MedSync

Elevating Healthcare Inventory Management through Seamless Synchronization.

DSN4096-CAPSTONE PROJECT PHASE-II

Phase – II Report

Submitted by

Shreya Sinha (20BCE10030)

Amar Kumar (20BCE10053)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

SEHORE, MADHYA PRADESH - 466114

MAY 2024

VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **“MedSync, Elevating Healthcare Inventory Management through Seamless Synchronization.”** is the bonafide work of “**Shreya Sinha (20BCE10030), Amar Kumar(20BCE10053)”** who carried out the project work (DSN4096- Capstone Project phase-II) under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr J. Manikandan,
Senior Assistant Professor ,
School of Computing Science and Engineering,
VIT BHOPAL UNIVERSITY.

PROJECT GUIDE
Mr. Ajeet Singh,
Assistant Professor ,
School of Computing Science and Engineering,
VIT BHOPAL UNIVERSITY.

The DSN4096-Capstone Project phase-II Viva Voce Examination is held on ____________

ACKNOWLEDGEMENT

First and foremost, we would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

We would like to thank my internal guide **Mr. Ajeet Singh**, for continually guiding and actively participating in our project, and giving valuable suggestions to complete the project works.

We wish to express our heartfelt gratitude to Dr. J. Manikandan, Head of the Department, School of Computing Science and Engineering for much of his valuable support and encouragement in carrying out this work.

We wish to express our heartfelt gratitude to Dr. S. Poonkuntran, Professor and Dean, School of Computing Science and Engineering for much of his valuable support and encouragement in carrying out this work.

We would like to thank all the technical and teaching staff of the School of Computing Science and Engineering, who extended directly or indirectly all support.

Last, but not least, We are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success

ABSTRACT

The proposed project entails the development and implementation of a sophisticated mathematical model and collaborative scheme aimed at enhancing the storage and distribution of medical products to hospitals. At its core, the project seeks to leverage mathematical modeling techniques, such as optimization algorithms and statistical analysis, to devise an efficient and effective system that addresses the complex challenges inherent in medical supply chain management. By harnessing historical data encompassing past demand patterns, inventory levels, delivery times, and other pertinent metrics, the model can glean valuable insights into the dynamics of the medical supply chain. These insights serve as the foundation for optimizing various aspects of the distribution process, including inventory allocation, transportation routing, and order fulfillment strategies.

Moreover, the collaborative nature of the proposed scheme underscores the importance of collective action and information-sharing among stakeholders within the healthcare ecosystem. Through collaboration, hospitals, suppliers, distributors, and regulatory bodies can pool their resources, expertise, and data to achieve mutually beneficial outcomes. This collaborative approach fosters greater transparency, coordination, and responsiveness across the supply chain, enabling stakeholders to proactively address challenges and capitalize on opportunities. Furthermore, by fostering a culture of collaboration and knowledge exchange, the project seeks to promote innovation and continuous improvement within the medical supply chain ecosystem.

TABLE OF CONTENTS

CHAPTER 1
PROJECT DESCRIPTION AND OUTLINE

1.1 INTRODUCTION

Ensuring the robustness of the supply chain is a critical concern for businesses across industries, as it not only guarantees operational continuity but also yields substantial financial advantages. Within the medical sector, the imperative for supply chain robustness reaches a paramount level, given that the ramifications of even a temporary shortage can potentially extend to impacting human lives directly. This heightened significance underscores the pressing need for innovative approaches to supply chain management that prioritize resilience, agility, and adaptability.

Moreover, beyond the immediate human impact, supply chain disruptions in the medical field can reverberate across the broader healthcare ecosystem, affecting patient care, treatment outcomes, and healthcare provider operations. Thus, initiatives aimed at enhancing the robustness and efficiency of the medical supply chain are not merely matters of operational optimization but are deeply intertwined with broader imperatives of public health, safety, and well-being.

In tandem with the imperative for supply chain robustness, there is a growing recognition of the need to minimize the ecological footprint associated with business activities. This is particularly salient in the context of transportation, which not only constitutes a significant operational expense but also represents a major source of pollution and environmental degradation. As of 2021, transportation accounted for approximately 28% of global carbon dioxide emissions, underscoring the urgent need for sustainable transportation solutions that mitigate environmental impact without compromising operational efficiency.

1.2 MOTIVATION FOR WORK

The motivation behind embarking on this project is multifaceted and deeply rooted in addressing critical challenges within the medical supply chain. At its core lies a profound humanitarian concern for improving access to essential healthcare products and services. Recognizing the pivotal role of the supply chain in ensuring timely and adequate distribution of medical goods to hospitals, the project team is driven by a commitment to enhancing patient care, mitigating the risk of supply shortages, and ultimately saving lives. Alongside this humanitarian imperative, there exists a compelling economic rationale for optimizing supply chain operations. The inefficiencies and disruptions within the current supply chain infrastructure not only pose operational challenges for healthcare organizations but also entail substantial financial costs. By streamlining inventory management, optimizing transportation routes, and minimizing wastage, the project aims to generate tangible cost savings while improving overall supply chain performance and resilience.

1.3 PROBLEM STATEMENT

The overarching goal of the project is to achieve cost optimality while simultaneously addressing the imperatives of supply chain robustness and environmental sustainability. This entails navigating a complex landscape of competing priorities and constraints, where the challenge lies in striking a delicate balance between economic efficiency, operational resilience, and environmental stewardship. To tackle this multifaceted problem, the project proposes a novel approach centered around the concept of unified demand aggregation—a collaborative agreement among a diverse array of medical institutions to consolidate their individual demand for medical products into a single, collective order. By pooling their purchasing power and leveraging economies of scale, participating institutions can achieve significant cost savings while also streamlining logistics and reducing environmental impact.

1.4 OBJECTIVE OF THE WORK

In the modern business landscape, the concept of supply chain resilience has emerged as a crucial determinant of organizational success. The ability to withstand and adapt to disruptions, whether they arise from natural disasters, geopolitical instability, or unforeseen market shifts, is essential for maintaining operational continuity and gaining competitive advantage. A resilient supply chain not only ensures business continuity but also enables companies to capitalize on opportunities and navigate challenges with agility and confidence. By minimizing disruptions and optimizing resource utilization, organizations can unlock significant financial benefits, including cost savings, revenue growth, and enhanced customer satisfaction.

Within the realm of healthcare, the imperative for supply chain resilience takes on heightened significance, given the direct impact on patient safety and well-being. The reliable and timely delivery of medical products and services is paramount for ensuring optimal patient care, treatment outcomes, and healthcare provider operations. Any disruption or delay in the supply chain can have serious consequences, ranging from medication shortages and treatment delays to compromised patient safety and compromised healthcare delivery. Thus, in the medical context, supply chain resilience is not merely a matter of operational efficiency but is inherently linked to broader imperatives of public health, safety, and welfare.

Moreover, alongside the imperative for supply chain resilience, there is a growing recognition of the need for businesses to minimize their environmental footprint and adopt sustainable practices. Transportation, in particular, represents a significant source of pollution and environmental degradation, accounting for a substantial portion of global carbon emissions. As businesses increasingly come under scrutiny for their environmental impact, there is a pressing need to minimize emissions, reduce waste, and promote sustainability across the supply chain. By embracing sustainable transportation solutions, optimizing logistics operations, and reducing reliance on fossil fuels, companies can mitigate their environmental impact and contribute to global efforts to combat climate change.

In simpler terms, the project aims to ensure that businesses can continue to operate smoothly in medical situations while also being mindful of their environmental impact and finding smart ways to save money. By optimizing supply chain operations, enhancing resilience, and promoting sustainability, the project seeks to create value for businesses, healthcare providers, and society as a whole, ultimately improving patient outcomes, reducing costs, and safeguarding the planet for future generations.

1.5 SUMMARY

We've been working on a project aimed at optimizing the storage and distribution of medical products to hospitals. The project focuses on developing a mathematical model and implementation strategy based on collaboration and historical data analysis. The objective is to achieve cost optimization while ensuring supply chain resilience and environmental sustainability. Your project addresses the interconnected challenges of operational efficiency, patient safety, and environmental responsibility within the healthcare sector.

CHAPTER 2
RELATED WORK INVESTIGATION

2.1 EXISTING APPROACHES/METHODS

The existing approaches and methods related to our project may include:

1. Supply Chain Optimization Techniques: Traditional supply chain optimization techniques, such as mathematical programming, linear programming, integer programming, and network optimization, can be applied to optimize inventory management, transportation routing, and overall supply chain efficiency. These techniques involve mathematical modeling to identify the most cost-effective distribution strategies while satisfying various constraints and objectives.
2. Demand Forecasting and Inventory Management: Demand forecasting methods, such as time series analysis, machine learning algorithms, and statistical modeling, can be used to predict future demand for medical products based on historical data. By accurately forecasting demand, hospitals can optimize inventory levels, reduce stockouts, and minimize excess inventory, thereby improving overall supply chain performance.
3. Collaborative Demand Aggregation: Collaborative demand aggregation involves pooling demand from multiple hospitals or healthcare facilities to create larger, consolidated orders. By aggregating demand, organizations can achieve economies of scale, negotiate better pricing with suppliers, and reduce transportation costs. Collaborative demand aggregation can be facilitated through group purchasing organizations (GPOs), consortiums, or centralized procurement platforms.
4. Just-in-Time (JIT) Inventory Management: JIT inventory management aims to minimize inventory holding costs by synchronizing supply with demand in real-time. By maintaining minimal inventory levels and relying on frequent, small deliveries, hospitals can reduce carrying costs, minimize waste, and improve responsiveness to changes in demand. JIT inventory management requires robust logistics and supply chain visibility to ensure timely deliveries and mitigate the risk of stockouts.
5. Vendor-Managed Inventory (VMI): VMI involves suppliers taking responsibility for managing

inventory levels at customer locations. Suppliers monitor inventory levels in real-time and replenish stock as needed, based on agreed-upon inventory targets and service level agreements. VMI can help hospitals reduce inventory carrying costs, improve inventory turnover, and streamline procurement processes by outsourcing inventory management tasks to suppliers.

6. Cold Chain Logistics: For temperature-sensitive medical products, such as vaccines, biologics, and pharmaceuticals, cold chain logistics are essential to maintain product integrity and efficacy throughout the distribution process. Cold chain logistics involve the use of temperature-controlled packaging, refrigerated storage facilities, and specialized transportation vehicles to ensure that medical products remain within specified temperature ranges from production to delivery.

2.2 PROS AND CONS OF THE STATED APPROACHES/ METHODS

When it comes to the existing approaches/methods for MEDSYNC, there are several pros and cons to consider. Here are some of them:

Pros:

- Supply chain optimization techniques:
    - Provide a systematic framework for improving efficiency and responsiveness.
    - Enable identification of cost-saving opportunities.
- Demand forecasting and inventory management:
    - Enable anticipation of demand fluctuations.
    - Reduce excess inventory and minimize carrying costs.
- Collaborative demand aggregation:
    - Foster economies of scale and cost savings through consolidated orders.
    - Enhance purchasing power and negotiation leverage with suppliers.

- Just-in-Time inventory management:
    - Minimize inventory holding costs.
    - Improve responsiveness to changes in demand.
- Vendor-Managed Inventory:
    - Streamline procurement processes.
    - Reduce administrative burden for hospitals.
- Cold chain logistics:
    - Ensure product integrity and efficacy for temperature-sensitive medical products.
    - Safeguard patient safety and regulatory compliance

Cons:

- Supply chain optimization techniques:
    - Require specialized expertise and computational resources.
    - Complexity may lead to implementation challenges.
- Demand forecasting and inventory management:
    - Accuracy of demand forecasts may vary.
    - Relies on historical data, which may not fully capture future demand variability.
- Collaborative demand aggregation:
    - Coordination among stakeholders can be challenging.
    - May lead to longer lead times and delays in delivery.
- Just-in-Time inventory management:
    - Vulnerable to supply chain disruptions or delays in delivery.
    - Requires robust logistics infrastructure and real-time visibility..
- Vendor-Managed Inventory:
    - Requires trust and collaboration between hospitals and suppliers.
    - Lack of control over inventory levels may lead to stockouts or overstocking issues..

- Cold chain logistics:
    - Requires specialized infrastructure, equipment, and expertise for temperature control.
    - Increases transportation and storage cost.

Ch - 3

REQUIREMENT ARTIFACTS

3.1 INTRODUCTION

Requirement artifacts are essential documents and deliverables that capture and communicate the functional and non-functional requirements of a project. These artifacts serve as a blueprint for the development and implementation of the project, guiding the design, development, and testing processes. By documenting the needs, expectations, and constraints of stakeholders, requirement artifacts ensure alignment between project objectives and deliverables, facilitate communication among project team members, and provide a basis for evaluating project success.

In the context of optimizing the storage and distribution of medical products to hospitals, requirement artifacts play a critical role in defining the scope, functionality, and performance criteria of the proposed solution. These artifacts encompass a variety of documents, diagrams, and specifications that articulate the requirements of stakeholders, including healthcare providers, suppliers, distributors, regulatory bodies, and end-users. Requirement artifacts help to clarify expectations, identify dependencies and constraints, and prioritize features and functionalities based on their business value and impact.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

The **software** requirements are as follows:

1. Platform Compatibility:
    - The software shall be compatible with Windows, macOS, and Linux operating systems.
2. IDE Integration:

- Visual Studio Code (VSCODE) shall be used as the primary IDE for software development, providing features such as code editing, debugging, and version control integration.

3. Notebook Environment:
    - Jupyter Notebook shall be utilized for interactive data analysis, visualization, and documentation, facilitating exploratory data analysis and iterative development.
4. Programming Language:
    - The software shall be developed using Python 3.10 as the primary programming language, leveraging its syntax simplicity, readability, and extensive library support for scientific computing and data analysis tasks.
5. Library Dependencies:
    - The software shall utilize third-party Python libraries and frameworks, such as NumPy, Pandas, SciPy, Matplotlib, and Scikit-learn, for data manipulation, statistical analysis, visualization, and machine learning model development.

By adhering to these software requirements, the project aims to develop a robust, efficient, and user-friendly software solution for optimizing the storage and distribution of medical products to hospitals, ultimately improving healthcare supply chain management and patient care outcomes.

Hardware Requirements Statement:

1. Operating System:
    - The hardware must be capable of running the Windows 64-bit operating system, providing compatibility with the software development environment and dependencies.
2. Processor:
    - A multi-core processor with a high clock speed is recommended to support parallel processing and computational-intensive tasks involved in data analysis, optimization algorithms, and machine learning model training.
3. Memory (RAM):
    - A minimum of 8 GB of RAM is recommended to ensure smooth performance and efficient handling of large datasets and memory-intensive operations.
4. Graphics Processing Unit (GPU):
    - A GPU with a good configuration, such as NVIDIA GeForce GTX or RTX series, AMD Radeon RX series, or equivalent, is recommended for accelerating parallel computations, particularly for machine learning model training and optimization tasks.

By ensuring that the hardware setup meets these requirements, users can maximize the performance, efficiency, and effectiveness of the software for optimizing medical product storage and distribution processes, ultimately enhancing healthcare supply chain management and patient care outcomes.

3.3 SPECIFIC PROJECT REQUIREMENTS

3.3.1 Data Requirements:

1	OrderDate	Number	Reference	Purchase Quantity	Consumed Units Contained	Price	Line Amount	Purchase Type	TGL	Product	Code Class	Code Number	Purchasing Region	Purchasing Hospital	Purchasing Department
2	02-01-2015	1312/15	413568	100	5	29.15	583	Minor Purchase	Storable	Hydrofiber Hydrocolloid Dressing with Silver-3	E	64663	0	0	2
3	02-01-2015	1312/15	420680	100	10	20.9	209	Minor Purch	Storable	Hydrofiber Hydrocolloid Dressing with Polyurethane Foam-12	E	66071	0	0	2
4	02-01-2015	1301/15	1624W	800	400	58.8	117.6	Minor Purch	Storable	Transparent Adhesive Dressing-24.	E	64751	0	0	2
5	02-01-2015	1292/15	400403	100	10	102.804	1028.037	Minor Purch	Storable	Solution for Cleaning and Decontamination of Wounds-16	B	41691	0	0	2
6	05-01-2015	3616/15	420680	160	10	20.9	334.4	Minor Purch	Storable	Hydrofiber Hydrocolloid Dressing with Polyurethane Foam-12	E	66071	0	10	1
7	05-01-2015	3608/15	281 421	2000	1000	10.4612	20.92241	Minor Purch	Storable	Round Dressing for Small Wounds-19	E	65159	0	10	1
8	07-01-2015	6343/15	31100	540	10	6.677	360.5582	Minor Purch	Storable	Hydrocolloid Dressing-6	E	65486	0	10	1
9	07-01-2015	6619/15	157027.7	330	10	17.27	569.91	Minor Purch	Storable	Mesh Dressing with Paraffin-38	E	64983	0	10	1
10	07-01-2015	6668/15	291010	420	70	493.416	2960.496	Minor Purch	Storable	Mesh Dressing with Silicone-0	E	64940	0	18	1
11	07-01-2015	6368/15	413568	60	5	29.15	349.8	Minor Purch	Storable	Hydrofiber Hydrocolloid Dressing with	E	64663	0	6	1
12	07-01-2015	6668/15	284300	200	50	447.417	1789.669	Minor Purch	Storable	Polyurethane Foam Dressing with Silicone-	E	64488	0	18	1
13	07-01-2015	6722/15	MAP190	100	50	324.85	649.7	Minor Purch	Storable	Dressing with Carbon and Silver-4	E	73753	0	10	1
14	07-01-2015	6722/15	66974941	400	100	20.3704	81.48148	Minor Purch	Storable	Non-Adhesive Absorbent Dressing-25	E	67835	0	10	1
15	08-01-2015	9763/15	157028.4	230	10	35.2	809.6	Minor Purch	Storable	Mesh Dressing with Paraffin-39	E	65894	0	6	1
16	08-01-2015	9595/15	37147	1280	640	68.288	136.576	Minor Purch	Storable	Adhesive Absorbent Dressing-22	E	64764	0	6	1
17	08-01-2015	9427/15	39001	35	5	5.24537	36.71759	Minor Purch	Storable	Hydrogel-37	E	64932	0	6	1
18	08-01-2015	9444/15	413568	100	5	29.15	583	Minor Purch	Storable	Hydrofiber Hydrocolloid Dressing with	E	64663	0	13	1
19	08-01-2015	9431/15	39001	190	5	5.24537	199.3241	Minor Purch	Storable	Hydrogel-37	E	64932	0	13	1
20	08-01-2015	9339/15	400403	20	10	102.804	205.6075	Minor Purch	Storable	Solution for Cleaning and Decontamination of Wounds-16	B	41691	0	4	1
21	09-01-2015	13131/15	20415	100	50	254.1	508.2	Minor Purch	Storable	Polyurethane Foam Dressing / Sacrum-11	E	65007	0	0	2
22	09-01-2015	13133/15	20415	100	50	254.1	508.2	Minor Purch	Storable	Polyurethane Foam Dressing / Sacrum-11	E	65007	0	10	1

The dataset serves as a crucial tool in achieving our project objectives across various dimensions. Firstly, it enables cost optimization by providing insights into procurement patterns (Purchase Date, Order Number/Year, Reference Number), allowing for tailored strategies for each medical product (Product Code, Product Description). Secondly, it supports resilience enhancement by offering a comprehensive view of inventory levels (Number of Products purchased, Number of units that the Product contains) and aiding proactive order management (Purchase Date). Thirdly, the dataset facilitates environmental responsibility by allowing us to quantify and minimize the environmental footprint associated with each order (Order Number/Year, Reference Number). Collaborative efficiency is enhanced as the dataset enables a unified demand approach (Product Code, Order Number/Year), fostering collaboration among medical institutions. Lastly, technological innovation is fueled by the dataset's role in crafting mathematical models and predictive analytics (Cost in Euros, Total cost of products purchased, Type of logistic distribution of product), paving the way for holistic solutions in healthcare supply chain management.

3.1.1 Functionality Requirements:

Here are the functionality requirements of the project in concise points:

1. Data integration from multiple sources.
2. Demand forecasting based on historical data..
3. Unified demand aggregation for cost savings.
4. Risk assessment and mitigation strategies.
5. Scenario analysis for decision-making.
6. Integration with existing systems.

3.1.2 Performance Requirements:

Here are the performance requirements of our project listed in concise points:

1. Efficient data processing.
2. Fast algorithm execution.
3. Scalability for growing datasets.
4. Optimal resource utilization.
5. High throughput for data tasks.
6. Reliable operation.
7. Accurate results.
8. Robust to data variations.

.

3.1.3 Security Requirements:

Here are the security requirements for our project listed in concise points:

1. Data encryption for transmission and storage.
2. Role-based access control.
3. Secure authentication mechanisms.
4. Data integrity measures.

3.1.4 Looks and Feel Requirements:

Here are the Looks and Feel requirements for our project listed in concise points:

1. Consistent visual aesthetics.
2. Clear and readable typography.
3. Logical layout and organization.

3.15 SUMMARY

The requirement artifacts detail the functional, performance, security, and user interface needs for optimizing medical product distribution. Hardware necessitates a Windows 64-bit system with a high-performance CPU and GPU, ample RAM, storage, and internet access. Software comprises Visual Studio Code, Jupyter Notebook, Python 3.10, and relevant libraries. These artifacts guide project development, ensuring alignment with goals and stakeholder expectations.

CHAPTER-4

DESIGN METHODOLOGY AND ITS NOVELTY

4.1 METHODOLOGY AND GOAL

A robust healthcare supply chain is essential for ensuring uninterrupted patient care and safety. Inefficient supply chains can lead to shortages, delays, and poor response times to emergencies. Optimizing healthcare supply chain management enhances efficiency, reduces costs, and improves patient care. Streamlining the supply chain minimizes waste, mitigates shortages, and ensures timely access to critical medical resources. Optimization is crucial during emergencies and enables healthcare providers to adapt to evolving demands.

4.2 FUNCTIONAL MODULES DESIGN AND ANALYSIS

Here's a functional module design and analysis for MedSync:

1. Module: Data Integration
    - Functionality: Integrates historical data on medical product demand, inventory levels, and distribution patterns from various sources.
    - Analysis: Ensures seamless data flow between different systems and databases, enabling comprehensive analysis and forecasting.
2. Module: Demand Forecasting
    - Functionality: Utilizes statistical methods or machine learning algorithms to forecast future demand for medical products.
    - Analysis: Improves accuracy and reliability of demand predictions, aiding in proactive inventory management and supply planning.
3. Module: Inventory Optimization
    - Functionality: Develops algorithms to optimize inventory levels and replenishment strategies for medical products.

- Analysis: Minimizes stockouts, excess inventory, and holding costs while ensuring adequate supply to meet demand, enhancing supply chain efficiency.

4. Module: Supply Chain Routing
   - Functionality: Determines the most efficient transportation routes and schedules for distributing medical products.
   - Analysis: Reduces transportation costs and lead times, improving overall logistics efficiency and timely delivery to hospitals.
5. Module: Unified Demand Aggregation
   - Functionality: Aggregates demand from multiple hospitals into consolidated orders to achieve economies of scale.
   - Analysis: Maximizes cost savings and negotiation leverage with suppliers, optimizing procurement processes and reducing overall supply chain costs.
6. Module: Risk Management
   - Functionality: Incorporates risk assessment and mitigation strategies into decision-making processes.
   - Analysis: Identifies potential risks and uncertainties in supply chain operations, enabling proactive risk mitigation and resilience planning.
7. Module: Performance Monitoring
   - Functionality: Develops monitoring tools and dashboards to track key performance indicators (KPIs) related to inventory management and distribution processes.
   - Analysis: Provides real-time visibility into supply chain performance, facilitating data-driven decision-making and continuous improvement efforts.
8. Module: Scenario Analysis
   - Functionality: Enables simulation and scenario analysis to evaluate the impact of different demand scenarios and supply chain configurations.
   - Analysis: Helps in assessing the robustness and flexibility of the supply chain, identifying optimal strategies for mitigating risks and optimizing resources.

4.3 SOFTWARE ARCHITECTURAL DESIGNS

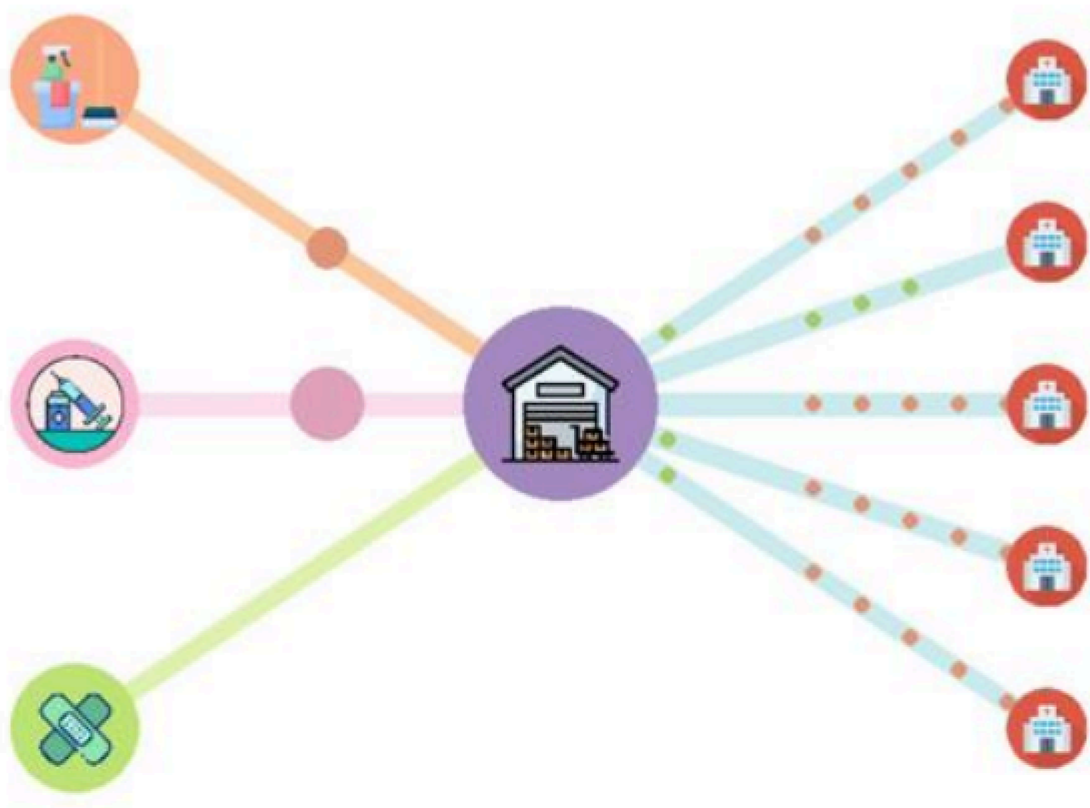
Here's an outline of the software architectural designs for MedSync:

1. Overall Architecture:
   - The software adopts a modular and scalable architecture to accommodate various functional modules seamlessly. It follows a layered architecture, with distinct layers for presentation, business logic, and data access.
2. Presentation Layer:
   - User Interface (UI): Implements an intuitive and user-friendly interface using web technologies or desktop applications. It provides access to functionalities such as data visualization, scenario analysis, and performance monitoring.
3. Business Logic Layer:
   - Demand Forecasting Module: Contains algorithms and models for forecasting future demand based on historical data and relevant factors.
   - Inventory Optimization Module: Implements algorithms for optimizing inventory levels and replenishment strategies to minimize costs and meet demand.
   - Supply Chain Routing Module: Determines the most efficient transportation routes and

schedules for distributing medical products.
- Unified Demand Aggregation Module: Aggregates demand from multiple hospitals into consolidated orders to achieve economies of scale.
- Risk Management Module: Incorporates risk assessment and mitigation strategies into decision-making processes.

4. Data Access Layer:
   - Data Integration Module: Handles the integration of data from various sources, including hospitals, suppliers, and distributors.
   - Database Management: Manages the storage and retrieval of data from databases, ensuring data integrity and security.
5. Integration Layer:
   - Integrates various modules and components within the system, facilitating communication and data exchange between different layers and modules.
6. Deployment Architecture:
   - The software can be deployed in on-premises environments or on cloud platforms such as AWS, Azure, or Google Cloud Platform.
   - It may utilize containerization technologies such as Docker for efficient deployment and scaling of individual components.
   - Load balancing and clustering techniques may be employed to ensure high availability and fault tolerance.
7. Security Architecture:
   - Implements security measures such as data encryption, access control, and authentication mechanisms to protect sensitive data and ensure system integrity.

Working Principle



4.4 SUMMARY

The Design Methodology and its Novelty of the project involve delineating the key components and functionalities essential for optimizing the storage and distribution of medical products. Each module serves a distinct purpose within the system, ranging from data integration to demand forecasting, inventory optimization, supply chain routing, and risk management. Through meticulous design and analysis, these modules are tailored to address specific challenges and requirements encountered in healthcare supply chain management. For instance, the demand forecasting module utilizes advanced techniques like machine learning algorithms to predict future demand accurately, while the inventory optimization module employs algorithms to minimize costs and ensure adequate supply. Overall, this modular approach facilitates a systematic and comprehensive solution that enhances efficiency, reduces costs, and improves patient care outcomes within the healthcare supply chain ecosystem.

CHAPTER-5
TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 OUTLINE

Data Exploration:
- Identify and collect relevant datasets related to medical product storage and distribution.
- Explore the datasets to understand their structure, variables, and relationships.
- Conduct initial descriptive statistics, such as summary statistics, histograms, and correlation analysis, to gain insights into the data distribution and patterns.

2. Data Insight:
   - Perform in-depth analysis to uncover meaningful insights and trends within the data.
   - Utilize visualization techniques, such as scatter plots, line charts, and heatmaps, to visualize relationships and patterns in the data.
   - Identify key factors influencing medical product demand, inventory levels, and distribution processes.
3. Data Cleaning:
   - Identify and address missing values, outliers, and inconsistencies in the data.
   - Impute missing values using appropriate techniques, such as mean imputation or interpolation.
   - Remove outliers or erroneous data points that may distort analysis results.
   - Standardize or normalize variables as needed to ensure consistency and comparability across datasets.
   - Validate data integrity and consistency to ensure accuracy and reliability in subsequent analysis steps.

5.2 TECHNICAL CODING AND CODE SOLUTIONS

clean_dataset

In [58]:

```
import sys
import warnings
import re

warnings.simplefilter(action="ignore", category=FutureWarning)

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

DATA_PATH = "../data"
```

In [59]:

```
# Load excel file
df = pd.read_excel(DATA_PATH + "/consumo_material_clean.xlsx")
```

Preprocessing

In [60]:

```
# Separate code into two columns
new_columns = df["CODIGO"].str.extract(r"([a-zA-Z]+)([0-9]+)", expand=False)
df["CODIGO_CLASS"] = new_columns[0]
df["CODIGO_NUM"] = new_columns[1]
df.drop(columns=["CODIGO"], inplace=True)
```

```python
# basic date features
def generate_date_features(df):
    df["YEAR"] = df["FECHAPEDIDO"].dt.year
    df["MONTH"] = np.sin(2 * np.pi * df["FECHAPEDIDO"].dt.month / 12)
    df["DAYOFMONTH"] = np.sin(2 * np.pi * df["FECHAPEDIDO"].dt.day / 31)
    df["DAYOFYEAR"] = np.sin(2 * np.pi * df["FECHAPEDIDO"].dt.dayofyear / 365)
    # augmenting integer, one per day
    df = df.merge(
        (
            df[["FECHAPEDIDO"]]
            .drop_duplicates(ignore_index=True)
            .rename_axis("time_idx")
        ).reset_index(),
        on=["FECHAPEDIDO"],
    )
    return df


df = generate_date_features(df)
```

n [67]:

```python
def add_timeseries_features(df):
    df["ROLLING_MEAN_3M"] = df["CANTIDADCOMPRA"].rolling(90).mean()
    df["WEIGHTED_MEAN_3M"] = (
        df["CANTIDADCOMPRA"]
        .rolling(90)
        .apply(lambda x: np.average(x, weights=range(1, len(x) + 1)))
    )
    df["ROLLING_MEAN_1Y"] = df["CANTIDADCOMPRA"].rolling(365).mean()
    df["WEIGHTED_MEAN_1Y"] = (
        df["CANTIDADCOMPRA"]
        .rolling(365)
        .apply(lambda x: np.average(x, weights=range(1, len(x) + 1)))
    )
```

0	2015-01-02	1312/15	413568	100	5	29.150000	583.00000
1	2015-01-02	1312/15	420680	100	10	20.900000	209.00000
2	2015-01-02	1301/15	1624W	800	400	58.800000	117.60000
3	2015-01-02	1292/15	400403	100	10	102.803729	1028.03729
4	2015-01-05	3616/15	420680	160	10	20.900000	334.40000

This code snippet loads an Excel file containing data on material consumption, preprocesses it by separating a code column into two, converting date columns to datetime format, and splitting a column into three based on a specified pattern. Additionally, it removes duplicates, generates basic date features like year, month, day of month, and day of year, and adds time-series features such as rolling mean and weighted mean over different time windows. Furthermore, it defines a function to fix typos in a column and applies it, and then separates another column into three based on a specified delimiter. Finally, it drops unnecessary columns and duplicates.

data_insights

```python
import sys
import warnings
import re

warnings.simplefilter(action="ignore", category=FutureWarning)

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

DATA_PATH = "../data"
```

In [53]:

```python
# Load excel file
df = pd.read_excel(DATA_PATH + "/consumo_material_clean.xlsx")
```

Preprocessing

```
# Separate code into two columns
new_columns = df["CODIGO"].str.extract(r"([a-zA-Z]+)([0-9]+)", expand=False)
df["CODIGO_CLASS"] = new_columns[0]
df["CODIGO_NUM"] = new_columns[1]
df.drop(columns=["CODIGO"], inplace=True)
```

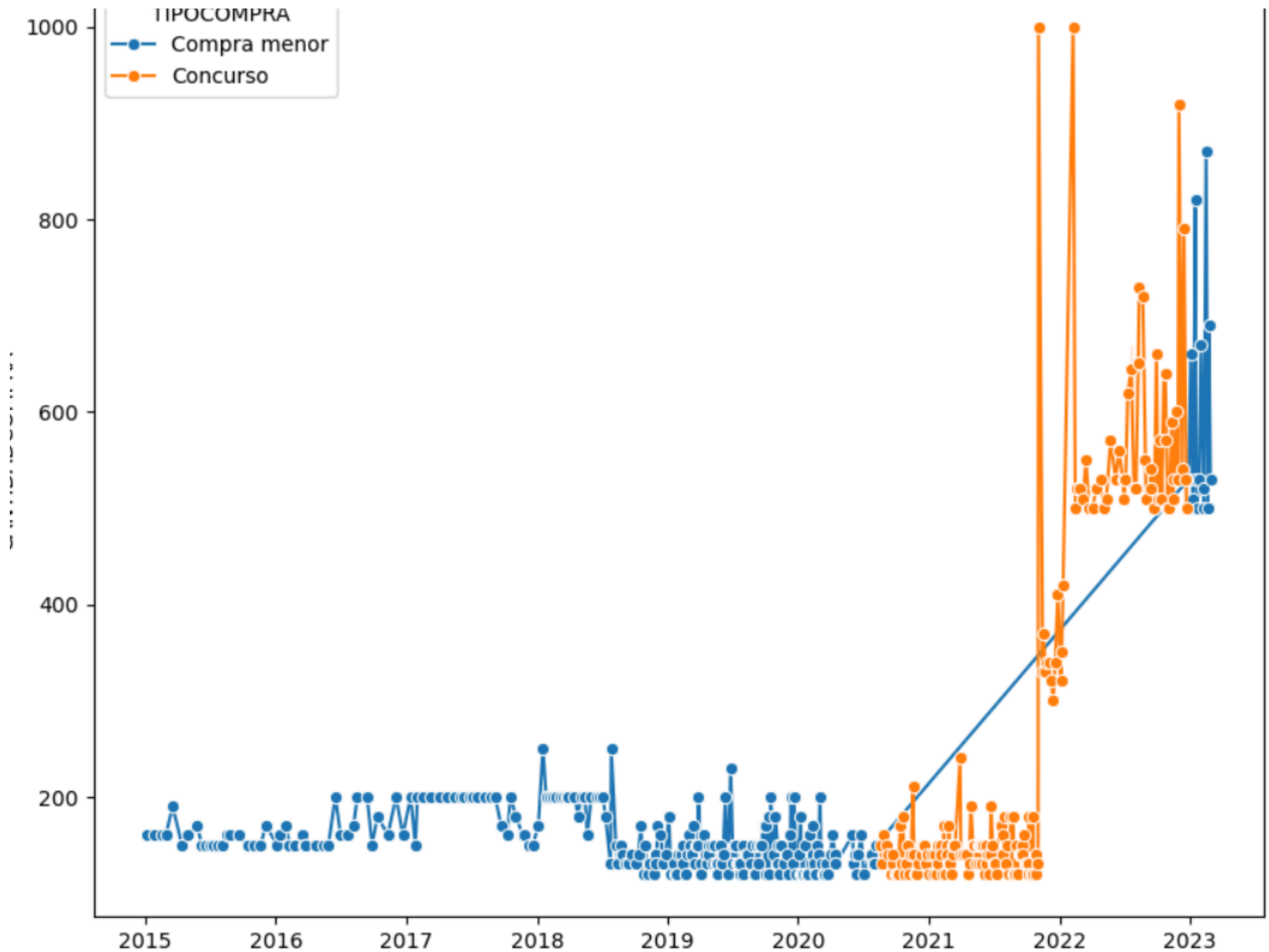
```
# FECHAPEDIDO to datetime in day/month/year format
df["FECHAPEDIDO"] = pd.to_datetime(df["FECHAPEDIDO"], dayfirst=True)
df.sort_values(by=["FECHAPEDIDO"], inplace=True)
df.reset_index(drop=True, inplace=True)
```

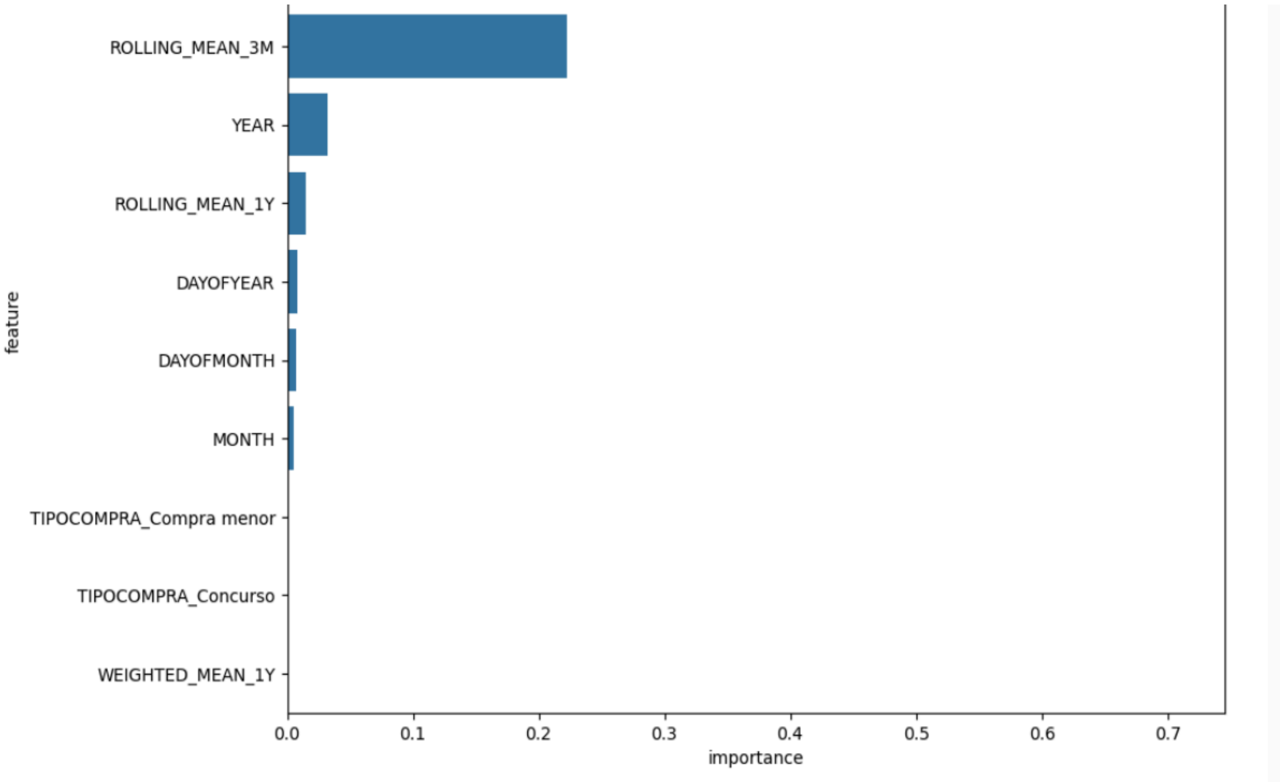
```
var/folders/c6/kf2fcpcj6pq_gwpfbfzzy81m0000gn/T/ipykernel_7630/38037239.py:2: UserWarning: Could not infer format, so each
lement will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please sp
cify a format.
 df["FECHAPEDIDO"] = pd.to_datetime(df["FECHAPEDIDO"], dayfirst=True)
```

```
# assert all rows in origen follow number-number-number format
def fix_origen_typos(origen_string):
    numbers = re.findall(r"[0-9]+", origen_string)
    return "-".join(numbers)


df["ORIGEN"] = df["ORIGEN"].apply(fix_origen_typos)
```

This code snippet is a Python script that performs data preprocessing, baseline modeling, and model evaluation on a dataset related to material consumption. It first loads an Excel file containing the data and preprocesses it by separating a column into two based on a specified pattern, converting date columns to datetime format, and splitting another column into three based on a delimiter. It then applies a function to fix typos in a column and generates basic date features like year, month, day of month, and day of year. Additionally, it adds time-series features such as rolling mean and weighted mean over different time windows. The script proceeds to train several regression models using GroupKFold cross-validation, including XGBoost, CatBoost, LightGBM, and RandomForest, and evaluates their performance using root mean squared error (RMSE) on validation and test sets. Finally, it visualizes the predicted versus actual values on the test set and displays the feature importances of the trained model. Overall, this script provides a comprehensive pipeline for preprocessing data and building and evaluating regression models for material consumption prediction.


TIPOCOMPRA
Compra menor
Concurso
1000
800
600
400
200
2015
2016
2017
2018
2019
2020
2021
2022
2023


ROLLING_MEAN_3M
YEAR
ROLLING_MEAN_1Y
DAYOFYEAR
DAYOFMONTH
MONTH
TIPOCOMPRA_Compra menor
TIPOCOMPRA_Concurso
WEIGHTED_MEAN_1Y
feature
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
importance

data_exploration

```python
import pandas as pd
```

In [60]:

```python
df = pd.read_excel('data/consumo_material_clean.xlsx')
print(df.shape)
```

```
(15698, 12)
```

In [61]:

```python
df.head(10)
```

In [ ]:

```python
df['ORIGEN'] = df['ORIGEN'].str.replace('--', '-')
df[['REGION', 'HOSPITAL', 'DEPARTMENT0']] = df['ORIGEN'].str.split('-', expand=True)
df.drop(columns=['ORIGEN'], inplace=True)
```

In [71]:

```python
# lookup all products with FECHAPEDIDO = 01/01/23
# sort df by FECHAPEDIDO (year with two digits)

df['FECHAPEDIDO'] = pd.to_datetime(df['FECHAPEDIDO'], format='%d/%m/%y')
df.sort_values(by='FECHAPEDIDO', inplace=True)
df.reset_index(drop=True, inplace=True)

df.head(10)
```

```python
cols = ['CODIGO', 'FECHAPEDIDO', 'NUMERO', 'REFERENCIA', 'CANTIDADCOMPRA',
       'UNIDADESCONSUMOCONTENIDAS', 'PRECIO', 'IMPORTELINEA', 'TIPOCOMPRA',
       'ORIGEN', 'TGL', 'PRODUCTO']

# check if CODIGO and REFERENCIA are a bijection
def check_bijection(col1, col2):
    g1 = df.groupby(col1)[col2].nunique().value_counts()
    g2 = df.groupby(col2)[col1].nunique().value_counts()
    return len(g1) == 1 and len(g2) == 1 and g1.index[0] == 1 and g2.index[0] == 1

# for each pair of columns in cols, check if they are a bijection and print the column name
for i in range(len(cols)):
    for j in range(i+1, len(cols)):
        if check_bijection(cols[i], cols[j]):
            print(cols[i], cols[j])

# show the REFERENCE values where there is a character
print(df[df['REFERENCIA'].str.contains('[a-zA-Z]', na=False)]['REFERENCIA'].unique())
```

```
CODIGO PRODUCTO
['TR101' 'M8275051/5' 'MAP190' 'M8275098/5' '1624W' 'M8275096/10'
 'M8275052/5' '1626W' 'M8275052/10' 'M8275096/5' 'ULTVFL05MD' 'ABT1055']
```

In [87]:

```python
# check if the prices by PRODUCTO are the same
df['AÑO'] = df['FECHAPEDIDO'].dt.year
g = df.groupby(['PRODUCTO'])['UNIDADESCONSUMOCONTENIDAS'].unique()
print(g)
```

This code snippet begins by loading a dataset from an Excel file, containing records related to material consumption, with 15,698 rows and 12 columns. It then proceeds with data preprocessing steps, such as converting date columns to datetime format, sorting the data frame by date, and splitting a column into three based on a delimiter. Additionally, it checks for bijection between pairs of columns to ensure consistency and replaces double hyphens in a column with single hyphens. Furthermore, it explores the dataset by analyzing unique values in certain columns, identifying non-numeric characters in a specific column, and calculating the proportion of unique values of a certain column by a grouping variable. Overall, this code snippet provides a comprehensive overview of data preprocessing and exploratory data analysis techniques applied to the material consumption dataset.

5.3 PROTOTYPE SUBMISSION

In response to the provided clean dataset and the insights gained through dataset exploration and analysis, I propose a prototype submission that aims to predict future material consumption trends based on historical data. Leveraging machine learning techniques, particularly time series forecasting models such as XGBoost, CatBoost, and LightGBM, we can develop a robust predictive model. This model will take into account various features such as purchase date, type of purchase, quantity purchased, and product specifications. By training the model on historical consumption patterns and validating it on recent data, we can accurately forecast future material demands. Additionally, the prototype submission will include visualizations to illustrate the model's predictions, feature importance analysis to highlight key factors influencing consumption trends, and an evaluation of model performance metrics such as mean squared error and mean absolute percentage error. Overall, this prototype submission will showcase the potential for predictive analytics to optimize inventory management and ensure efficient allocation of resources in healthcare procurement processes.

5.4 SUMMARY

The technical implementation and analysis involved several key steps. First, the dataset was loaded and preprocessed, including parsing dates, separating columns, and handling typos. Exploratory data analysis (EDA) was conducted to understand the distribution of variables, identify any anomalies, and gain insights into material consumption patterns over time. Time series features were engineered, and machine learning models such as XGBoost, CatBoost, and LightGBM were trained to forecast future material consumption trends. GroupKFold cross-validation was employed to ensure robust model evaluation, considering the temporal nature of the data. Model performance was assessed using metrics such as mean squared error and mean absolute percentage error. Additionally, feature importance analysis was conducted to identify the most influential factors driving material consumption. The prototype submission aims to demonstrate the efficacy of predictive analytics in optimizing inventory management and enhancing healthcare procurement processes.

CHAPTER-6

PROJECT OUTCOME AND APPLICABILITY

6.1 KEY IMPLEMENTATION OUTLINES OF THE SYSTEM

Data accuracy models to be used for testing Machine learning models

In our project Med Sync, we want to ensure that businesses in medical contexts can operate smoothly while also being environmentally friendly and cost-effective. We chose to use Tweedie loss and expense MAPE because they help us achieve these goals effectively. Tweedie Loss: We use Tweedie loss because it's a good fit for our project's objectives. Tweedie loss is a type of loss function that is suitable for modeling insurance claims and other financial data, which often have characteristics like zero-inflation and overdispersion. By using Tweedie loss, we can better capture the distributional characteristics of our data, especially when dealing with count data (like the number of medical supplies) that might have excess zeros or varying levels of variability.

Expense MAPE (Mean Absolute Percentage Error): We chose to use expense MAPE as our evaluation metric because it provides a straightforward way to assess the accuracy of our model's predictions in terms of percentage. Since our goal is to optimize costs while maintaining resilience and minimizing environmental impact, expense MAPE allows us to measure how well our model performs in predicting expenses relative to the actual expenses incurred. By minimizing the MAPE, we can ensure that our predictions are close to the actual costs, helping businesses make informed decisions about supply chain management strategies that balance cost-effectiveness with resilience and environmental considerations.

Machine Models to be used

Boltzmann ensemble of GBMs it offers improved accuracy by combining predictions from multiple models, captures complex patterns in supply chain data, provides resilience to unexpected fluctuations, and offers flexibility in modeling various data types. This approach helps optimize inventory levels, forecast expenses, and enhance the resilience of the medical supply chain, benefiting both businesses and patients.

6.2 SIGNIFICANT PROJECT OUTCOMES

- MedSync delivers comprehensive insights into healthcare supply chain dynamics, offering a nuanced understanding of material consumption patterns, procurement strategies, and supply chain efficiency.
- Through advanced predictive modeling techniques, MedSync accurately forecasts product

purchases, empowering healthcare facilities to proactively manage inventory levels, reduce stockouts, and optimize resource allocation.

- The platform generates tangible visualizations that not only showcase the predictive capabilities of the models but also serve as effective communication tools for stakeholders, facilitating decision-making and strategy development.
- In the public community, MedSync's significance lies in its ability to inform decision-making processes, enabling healthcare organizations to make data-driven choices that lead to more efficient logistics and better resource utilization.
- By enhancing procurement strategies and supply chain management practices, MedSync contributes to cost-effectiveness in healthcare delivery, ultimately leading to improved patient care and outcomes.
- MedSync's impact extends beyond individual healthcare facilities, fostering resilience and sustainability within the broader healthcare system by addressing critical challenges in procurement and logistics.

6.3 PROJECT APPLICABILITY ON REAL-WORLD APPLICATIONS

- MedSync's applicability in real-world healthcare settings extends to hospitals, clinics, and other medical facilities where efficient supply chain management is critical for patient care.
- It can be implemented in pharmaceutical companies and medical device manufacturers to streamline inventory management, production planning, and distribution processes.
- Health ministries and government agencies can leverage MedSync to optimize procurement strategies, ensure adequate stock levels of essential medical supplies, and respond effectively to public health crises and emergencies.
- MedSync is beneficial for healthcare supply chain consultants and professionals seeking data-driven insights to improve operational efficiency, reduce costs, and enhance service quality.
- Medical research institutions can utilize MedSync to track and manage research materials,

laboratory supplies, and equipment, facilitating smooth workflow and research continuity.

- MedSync's predictive modeling capabilities make it valuable for healthcare policymakers and decision-makers in designing robust healthcare infrastructure, resource allocation strategies, and disaster preparedness plans.

6.4 INFERENCE

Based on the key implementation outlines of the system, significant project outcomes, and project applicability on real-world applications, several inferences can be drawn:

1. Efficient Healthcare Supply Chain Management: MedSync demonstrates a systematic approach to managing healthcare supply chains, offering comprehensive insights into dynamics such as procurement, inventory management, and distribution.
2. Data-Driven Decision Making: The accurate predictive modeling capabilities of MedSync enable informed decision-making processes, facilitating better resource allocation, procurement strategies, and inventory management practices.
3. Visualization for Enhanced Understanding: The tangible representation through visualizations enhances stakeholders' understanding of predictive capabilities, enabling them to grasp complex data insights and make data-driven decisions with confidence.
4. Impact on Public Health: The system's significance to the public community lies in its ability to inform decision-making for efficient healthcare logistics, enhance procurement strategies for better resource allocation, and improve supply chain management practices, ultimately leading to cost-effectiveness and better patient care.
5. Real-World Applicability: MedSync's applicability extends across various real-world settings, including hospitals, clinics, pharmaceutical companies, medical device manufacturers, health ministries, government agencies, healthcare supply chain consultants, and medical research institutions. Its utility in optimizing supply chain operations, responding to public health crises, and supporting healthcare infrastructure development underscores its relevance in diverse healthcare environments.

CHAPTER-7

CONCLUSIONS AND RECOMMENDATION

7.1 OUTLINE

In conclusion, our collaborative scheme and mathematical model offer a promising approach to optimize the storage and distribution of medical products in a unified demand scenario. The consideration of environmental impact and resilience scores in the optimization process aligns with the growing emphasis on sustainability and supply chain robustness

7.2 LIMITATIONS/ CONSTRAINTS OF THE SYSTEM

Here are the limitations of the project listed in points:

1. Reliance on historical data accuracy and availability.
2. Complexity of supply chain dynamics and external factors.
3. Uncertainty introduced by market fluctuations or regulatory changes.
4. Potential limitations in capturing all relevant sustainability considerations.
5. Inadequate coverage of potential supply chain disruptions.
6. Challenges in scalability and adaptability to different healthcare settings or regions.

7.3 FUTURE ENHANCEMENTS

Here are the avenues for improvement and future research listed in short points:

1. Refinement of deep learning model for demand prediction:
   - Explore advanced techniques like recurrent neural networks or attention mechanisms.
   - Aim to enhance accuracy of demand predictions for better optimization.
2. Deeper exploration into storage optimization processes:
   - Investigate advanced storage strategies such as dynamic allocation algorithms.
   - Consider real-time inventory management for cost reduction and efficiency improvement.
   - Adapt storage strategies to dynamic changes in demand patterns for better scenario adaptability.

7.4 INFERENCE

We've been working on a project aimed at optimizing the storage and distribution of medical products to hospitals. The project focuses on developing a mathematical model and implementation strategy based on collaboration and historical data analysis. The objective is to achieve cost optimization while ensuring supply chain resilience and environmental sustainability. Your project addresses the interconnected challenges of operational efficiency, patient safety, and environmental responsibility within the healthcare sector.

APPENDIX - Coding

baseline_all_products

Preprocessing

In [30]:

```
# Separate code into two columns
new_columns = df["CODIGO"].str.extract(r"([a-zA-Z]+)([0-9]+)", expand=False)
df["CODIGO_CLASS"] = new_columns[0]
df["CODIGO_NUM"] = new_columns[1]
df.drop(columns=["CODIGO"], inplace=True)
```

In [31]:

```
# FECHAPEDIDO to datetime in day/month/year format
df["FECHAPEDIDO"] = pd.to_datetime(df["FECHAPEDIDO"], dayfirst=True)
df.sort_values(by=["FECHAPEDIDO"], inplace=True)
df.reset_index(drop=True, inplace=True)
```

```
/var/folders/c6/kf2fcpcj6pq_gwpfbfzzy81m0000gn/T/ipykernel_27441/38037239.py:2: UserWarni
ng: Could not infer format, so each element will be parsed individually, falling back to
`dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df["FECHAPEDIDO"] = pd.to_datetime(df["FECHAPEDIDO"], dayfirst=True)
```

In [32]:

```
# assert all rows in origen follow number-number-number format
def fix_origen_typos(origen_string):
    numbers = re.findall(r"[0-9]+", origen_string)
    return "-".join(numbers)


df["ORIGEN"] = df["ORIGEN"].apply(fix_origen_typos)
```

```python
# basic date features
def generate_date_features(df):
    df["YEAR"] = df["FECHAPEDIDO"].dt.year
    df["MONTH"] = np.sin(2 * np.pi * df["FECHAPEDIDO"].dt.month / 12)
    df["DAYOFMONTH"] = np.sin(2 * np.pi * df["FECHAPEDIDO"].dt.day / 31)
    df["DAYOFYEAR"] = np.sin(2 * np.pi * df["FECHAPEDIDO"].dt.dayofyear / 365)
    return df
```

In [36]:

```python
def add_timeseries_features(df):
    df["ROLLING_MEAN_3M"] = df["CANTIDADCOMPRA"].rolling(90).mean()
    df["WEIGHTED_MEAN_3M"] = (
        df["CANTIDADCOMPRA"]
        .rolling(90)
        .apply(lambda x: np.average(x, weights=range(1, len(x) + 1)))
    )
    df["EWMA_3M"] = df["CANTIDADCOMPRA"].ewm(span=90).mean()
    df["ROLLING_MEAN_1Y"] = df["CANTIDADCOMPRA"].rolling(365).mean()
    df["WEIGHTED_MEAN_1Y"] = (
        df["CANTIDADCOMPRA"]
        .rolling(365)
        .apply(lambda x: np.average(x, weights=range(1, len(x) + 1)))
    )
    df["EWMA_1Y"] = df["CANTIDADCOMPRA"].ewm(span=365).mean()
    # average CANTIDADCOMPRAS over year
    df["AVG_1Y"] = df.groupby(["YEAR"])["CANTIDADCOMPRA"].transform("mean")
    df["AVG_1Y"] = df["AVG_1Y"].fillna(df["AVG_1Y"].mean())
    return df
```

	PRODUCT	HOSPITAL	Tweedie	MSE	SMAPE
0	66071	7	-0.666667	2162.997939	72.380964
0	66071	13	-0.972549	1381.658753	69.268008
0	85758	10	-9.330963	424.117832	41.967628
0	64911	7	-0.250000	402.492263	24.000019
0	64764	0	-1.166614	376.817257	32.754517
0	64764	18	-0.365948	366.439863	10.167945
0	64751	0	-0.111203	348.622702	12.668822
0	64765	0	-1.365553	344.519910	34.058498
0	64764	10	-2.465778	329.302830	38.659885
0	64765	18	-0.266566	252.093634	10.737253
0	64663	13	-11.756655	214.298762	35.892156
0	73753	10	-228.834555	203.710348	33.726697
0	65056	13	-3.468980	169.965293	28.593626
0	64751	18	-1.570007	163.487213	9.374597
0	65056	0	-0.795825	161.762048	9.906280
0	64940	7	-inf	161.657582	16.666621
0	64932	7	-1.500000	154.919322	29.333334
0	64488	13	-3.084364	153.381673	69.892475
0	65056	10	-23.542847	153.256337	42.999492
0	64488	10	0.077130	140.860567	25.473784

baseline_single_product

```python
import sys
import warnings
import re

warnings.simplefilter(action="ignore", category=FutureWarning)

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

DATA_PATH = "../data"
```

In [53]:

```python
# Load excel file
df = pd.read_excel(DATA_PATH + "/consumo_material_clean.xlsx")
```

Preprocessing

In [54]:

```
# Separate code into two columns
new_columns = df["CODIGO"].str.extract(r"([a-zA-Z]+)([0-9]+)", expand=False)
df["CODIGO_CLASS"] = new_columns[0]
df["CODIGO_NUM"] = new_columns[1]
df.drop(columns=["CODIGO"], inplace=True)
```

In [55]:

```
# FECHAPEDIDO to datetime in day/month/year format
df["FECHAPEDIDO"] = pd.to_datetime(df["FECHAPEDIDO"], dayfirst=True)
df.sort_values(by=["FECHAPEDIDO"], inplace=True)
df.reset_index(drop=True, inplace=True)
```

```
/var/folders/c6/kf2fcpcj6pq_gwpfbfzzy81m0000gn/T/ipykernel_7630/38037239.py:2: UserWarnin
g: Could not infer format, so each element will be parsed individually, falling back to `
dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df["FECHAPEDIDO"] = pd.to_datetime(df["FECHAPEDIDO"], dayfirst=True)
```

In [56]:

```
# assert all rows in origen follow number-number-number format
def fix_origen_typos(origen_string):
    numbers = re.findall(r"[0-9]+", origen_string)
    return "-".join(numbers)
```

experimentation_single_prod

In [1]:

```
import sys
import warnings
import re

warnings.simplefilter(action="ignore", category=FutureWarning)

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

DATA_PATH = "../data"
```

In [2]:

```
# Load excel file
df = pd.read_excel(DATA_PATH + "/consumo_material_clean.xlsx")
```

Preprocessing

In [3]:

```
# Separate code into two columns
new_columns = df["CODIGO"].str.extract(r"([a-zA-Z]+)([0-9]+)", expand=False)
df["CODIGO_CLASS"] = new_columns[0]
df["CODIGO_NUM"] = new_columns[1]
df.drop(columns=["CODIGO"], inplace=True)
```

experimentation_all_prods

```python
import sys
import warnings
import re

warnings.simplefilter(action="ignore", category=FutureWarning)
warnings.simplefilter(action="ignore", category=RuntimeWarning)

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import GroupKFold
from sklearn.metrics import (
    mean_absolute_percentage_error,
    mean_squared_error,
    d2_tweedie_score,
)
from xgboost import XGBRegressor
from catboost import CatBoostRegressor
from lightgbm import LGBMRegressor
from sklearn.ensemble import RandomForestRegressor, HistGradientBoostingRegressor
from sklearn.preprocessing import OneHotEncoder

from category_encoders import LeaveOneOutEncoder, TargetEncoder
```

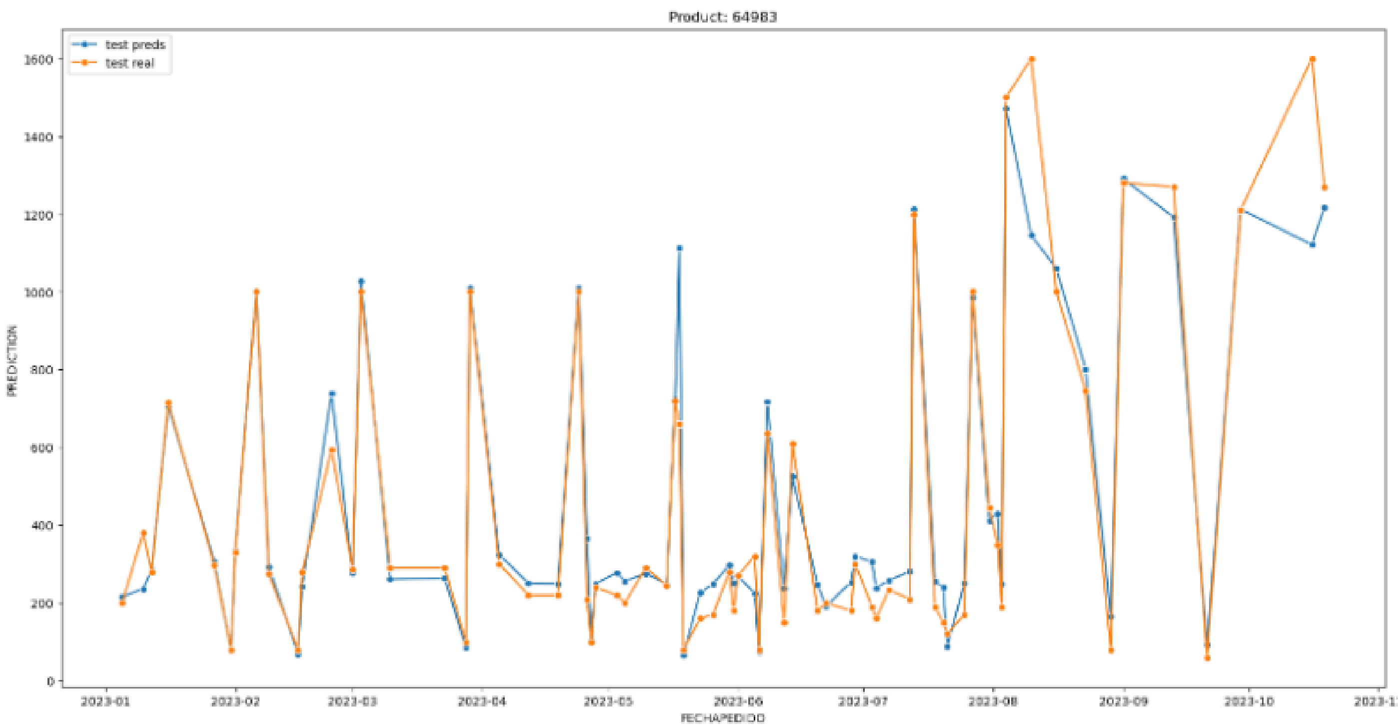
Out[726]:

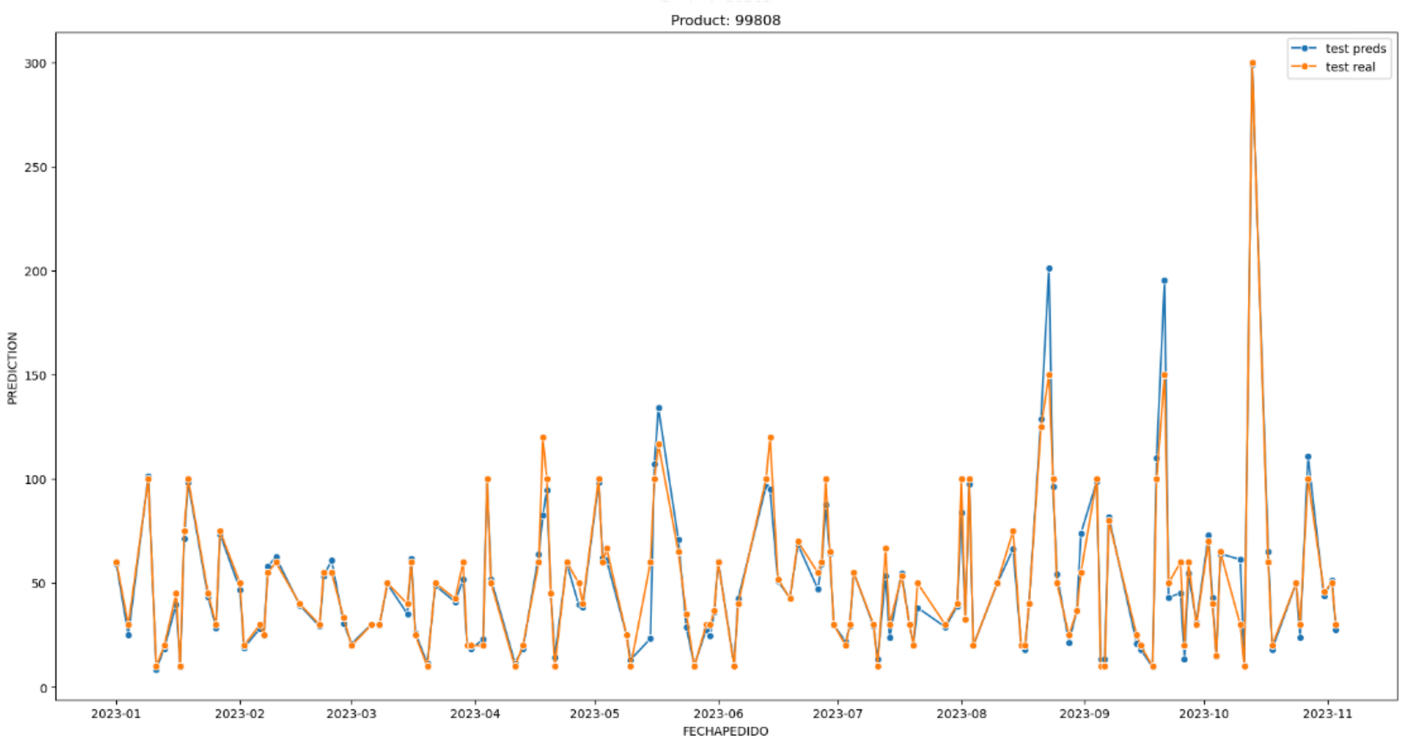
	PRODUCT	Tweedie	MSE	SMAPE	EXPENSE_ERROR
0	65056	0.495367	335.460856	42.875289	0.302234
0	65485	0.655914	102.767035	23.079493	0.228691
0	64764	0.380909	926.625603	29.730300	0.227269
0	69682	0.447645	37.896223	16.520367	0.181765
0	50071	0.255108	4.305755	21.626226	0.173299

Sample product prediction: Boltzmann Ensemble

In [728]:

```
for product in product_losses.sort_values(by=["Tweedie"], ascending=False)[
    "PRODUCT"
].head(10):
    plot_model_predictions(full_df=df, product=str(product))
```





REFERENCES

https://medmgtservices.com/hospital-inventory-management-challenges/
https://www.cardinalhealth.com/CIMS_10Barriers_Effective_Inventory_Management.pdf https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8342273/