to advanced analytics to drive efficiency and unlock insights. The oil and gas sector is one of the leading industries that rely on data-driven decision-making and complex operations. Therefore, the sector is at the forefront of this trend and continuously seeks innovative solutions to overcome its challenges.

This project is entirely focused on implementing data science to address key challenges within the oil and gas industry. As data scientists, our mission is to construct comprehensive models that can anticipate order demands, estimate accurate oil stock levels, optimize delivery routes, detect leaks, and identify malfunctioning equipment within tanks.

Our team will leverage advanced analytics techniques such as machine learning algorithms and predictive modeling to develop these models. These models will be designed to provide accurate predictions and insights that enable decision-makers to make informed decisions that drive operational efficiency, reduce risks, and enhance safety.

Our goal is to revolutionize how the industry manages its operations, moderates risks, and enhances efficiency through the adoption of advanced analytics and machine learning techniques. We believe that our approach will help the industry to unlock untapped potential and achieve a competitive advantage.

1 Business Understanding

1.1 Company Introduction

1.1.1 Overview:

Established in Tunisia, PGS is a dynamic company boasting a turnover exceeding ten million euros, showcasing over 15 years of industry expertise. With a dedicated team of over one hundred employees, PGS has established itself as a key player in the market. The company has successfully served over thirty customers, solidifying its reputation as a reliable partner.

Figure 1: PGS Logo

1.1.2 Global Presence:

PGS operates internationally, maintaining a strong presence in over 10 countries. The company's headquarters are strategically located in Tunisia, and it has further expanded its reach with branches in Algeria and Morocco. Additionally, PGS has established its footprint in diverse regions, including Libya, Egypt, Senegal, Cote d’Ivoire, Cameroon, and Nigeria.

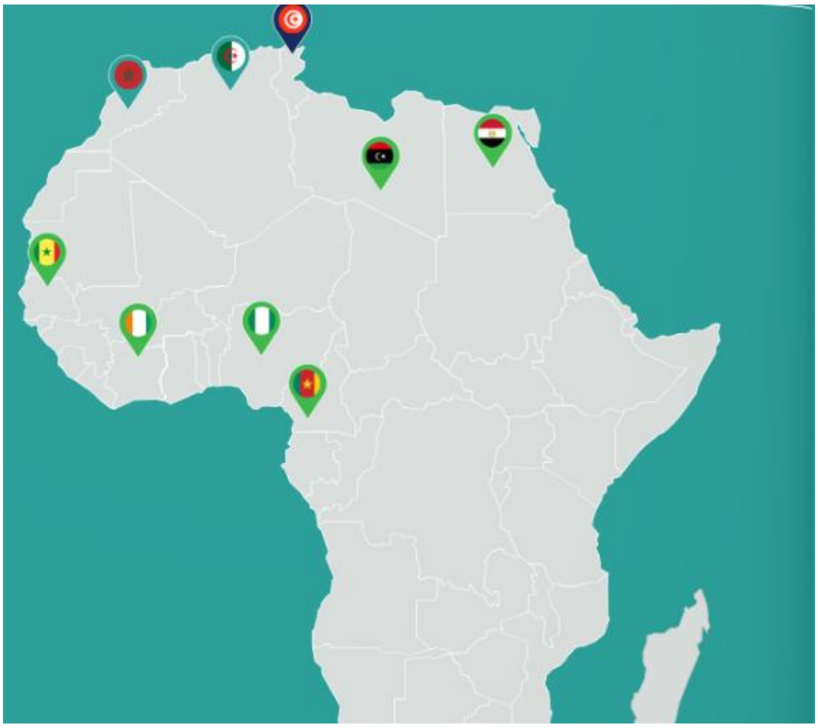


Figure 2: PGS Global Presence

1.1.3 Business Domain:

PGS is committed to serving a diverse array of markets, demonstrating adaptability and versatility in its solutions. The company's primary sectors of expertise encompass the following industries:

- Oil and Gas Excellence:

Within the Oil and Gas industry, PGS stands as a beacon of excellence, providing fitted solutions that address the complex and dynamic needs of this critical sector. PGS leverages cutting-edge technologies and industry best practices to offer comprehensive services that enhance operational efficiency, safety, and overall performance.

- Chemicals and Petrochemicals:

PGS extends its expertise to the chemicals and petrochemicals sector, offering specialized solutions that enhance processes and maximize productivity.

- Power and Energy:

PGS plays a pivotal role in the Power and Energy industry, providing innovative solutions that contribute to sustainable and efficient energy production.

- Water Treatment:

Acknowledging the critical importance of water management, PGS offers solutions in water treatment to ensure responsible and sustainable practices.

- Green Energy:

In line with global efforts towards sustainability, PGS actively engages in the Green Energy sector, offering solutions that promote environmentally friendly practices and renewable energy sources.

1.1.4 Services:

PGS is on a mission to offer comprehensive and modular process automation solutions tailored for the Oil and gas, petrochemical, power, Green Energy, water treatment, and energy industries. Their commitment is to provide cutting-edge services that enhance efficiency, reliability, and sustainability across diverse sectors.

PGS offers a range of services designed to meet the unique needs of our clients:

- **Installation and Commissioning:** We specialize in the seamless installation and commissioning of systems, ensuring a smooth transition to enhanced automation solutions.
- **Maintenance:** PGS is dedicated to the ongoing maintenance of systems, keeping them in the best condition for sustained performance.
- **Troubleshooting and Repair:** the expert team is equipped to identify and address issues promptly, minimizing downtime and ensuring uninterrupted operations.
- **Meters and Metering Systems Calibration:** Precision is key in our industries. PGS provides calibration services for meters and metering systems, ensuring accurate measurements for critical processes.
- **Manufacturing and Assembling:** PGS engages in the manufacturing and assembling of specialized components, contributing to the development of efficient and reliable systems.

PGS expertise extends across various stages of the industrial processes, including:

- **Exploration and Production:** PGS excels in providing solutions that enhance the efficiency and safety of exploration and production activities in the Oil and gas industry.
- **Pipeline Transport:** PGS offers fitted solutions for transporting materials through pipelines, ensuring a secure and efficient flow.
- **Refining:** PGS contributes to refining processes by implementing advanced automation solutions that optimize production and quality control.
- **Storage and Distribution:** PGS services cover the entire spectrum of storage and distribution processes, facilitating the safe and reliable movement of materials.

In summary, PGS is dedicated to delivering a complete suite of services and process solutions, supporting its clients in achieving operational excellence and sustainability in their respective industries.

1.1.5 Key Achievements:

- **Precision Calibration System Certification:** PGS achieved a significant milestone by developing a high-precision calibration system that adheres to stringent quality standards. This accomplishment was highlighted by the successful certification at the inauguration of the European Center for Flow Measurement VSL in Rotterdam. PGS's dedication to precision and quality has been recognized on an international level.
- **Turnkey Project for the World's Largest Oil Terminal:** Acknowledging PGS's expertise, the company was awarded a turnkey project for one of the world's largest oil terminals located in Morocco. This impressive project encompasses a substantial storage capacity of 600,000 cubic meters, showcasing PGS's capability to handle and deliver on a grand scale in the oil and gas sector.
- **Automated Truck Loading Skids for Storage Tank Farm:** Demonstrating PGS's commitment to automation and efficiency, the company has been responsible for supplying automated truck loading skids for a storage tank farm with a capacity of 110,000 cubic meters. This project underscores PGS's role in providing innovative solutions for the storage and distribution processes within the industry.

These notable achievements highlight PGS's ability to innovate, execute large-scale projects, and contribute to advancements in precision calibration and automation within the oil and gas sector. They stand as a testament to PGS's commitment to excellence and its significant impact on critical industrial processes.

1.1.6 Partners

PGS values strong collaborations, and key partnerships with industry leaders such as ABB, Krohne, Schneider Electric, Milton Roy, and Emerson have been integral to its success. These partnerships signify a commitment to leveraging the expertise and cutting-edge technologies offered by these renowned companies. Through long-term collaborations with world-class Original Equipment Manufacturers (OEMs), PGS has not only established solid relationships but has also significantly enhanced its knowledge and expertise in the market. The exchange of ideas, access to state-of-the-art technologies, and shared insights from these partnerships have played a pivotal role in PGS's continuous improvement and growth. These collaborations underscore PGS's dedication to staying at the forefront of industry advancements, ensuring the delivery of top-notch solutions to its clients.

1.2 About Project

1.2.1 Challenges:

Inefficient inventory management in oil depots poses significant challenges to operational efficiency, cost-effectiveness, and risk mitigation. The lack of forecasting methods often leads to failure to account for dynamic market conditions, supply chain disruptions, and fluctuating demand patterns, leading to suboptimal inventory levels, stockouts, excess inventory, and increased operational costs. Therefore, there is a pressing need to develop an integrated forecasting system tailored specifically for oil depots that can accurately predict demand, optimize inventory levels, and enhance decision-making processes to ensure efficient operations, minimize costs, and mitigate risks.

Additionally, there is a lack of an effective logistics strategy for transporting petroleum products from production to distribution.

1.2.2 Solution:

To address the outlined challenges, multiple solutions have been proposed:

- Develop and implement a comprehensive forecasting system tailored specifically for oil depots. This system aims to eliminate stockouts, streamline ordering processes, and analyze tank performance across multiple parameters, ensuring optimal inventory management and enhancing customer satisfaction.
- Conduct a thorough analysis of critical operational and logistical factors within oil depots to identify key criteria influencing inventory management and distribution efficiency.
- Generate and deploy a user-friendly reporting dashboard to visually present forecasting results, providing stakeholders with actionable insights.
- Utilize analytical techniques to understand tank behavior based on various parameters such as volume, meter readings, and injector performance. This understanding will optimize inventory levels and replenishment processes accordingly.
- Enhance order processes by determining optimal quantities for replenishment, considering factors like density, pressure, and temperature. This optimization will improve operational efficiency and cost-effectiveness.
- Optimize logistics chains to facilitate efficient transportation of petroleum products from production sites to depots and distribution centers. This optimization will minimize transit times and reduce costs.

1.2.3 Methodology used (TDSP):

The Team Data Science Process (TDSP) comprises five main stages: Start, Business Understanding, Data Acquisition and Understanding, Modeling, and Deployment:

- **Start**: The Start stage involves initiating the project, assembling the project team, defining the project objectives, and establishing the project plan. This stage sets the foundation for the entire data science project.
- **Business Understanding**: In the Business Understanding stage, the focus is on gaining a comprehensive understanding of the business problem or opportunity. This involves identifying stakeholders, defining success criteria, and formulating hypotheses to guide the analysis.
- **Data Acquisition and Understanding**: During the Data Acquisition and Understanding stage, data sources are identified, data is collected, cleaned, and prepared for analysis. Exploratory data analysis techniques are applied to understand the structure, quality, and relationships within the data.
- **Modeling**: The Modeling stage involves selecting appropriate machine learning algorithms, building, and training models using the prepared data, and evaluating their performance against the project objectives. Iterative experimentation may be conducted to fine-tune models and improve their accuracy.
- **Deployment**: Once a satisfactory model is developed, it needs to be deployed into production systems to generate actionable insights or support decision-making processes. Deployment considerations include scalability, reliability, and integration with existing systems.

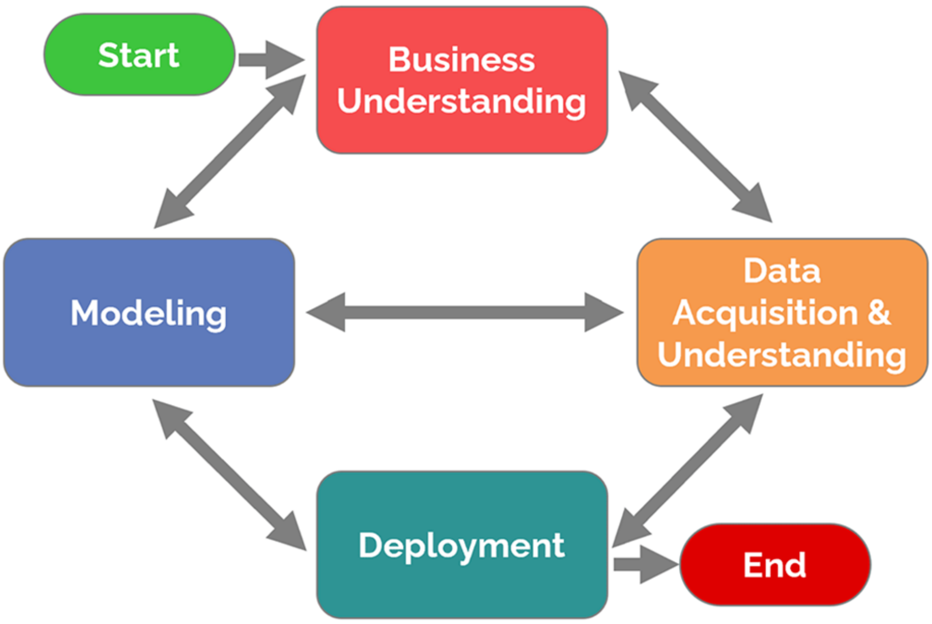


Figure 3:TDSP

The Team Data Science Process (TDSP) operates through a seamless transition from one stage to the next, ensuring a cohesive flow of activities. Beginning with the "**Start**" stage, where the project is initiated and objectives are set, the process then moves into "**Business Understanding**." Here, stakeholders collaborate to define the problem and hypotheses. This understanding informs the subsequent "**Data Acquisition and Understanding**", where data is gathered and prepared. With prepared data, the team enters the "Modeling" phase, developing and refining models to address the defined problem. Finally, in the "**Deployment**" phase, models are implemented into production systems.

1.2.4 Study of the existing:

In the analysis of current offerings, two platforms stand out:

Novi Labs Platform:

Novi Labs offers cutting-edge development planning software that provides forecasting, analytical, and predictive analysis tools in the oil and gas industry developed by the Novi company located in the USA. Novi Labs provides accurate forecasts to help oil and gas companies optimize their operations. The strength of the Novi Labs Platform lies in its ability to deliver precise predictions, enabling companies to make informed decisions for efficient resource allocation and overall operational improvement.

Figure 4: Novi Labs Logo

Speedy Fuels:

Speedy Fuels is a company located in the UK that offers a comprehensive range of oil delivery services tailored to unique requirements, with a nationwide 24/7 supply. They offer a variety of products, including red diesel (gas oil), standard white diesel, industrial heating oil, kerosene, and lubricants. The company ensures timely deliveries, with 92% of orders fulfilled within 24 hours, and they also provide emergency oil delivery services for those who have unexpectedly run out of fuel.

Figure 5: Speedy Fuels Logo

1.2.5 Sustainable Development Goals:

Figure 6: Project Sustainable Development Goals

SDG 3 - Good Health and Well-being

Implementing an Integrated Forecasting System for Inventory Management in Oil Depots can indirectly contribute to SDG 3 by ensuring the efficient management of resources, including oil and gas products, which are crucial for various health-related activities such as transportation of medical supplies and providing energy for healthcare facilities.

SDG 6 - Clean Water and Sanitation

Efficient inventory management in oil depots can prevent spills and leaks, minimizing the risk of water contamination. Additionally, by optimizing delivery routes and reducing fuel consumption, the project can indirectly contribute to reducing water pollution and conserving water resources.

SDG 7 – Affordable and Clean Energy

By optimizing inventory management and delivery processes, the project can contribute to ensuring the availability of energy resources, such as oil and gas products, sustainably and efficiently, thereby supporting SDG 7.

SDG 8 - Decent Work and Economic Growth

Implementing an integrated forecasting system for inventory management can lead to increased efficiency in the oil depot operations, potentially creating employment opportunities and contributing to economic growth in the oil and gas sector.

SDG 9 - Industry, Innovation, and Infrastructure

The project involves the development and implementation of innovative solutions for inventory management and route optimization, contributing to advancements in industry practices and infrastructure within the oil and gas sector.

SDG 12 - Responsible Consumption and Production

The project contributes to SDG 12 by promoting responsible consumption and production in the oil and gas industry. Through optimized inventory management and efficient delivery routes, it minimizes waste and resource usage, aligning with goals for sustainable practices.

SDG 13 - Climate Action

By optimizing delivery routes and reducing fuel consumption, the project can contribute to mitigating greenhouse gas emissions and promoting sustainable transportation practices, aligning with the objectives of SDG 13.

SDG 17 - Partnerships for the Goals

Collaboration between stakeholders, including government agencies, private companies, and technology providers, is essential for the successful implementation of the project, highlighting the importance of partnerships to achieve sustainable development goals.

1.3 Business Objectives

In the pursuit of operational excellence within the oil and gas industry, the establishment of clear strategic business objectives is crucial to an organization's success. Business objectives serve as the guiding principles that shape the direction and actions of a company, providing a roadmap for growth, sustainability, and achievement.

These objectives encapsulate the overarching goals and aspirations of an organization, reflecting its vision, mission, and core values.

1.3.1 Inventory Management:

Inventory management involves more than just keeping track of the oil stock, it's about maintaining a dynamic system that guarantees control and precision at every stage of the oil supply chain.

This involves overseeing the bulk movements of oil, from its initial receipt through import via barge and pipelines to its subsequent transfer between storage tanks and eventually to trucks for distribution.

Continuously tracking and managing stock levels allows identification of potential shortages before occurrence. This strategic approach optimizes the distribution of oil, preventing supply disruptions.

1.3.2 Orders Management:

Ensure Oil Product Availability: Showing commitment to a consistent and smooth supply of oil products, the utilization of advanced techniques plays a crucial role in accurately predicting demand, mitigating the risk of shortages. This strategic approach ensures inventory alignment with market needs, enhancing customer satisfaction and overall business performance.

Optimize Dispatching: The goal is to enhance the efficiency of dispatching operations by systematically checking and managing oil orders. This involves meticulously analyzing demand, strategically allocating orders to trailers, and ensuring a well-organized and optimized dispatching system.

1.3.3 Delivery Management:

Determine Optimal Delivery Routes: Efficiency in oil deliveries remains an objective. The focus is on factors such as road conditions, time constraints, and weather updates. This strategic initiative not only optimizes fuel consumption but also actively contributes to environmental sustainability.

Optimize Delivery Schedules: Recognizing the importance of timely and efficient deliveries, the target is to ensure a smooth and timely flow of oil products.

Select Suitable Driver-Trailer Pair: In the pursuit of enhancing the efficiency and reliability of deliveries, to match drivers with trailers. Considerations will include historical performance, skills, and specific delivery requirements, ensuring a harmonious pairing that contributes to safe and efficient operations.

1.3.4 Equipment Monitoring:

Anomaly Detection: High priority is placed on the security and integrity of equipment. The goal is to detect anomalies corresponding to irregularities. This includes surveillance and monitoring to identify and address issues such as thefts, leaks, evaporation, and malfunctions in meters and injectors.

1.4 Data Science Objectives

1.4.1 Data Science Objective 1: Predictive Model for Oil Stock Levels

To address inventory management challenges related to bulk movements, the goal is to develop a predictive model ensuring accurate estimation of oil stock levels in individual tanks. This model aims to predict tank stock levels over time, facilitating the prevention of stockouts. By using bulk movement data, historical information, and external factors, the model will enable employees to find the optimal time for ordering oil shipments, thereby enhancing operational efficiency and preventing disruptions in stock availability.

Metrics

Percentage of Tank Capacity Utilized:

- **Definition**: Measures the percentage of each tank's capacity that is filled at a given time.
- **Purpose**: Provides insights into how efficiently each tank is utilized, guiding decisions on optimal ordering times to prevent stockouts.

Stockout Rate:

- **Definition**: Calculates the rate at which stockouts occur based on model predictions.
- **Purpose**: Quantifies the effectiveness of the predictive model in preventing stockouts and maintaining continuous stock availability.

1.4.2 Data Science Objective 2: Demand Prediction Model

The goal is to create a demand prediction model that estimates the demanded orders, addressing challenges in orders management. This model will consider factors such as availability, borrowing, and skid-based distribution, contributing to improved order processes and preventing stockouts. In addition, the model will help predict future customer demands, allowing us to take proactive steps to maintain the right stock levels. This ensures timely and efficient order fulfillment for better customer satisfaction.

Metrics

Order Client Quantity

- **Definition**: Measures the accuracy of predicted order quantities compared to the actual quantities demanded by clients.
- **Purpose**: Assesses the reliability of the demand prediction model in accurately estimating the quantity of orders.

1.4.3 Data Science Objective 3: Route Optimization Model

To address challenges in delivery management, our objective is to develop a Route Optimization Model that predicts the best delivery routes based on road conditions, time constraints, and real-time weather updates. By optimizing travel paths, reducing delivery times, and minimizing fuel consumption, this model enhances overall delivery efficiency and significantly contributes to long-term customer loyalty.

Metrics

Travel Time Reduction

- **Definition**: Measures the percentage reduction in travel time achieved by optimized routes compared to baseline routes.
- **Purpose**: Quantifies the improvement in delivery efficiency and reduced travel times achieved through optimized routes.

Optimal Oil Transport Condition

- **Definition**: Evaluate the suitability of transportation conditions for oil products, considering factors such as temperature, pressure, and safety regulations.
- **Purpose**: Ensures that oil products are transported under optimal conditions, minimizing risks of spillage, contamination, and product degradation during transit.

1.4.4 Data Science Objective 4: Anomaly Detection System

To enhance equipment monitoring, an anomaly detection system will be implemented to identify and respond to irregularities such as thefts, leaks, evaporation issues, meter malfunctions, and injector dysfunction.

Metrics:

Sensitivity

- **Definition**: Measures the proportion of true anomalies correctly identified by the system.
- **Purpose**: Evaluate the effectiveness of the system in detecting genuine anomalies, ensuring comprehensive monitoring and response capabilities.

Injector Efficiency

- **Definition**: Assesses the performance of injectors in accurately adding additives or other substances into the system.
- **Purpose**: Ensures injectors are functioning optimally, contributing to the accuracy and integrity of product measurements and transfers.

Volume Discrepancy

- **Definition**: Quantifies the difference between the expected and measured volumes of products in the system.
- **Purpose**: Identifies discrepancies in product volumes, indicating potential issues such as leaks, thefts, or inaccuracies in measurement, facilitating proactive maintenance and intervention measures.

2 Data Understanding

2.1 Introduction

In this section, we'll explore the data provided for our project on forecasting inventory management in oil depots. Understanding our data is essential for building accurate predictions and making informed decisions. We'll take a closer look at the data's characteristics, quality, and how it relates to our project goals. Let's dive in and uncover insights that will guide our next steps.

2.2 -Data Description

In this subsection, we'll explore the different aspects of our data, explaining the various key terms used and detailing the steps of how the system works. We'll uncover the inner workings of our data, shedding light on its structure and functionality. Through this exploration, we'll gain a deeper understanding of the data provided by PGS and its relevance to our project on forecasting inventory management in oil depots.

- Terminologies

Carriers: Carriers are companies responsible for organizing the transportation of products at a terminal.

Containers: Containers are used to identify the physical locations where products are held or moved.

They utilize various container types to facilitate transportation such as:

1. tanks
2. trailers
3. vapor recovery units
4. Ships
5. Barges
6. railcars

CONTAINER_CODE	CONTAINER_TYPE
TK-307	1
TK-400	1
TK-500	1
TK-600	1
10113-B-7	2
1025-03	2
1029-010	2
1037-02	2

Terminal Products: Terminal products are the base products stored at the terminal and constitute the components of supplier, customer, and destination products.

Suppliers and Customers: Suppliers are companies that own or sell products at a terminal, while customers purchase products from suppliers.

Drivers: Drivers are personnel employed by carriers who are authorized to drive transport vehicles and haul products into and out of terminals.

Product Groups: Product groups define sets of related products such as additives, diesel, gasoline, and heavy fuel.

Destinations: Destinations refer to specific locations where products are delivered, which can be either specific addresses or general areas.

Folios: Folios represent one business day and are used to group orders and bulk movements for that day.

End of Day (EOD) and End of Month (EOM):

EOD signifies the closure of one business day's folio and the opening of a new one for the next business day. EOM refers to the closure of the carryover folio and the distribution of inventory gains and losses if configured.

Bulk Movements:

Bulk movements record product movements in bulk quantities and include types such as bulk **receipt**, **disposal**, **transfer**, and **external transfer.**

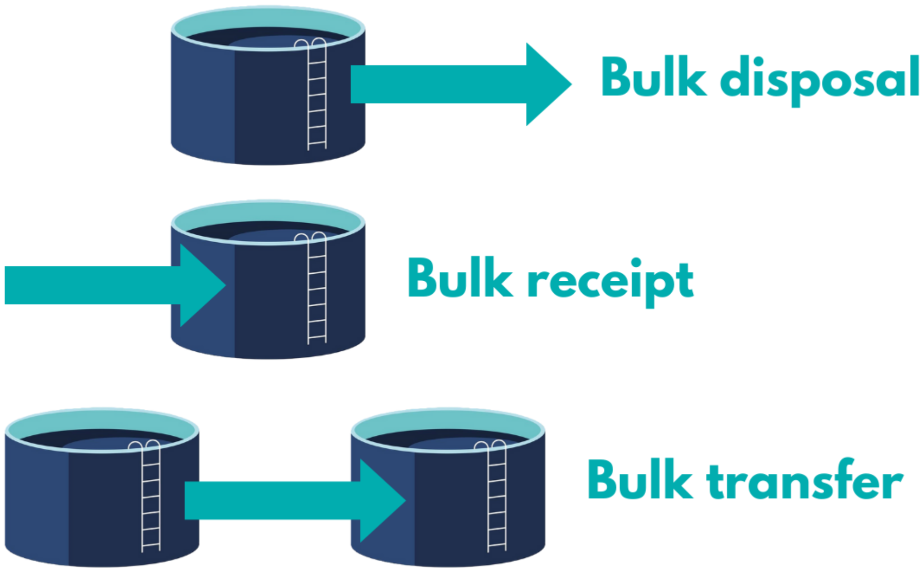


Figure 7: Bulk Movements

- **Bulk receipt:** A significant movement of product from an external container into an internal container (for example, from a ship to a tank).

- **Bulk disposal:** A substantial movement of product from an internal container to an external mobile container (for example, from a tank to a railcar).

- **Internal transfer:** A significant movement of product between two internal containers (such as between two tanks). This transfer can involve the physical movement of product from one tank to another, or it may involve book-entry transfers of terminal product ownership between suppliers. It can also occur due to the regrading of a terminal product.

- **External transfer:** A significant movement of product between two external mobile containers (for example, from a barge to a ship).

Order Statuses:

Orders undergo various statuses such as:

1: **ENTERING:** The order or trip has been created, but not yet dispatched, or is being corrected. Only orders and trips with an Entering status can be modified or deleted.

2: **DISPATCHED:** Order shipped and ready for loading

3: **LOADING:** When the order is validated and FAN LOADING

4: **LOADING AT BAY:** Compartments are loading

5: **VALIDATING:** The system is performing AIC checks on the order

6: **LOADED:** The quantity shipped is fully loaded, the BOL is ready for printing

7: **CANCELED:** Only loaded orders can be canceled

Presets/Meters and Injectors:

Presets, meters, and injectors play crucial roles in tracking product movements and quantities at different stages.

Product Inventory:

Inventory is managed with **fractional quantities**, **weights**, and other measures to ensure accurate tracking.

Product Group Codes:

Product group codes categorize products into groups such as :

ADD	ADDITIVE
DSL	DIESEL
GAS	GASOLINE
HFO	HEAVY FUEL

Host Types:

Host types include **terminal owners**, **suppliers**, **customers**, and **product groups**, representing different entities in the system.

2.3 -Data Classification

Inventory management:

BULK_PRODUCT.xls: This file contains detailed information on bulk products, including movements and transactions, essential for tracking inventory flow and understanding product movements within the oil management system.

Bulk_movement.xls: This dataset records bulk movements of products in the oil management system, categorizing various types of movements such as bulk receipts, disposals, transfers, and external transfers. It tracks the movement of products between different containers and locations.

CONTAINER.xls: This dataset offers detailed information about containers used in the oil management system, including their types, capacities, and specifications. It helps in understanding the physical locations where products are held or moved within the system.

ORDERS_TANK_INV[1-2].xls: These files capture order details related to tank inventory, providing essential information for managing current inventory levels and planning for future needs within the oil management system.

ORDER_RECIPE [1-8].xls:

Orders[1-2].xls

Orders management:

ORDER_RECIPE [1-8].xls: These files contain detailed recipes for orders, specifying the requirements, formulations, or blends needed for each order. They provide essential information for precise order fulfillment and inventory planning within the oil management system. The data includes details such as tank codes, preset codes, meter codes, injector codes, terminal product numbers, ordered quantities, gross quantities, blend percentages, rates, weights, densities, totalizer readings, Reid vapor pressure, and weights in vacuum.

Orders[1-2].xls: These files contain comprehensive details on orders placed within the oil management system, including information such as order IDs, loaded driver numbers, carrier numbers, folio numbers, tractor numbers, delivered carrier and driver numbers, supplier and customer numbers, destination numbers, movement and revision numbers, order statuses, status dates, purchase order numbers, start and end load times, BOL printing status, order entry times, freight billing status, authorization request IDs, TABS order numbers, previous order IDs, and signature paths. They facilitate efficient order management and forecasting of demand within the system.

Delivery management:

ORDER_RECIPE [1-8].xls:

Orders[1-2].xls

TRACTOR.xls: This file contains data related to tractors used for transportation within the terminal automation system. It includes information such as carrier numbers, tractor numbers, fueling supplier and product numbers, license plate details, state information, loading lockout status and reasons, card numbers, last modification details, current visit IDs, vehicle numbers, and unladen weights. Tractors are essential components in the transportation process, facilitating the movement of products within the terminal.

TRAILER.xls: This file contains details about trailers utilized for transportation purposes within the terminal automation system. It includes data such as the last modification timestamp, loading type, host trailer codes, alternate trailer codes, truck numbers, usable capacities, vehicle types, loading limits, card numbers, maximum tare weights, requirements for restricted flow, repeat order IDs, global lockout status and reasons, global lockout user and terminal IDs, current visit IDs, unladen weights, maximum weights, and trailer loading time IDs. Trailers play a crucial role in transporting bulk quantities of products within the terminal, ensuring efficient logistics operations.

COMPARTMENT.xls: This file contains information about compartments within the terminal automation system. It includes data such as the owner ID, container code, compartment number, container type, capacity, product group code, supplier number, product number, octane level, last modification timestamp, order ID, valid product group code, tag ID, inventory tracking status, trans-loading tank code, import/export quantity, terminal product number, import/export quantity type, manual temperature, manual density, supplier number for import/export, manual coefficient of expansion, manual pressure, visit number, and ethanol expansion. Compartments are crucial for storing and transporting terminal products within the terminal, and this dataset provides details about their characteristics and usage.

TRAILER.xls:

Tank.xls

Equipment monitoring:

INJECTOR_HISTORY.xls: This file contains historical records related to injectors within the terminal automation system. It includes information such as the meter preset code, preset code, injector code, folio number, totalizer quantity, terminal product number, meter code, throughput, unaccounted quantity, product throughput, certification rate, fractional totalizer, fractional throughput, fractional unaccounted quantity, totalizer for beginning-of-day (BOD), fractional totalizer for BOD, actual throughput, fractional actual throughput, and loading quantity type. Injectors play a crucial role in injecting additives into terminal products, and this dataset provides insights into their historical performance and usage.

METER_HISTORY.xls: This file contains historical records related to meters within the terminal automation system. It includes details such as the preset code, folio number, meter code, gross totalizer quantity, terminal product number, gross throughput, net totalizer quantity, gross unaccounted quantity, net throughput, net unaccounted quantity, gross totalizer for beginning-of-day (BOD), net totalizer for BOD, actual gross throughput, actual net throughput, and loading quantity type. Meters are essential for measuring the flow of terminal products, and this dataset offers insights into their historical data and performance.

ORDER_RECIPE_[1-8].xls: In these datasets, we will focus solely on the order_id, injector_code, meter_code, committed_net_quantity, and ordered_quantity columns for this business objective. We will utilize these columns to know which injector or meter was used in each order and to investigate any disparities between the ordered quantity and the quantity calculated by the equipment.

Orders[1-2].xls: In this dataset, we only utilize the order_id and folio_number columns to perform a join with the meter_history and injector_history datasets, using the folio_number as the linking factor. This process allows us to determine the date the order was placed, and which injector or meter was utilized. Subsequently, we can merge this information into two distinct datasets: one for meters and another for injectors.

2.4 Mockups:

2.4.1 Dashboard Interface:

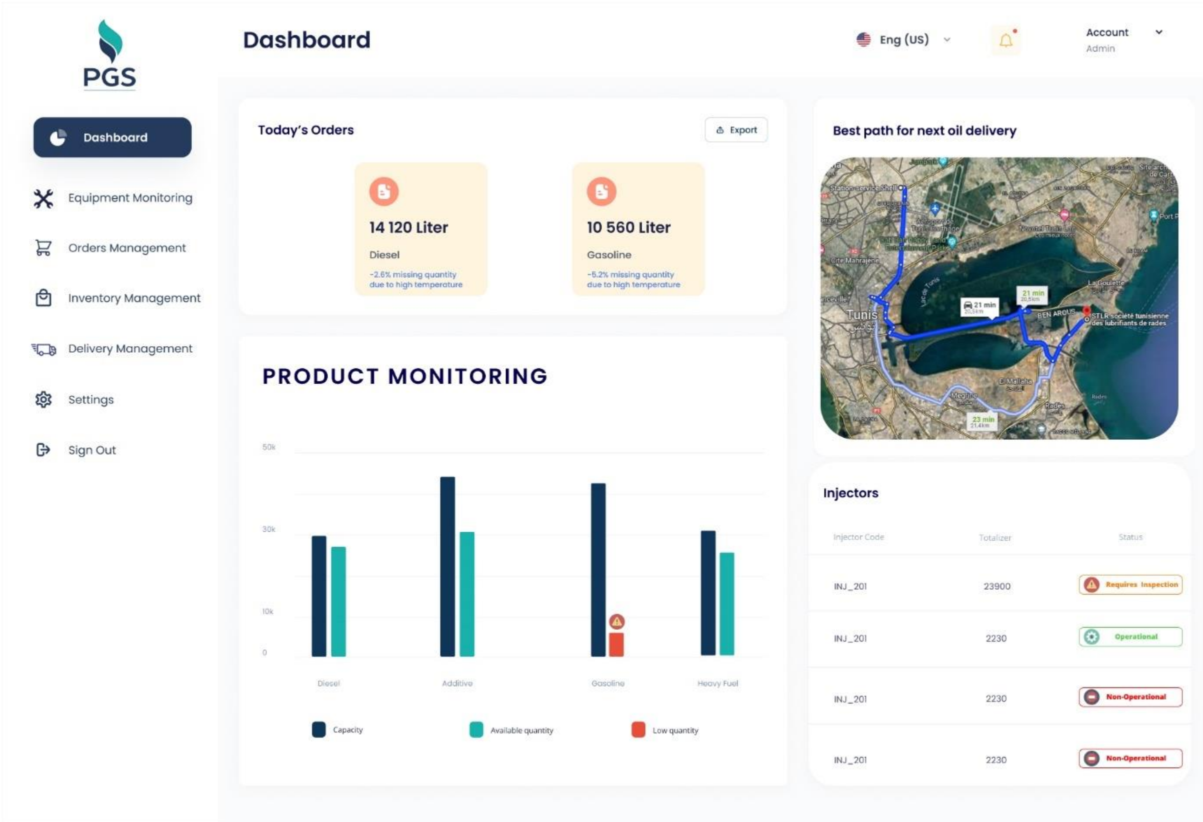


Figure 8: Dashboard Interface

The application dashboard features a clean and organized layout designed to provide quick access to important information and functionalities. At the center of the dashboard, there are four distinct sections, each serving a specific purpose:

- **Today's Orders:** Positioned prominently at the top-left corner, the "Today's Orders" section provides a snapshot of orders placed on the current day.
- **Product Monitoring:** Directly below the "Today's Orders" section is the "Product Monitoring" panel. This section offers real-time insights into the inventory levels of products such as gasoline, diesel, heavy fuel, and additives.
- **Best Path:** Located at the top-right corner of the dashboard, the "Best Path" module offers intelligent routing recommendations for upcoming deliveries. By analyzing various factors such as distance, traffic conditions, and weather, this feature suggests the most optimal routes to ensure timely and efficient deliveries.
- **Injectors:** Positioned below the "Best Path" section, the "Injectors" panel provides insights into the status of fuel injectors. It categorizes injectors into three main statuses: Operational, Requires Inspection, and Non-Operational.

In addition to the central sections, the dashboard features a sidebar on the left-hand side, housing four management options corresponding to the different aspects of the application:

- Orders Management
- Inventory Management
- Routing & Delivery
- Injector Maintenance

2.4.2 Equipment Monitoring Interface:

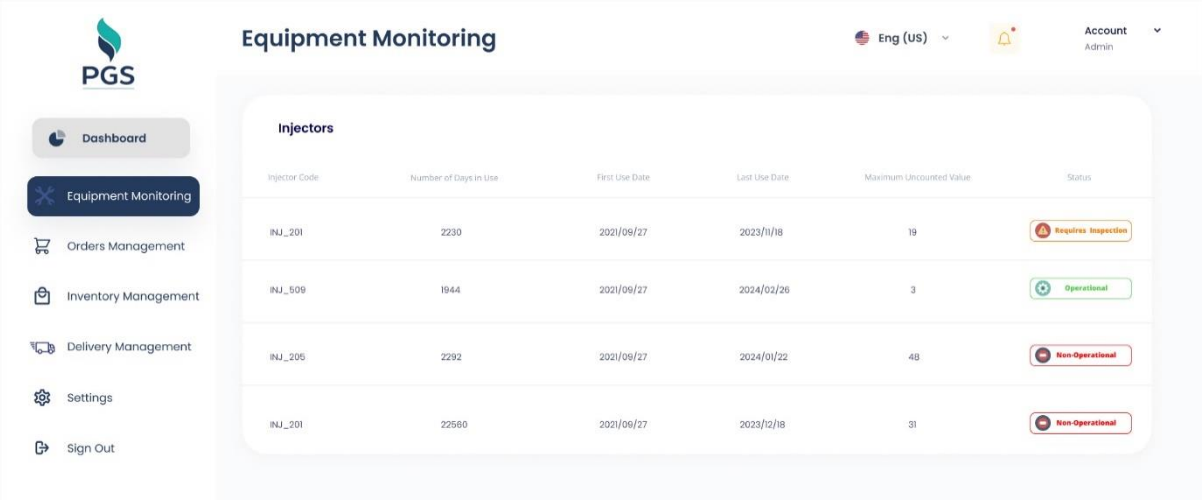


Figure 9: Equipment Monitoring

The Equipment Monitoring Interface provides a comprehensive overview of the performance and status of fuel injectors utilized in the operational processes. Designed with clarity and efficiency in mind, this interface offers detailed insights into each injector's specifications and operational histories.

- The Injector Code:

Each injector is uniquely identified by its code, allowing for easy reference, and tracking within the system.

- The Totalizer:

The Totalizer column displays the total amount of fuel dispensed by each injector since its installation or last reset

- The First Use Date:

Indicates the date when the injector was first put into service.

- Last Use Date:

Reflects the most recent date on which the injector was utilized.

- Maximum Uncounted Value:

This parameter represents the maximum quantity of fuel that the injector did not count in a single day

- The Status:

The Status column provides real-time updates on the current operational status of each injector. Status indicators may include:

  - Operational
  - Requires Inspection
  - Non-Operational

By presenting key metrics and status updates in a clear and organized manner, the Equipment Monitoring Interface empowers users to monitor and manage fuel injectors effectively.

2.4.3 Orders Management Interface:

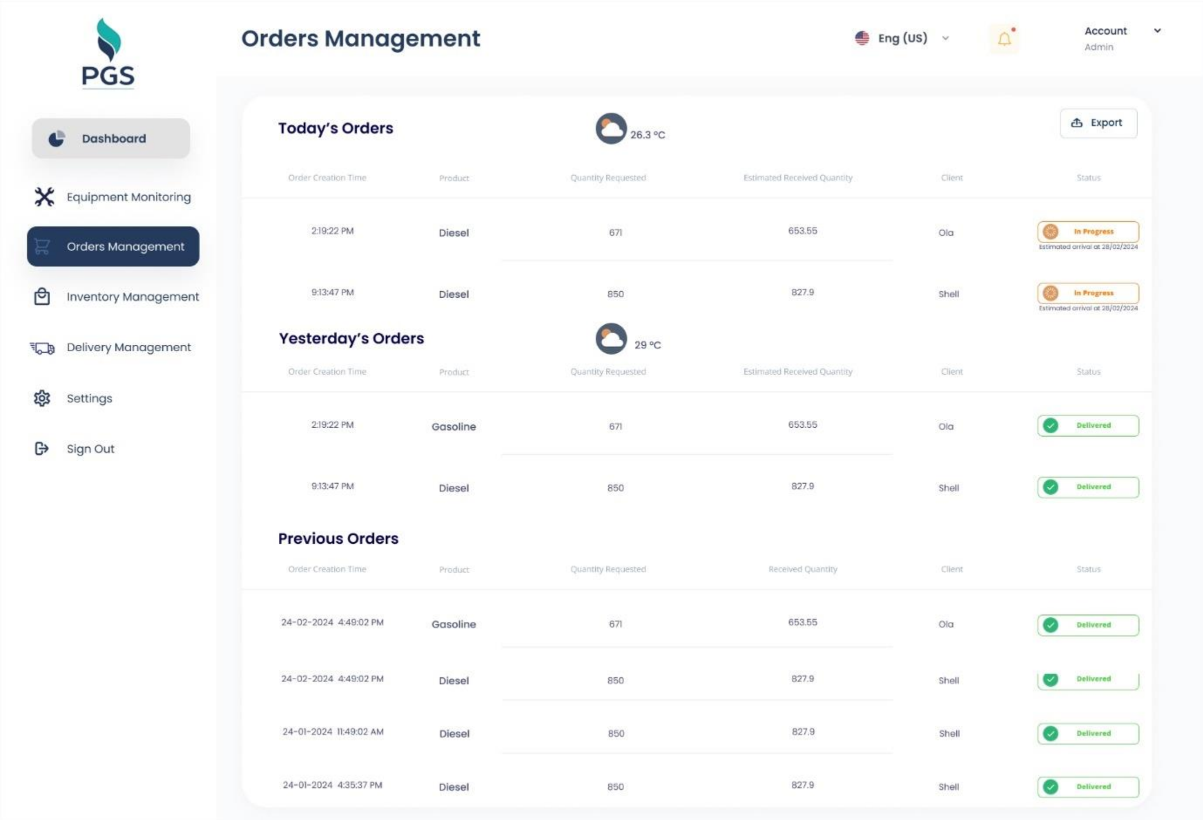


Figure 10: Orders Management

The Order Management Interface offers a comprehensive overview of today's orders, yesterday's orders, and previous orders, providing users with valuable insights to efficiently track and manage the orders. The interface is designed to display essential order details alongside weather information, facilitating informed decision-making and logistics planning.

For each order, the following details are displayed:

- Order Creation Time:

Indicates the date and time when the order was created.

- Product:

Specifies the type of product requested in the order (e.g., gasoline, diesel, heavy fuel, additives).

- Quantity Requested:

Displays the quantity of the product requested by the client.

- Estimated Received Quantity (In-Progress Orders):

For orders in progress, this field provides an estimated quantity of the product expected to be received based on current weather conditions.

- Received Quantity (Delivered Orders) :

For orders that have been delivered, this field indicates the actual quantity of the product received by the client.

- Client:

Identifies the gas station (client) placing the order.

- Status:

Indicates the status of the order, distinguishing between orders that have been delivered and those that are still in progress.

This interface provides a user-friendly platform for order management, facilitating informed decision-making and streamlined logistics operations.

2.4.4 Inventory Management Interface:

Figure 11: Inventory Management

The Product Monitoring Interface offers a comprehensive overview of the total available quantity of products stored across all tanks, providing users with real-time insights to efficiently manage inventory levels.

The Total Available Quantity section displays the aggregate quantity of all products currently available in the storage tanks. It serves as a key indicator of overall inventory status and availability.

Directly below the total available quantity, a detailed table presents information for each tank, including its unique code, the product it stores, its capacity, and the remaining quantity.

This interface provides a user-friendly platform for product monitoring, enabling users to access crucial information immediately.

2.4.5 Delivery Management Interface:

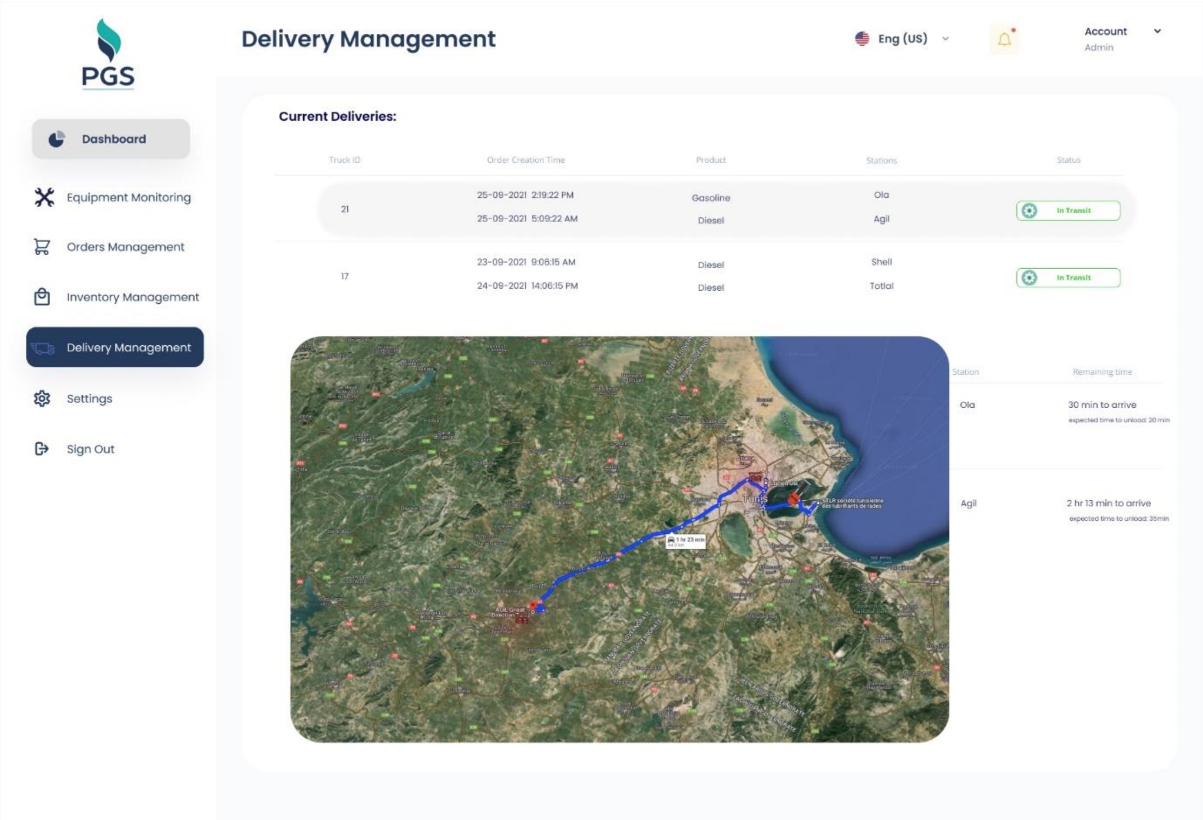


Figure 12: Delivery Management

The Delivery Management Interface provides an integrated platform for monitoring deliveries, offering valuable insights into the status and progress of each one.

The table provides detailed information for each ongoing delivery, including:

- Truck ID:

A unique identifier is assigned to each delivery truck.

- Order Creation Time:

A timestamp indicating when the delivery order was created.

- Products Being Delivered:

Specifies the products included in the delivery, as each truck can transport multiple orders simultaneously.

- Clients (Gas Stations):

Lists the gas stations or clients associated with the delivery.

- Status:

Indicates the current status of the delivery, distinguishing between:

  - In Transit
  - Delivered
  - Pending

By selecting a truck from the table, the map dynamically updates to display the delivery route, including the location of each station (client) along the way. Each station is marked on the map, providing clear visual cues for route navigation and progress tracking. Additionally, the map interface calculates and displays the estimated remaining time to arrive at each station.

This interface offers a comprehensive solution for delivery management, providing users with the tools and insights needed to streamline operations.

3 Data Acquisition

3.1 Inventory Management

3.1.1 Used Datasets

We used two data frames to achieve Data Science Objectives: **merged_bulk** and **merged_orders**.

merged_bulk is obtained by merging **Bulk_Movement.xls, Bulk_Product.xls and Tank1.xls**. It contains useful features such as FOLIO_NUMBER, GROSS_QUANTITY, and SHELL_CAPACITY.

- **merged_orders** are obtained by concatenating **ORDER_RECIPE [1-8].xls** and then merging it with **Orders.xls**. It contains features such as **GROSS_QUANTITY** and **FOLIO_NUMBER**.

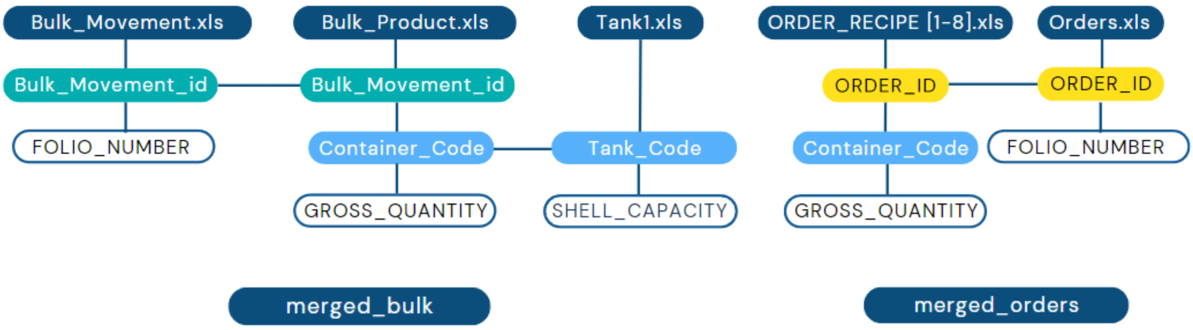


Figure 13:Inventory Management Dataset Schema

3.1.2 Data Cleaning Phase

Step 0:

We start by Loading the Datasets into the environment.

Step 1:

We select the required features from each dataset.

```
columns_to_keep_tank1 = ['TANK_CODE','NAME','SHELL_CAPACITY','TERMINAL_PRODUCT_NUMBER']
```

```
columns_to_keep_bulk_movement = ['BULK_MOVEMENT_ID','FOLIO_NUMBER','FROM_SUPPLIER_NUMBER',
                                 'TO_SUPPLIER_NUMBER','TYPE','MOVEMENT_NUMBER','MOVEMENT_DATE','AIC_STATUS']
```

```
columns_to_keep_bulk_product = ['BULK_MOVEMENT_ID','TERMINAL_PRODUCT_NUMBER','CONTAINER_CODE',
                                'FIGURES_TYPE','GROSS_QUANTITY','NET_QUANTITY','TEMPERATURE','END_DATE']
```

```
columns_to_keep_orders = ['ORDER_ID','MOVEMENT_NUMBER','SUPPLIER_NUMBER','FOLIO_NUMBER']
```

Step 2:

We concatenate order recipe from 1->8 then we merge it with orders.

```
order_recipe_merge = pd.concat([order_recipe1, order_recipe2, order_recipe3, order_recipe4,
                                order_recipe5, order_recipe6,order_recipe7,order_recipe8], ignore_index=True)
```

```
merged_orders = pd.merge(order_recipe_merge, orders, on='ORDER_ID', how='inner')
```

Step 3:

We merge Bulk movement, bulk product, and tank1.

```
merged_bulk = pd.merge(bulk_movement, bulk_product, on='BULK_MOVEMENT_ID', how='inner')
```

```
merged_bulk_tank = pd.merge(merged_bulk, tank1[['TANK_CODE', 'NAME']], left_on='CONTAINER_CODE',
                            right_on='TANK_CODE', how='inner')
```

Step 4:

We check for outliers using a boxplot:

gross quantity on merged_orders:

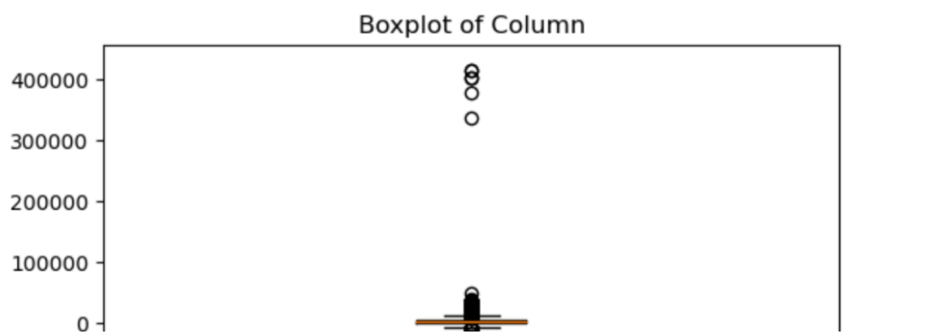


Figure 14: Boxplot for gross quantity on merged_orders in Inventory Management

gross quantity on merged_bulk:

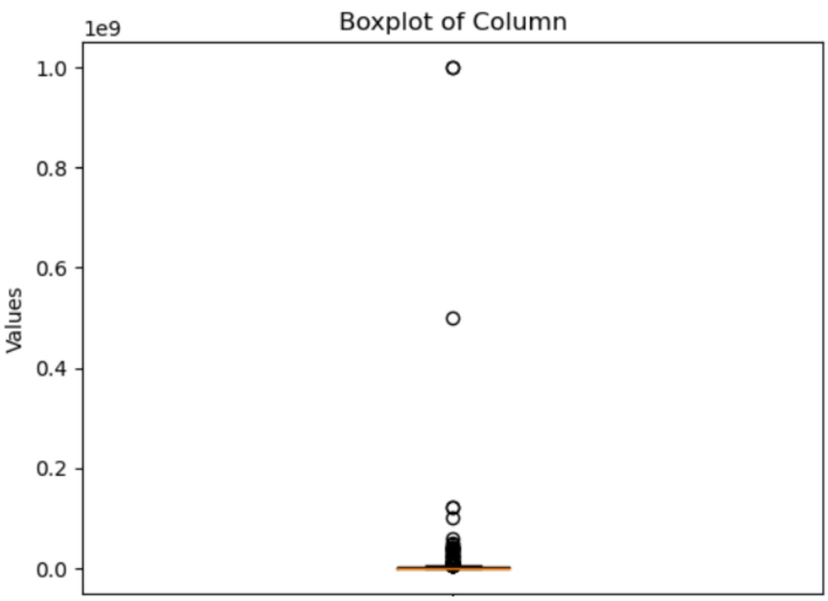


Figure 15: Boxplot for gross quantity on merged_bulk in Inventory Management

We discovered values beyond the Q3 level and subsequently eliminated them.

3.1.3 Challenges Encountered

Challenge 1:

We expected to obtain the currently available quantity after subtracting the outgoing quantities from the incoming quantities in each tank. However, we were taken aback to discover negative values and excessively large values instead.

```
Tank TK-102: -81798086
Tank TK-103: 86676401
Tank TK-104: 338414086
Tank TK-105: 462438
Tank TK-106: -130163244
Tank TK-201: -9901007
Tank TK-202: -12683941
Tank TK-305: -2115771
Tank TK-500: -27037307
Tank TK-600: -6218233
```

Figure 16: Challenge 1 in Inventory Management

3.2 Orders Management

3.2.1 Used Datasets

We used two data frames to achieve Data Science Objectives: **order_recipe[1-8]** and **orders**.

Quantity_Sum is obtained by merging **order_recipe[1-8].xls and orders.xls**. It contains useful features such as ORDERED_QUANTITY, TERMINAL_PRODUCT_NUMBER, and CUSTOMER_NUMBER.

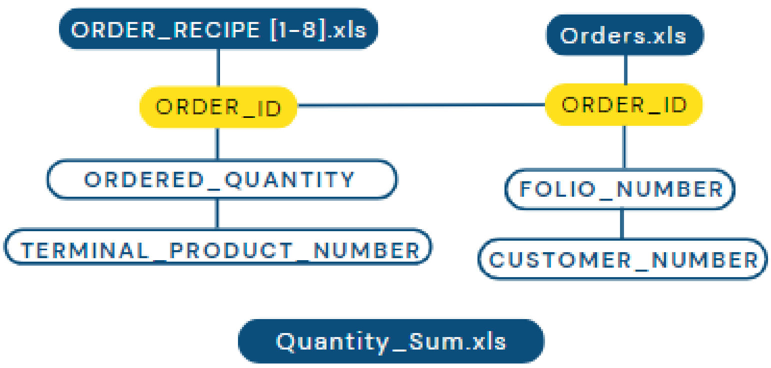


Figure 17: Orders management Dataset schema

3.2.2 Data Cleaning Phase

Step 0:

We start by Loading the Datasets into the environment.

Step 1:

```
order_recipe = ['ORDER_ID','ORDER_RECIPE_ID','PRODUCT_GROUP_CODE','ORDERED_QUANTITY','TERMINAL_PRODUCT_NUMBER']
columns_to_keep_orders = ['ORDER_ID','FOLIO_NUMBER','CUSTOMER_NUMBER']
```

We select the required features from each dataset.

Step 2:

```
order_recipe = pd.concat([order_recipe1, order_recipe2, order_recipe3, order_recipe4,
                         order_recipe5,order_recipe6,order_recipe7, order_recipe8], ignore_index=True)
orders = pd.concat([orders, orders2], ignore_index=True)

#merging orders
Orders_merged = pd.merge(order_recipe, orders, on="ORDER_ID", how="inner")
```

We concatenated and merged order_recipe with orders.

Step 3:

```
sum_ordered_quantity = merged_data.groupby(['TERMINAL_PRODUCT_NUMBER'])['ORDERED_QUANTITY'].sum()
```

We grouped orders by Folio Number, Customer Number, Terminal Number, and Ordered Quantity and calculated the ORDERED QUANTITY sum.

Step 4:

We checked for outliers

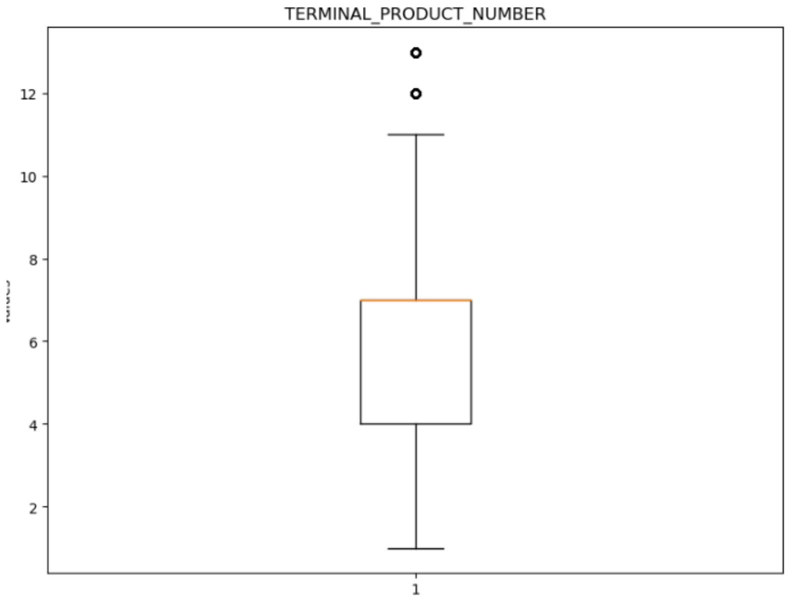


Figure 18: Boxplot for TERMINAL_PRODUCT_NUMBER in Quantity_Sum.xls

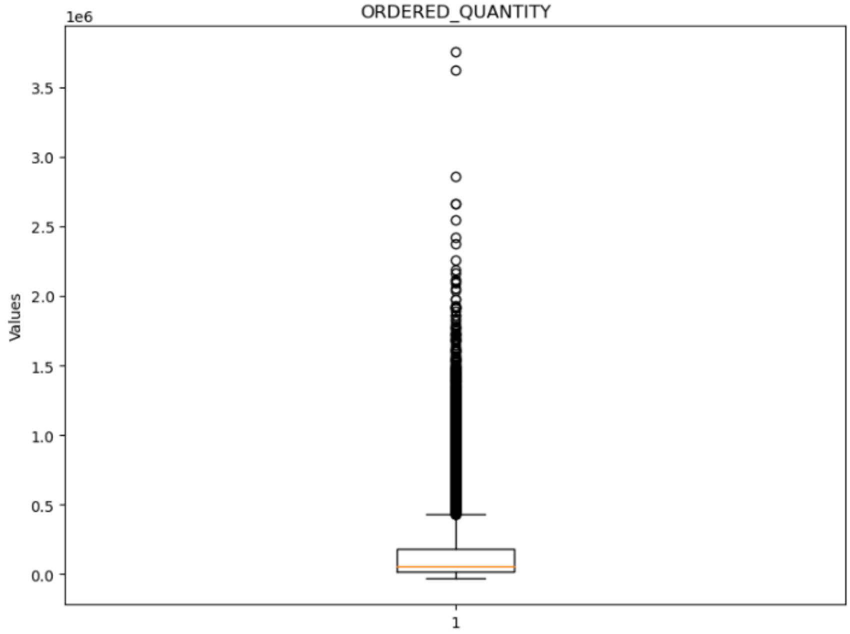


Figure 19: Boxplot for ORDERED_QUANTITY in Quantity_Sum.xls

We decided to keep the outliers.

Step 5:

We checked for negative values in ORDERD_QUANTITY.

	FOLIO_NUMBER	CUSTOMER_NUMBER	TERMINAL_PRODUCT_NUMBER	ORDERED_QUANTITY
3888	2022-03-27	15	4	-5000
3889	2022-03-27	15	7	-28000

We decided to remove them (canceled orders).

3.2.3 Challenges Encountered

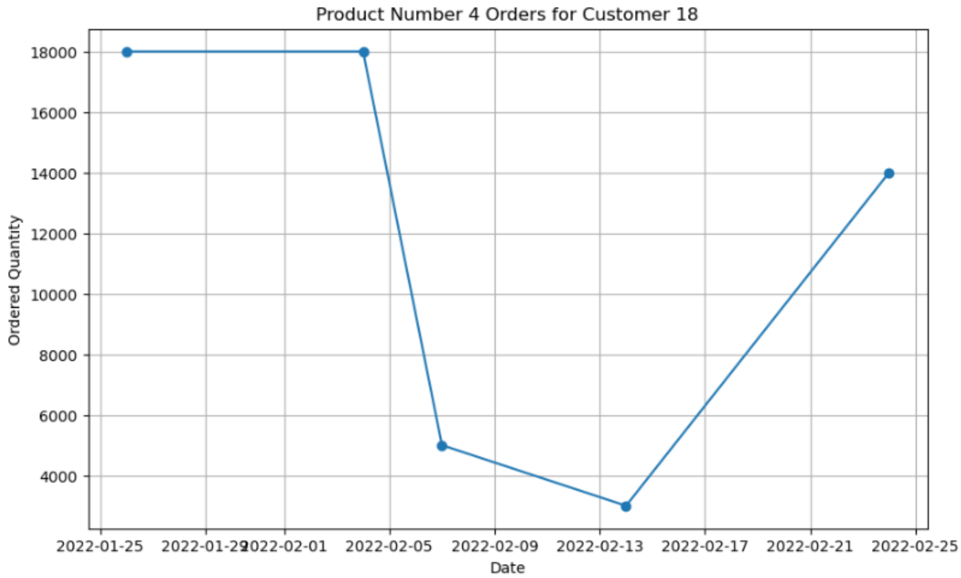


Figure 20: ORDERS for customer 18 product number 4

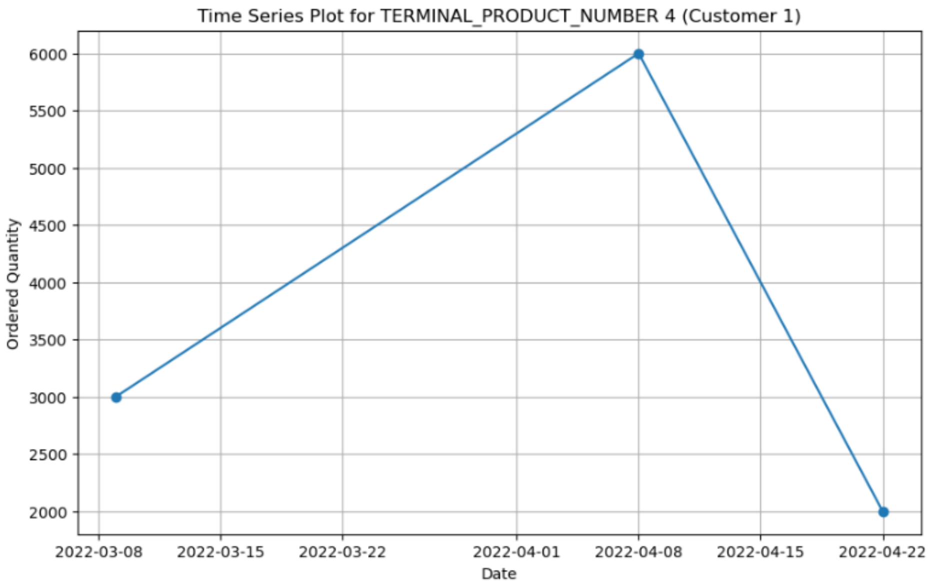


Figure 21: ORDERS for customer 1 product number 4

We encountered insufficient data available to construct a predictive model for some customers and product types.

3.3 Delivery Management:

3.3.1 Used Datasets

To achieve Data Science Objectives, we used four data frames: **order_recipe[1-8]** and **orders**, **trailer,** and **compartment.**

Order_delivery is obtained by merging order_recipe[1-8].xls and orders.xls.

TrailerCompartementMerge is obtained by merging **Trailer.xls** and **Compartment.xls**. It contains useful features such as USABLE_CAPACITY, COMPARTMENT_NUMBER, and CAPACITY.

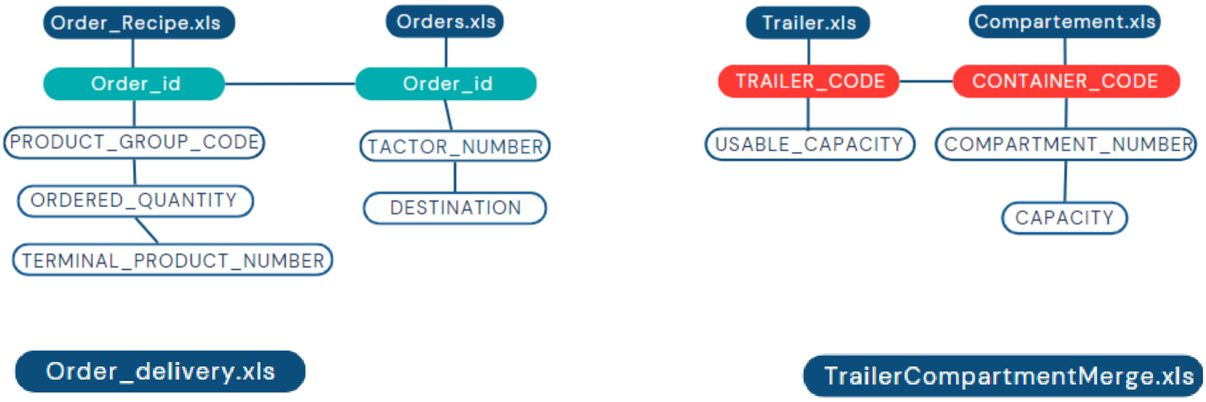


Figure 22: Delivery management Dataset schema

3.3.2 Data Cleaning Phase

Step 0:

We started by loading the datasets.

Step 1:

We select the required features from each dataset.

```
columns_to_keep_order_recipe1 = ['ORDER_ID','PRODUCT_GROUP_CODE','TERMINAL_PRODUCT_NUMBER','ORDERED_QUANTITY']
columns_to_keep_compartment = ['CONTAINER_CODE','COMPARTMENT_NUMBER','CAPACITY']

columns_to_keep_tractor = ['TRACTOR_NUMBER']

columns_to_keep_trailer = ['TRAILER_CODE','USABLE_CAPACITY']
```

Step 2:

```
order_recipe = pd.concat([order_recipe1, order_recipe2, order_recipe3, order_recipe4,
                          order_recipe5,order_recipe6,order_recipe7, order_recipe8], ignore_index=True)
orders = pd.concat([orders, orders2], ignore_index=True)

#merging orders
Orders_merged = pd.merge(order_recipe, orders, on="ORDER_ID", how="inner")
```

We concatenated orders with order_recipe.

Step 3:

We checked for Nan values, and we didn't find any.

Step 4:

We checked for outliers, and we decided to remove them.

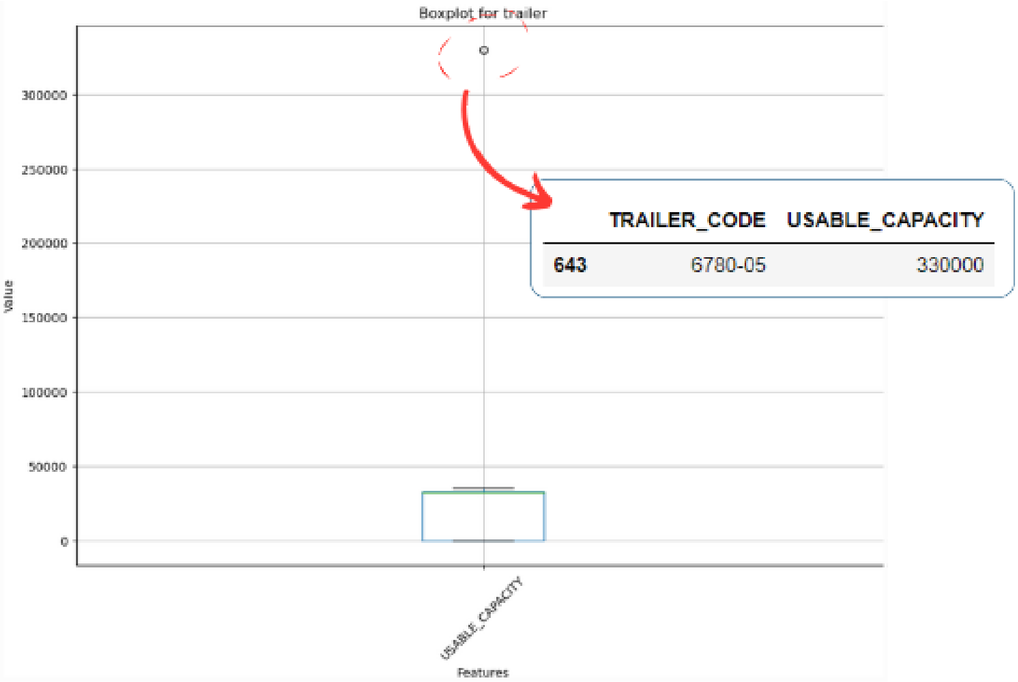


Figure 24: Boxplot for USABLE_CAPACITY in Trailer.xls before removing outliers

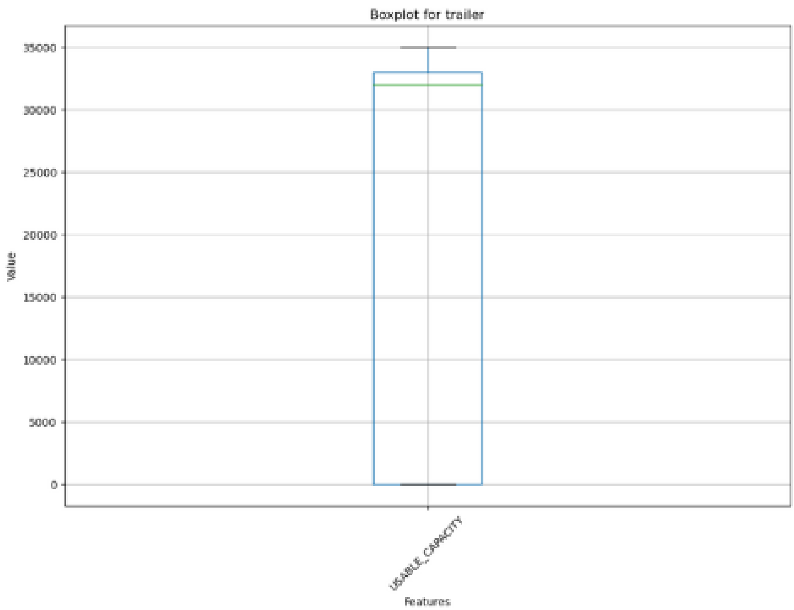


Figure 23: Boxplot for USABLE_CAPACITY in Trailer.xls after removing outliers

3.3.3 Challenges Encountered

We lacked information on departure points and destinations. So, we chose **JPS JORF PETROLIUM STORAGE** as a departure point and **19 service stations.**

Figure 25: JORF PETROLEUM STORAGE

Figure 26: Destination service stations

```
sorted(orders['DESTINATION_NUMBER'].unique())
[1, 2, 3, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 17, 18, 19]
```

	DESTINATION_NUMBER	DESTINATION_NAME	DESTINATION_LOCATION
0	1	Station Service Afriquia	https://maps.app.goo.gl/DFDSmJJKUgkEBAAs7
1	2	Station Service Cepsa	https://maps.app.goo.gl/f6Q9BAL16irfHR518
2	3	Okay Energy	https://maps.app.goo.gl/t754Yow6ZuL1ab1v6
3	5	Shell	https://maps.app.goo.gl/nCNSydn62ZWbQJwn8
4	6	TotalEnergies ISMAILIYA	https://maps.app.goo.gl/BgtVZb3mvS7W3q5cA
5	7	TotalEnergies JORF	https://maps.app.goo.gl/Rrr8HEH75YXcuT7P9
6	8	Station Service Petromin	https://maps.app.goo.gl/PVuANCgQnR2v1n3a9
7	9	Guiri Afriquia Fuel Station	https://maps.app.goo.gl/ZicpnHz2ye5UPVyM9
8	12	TotalEnergies AOUAL MAHFOUD	https://maps.app.goo.gl/2eUEALRU9v5pPQ8x5
9	13	Station Service AM Petro Service	https://maps.app.goo.gl/MwTCH1HjrDfQkroh9
10	14	Station Service Petrofib	https://maps.app.goo.gl/2Y7RvfvWqGgyXTQc7
11	15	TotalEnergies RELAIS MAZAGAN	https://maps.app.goo.gl/PgLqY8NujuM3kTkM8
12	16	TotalEnergies AL FATH	https://maps.app.goo.gl/askpVyhjRKptzCD36
13	17	Total service station Doukkala	https://maps.app.goo.gl/fFgLA1y7t13dJF8C8
14	18	Service Station Tangueda	https://maps.app.goo.gl/KeH1D4zcLiy7RLsm9
15	19	Station Service Samir	https://maps.app.goo.gl/pkcqCUxZPmtE2g7D6

Figure 27: Destination service stations

We linked each destination number to a service station randomly.

3.4 Equipment Monitoring

3.4.1 Used Datasets

To achieve Data Science Objectives, we used four data frames: **injector_history**, **meter_history**, **orders,** and **order_recipe[1-8].**

Merged_meter is obtained by merging **order_recipe[1-8].xls**, **orders.xls** and **meter_history.xls**. It contains useful features such as GROS_UNNCOUNTED, ACTUAL_GROSS_THRUPUT, GROSS_THRUPUT, COMMITED_NET_QUANTITY, and ORDERED_QUANTITY.

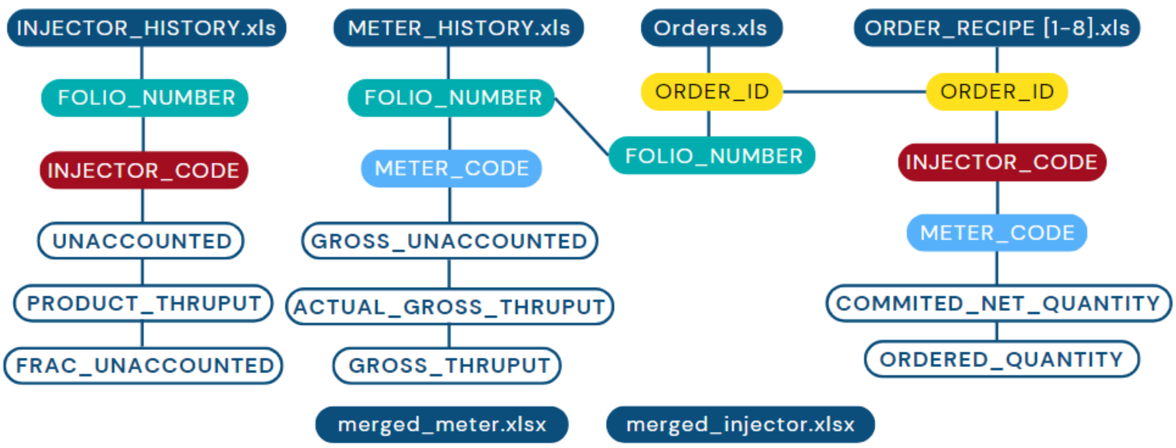


Figure 28: Equipment Monitoring Dataset Schema

3.4.2 Data Cleaning Phase

Step 0:

We started by loading the datasets.

Step 1:

We select the required features from each dataset.

```
columns_to_keep_injector_history = ['INJECTOR_CODE','FOLIO_NUMBER','UNACCOUNTED','FRAC_UNACCOUNTED','PRODUCT_THRUPUT']
columns_to_keep_meter_history = ['METER_CODE','FOLIO_NUMBER','NET_UNACCOUNTED','GROSS_UNACCOUNTED','GROSS_THRUPUT','NET_THRUPUT,
                                 'ACTUAL_GROSS_THRUPUT']
columns_to_keep_order_recipe1 = ['ORDER_ID','INJECTOR_CODE','METER_CODE','ORDERED_QUANTITY','COMMITTED_NET_QUANTITY']
columns_to_keep_orders = ['ORDER_ID','FOLIO_NUMBER']
```

Step 2:

```
order_recipe = pd.concat([order_recipe1, order_recipe2, order_recipe3, order_recipe4, order_recipe5,
                          order_recipe6, order_recipe7, order_recipe8], ignore_index=True)
```

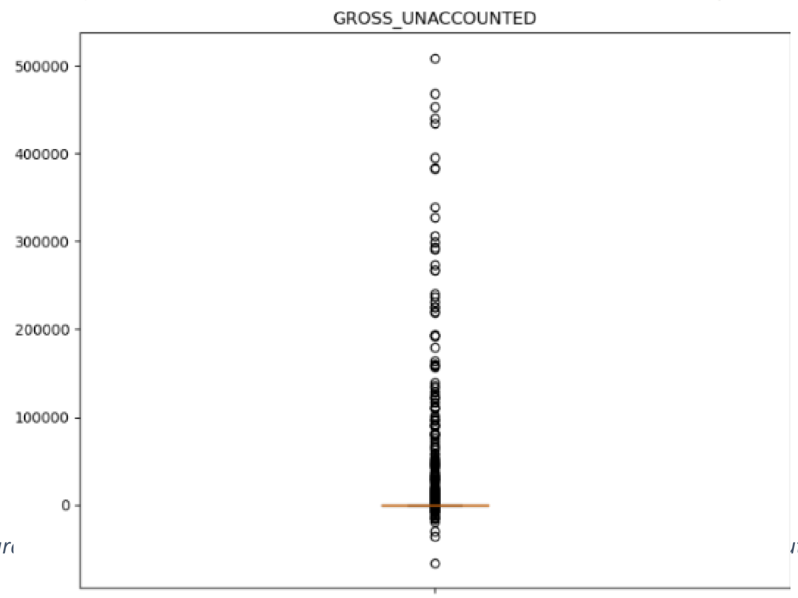
We concatenated orders with order_recipe.

Step 3:

We checked for NaN values.

Step 4:

We checked for outliers, and we decided to remove them.



Figur ... ıtliers

Figure 30: Boxplot for GROSS_UNACOUNTED in merged_meter.xls after removing outliers

3.4.3 Challenges Encountered

```
merged_meter['Theft1'] = merged_meter['ORDERED_QUANTITY'] - merged_meter['GROSS_THRUPUT']
merged_meter['Theft2'] = merged_meter['COMMITTED_NET_QUANTITY'] - merged_meter['GROSS_THRUPUT']
merged_meter['Theft3'] = merged_meter['COMMITTED_NET_QUANTITY'] - merged_meter['ACTUAL_GROSS_THRUPUT']
```

```
             Theft1         Theft2         Theft3
count  7.187000e+03    7187.000000    7187.000000
mean   2.628825e+04   -1109.842076   -3135.848198
std    6.009380e+04    8875.662142   23306.278883
min   -2.418590e+05 -243827.000000 -509529.000000
25%    5.870000e+02   -1808.000000   -1948.000000
50%    1.061200e+04    -449.000000    -482.000000
75%    2.801400e+04      36.000000      20.000000
max    1.005531e+06   66245.000000   66244.000000
```

Theft1	Theft2	Theft3
-165771	-172173	-180171
20270	7443	-6552
33029	-8341	-8345
34763	-11824	-11825
8295	-17052	-17055
...	...	...
390058	-2420	-2437
12	-655	-656
590784	-9787	-9811
344476	19047	19033
100340	5616	5611

There's a significant variance between the commited_net_quantity or ordered_quantity and the calculations made by meter or injector.

4 Modeling

4.1 Introduction

The initial stage of our project involved harnessing cutting-edge technology to gather vital data necessary for route optimization and predictive assessment of travel durations. This phase played a pivotal role in facilitating streamlined logistics activities and bolstering the predictive prowess of our analytical models. Here, we delineate the fundamental procedures and approaches utilized during the data acquisition endeavor.

4.2 Inventory Management

4.2.1 Tank Quantity Prediction

In our inventory management strategy, we implemented various predictive modeling techniques to achieve precise estimations of oil stock levels leveraging bulk movement data. Specifically, we utilized Exponential Smoothing (ETS), Autoregressive Integrated Moving Average (AutoReg), and Seasonal Autoregressive Integrated Moving Average (SARIMA) methods to forecast future stock levels with accuracy.

After conducting predictive modeling using Exponential Smoothing (ETS), Autoregressive (AutoReg), and Seasonal Autoregressive Integrated Moving Average (SARIMA) techniques, we evaluated their performance based on the Root Mean Squared Error (RMSE) metric. The RMSE values obtained were 1.4 * 10^8 for ETS, 1.9 * 10^8 for AutoReg, and notably lower at 2.5 * 10^7 for SARIMA. These results indicate that SARIMA outperformed both ETS and AutoReg in terms of accuracy, demonstrating its superior ability to forecast oil stock levels based on bulk movement data. Therefore, SARIMA stands out as the most effective model for our inventory management system, providing the most reliable estimates for optimizing stock levels and ensuring efficient operations.

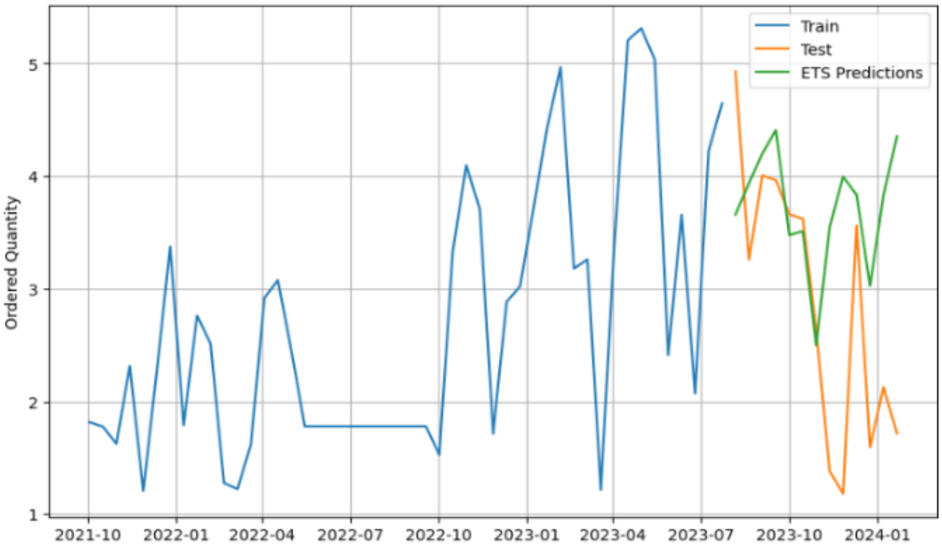


Figure 32: ETS Forecast for Gross Quantity TK-307

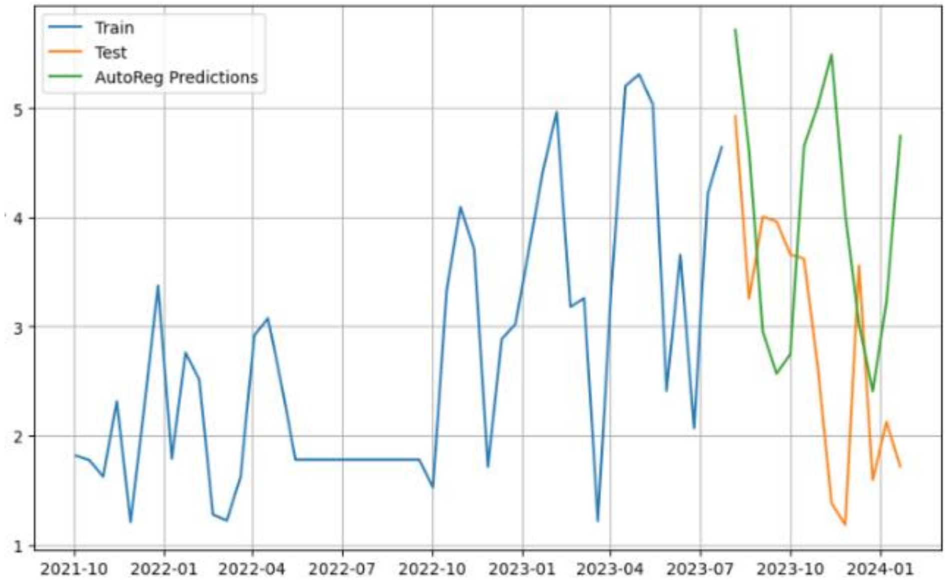


Figure 33: AutoReg Forecast for Gross Quantity TK-307

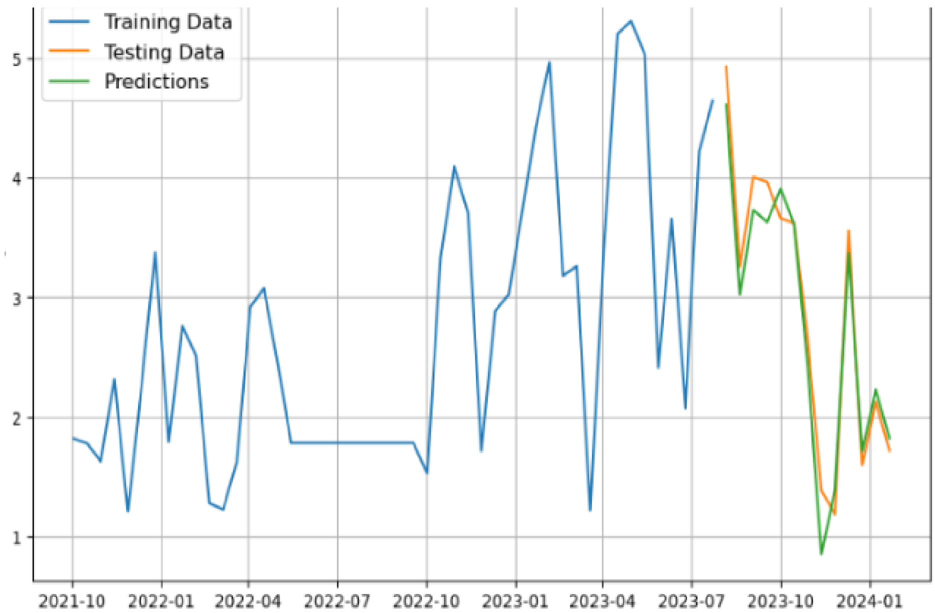


Figure 31: Sarima Forecast for Gross Quantity TK-307

In our inventory management project, SARIMA emerges as the top performer in forecasting oil stock levels. SARIMA, a variant of ARIMA tailored for seasonal data, utilizes hyperparameters p, d, and q for non-seasonal trends, along with P, D, and Q for seasonal patterns. By analyzing the autocorrelation and partial autocorrelation functions, we optimize these parameters. The PACF and ACF plots guide us in selecting the optimal configuration. SARIMA's adaptability to capture both seasonal and non-seasonal dynamics makes it a reliable choice for precise stock level predictions, empowering efficient resource management in our system.

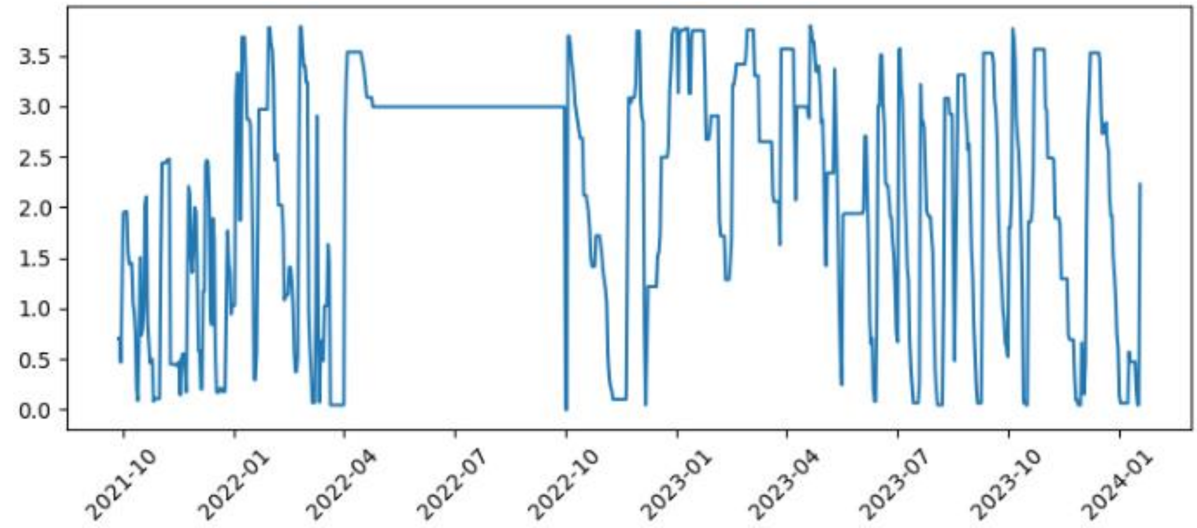


Figure 36: Time Series for Tank-304

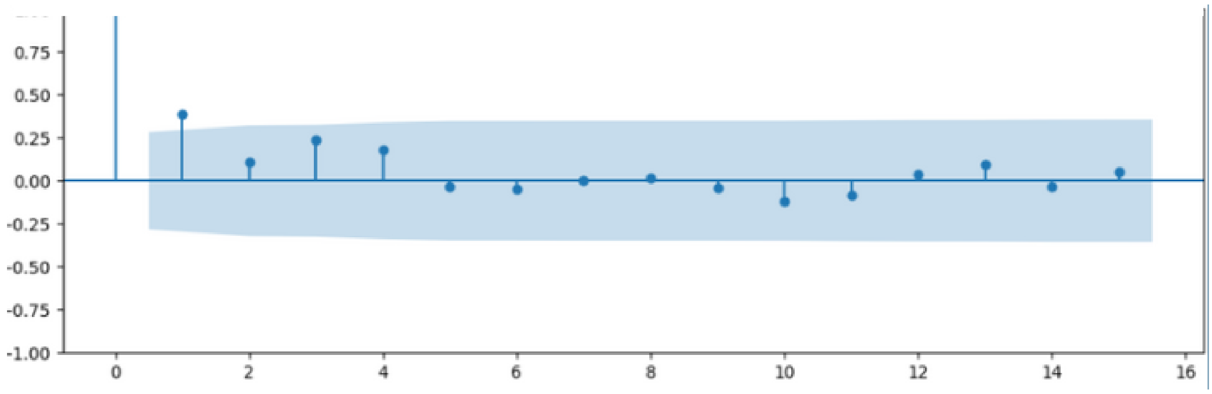


Figure 35: Autocorrelation for TS Tank-304

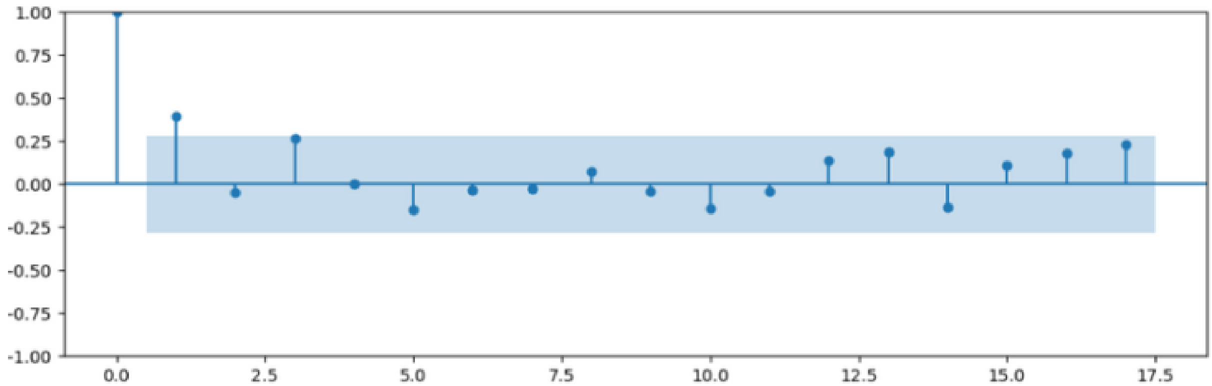


Figure 34: Partial Autocorrelation for TS Tank-304

Our predictive modeling efforts culminate in a comprehensive forecast, extending four months into the future. Leveraging SARIMA, our system provides a nuanced prediction of oil stock levels, offering insights that guide proactive decision-making. This predictive capability enables stakeholders to anticipate future demand trends, optimize resource allocation, and streamline logistical operations. With a forward-looking perspective, our system empowers organizations to adapt and thrive in a dynamic and ever-changing environment.

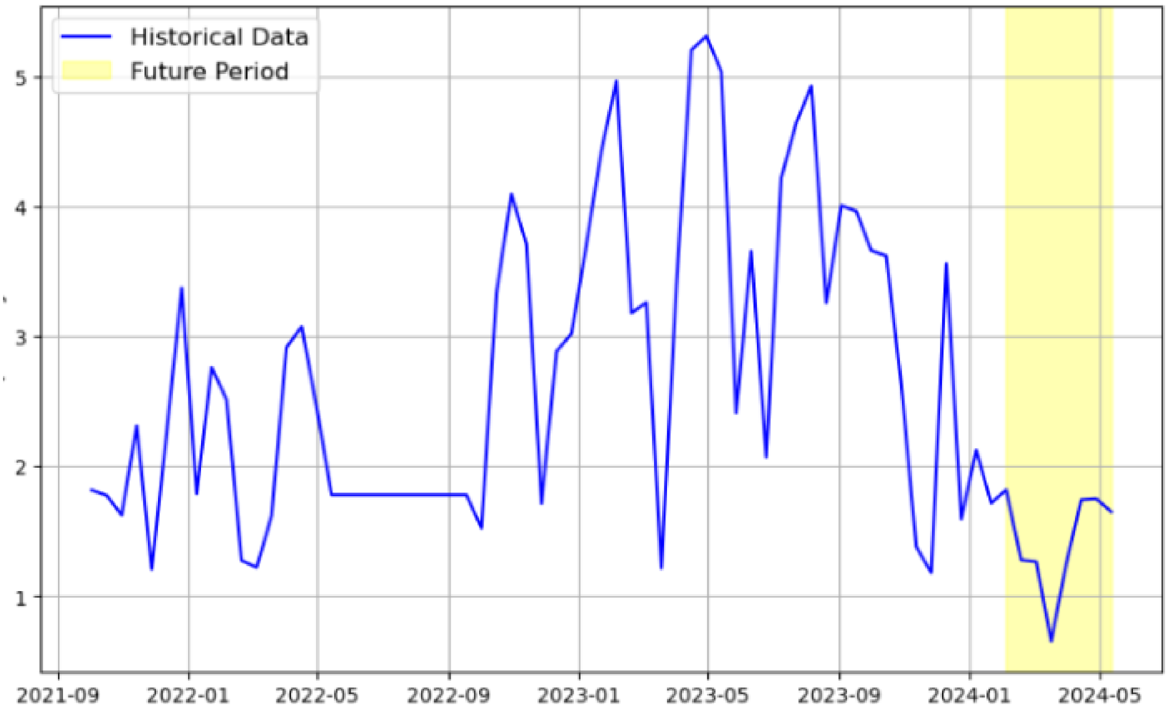


Figure 37: Tank 307 gross quantity with 4-month prediction

4.2.2 Scraping Temperature Data

Upon completing the data scraping process, we acquired temperature data specific to El Jadida, Morocco. To refine our dataset, we implemented filtering criteria aimed at isolating days characterized by optimal weather conditions. Specifically, we focused on identifying days when the wind speed remained below 20 km/h and the temperature stayed below 23°C. This meticulous filtering ensures that our subsequent analyses and modeling efforts are built upon a foundation of weather data representative of favorable conditions for our intended applications. By selecting data points meeting these specific thresholds, we enhance the precision and relevance of our analyses, facilitating more accurate insights and informed decision-making processes.

	day	date	Weather	Wind
0	Mon	Mon29 Apr	18 °C	22 km/h
1	Tue	Tue30 Apr	19 °C	19 km/h
2	Wed	Wed1 May	20 °C	20 km/h
3	Thu	Thu2 May	19 °C	34 km/h
4	Fri	Fri3 May	21 °C	26 km/h
5	Sat	Sat4 May	24 °C	34 km/h
6	Sun	Sun5 May	24 °C	28 km/h
7	Mon	Mon6 May	23 °C	18 km/h
8	Tue	Tue7 May	24 °C	14 km/h
9	Wed	Wed8 May	24 °C	21 km/h
10	Thu	Thu9 May	24 °C	20 km/h
11	Fri	Fri10 May	21 °C	19 km/h
12	Sat	Sat11 May	21 °C	24 km/h
13	Sun	Sun12 May	25 °C	20 km/h
14	Mon	Mon13 May	25 °C	20 km/h

Figure 39: Data scrapping from a weather app

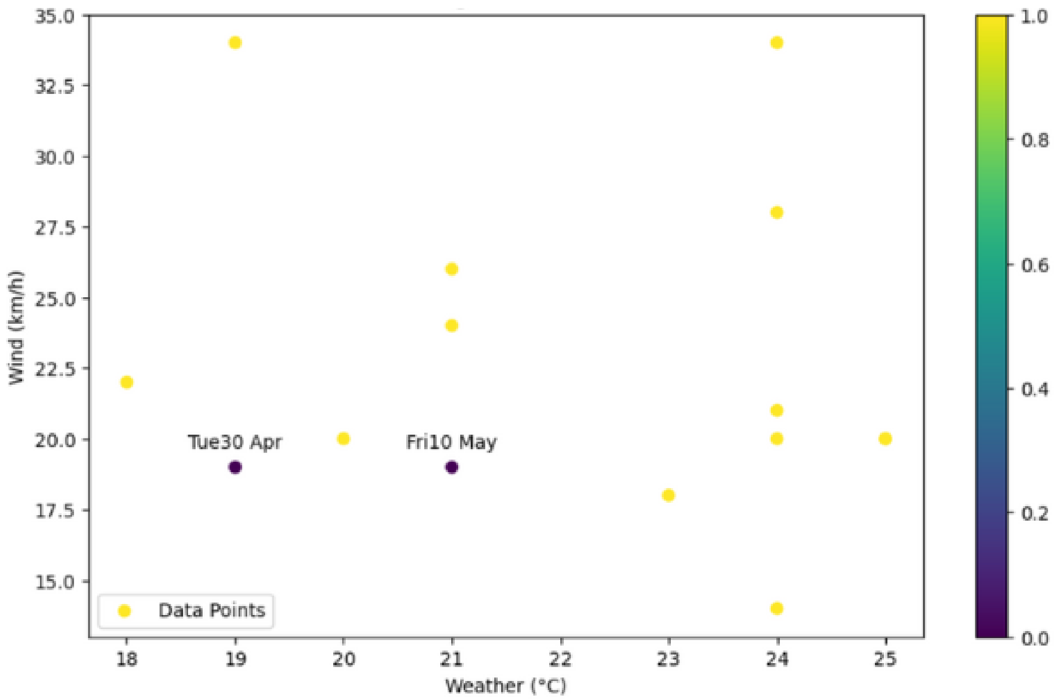


Figure 38: Weather and Wind Clustering

4.3 Orders Forecasting:

In our pursuit of optimizing inventory management, we extended our predictive modeling endeavors to forecast orders with precision. Employing a similar approach to our stock level predictions, we utilized a combination of advanced techniques, including Exponential Smoothing (ETS), Autoregressive Integrated Moving Average (AutoReg), and Seasonal Autoregressive Integrated Moving Average (SARIMA).

Our findings underscore SARIMA's effectiveness in delivering precise predictions tailored to our inventory management needs. Leveraging its sophisticated framework, SARIMA optimizes hyperparameters such as p, d, and q for non-seasonal trends, along with P, D, and Q for seasonal patterns. By harnessing insights from autocorrelation and partial autocorrelation functions, we fine-tuned SARIMA's parameters to capture both seasonal and non-seasonal dynamics effectively.

With SARIMA at the helm of our predictive modeling efforts, we extend our forecasting capabilities into the realm of order quantities, providing stakeholders with nuanced insights that inform proactive decision-making. This predictive prowess empowers organizations to anticipate demand trends, optimize resource allocation, and enhance logistical operations, fostering adaptability and resilience in a rapidly evolving business landscape.

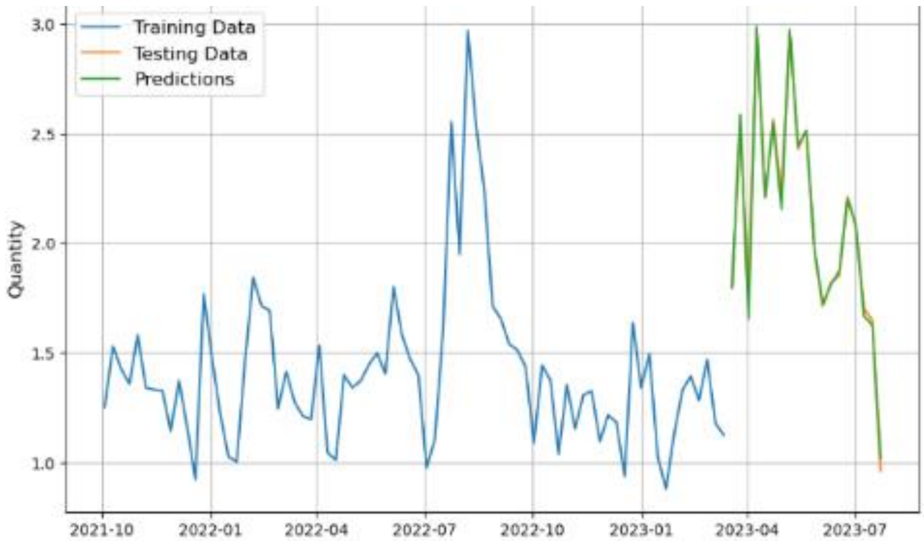


Figure 40: Gross quantity forecast for customer 3 Product 4

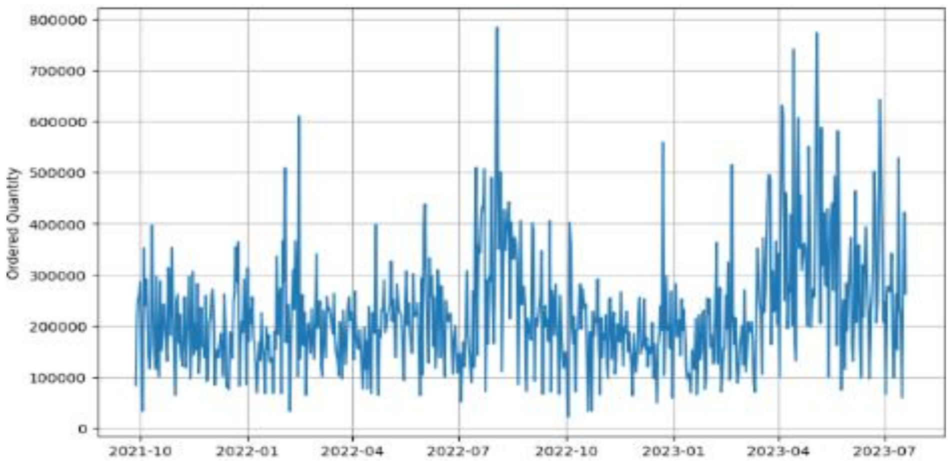


Figure 41: Time series product customer 3 product 4

4.4 Delivery Optimization:

In our quest for delivery optimization, we employed the Travelling Salesman Problem (TSP) algorithm to devise the most efficient routes for visiting all locations precisely once, starting and ending at the departure point. The first step involved constructing a comprehensive Data Frame encapsulating all possible connections between two locations. To enrich this dataset, we integrated distance, duration, and the corresponding path for each route, leveraging the capabilities of OSRM (Open-Source Routing Machine). Subsequently, we implemented an algorithm tasked with traversing through each potential route, assigning a score to each based on normalized duration and consumption metrics. The optimal route was determined as the one with the lowest cumulative score, signifying the most efficient itinerary for our delivery operations. By harnessing the power of the TSP algorithm and strategic data integration, we aimed to streamline our logistical processes and enhance delivery efficiency across all destinations.

Figure 42: Example of applying the TSP algorithm

4.5 Equipment Monitoring:

In our equipment monitoring, we have two objectives: the first is to develop a clustering model for detecting malfunctioning meters and injectors. The second objective is to create another clustering model to detect leaks in tanks.

To achieve these goals, we developed an algorithm based on GROSS-UNACCOUNTED and FRAC_UNACCOUNTED values to deduce the number of days since the last anomaly, the number of consecutive anomalies, and the total liters of product for the sequence of anomalies, considering a threshold of ± 20 liters for anomalies.

Meters data

METER_CODE	FOLIO_NUMBER	ORDERED_QUANTITY	COMMITTED_NET_QUANTITY	GROSS_UNACCOUNTED
MTR_201	20210927.0	172100	165698	7998
MTR_201	20210928.0	469000	456173	13995
MTR_201	20210929.0	564000	522630	4
MTR_201	20210930.0	598000	551413	1
MTR_201	20211001.0	308100	282753	3
...	...	...	...	...
MTR_503	20230715.0	655000	262522	17
MTR_503	20230716.0	71000	70333	1
MTR_503	20230717.0	1055000	454429	24
MTR_503	20230718.0	771000	445571	14
MTR_503	20230719.0	400000	305276	5

GROSS_Unaccounted = Actual_Gross_Thruput - Gross_Thruput

Injectors data

INJECTOR_CODE	FOLIO_NUMBER	ORDERED_QUANTITY	COMMITTED_NET_QUANTITY	UNACCOUNTED	FRAC_UNACCOUNTED
INJ_201	20210928.0	10200	10200	0	-1
INJ_201	20210929.0	24600	22200	0	1
INJ_201	20210930.0	16200	16200	0	0
INJ_201	20211001.0	25800	20700	0	-1
INJ_201	20211002.0	26400	26400	0	1
...	...	...	...	...	...
INJ_509	20230118.0	4250	4250	0	2
INJ_509	20230120.0	3250	3250	0	0
INJ_509	20230121.0	4500	4500	0	2
INJ_509	20230301.0	46000	5000	0	0
INJ_509	20230307.0	42500	3250	0	0

Frac_Unaccounted = Actual_Frac_Thruput - Frac_Thruput

METER_CODE	successive_non_zero	successive_non_zero_abs_sum	last_date_non_zero
MTR_201	1	14995	356
MTR_202	1	4987	356
MTR_203	1	11991	356
MTR_301	1	7999	2
MTR_302	1	13589	356
MTR_303	2	54	286
MTR_401	2	82	285
MTR_402	1	144	298
MTR_403	1	40	286
MTR_501	2	47	285
MTR_502	1	29	286
MTR_503	1	24	286

Subsequently, we applied the elbow method to determine the optimal number of clusters and found that the optimal number was 4 clusters for the meters and 3 clusters for the injectors.

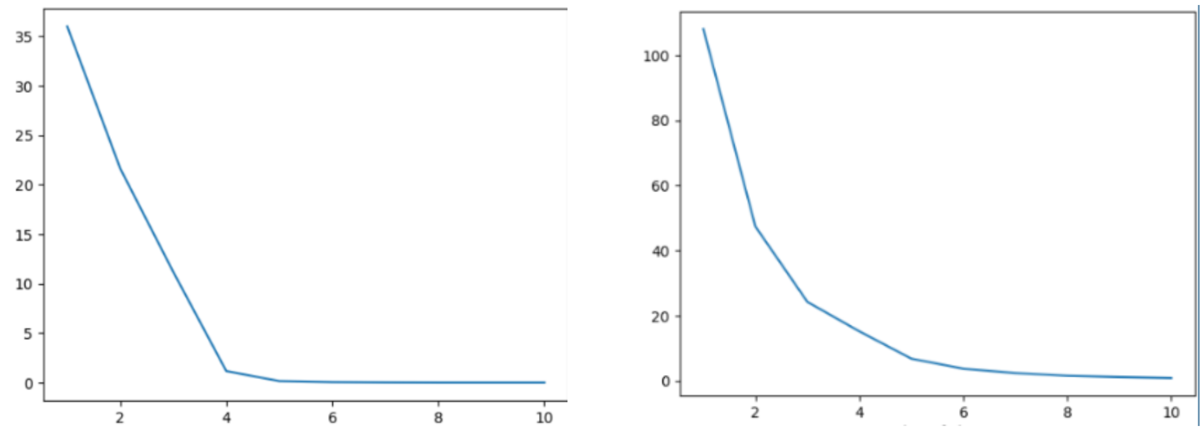


Figure 43: Application of the Elbow Method

After applying the KMeans clustering, we obtained the following results.

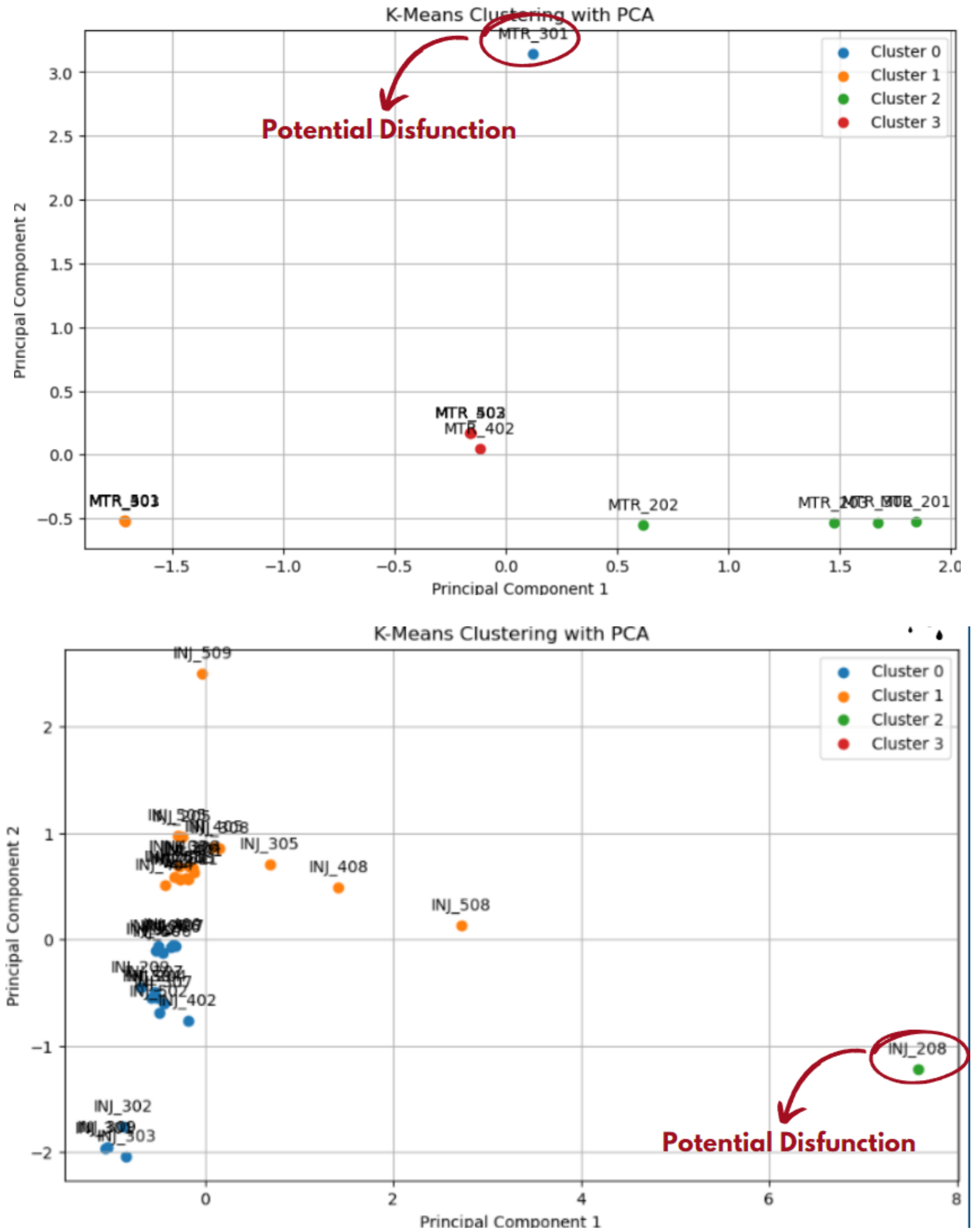


Figure 44: Results of applying the KMeans clustering

As for leak detection in tanks, we focused on the quantity of product recorded on each order (Net_Quantity) and the quantity of product leaving each tank (Rack_Disposals). The difference between these two variables is termed (Quantity_difference).

FOLIO_NUMBER	TANK_CODE	TERMINAL_PRODUCT_NUMBER	NET_QUANTITY	RACK_DISPOSALS	quantity_difference
20210927	TK-100	8	51	92	-41
20210928	TK-100	8	115	122	-7
20210929	TK-100	8	204	216	-12
20210930	TK-100	8	186	200	-14
20211001	TK-100	8	181	191	-10
...	...	...	...	...	...
20230714	TK-600	11	6	11	-5
20230715	TK-600	11	5	9	-4
20230717	TK-600	11	0	1	-1
20230718	TK-600	11	19	26	-7
20230719	TK-600	11	14	20	-6

Figure 45: Quantity diffrence

Our algorithm will be based on this feature, calculating the successive number of days with leaks (quantity_difference < -20), the total quantity of these leaks, and the date of the last leak.

TANK_CODE	successive_less_minus20	successive_less_minus20_abs_sum	last_date_less_minus20	max_folio_date
TK-100	1	21	295	2023-07-08
TK-101	5	121	296	2023-07-07
TK-102	3	395	338	2023-05-26
TK-103	2	57	284	2023-07-19
TK-104	1	25	307	2023-06-26
TK-105	2	731	411	2023-03-14
TK-106	1	91	300	2023-07-03
TK-200	0	0	0	No less than -20 values found
TK-201	1	61	361	2023-05-03
TK-202	1	21	288	2023-07-15
TK-203	1	29	286	2023-07-17
TK-205	4	94	345	2023-05-19
TK-300	0	0	0	No less than -20 values found
TK-305	0	0	0	No less than -20 values found
TK-306	0	0	0	No less than -20 values found
TK-400	0	0	0	No less than -20 values found
TK-500	1	39	944	2021-09-27
TK-600	0	0	0	No less than -20 values found

Figure 46: Output of the leak detection algorithm

We then applied the elbow method and noticed that the optimal number of clusters is 5.

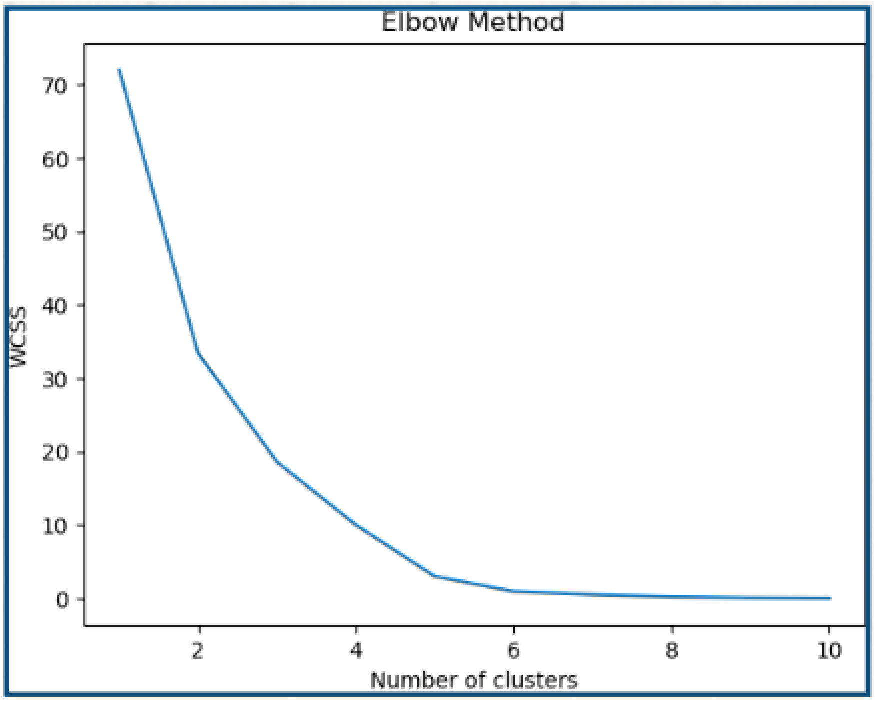


Figure 47: Elbow Method Application

Here are the results of the KMeans clustering:

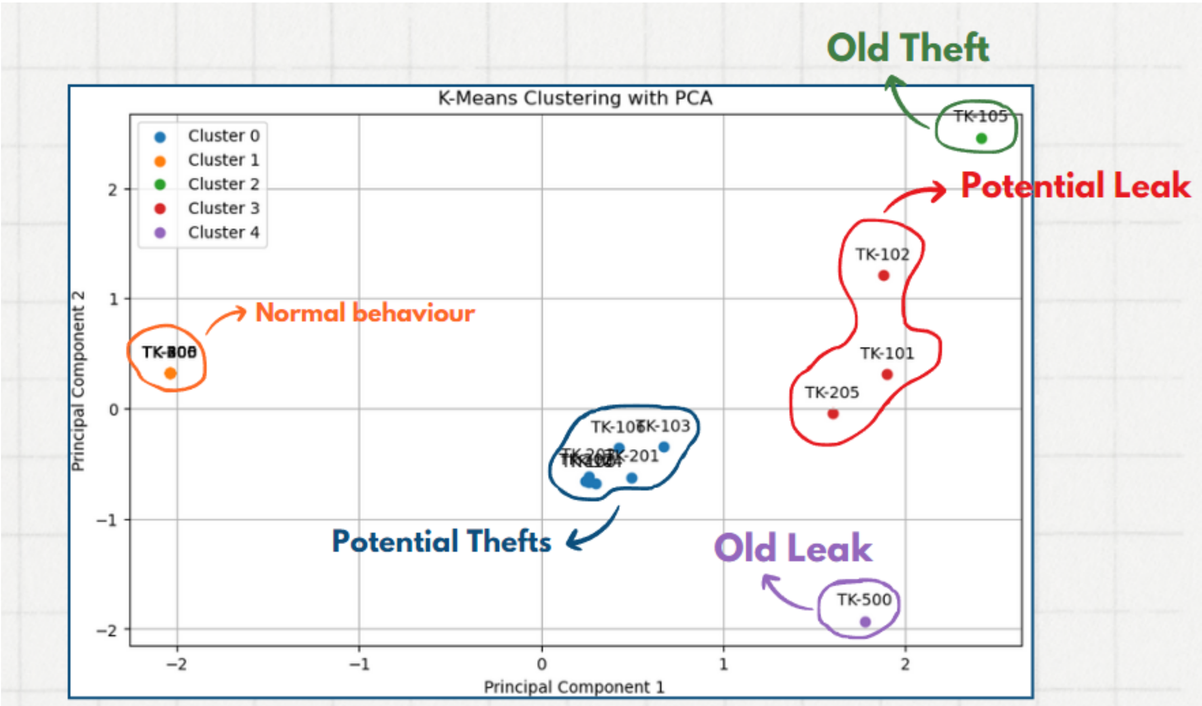


Figure 48: plotting the output of the leak detection algorithm

5 Deployment

5.1 Technology Used:

In developing our web application, we leveraged Flask for web development and Anaconda for model construction, utilizing Python as the primary programming language. This integration facilitated a smooth integration of features across our development phases.

5.2 Web Site Application:

5.2.1 Inventory Management:

The Inventory Management interface is the place where users can access all information and functionalities relating to the tanks.

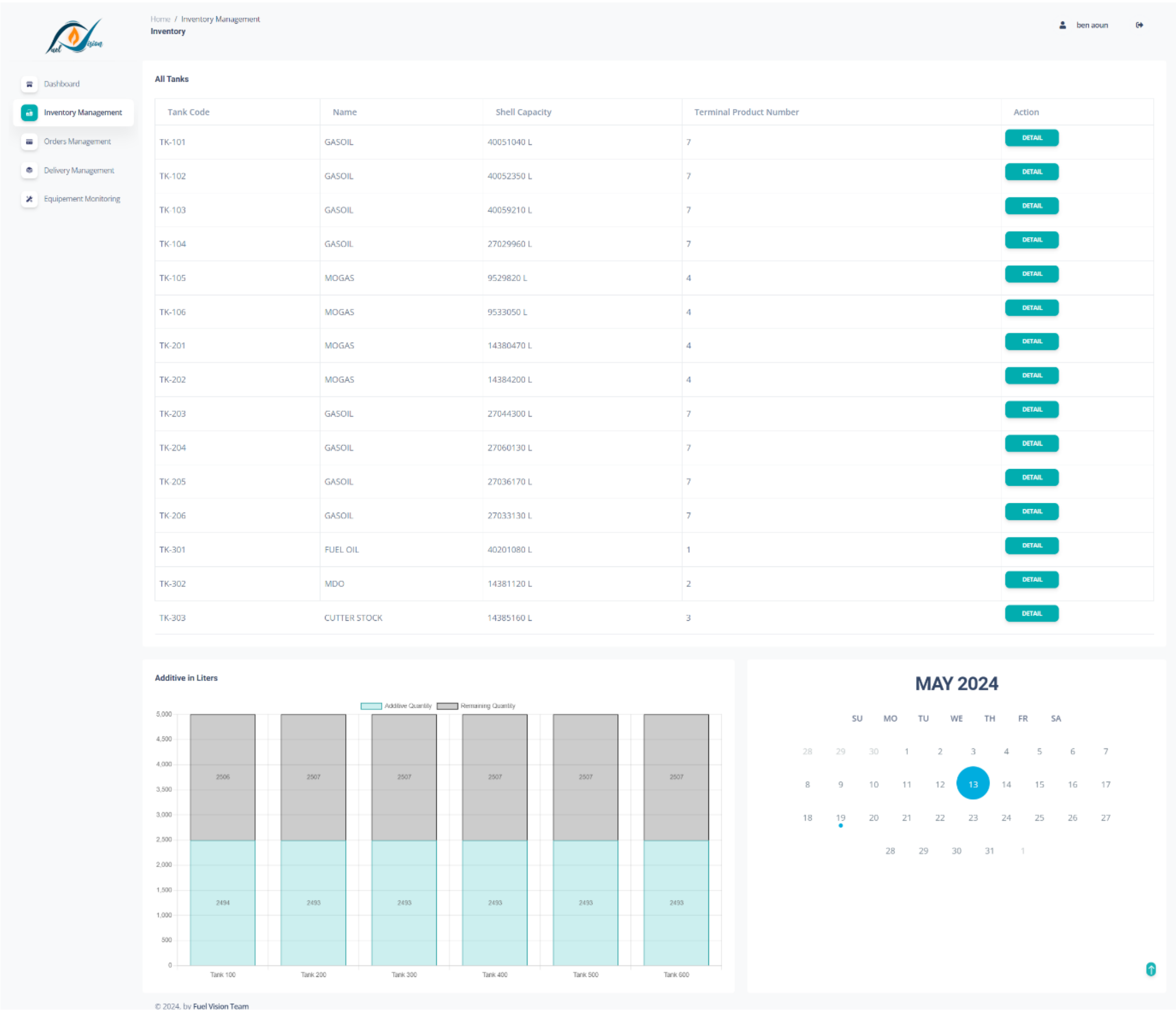


Figure 49: The Inventory Management Interface

At the top of the interface, the primary tanks hold essential information such as product name, shell capacity, and an option to delve into more detailed information by clicking on the details button.

Below this, users can easily view tanks designated for additive products. Each tank is listed with its respective quantity of additives and the remaining capacity available.

Additionally, the system includes a calendar feature crucial for planning logistics. The calendar highlights days throughout the month when weather conditions are optimal for unloading or moving oil from the harbor. This feature aids in efficient scheduling, ensures smooth operations, and reduces oil evaporation.

5.2.2 Order Management:

The Order Management Interface provides effortless access to all information regarding orders, ensuring seamless retrieval and management.

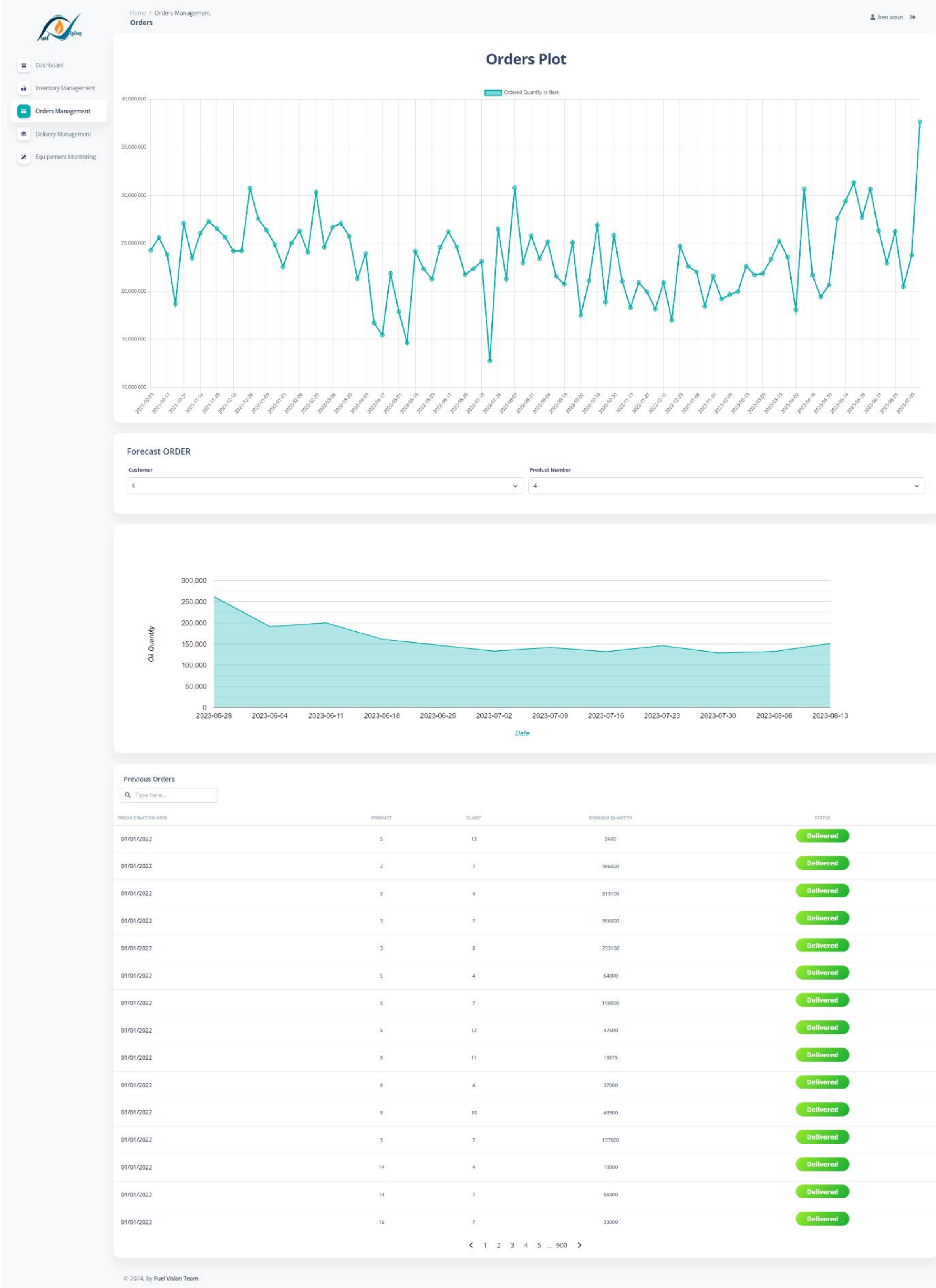


Figure 50: Order Management Interface

The first section of the interface focuses on presenting a plot showcasing the order quantity over time.

Following this, the forecast section allows users to select a customer and a product, generating a plot that displays all orders for that specific client and product along with a 4 to 8-week forecast.

In the final section, all orders are arranged chronologically by their creation date. Each row presents essential details about the order, including delivery status, order quantity, client, and product type.

5.2.3 Delivery Management:

The Delivery Management section serves as the repository for delivery archives. Within this interface, each row provides details such as the product type, the number of stops, and the total quantity loaded onto the trailer for each delivery.

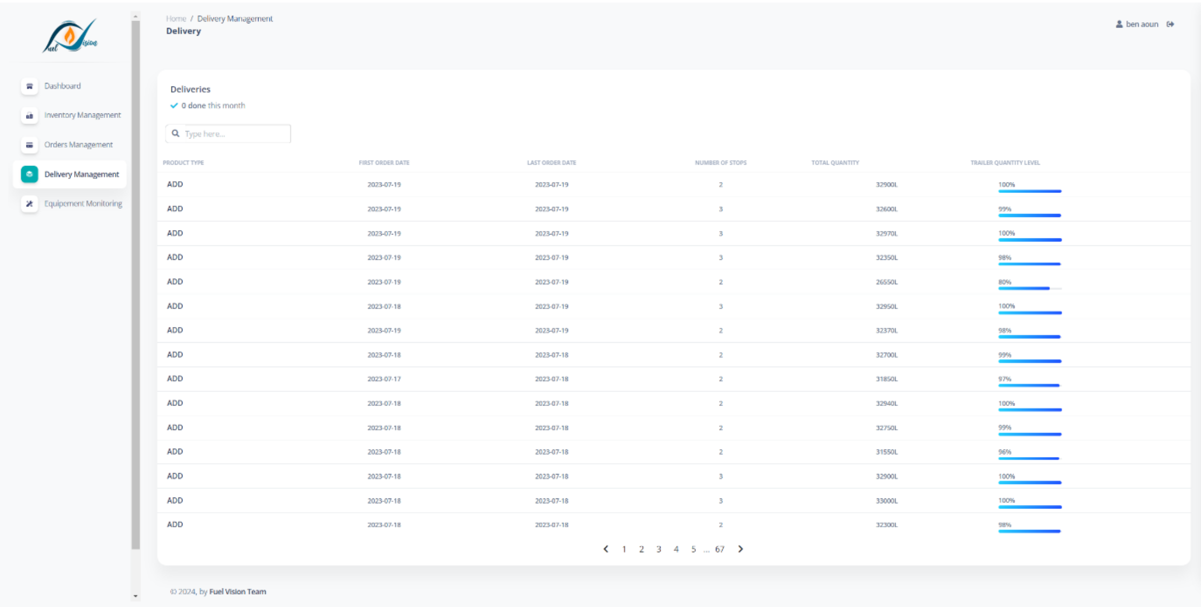


Figure 51: Delivery Management Interface

By selecting a specific delivery, users gain access to crucial information including the optimal path, station order, time intervals between stops, and fuel consumption for the entire trip measured in liters. Additionally, an interactive map feature is provided to enhance visualization, displaying the locations of stations and routes for better navigation and planning.

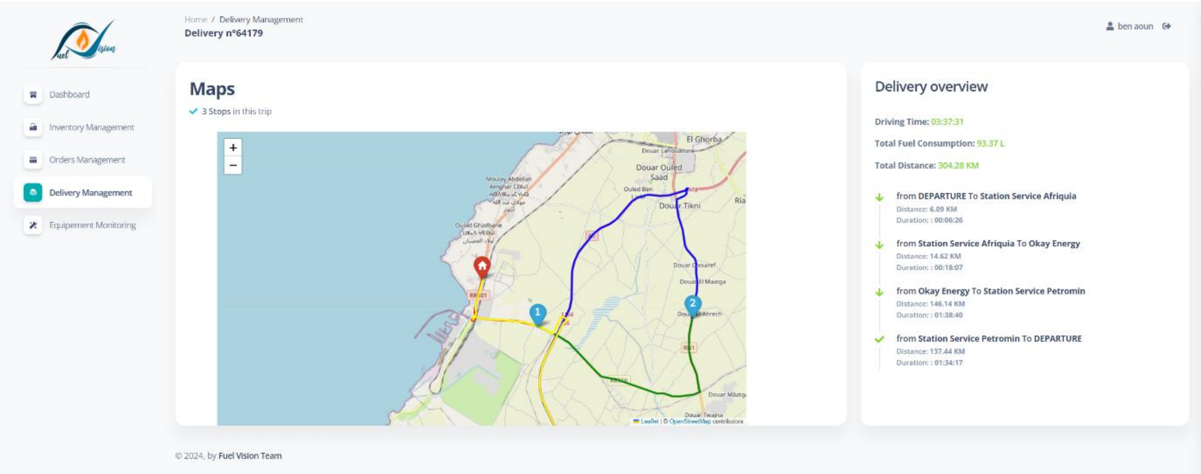


Figure 52: Map Preview in Delivery Management

5.2.4 Equipment Monitoring:

The Equipment Monitoring interface is structured into three primary sections: meter monitoring, injector monitoring, and tank monitoring. Each section offers fast access for adding new records.

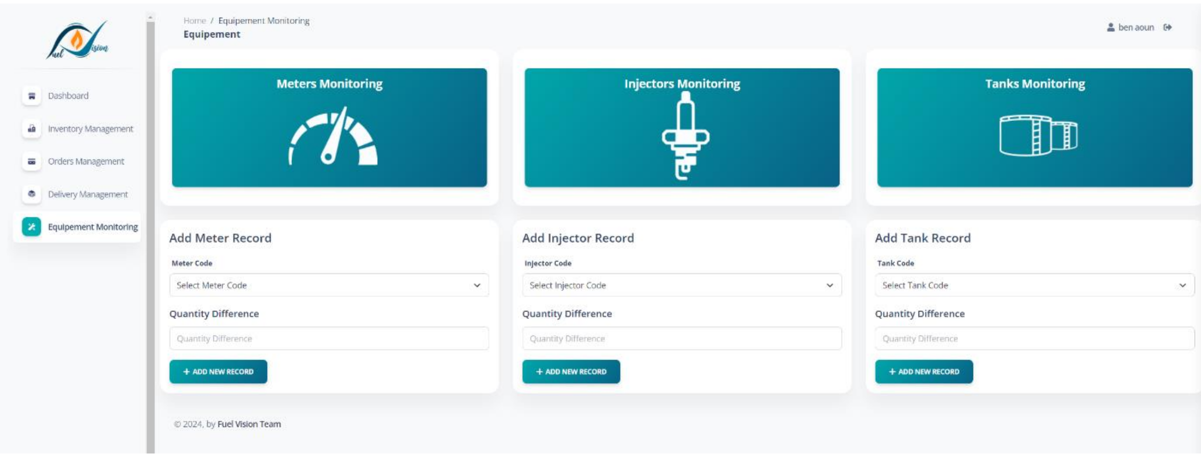


Figure 53: Equipment Monitoring Interface

Within each of the meter monitoring, injector monitoring, and tank monitoring interfaces, users can access three main sections:

- Monitoring: This section provides real-time monitoring of the respective equipment, offering insights into its status.
- Statistics: Users can access statistical data related to the monitored equipment.
- Add Record: This section allows users to effortlessly add new records or observations related to the equipment.

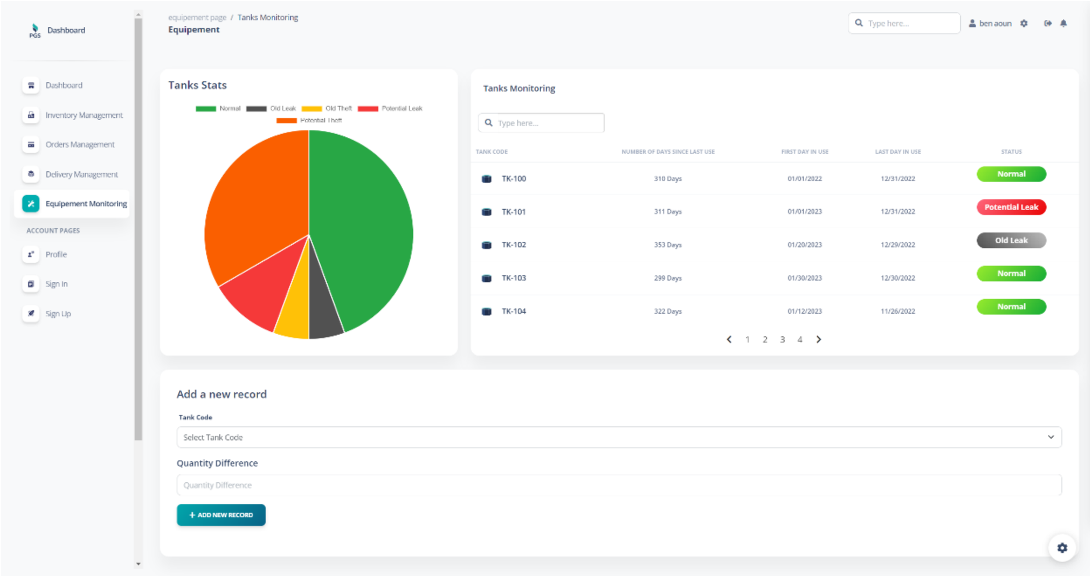


Figure 54: Tank Interface in Equipment Monitoring

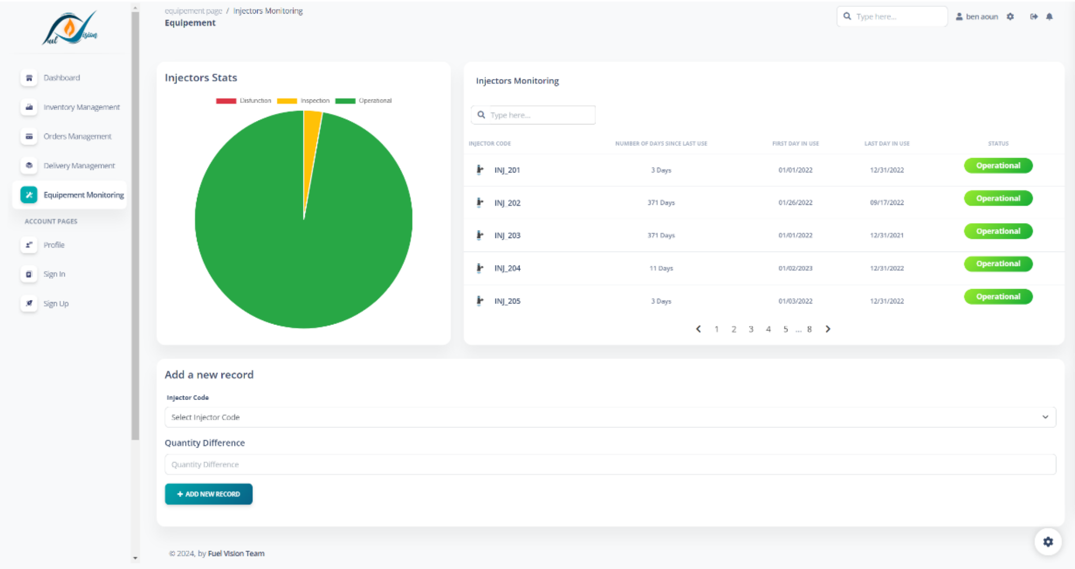


Figure 55: Injector Interface in Equipment Monitoring

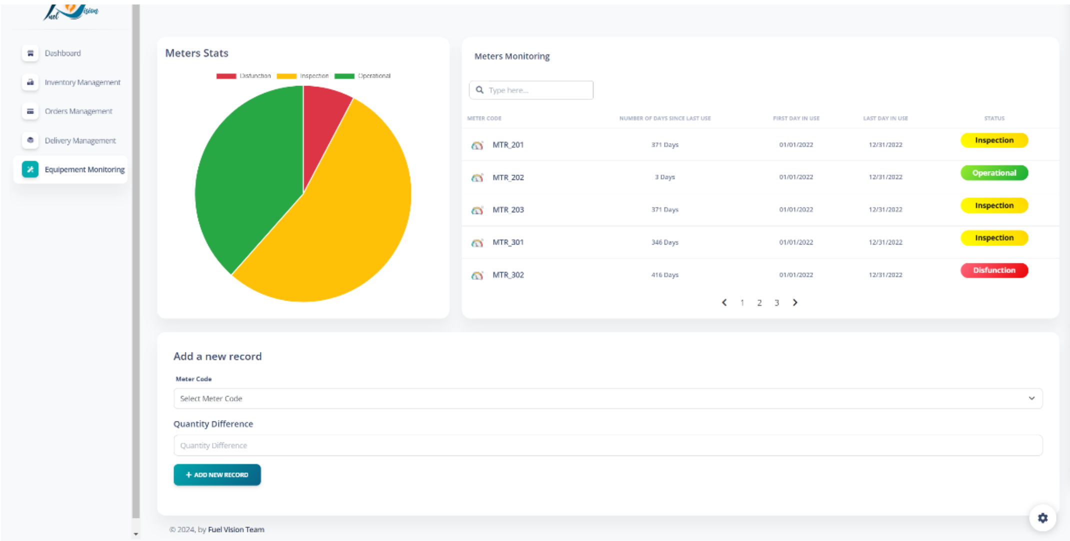


Figure 56: Meter Interface in Equipment Monitoring

5.2.5 Dashboard:

The dashboard serves as a user-friendly interface, providing easy access to all sections of the website, with a sidebar facilitating navigation. Positioned in the middle of the dashboard is a summary of key information, such as the number of injectors, tank orders, and other pertinent details.

Additionally, quick access to additive-level information in the tanks is included for efficient monitoring. The dashboard is thoughtfully designed to showcase the most relevant and useful information for quick access and decision-making.

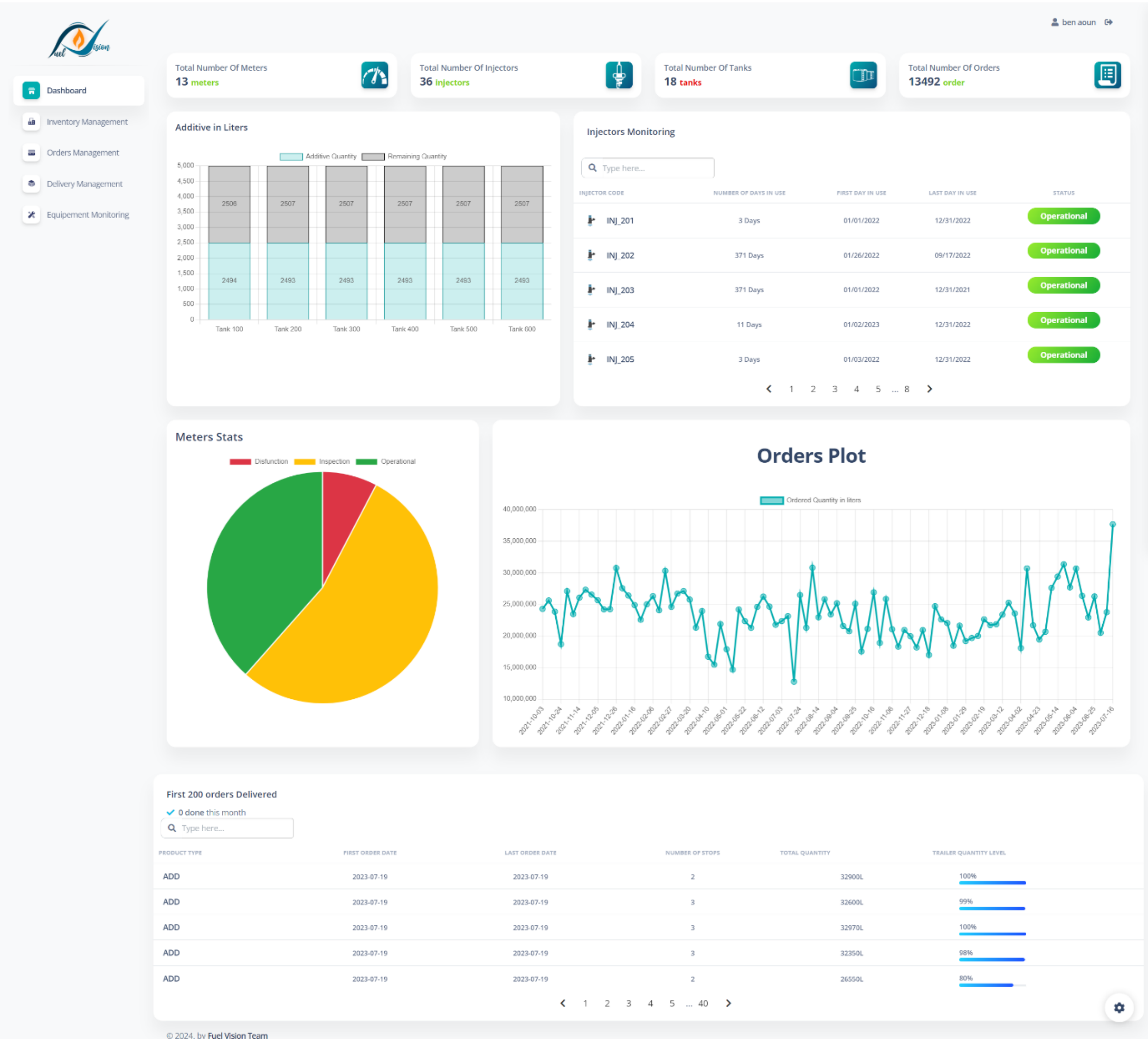
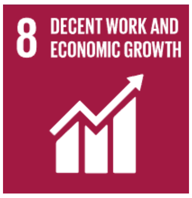


Figure 57: Dashboard Interface

5.3 Achieved SDGs:

Prediction on oil stock levels:

Predicting the amount of oil in each tank will give us an insight into the remaining quantity, aid in decision-making, and subsequently enhance the company's profitability.



Recommendation of the best date for transferring from ships to tanks:

Knowing the best date for transferring oil from barges to tanks, based on wind and temperature conditions, is crucial to minimize the risk of accidents that could lead to oil spills or other environmental and health damages.

Prediction on oil orders

Predicting the quantities of products ordered will help us understand the various patterns and seasonality of the orders. This will enhance our productivity and our knowledge of the market, ultimately leading to increased profitability.

Optimal root based on fuel consumption

Determining the best path while taking into consideration the truck's fuel consumption will reduce our impact on the environment and make our consumption and production more responsible.

Optimal root based on duration

Determining the best path while taking into consideration the duration of the trip will promote economic growth, boost delivery productivity, and enhance partnerships with gas stations.

Leaks Detection

To prevent leaks into groundwater and avoid degrading life on Earth and land.

Meters & Injectors Clustering

The anomaly detection system in injectors and meters will improve productivity and work efficiency, and it will reduce the maintenance time associated with the equipment.

Conclusion

In this report, we detailed the development and implementation of a robust forecasting system aimed at solving the critical challenges in oil depot management. The system is designed to enhance stock management, logistics optimization, and customer satisfaction, utilizing advanced analytics and machine learning algorithms to provide data-driven solutions.

Both functional and non-functional requirements have been precisely addressed to ensure that the solution meets operational needs whilst maintaining high standards of performance, scalability, and reliability. The emphasis on scalability has enabled the system to adapt to future growth and evolving industry dynamics, positioning it to remain relevant and effective in the long term.

A user-friendly web application has been created, seamlessly integrating the forecasting system into the daily operations of the oil depot. The project not only meets current requirements but also paves the way for continued innovation and excellence in this ever-evolving market.

List of References

Data Science and Analysis:

- Microsoft Azure Architecture - Data Science Process Overview: https://learn.microsoft.com/en-us/azure/architecture/data-science-process/overview
- The Data Science Process - Data Science Project Lifecycle: https://www.datascience-pm.com/tdsp/
- Kaggle - Data Cleaning: https://www.kaggle.com/learn/data-cleaning
- Coursera - Time Series and Survival Analysis: https://www.coursera.org/learn/time-series-survival-analysis
- Investopedia - Autoregressive Integrated Moving Average (ARIMA): https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

Development and Programming:

- Twilio - How to Set Up Your Python and Flask Development Environment: https://www.twilio.com/docs/usage/tutorials/how-to-set-up-your-python-and-flask-development-environment

Data Visualization:

- Folium Documentation: https://python-visualization.github.io/folium/latest/
- Google Charts Developer Documentation: https://developers.google.com/chart
- Chart.js Documentation: https://www.chartjs.org/

Geographic Information Systems (GIS):

- Open-Source Routing Machine (OSRM) API Documentation: https://project-osrm.org/docs/v5.5.1/api/#general-options

Database Management:

- MongoDB Manual - Install MongoDB on Windows: https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/

Miscellaneous:

- ChatGPT: https://chatgpt.com/chat
- PGS International: https://pgsintl.com/
- United Nations Sustainable Development Goals: https://sdgs.un.org/goals