
Ministère de l'Enseignement
Supérieur et de la Recherche
Scientifique

**** *

Université de Carthage

**** *

Ecole Polytechnique de Tunisie



وزارة التعليم العالي و البحث العلمي

**** *

جامعة قرطاج

**** *

المدرسة التونسية للتقنيات

ENGINEERING INTERNSHIP REPORT

Optimizing Monthly Objective Distribution for Pharmaceutical Wholesalers An OR and ML Approach

1st July - 15th August, 2023

Hosting Company

MEDICACOM

Elaborated by

Youssef KHARRAT

Graduate Engineering Student - SISY

Supervised by

Dr. Majdi KARRAY

CEO at MEDICACOM

Academic Year

2022/2023

Rue Elkhawarezmi BP 743 La Marsa 2078

Tel: 71 774 611 -- 71 774 699 Fax: 71 748 843

Site Web: www.ept.tn

نهج الخوارزمي ص.ب 743 المرسى 2078

الهاتف : 71 774 611 - 71 774 699 الفاكس : 71 748 843

موقع الواب : www.ept.tn

Abstract

This report provides a detailed description of the steps followed to formulate the objective distribution problem mathematically, a type of **NP**-hard resource allocation problem, and subsequently solve it using a metaheuristic approach.

The fundamental outcome of this effort is the development of a method to determine the optimal objective distribution vector, not reliant on the wholesaler's expertise but based on rigorous analysis of pharmaceutical wholesaler behavior, medicine trends, demand fluctuations, and seasonal patterns, employing techniques from various disciplines, including statistics, estimation, machine learning, data analysis, data preprocessing, operational research, and optimization.

Keywords : Objective Distribution Problem, Bounded Knapsack Problem, Operational Research, Gradient Boosting Regressor, Metaheuristic Approaches, Data Preprocessing, Pharmaceutical Supply Chain.

“The only constant variable is the unexpected. We can’t control it. It’s only when you look backwards, that you see how it was all connected.”

– Lessons in Chemistry, 2023

Acknowledgements

Foremost, we wish to express our sincere appreciation to the dean of **Ecole Polytechnique de Tunisie** for giving us the opportunity to engage in this engineering internship within the startup, Medicacom.

Bearing in mind the above, we extend our deepest gratitude and admiration to our supervisor, **Dr. Majdi KARRAY**, Co-Founder, and CEO at Medicacom, for entrusting us with the privilege to contribute to this project within a professional setting. His unwavering guidance, demonstrated through multiple sessions, has been instrumental to our progress.

We must also acknowledge that the successful completion of this work was made possible through the support and supervision of **Mr. Iskander REGAIEG**, Data Scientist at Medicacom. His expert guidance, counsel, and profound knowledge in high-performance computing greatly enriched our experience throughout this internship.

Lastly, we want to express our gratitude to everyone who played a role in ensuring the seamless execution and success of our engineering internship.

Contents

Abstract	i
Acknowledgements	iii
List of Figures	viii
List of Tables	ix
List of Acronyms	x
General Introduction	xi
1 Project Overview	1
1.1 Host Organization	1
1.2 Project Presentation	2
1.2.1 Context	2
1.2.2 Problematic	2
1.2.3 Description of the Current Scenario	2
1.2.4 Main Mission of the Project	3
1.3 Internship Progress	3
2 Problem Reformulation and Modeling	5
2.1 Pharmaceutical Supply Chain Process	5
2.2 Mathematical Reformulation	8
2.2.1 Problem Parameters	8
2.2.1.1 Industry Parameter	8
2.2.1.2 Wholesaler Parameter	8
2.2.1.3 Month Parameter	8
2.2.2 Problem Variables	8
2.2.2.1 Objective Variable	8
2.2.2.2 Distribution Vector	9
2.2.2.3 Stock Vector	9
2.2.2.4 Sales Vector	9
2.2.3 Problem Constraints	9
2.2.3.1 Budget Constraint	9
2.2.3.2 Forecast Demand Vector	9

2.2.3.3	Understocking Constraint	10
2.2.3.4	Threshold Vector	10
2.2.3.5	Overstocking Constraint	10
2.2.4	Problem Formulation	11
2.2.4.1	Trend Vector	11
2.2.4.2	Value Vector	11
2.2.4.3	Criterion Function	11
2.2.4.4	Problem Statement	11
2.3	Resolution	12
2.3.1	Solving Steps	12
2.3.2	Sequence Diagram	12
2.4	State of the Art	14
2.4.1	Sales Average	14
2.4.2	Inventory Ratio	14
2.4.3	Mediacom Formula	14
2.4.4	Discussion	14
3	Data Preprocessing and Analysis	15
3.1	Dataset Overview	15
3.1.1	Dataset Presentation	15
3.1.2	Data Acquisition	16
3.1.3	Dataset Issues	16
3.1.3.1	Dirty Data	17
3.1.3.1.1	Missing Values	17
3.1.3.1.2	Noisy Data	17
3.1.3.2	Categorical Features	17
3.1.3.3	Redundant Features	17
3.1.3.4	Missing Features	18
3.1.3.5	Data Complexity	18
3.1.3.6	Heterogeneous Input Features	18
3.2	Data Preprocessing	18
3.2.1	Inventory Dynamics	18
3.2.2	Filtering Data	18
3.2.3	Data Transformation	19
3.2.3.1	Converting Nominal to Numeric	19
3.2.3.2	Redundant Data Elimination	19
3.2.4	Data Cleaning	20
3.2.4.1	Handling Missing Values	20
3.2.4.2	Handling Noisy Data	20
3.2.5	Data Integration	20
3.2.5.1	Price Vector Integration	20
3.2.5.2	Monthly Objective Integration	20
3.2.5.3	Integrating the Sales Vector of the Last Month	20
3.2.6	Data Reformatting	20

3.3	Data Analysis	21
3.3.1	Correlation Analysis	21
3.3.2	Feature Engineering	22
3.3.2.1	Incrementation Matrix	22
3.3.2.2	Dynamically Predicted Sales Vector	23
3.3.2.3	Priority Matrix	23
3.3.2.4	Traditionally Predicted Sales Vector	24
3.3.2.5	Seasonal Trend Matrix	24
3.3.2.6	Sales Vector Predicted by Seasonal Trend	24
3.3.2.7	Global Trend Vector	25
3.3.2.8	Sales Vector Predicted by Global Trend	25
3.3.3	Feature Selection	25
4	Sales Forecasting Model	27
4.1	Model Specification	27
4.2	Model Selection	28
4.2.1	Regression	28
4.2.2	Possible Models	28
4.2.3	Model Comparison	29
4.2.4	Gradient Boosting Regressor	29
4.3	Hyperparameter Tuning	30
4.3.1	Empirical Test	31
4.3.2	Model Performance	31
4.4	Finalizing the Objective Distribution Problem Statement	32
4.4.1	Sales Vector Forecasting	32
4.4.2	Threshold Vector Approximation	33
4.4.3	Complete Problem Statement	34
5	Solving the Objective Distribution Problem	35
5.1	Bounded Knapsack Problem	35
5.2	Converting the ODP to a BKP	36
5.2.1	Simplification by the Understocking Constraint	37
5.2.2	Modification on the Overstocking Constraint	37
5.2.3	Bound Vector	37
5.2.4	Analogy Between ODP and BKP	38
5.2.5	ODP Statement in the Framework of a BKP	38
5.3	Solving the Bounded Knapsack Problem	39
5.3.1	Dynamic Programming	39
5.3.2	Metaheuristic Algorithm	40
6	Implementation and Validation	42
6.1	Notebook and Matrices	42
6.2	Instructions for Using the Notebook	43
6.3	Testing the Implemented Solution	44

6.3.1	Input	44
6.3.2	Output	45
6.3.3	Discussion	46
General Conclusion and Perspectives		48
References		48

List of Figures

1.1	Gantt Chart	4
2.1	Schematic representation of the pharmaceutical supply chain	7
2.2	Sequence Diagram	13
3.1	Screenshot of the Initial Dataset	16
3.2	Screenshot of the wholesaler matrix (right) and the medicine matrix (left)	19
3.3	Screenshot of the dataset after applying preprocessing	21
3.4	Importance of each feature for forecasting the sales vector	21
3.5	Correlation Heatmap	22
3.6	Screenshot of the incrementation matrix	23
3.7	Screenshot of the priority matrix	24
3.8	Importance of each proposed feature for forecasting the sales vector . . .	25
3.9	Screenshot of the Final Dataset	26
4.1	Example of the grid search application	31
4.2	Gradient Boosting Regressor Performance	32
6.1	Input of the Sample Coordinates	44
6.2	Screenshot of the Input Data of the Sample	45
6.3	Screenshot of the Model Output	46
6.4	Comparing the forecast sales vector and the sales vector	47

List of Tables

4.1	Model Comparison	29
5.1	Analogy Between ODP and BKP	38

List of Acronyms

BKP Bounded Knapsack Problem.

GBR Gradient Boosting Regressor.

IT Information Technologies.

MAE Mean Absolute Error.

ML Machine Learning.

NP Nondeterministic Polynomial.

ODP Objective Distribution Problem.

OR Operational Research.

RMSE Root Mean Squared Error.

General Introduction

As part of the training and preparation of engineering students for their prospective careers, with the aim of qualifying them for the job market using applied methods, the Ecole Polytechnique de Tunisie offers an obligatory engineering internship, enabling its students to explore the environment of engineering work within a company in its authentic setting, and allowing them to immerse themselves in its corporate environment, concretize theoretical knowledge, hone practical skills, and cultivate a sense of responsibility.

During the period from July 1 to August 15, 2023, we completed an engineering internship in the data science department of Medicacom, a startup located in Sfax, specializing in IT services within the field of e-health.

The assigned project aimed to address the issue of chaotic monthly medicine purchases, including problems of overstocking and understocking faced by pharmaceutical wholesalers. The objective was to propose an optimal objective distribution. As a matter of fact, the pharmaceutical industry, a cornerstone of healthcare, operates within a complex network of supply chains involving diverse stakeholders and intricate logistics. This realm is marked by various challenges—demand fluctuations, inventory management intricacies, and resource allocation dilemmas—which directly impact the availability of medications to patients, a non-negotiable exigency in healthcare.

In the pursuit of optimizing these processes, our engineering internship report offers solution based on data analysis, machine learning, and operational research to optimize pharmaceutical supply chains. The report unfolds methodically, commencing with an overview of the sector and the hosting company, followed by a detailed presentation of the project's goals and a summary of the internship's circumstances and progress.

The subsequent chapter enumerates the steps of the pharmaceutical supply chain to comprehend our project's position and identify the wholesaler's needs, thereby clarifying the project's objectives. We then proceed to reformulate the problem mathematically, concretizing the concepts.

The third chapter focuses on exploring the dataset provided by Medicacom, with the aim of preparing it for integration into a machine learning model designed for forecasting the sales vector. This step consumed nearly half of our time during the engineering

internship, driven by our belief in the principle of 'Garbage In, Garbage Out.'

In the subsequent chapter, our attention shifts to the development of a sales forecasting machine learning model capable of generating a forecast sales vector, enabling us to completely state the objective distribution problem.

The fifth chapter centers on solving the objective distribution problem using an operational research approach, by projecting it onto another resource allocation problem, the bounded knapsack problem.

The subsequent chapter provides detailed instructions for using the notebook and the implemented solution. We showcase and discuss the outcomes achieved thus far. Finally, we will wrap up our work with a conclusion and some recommendations for the future.

It is worth mentioning that through this internship experience, we applied our multidisciplinary background and the theoretical knowledge acquired during our training at EPT, specifically in data analysis, machine learning, operational research, optimization, and estimation, demonstrating the multidisciplinary nature of polytechnic engineering students—a key quality that we do believe is highly demanded. Beyond enriching our expertise, this internship allowed us to refine our scientific rigor, critical thinking, open-mindedness, and taught us the adaptability required of an engineer.

Chapter 1

Project Overview

Introduction

In this chapter, we will present an overview of the hosting company, Medicacom, where our engineering internship took place. Next, we will provide details about the project's context, the identified problematics, the existing solution, and the project's primary objectives. Afterward, we will illustrate the progress of the internship using a Gantt chart.

1.1 Host Organization

Medicacom, founded in 2020 and based in Sfax, is an IT services startup specialized in delivering innovative technological products covering a wide range of business. With a keen focus on developing the field of e-health, it adopts advanced methodologies inspired by latest technologies, notably artificial intelligence, to optimize patient care pathways.[9]

Its multidisciplinary team, comprising medical experts, software engineers, and data scientists, seek to help healthcare professionals explore the world of artificial intelligence, including machine learning and deep learning, and harness its incredible advantages in the health sector, which have proven to be highly effective.[9]

Medicacom's vision revolves around designating the patient as the main player in their health journey and offering them remote support from health experts through advanced e-health tools, all while maintaining strict confidentiality policies. Besides, the startup advocates for prevention over treatment, emphasizing the importance of preventing diseases before they occur and taking necessary precautions to minimize potential risks.[9]

1.2 Project Presentation

1.2.1 Context

In the complex world of pharmaceutical supply chains, medicines journey from manufacturing industries, to monthly bulk purchases by pharmaceutical wholesalers, and, ultimately, to pharmacies, reaching eventually individual consumers.

More concretely, the purchasing operation between the industry and pharmaceutical wholesalers operates as follows: every month, each wholesaler sets an objective, or turnover, that they aim to achieve by purchasing appropriate medicines. In simple terms, they have a budget they want to use for buying the right medicines, considering seasonal trends and their current stock. So, they allocate their budget among medicines based on intuition and their expertise in the field.

1.2.2 Problematic

However, with this approach, pharmaceutical wholesalers have found themselves dealing with issues of overstocking and understocking. In fact, some wholesalers, midway through the month, run out of the products pharmacies need, which ends up costing them customers. Additionally, this chaotic stock management results in certain products ending up sitting in storage for months before they're sold, occupying unprofitable space, and causing financial losses.

The industry, too, faces its set of challenges in preparing orders. The sales operation is essentially a retail process, not made-to-order manufacturing. Wholesalers place their monthly orders with the industry and expect delivery within a week, necessitating orders estimation in advance and manufacturing products accordingly. However, the industry's estimation methods, rooted in experience, lack the rigor required for such precision, leading to potential customer loss.

In short, we need a method or model to optimize the distribution of wholesalers' monthly objectives among medicines, considering their stock status and seasonal patterns, and at the same time, it provides the industry with precise quantities for manufacturing, thereby avoiding excess or shortage in production.

1.2.3 Description of the Current Scenario

Medicacom plays as an intermediary between the industry and pharmaceutical wholesalers, handling the digitalization of product sales operations. In fact, at the beginning of each month, after receiving the Monthly Statements from each wholesaler – which detail the quantities of medicines sold in the previous month, current stock status, and this

month's objective – the industry records this data on a platform developed by Medicacom.

Medicacom has also developed a formula for objective distribution proposal, based on least-squares method and stock ratios. This formula indirectly aids the industry in determining the appropriate quantities to be manufactured in advance. More importantly, the industry utilizes it to propose the objective distribution to wholesalers or to suggest adjustments to their orders, as orders are negotiable as long as the distribution aligns with the objective.

1.2.4 Main Mission of the Project

The main mission of this project is to create a predictive model based on machine learning to optimize the distribution of pharmaceutical wholesalers' monthly objectives among medicines. The model considers factors like the sales history from the past three years, seasonal patterns, inventory capacities, clearance ratios, and product prices. By doing so, this approach ensures effective inventory management for both pharmaceutical wholesalers and the industry.

1.3 Internship Progress

Throughout our internship, we adhered to an organized and structured schedule, as illustrated in the following Figure 1.1. Roughly half of our internship was focused on problem modeling and data preprocessing, driven by our belief in the postulate of 'Garbage In, Garbage Out'.

Week	July				August	
	1	2	3	4	1	2
Understanding the nature of the problem statement Reformulation and Modeling of the Problem Tasks Identification						
Data Preprocessing Converting the dataset into a usable format Analyzing data and selecting features						
Comparison and evaluating different machine learning models for forecasting Choosing the Gradient Boost Regression Model Tuning model hyperparameters and feature selection						
Development of the RO distribution model Reformulating mathematically the problem into the knapsack problem						
Implementation of the solution Ensuring consistency and accuracy in the implemented model Testing and Validation						

Figure 1.1: Gantt Chart

Conclusion

In this chapter, we presented an overview of the hosting company, Medicacom. Next, we provided details about the project's context, including the identified problems, the existing solution, and the project's primary objectives. Afterward, we illustrated the progress of the internship using a Gantt chart. The next chapter will focus on reformulating the problem mathematically and rigorously concretizing the concepts.

Chapter 2

Problem Reformulation and Modeling

Introduction

In this chapter, we will enumerate the steps of the pharmaceutical supply chain in order to understand our project's position and identify the wholesaler's needs, thereby clarifying the project's output. Next, we will reformulate the problem mathematically, concretizing the concepts and defining the problem parameters and variables. This step is at **the core of our engineering internship** since the engineer's fundamental task is modeling before solving. Subsequently, we will define the problem constraints and the criterion function. We will then state the problem, which we refer to as the objective distribution problem (ODP). Afterward, we will break down the problem resolution into steps that we will follow in the remainder of our work, clarifying the solution's architecture through a sequence diagram. Finally, we will discuss the weaknesses of the Medicacom formula.

2.1 Pharmaceutical Supply Chain Process

The pharmaceutical supply chain process, which outlines the steps and illustrates the trajectory of medicines from manufacturing to consumption, is quite complex in reality, involving numerous intricate details. One way to schematize it simplistically is as shown in Figure 2.1, and the following steps outline this process:

1. **Demand Estimation:** The industry¹ estimates in advance the quantities of products that are likely to be ordered by its regular customers, in our case, 32 different pharmaceutical wholesalers. Subsequently, the industry fabricates the medicines predicated on these estimations.
2. **Monthly Objective Setting:** At the beginning of each month, the wholesaler sets its objective for the month. Essentially, a Monthly Objective refers to the turnover

¹In precise terms, "the industry" refers to the manufacturing entities of medicines. In our case, Medicacom collaborates with two entities. However, for methodological simplicity, we consider them as one industry, as our developed model depends on the type of medicine, not the manufacturer.

that a wholesaler aims to achieve this month or even surpass. It can also be defined—and this is the definition we shall adopt for the remainder of the report—as the allocated budget for the month, earmarked by the wholesaler for procuring the appropriate medicines.

3. **Order Creation:** To create an order, the wholesaler either autonomously distributes the objective among medicines, considering its current stock status, seasonal patterns, inventory capacity, and its own expertise, or it may solicit industry guidance for objective distribution and ask for a suggestion based on collective expertise gained from interactions with other wholesalers. In both cases, the order remains negotiable as long as the distribution aligns with the objective.
4. **Purchasing Operation:** After the order is placed, the purchasing operation occurs. The wholesaler formally submits the order and expects delivery within a week. This underscores the necessity for the industry to prepare and manufacture products in advance.
5. **Inventory Management:** Subsequently, the wholesaler stores the bulk quantities of medicines purchased in its inventory until each product is sold to a pharmacy.
6. **Consumer Dispensation:** Finally, the medicines reach individual consumers through pharmacies.

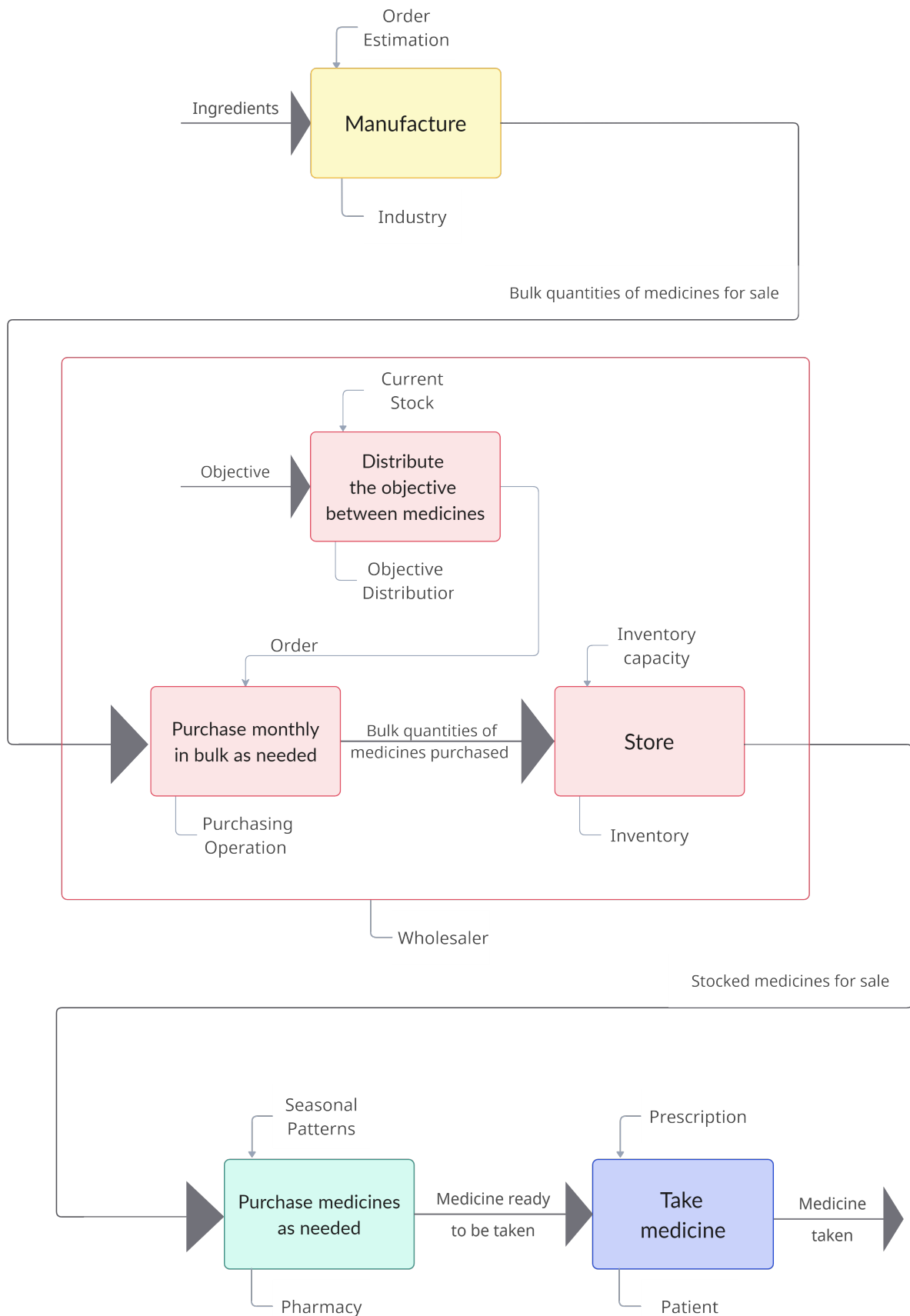


Figure 2.1: Schematic representation of the pharmaceutical supply chain

2.2 Mathematical Reformulation

One way to formulate the objective distribution problem mathematically is as follows:

2.2.1 Problem Parameters

First, we define the problem parameters:

2.2.1.1 Industry Parameter

Let M be the number of medicines from which the wholesaler selects what to purchase in bulk from the industry; in our case, $M = 154$ *medicines*. The iteration through M is executed using the medicine identifier $i \in \llbracket 1; M \rrbracket$.

Let $P = (p^1, \dots, p^M) \in \mathbb{R}^M$ denote the price vector, where each element, p^i , represents the price of the i^{th} medicine expressed in *TND*.

2.2.1.2 Wholesaler Parameter

The wholesaler is identified by w , the wholesaler identifier, where $w \in \llbracket 1; W \rrbracket$, with W being the number of regular customers wholesalers of the industry; in our case, $W = 32$ *wholesalers*.

2.2.1.3 Month Parameter

Let t represent the month identifier, where $t \in \llbracket 1; 12 \rrbracket$, corresponding to the twelve months of the year.

We define the generic month, denoted as $\tilde{t} \in \mathbb{N}^*$, as the identifier of the month accounting for the current year. This notation proves valuable for integrating historical data from the previous 3 years.

Given that our dataset originates from the year 2019, we establish $\tilde{t}$ as follows:

$$\tilde{t} = t + 12 \cdot (y - 2019) , \text{ where } y \text{ signifies the year} \quad (2.1)$$

2.2.2 Problem Variables

Second, we define the problem variables:

2.2.2.1 Objective Variable

Let $O_{w,t}$ represent the objective of wholesaler w in month t , typically around the magnitude of 100,000 *TND*.

2.2.2.2 Distribution Vector

Let the vector of decision variables, referred to as the distribution vector, be denoted by $X_{w,t} = (x_{w,t}^1, \dots, x_{w,t}^M) \in \mathbb{N}^M$, where $x_{w,t}^i$ represents the quantity of the i^{th} medicine to be purchased by wholesaler w at the onset of month t .

We shall distinguish between the distribution vector that did happen, denoted as $X_{w,t}$, and the proposed ones calculated before the purchasing operation occurred, denoted as $\hat{X}_{w,t}$ or $X_{w,t}^*$.

2.2.2.3 Stock Vector

Let $S_{w,t} = (s_{w,t}^1, \dots, s_{w,t}^M) \in \mathbb{N}^M$ represent the stock vector, where each element $s_{w,t}^i$ signifies the quantity of the i^{th} medicine stored in the inventory of wholesaler w , at the onset of month t .

2.2.2.4 Sales Vector

Let $V_{w,t} = (v_{w,t}^1, \dots, v_{w,t}^M) \in \mathbb{Z}^M$ represent the sales vector, where each element $v_{w,t}^i$ signifies the quantity of the i^{th} medicine sold during month t by wholesaler w . The sales vector $V_{w,t}$ remains unknown during the month t and will only be known at the end of the month. Thus, we define the forecast sales vector $\hat{V}_{w,t}$.

2.2.3 Problem Constraints

Then, we define the problem constraints:

2.2.3.1 Budget Constraint

Since we defined the monthly objective $O_{w,t}$ as the allocated budget for the month, earmarked by wholesaler w for procuring the appropriate medicines, the cost of the bulk quantities of medicines to be purchased by the wholesaler should match² the objective. Hence, the mathematical translation is formulated as follows:

$$\sum_{i=1}^M x_{w,t}^i \cdot p^i = O_{w,t} \quad (2.2)$$

2.2.3.2 Forecast Demand Vector

Let $D_{w,t} = (d_{w,t}^1, \dots, d_{w,t}^M) \in \mathbb{N}^M$ represent the demand vector, where each element $d_{w,t}^i$ signifies the quantity of the i^{th} medicine requested from the wholesaler w during month t .

The demand vector $D_{w,t}$ remains unknown during month t and will only be known at the end of the month. Thus, we define the forecast demand vector $\hat{D}_{w,t}$.

²equal or slightly higher than the objective

2.2.3.3 Understocking Constraint

By definition, understocking refers to the situation in which a wholesaler fails to maintain adequate inventory levels of its products to fulfill pharmacy demands.

It occurs when the quantity of medicines on hand is not sufficient to meet the immediate needs of customers, leading to stockouts and potential lost sales opportunities.[16]

In other words, to avoid understocking, the quantity of each medicine stored in the inventory of wholesaler w at the onset of month t , combined with the quantity to be purchased by the wholesaler at the onset of the month, should be higher than the quantity of the medicine requested from the wholesaler during the month. Namely:

$$s_{w,t}^i + x_{w,t}^i \geq d_{w,t}^i, \text{ for } i = 1, 2, \dots, M \quad (2.3)$$

Since the demand vector $D_{w,t}$ is unknown at the onset of month t , we forecast it by analyzing historical sales data and seasonal patterns. This necessitates the application of the machine learning model, which assumes a crucial role in the analysis.

Furthermore, assuming the entire demand quantity will be sold, the understocking constraint can be expressed as follows, using the forecast sales vector $\hat{V}_{w,t}$:

$$s_{w,t}^i + x_{w,t}^i \geq \hat{v}_{w,t}^i, \text{ for } i = 1, 2, \dots, M \quad (2.4)$$

2.2.3.4 Threshold Vector

Let $H_{w,t} = (h_{w,t}^1, \dots, h_{w,t}^M) \in \mathbb{Z}^M$ represent the threshold vector, where each element $h_{w,t}^i$ signifies the maximum quantity of the i^{th} medicine beyond which, if the stored quantity of the i^{th} medicine in the inventory of the wholesaler w , at the onset of month t , surpasses it, this medicine will remain dormant for more than 2 months.

Evidently, this vector is unknown; hence, we define the approximate threshold vector denoted by $\hat{H}_{w,t}$.

2.2.3.5 Overstocking Constraint

By definition, overstocking refers to the situation in which a wholesaler holds excess inventory beyond what is necessary to meet current or future demand. While maintaining a sufficient stock level is crucial for uninterrupted operations, overstocking can lead to various challenges that impact a company's financial health and overall efficiency.[11]

It occurs when the medicine remain dormant for more than 2 months. In other words, to avoid overstocking, the quantity each medicine stored in the inventory of wholesaler w at the onset of month t , combined with the quantity to be purchased by the wholesaler at the onset of the month, should be lower than the maximum quantity of the medicine beyond which, if the stored quantity of the medicine in the inventory of the wholesaler w , at the onset of month t , surpasses it, this medicine will remain dormant for more than 2 months. Namely:

$$s_{w,t}^i + x_{w,t}^i \leq h_{w,t}^i, \text{ for } i = 1, 2, \dots, M \quad (2.5)$$

Since the threshold vector $H_{w,t}$ is unknown, and difficult to calculate by a heuristic method, we approximate it by analyzing historical sales data, storage costs and seasonal patterns.

2.2.4 Problem Formulation

The **objective distribution problem (ODP)** is an optimization problem in which a set of medicines, each with a price p^i , a value $\nu_{w,t}^i$ and a trend coefficient $\tau_{w,t}^i$, must be purchased in bulk with a limited budget $O_{w,t}$ in such a way as to maximize the total value of the medicines purchased. The challenge is to determine the optimal quantity of each medicine to be purchased $x_{w,t}^i$, with these quantities being the decision variables.

2.2.4.1 Trend Vector

Let $\tau_{w,t} = (\tau_{w,t}^1, \dots, \tau_{w,t}^M) \in [0, 1]^M$ represent the trend vector, where each element $\tau_{w,t}^i$ signifies the percentage of the seasonal sales trend of the i^{th} medicine by the wholesaler w during month t .

2.2.4.2 Value Vector

Let $\nu_{w,t} = (\nu_{w,t}^1, \dots, \nu_{w,t}^M) \in \mathbb{R}^M$ represent the value vector, where each element $\nu_{w,t}^i$ signifies the value of the i^{th} medicine for the wholesaler w during month t , computed as the product of the price and the respective trend coefficient:

$$\nu_{w,t}^i = p^i \cdot \tau_{w,t}^i \quad (2.6)$$

2.2.4.3 Criterion Function

The goal is to determine the optimal distribution vector $X_{w,t}^*$ in such a way as to maximize the probability of each purchased medicine $x_{w,t}^{*,i}$ being sold in this month t or the next $t + 1$, at most. In other words, we aim to maximize the total value of the medicines purchased. Thus, the criterion function is defined as follows:

$$\text{maximize } \sum_{i=1}^M x_{w,t}^i \cdot \nu_{w,t}^i \quad (2.7)$$

2.2.4.4 Problem Statement

One way to state the objective distribution problem is as follows:

Find the optimal distribution vector $X_{w,t}^*$ in order to:

$$\text{maximize } \sum_{i=1}^M (x_{w,t}^i \cdot p^i) \cdot \tau_{w,t}^i \quad (2.8)$$

subject to constraints:

$$\begin{cases} \sum_{i=1}^M x_{w,t}^i \cdot p^i = O_{w,t} \\ s_{w,t}^i + x_{w,t}^i \geq \hat{v}_{w,t}^i, & \forall i \in \llbracket 1; M \rrbracket \\ s_{w,t}^i + x_{w,t}^i \leq \hat{h}_{w,t}^i, & \forall i \in \llbracket 1; M \rrbracket \\ x_{w,t}^i \in \mathbb{N}, & \forall i \in \llbracket 1; M \rrbracket \end{cases} \quad (2.9)$$

2.3 Resolution

In the problem statement, we have three unknown vectors $\tau_{w,t}$, $\hat{V}_{w,t}$, and $\hat{H}_{w,t}$ apart from the distribution vector we are searching for. Thus, we need to calculate or estimate these vectors initially before proceeding to solve the ODP.

2.3.1 Solving Steps

To solve the ODP, we have put together a step-by-step approach to be employed:

1. **Trend Vector Calculation:** Compute the trend vector $\tau_{w,t}$ by analyzing historical sales data. (Skill: Data Analysis)
2. **Sales Vector Forecasting:** Forecast the sales vector $\hat{V}_{w,t}$ by analyzing historical sales data and seasonal patterns. (Skill: Machine Learning)
3. **Threshold Vector Approximation:** Approximate the threshold vector $\hat{H}_{w,t}$ by analyzing historical sales data, storage costs, and seasonal patterns. (Skill: Machine Learning)
4. **Operational Research Problem Solving:** Determine the optimal distribution vector $X_{w,t}^*$ by solving the ODP as an operational research problem equivalent to the UPK. (Skill: Operational Research)
5. **Implementation:** Implement the solution in a notebook to make it practical for the industry. (Skill: Information System)

2.3.2 Sequence Diagram

To better understand the link between the ODP solution and the wholesaler's needs, we employed a sequence diagram, illustrated in Figure 2.2, to visualize the solution's architecture.

A sequence diagram simply depicts interactions between objects in a sequential order, detailing the sequence in which these interactions occur. It describes how the objects in

a system function and in what order.[14]

In fact, this sequence diagram stands as **a map schematizing all of our work done during the engineering internship.**

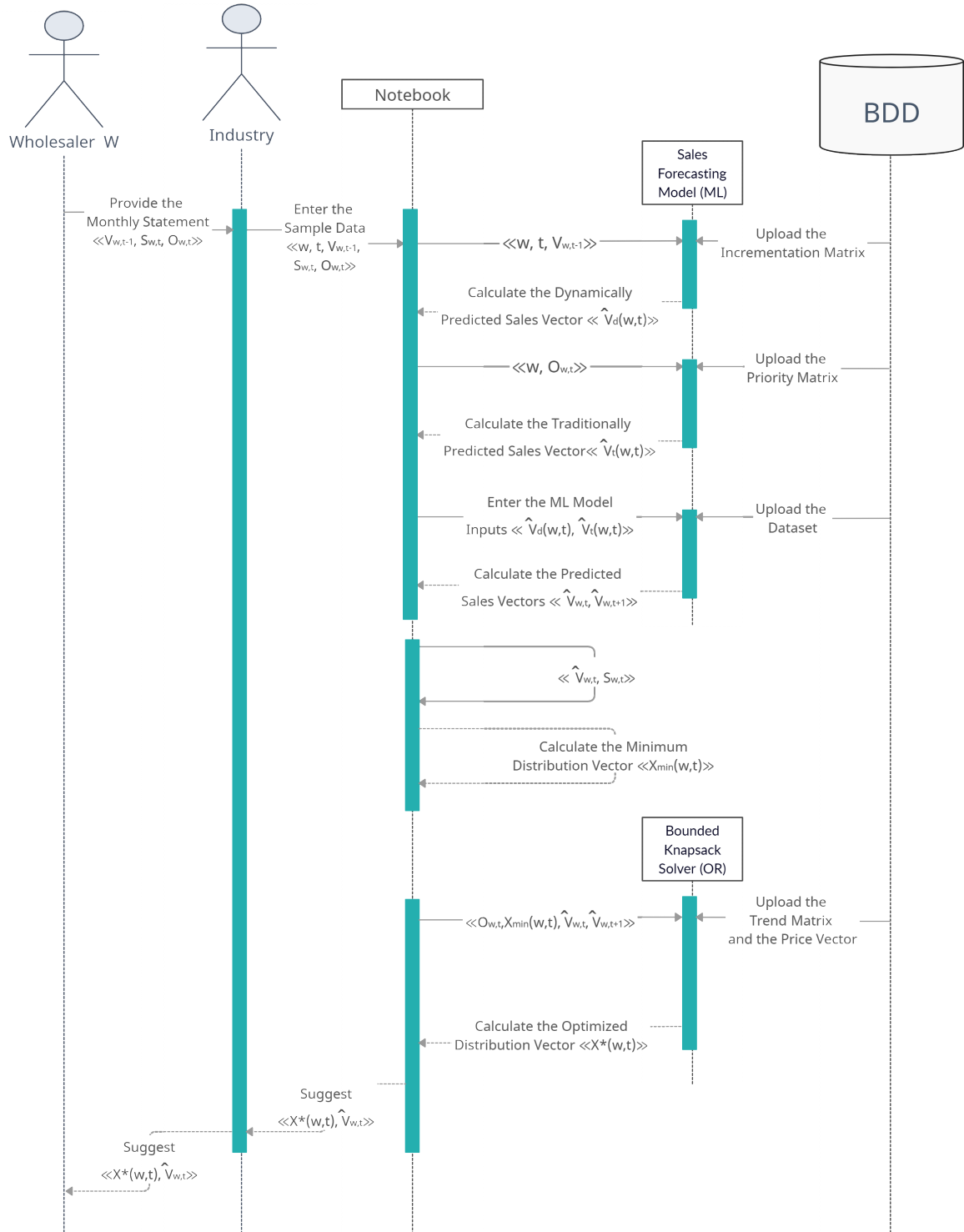


Figure 2.2: Sequence Diagram

2.4 State of the Art

Conclusion

In this chapter, we detailed the pharmaceutical supply chain, positioned our project, and clarified its output compared to the wholesaler's needs. Next, we rigorously reformulated the problem mathematically and concretized the concepts. Subsequently, we defined the problem constraints, the criterion function, and stated the objective distribution problem. Afterward, we divided the problem resolution into steps and visualized the solution's architecture through a sequence diagram. Finally, we discussed the limitations of the Medicacom formula. The next chapter will detail the data preprocessing and analysis, which is the first step of the ODP resolution.

Chapter 3

Data Preprocessing and Analysis

Introduction

In this chapter, we will focus on the exploration of the dataset provided by Medicacom, aiming to prepare it for integration into a machine learning model designed for forecasting the sales vector $\hat{V}_{w,t}$. First, we will present and understand the dataset and each attribute, and we will explain how the data was collected in order to understand how to correct the errors. Then, we will enumerate the issues of the dataset, which arrived in a less-than-ideal state, such as missing values, noisy data, and redundant features. Afterward, we will explain the strategic steps we followed in data preprocessing, which aim to transform raw data into a refined, useful format, ultimately facilitating the forecasting of the sales vector, and to correct the dataset errors based on the inventory dynamics formula that we shall define. After handling the dataset issues, we will analyze the data to uncover hidden connections and identify the correlation between features, such as wholesaler behavior and seasonal patterns. Then, we will transform the features into homogeneous features with the sales vector by proposing four ways to approximate the sales vector. We will evaluate these proposed four features and identify the most relevant and significant ones to create our final dataset. It is noteworthy that the work explained in this chapter took almost **half of our time during the engineering internship**, driven by our belief in the postulate of 'Garbage In, Garbage Out'.

3.1 Dataset Overview

To forecast the sales vector $\hat{V}_{w,t}$, we will construct a machine learning model. To accomplish this, we have been given a dataset comprising recorded purchasing operations from the past three years. This dataset serves as a valuable repository of wholesaler behavior, enabling us to gain insights into seasonal patterns.

3.1.1 Dataset Presentation

The dataset provided by Medicacom for exploitation consists of purchasing, sales, and inventory data from more than 32 regular wholesalers, spanning the period between Jan-

uary 2019 and May 2023. Figure 3.1, extracted from this database, illustrates the first 22 lines out of 1,014,839 lines. Each line contains 8 attributes: the wholesaler's name, the medicine's name, the type of operation, the date, the medicine's code, the measure, the type of measure, and the way of recording.

	A	B	C	D	E	F	G	H
1	Wholesaler Name	Medicine Name	Type	Date	Code	Measure	Measure Type	Check-in
2			STOCK	jeudi 1 décembre 2022		13	Volume	début mois
3			STOCK	jeudi 1 décembre 2022		39,871	Valeur	début mois
4			STOCK	jeudi 1 décembre 2022		144,149	Valeur	début mois
5			STOCK	jeudi 1 décembre 2022		47	Volume	début mois
6			STOCK	jeudi 1 décembre 2022		328,169	Valeur	début mois
7			STOCK	jeudi 1 décembre 2022		107	Volume	début mois
8			STOCK	jeudi 1 décembre 2022		349,638	Valeur	début mois
9			STOCK	jeudi 1 décembre 2022		114	Volume	début mois
10			STOCK	jeudi 1 décembre 2022		1272,805	Valeur	début mois
11			STOCK	jeudi 1 décembre 2022		415	Volume	début mois
12			STOCK	jeudi 1 décembre 2022		990,641	Valeur	début mois
13			STOCK	jeudi 1 décembre 2022		323	Volume	début mois
14			STOCK	jeudi 1 décembre 2022		170	Volume	début mois
15			STOCK	jeudi 1 décembre 2022		127	Volume	début mois
16			STOCK	jeudi 1 décembre 2022		521,39	Valeur	début mois
17			STOCK	jeudi 1 décembre 2022		389,509	Valeur	début mois
18			STOCK	dimanche 1 janvier 2023		1196,13	Valeur	début mois
19			STOCK	dimanche 1 janvier 2023		390	Volume	début mois
20			STOCK	dimanche 1 janvier 2023		125	Volume	début mois
21			STOCK	dimanche 1 janvier 2023		383,375	Valeur	début mois
22			STOCK	dimanche 1 janvier 2023		361,906	Valeur	début mois

Figure 3.1: Screenshot of the Initial Dataset

For instance, the first line of the Figure 3.1, reads as follows: the wholesaler " ", at the "onset of the month" on "Thursday, December 1, 2022", had a medicine " " identified by the code " " in its "inventory", with a "volume" of "13".

3.1.2 Data Acquisition

This dataset is formed by the accumulation of recordings of the Monthly Statements of each wholesaler, which contain the last sales vector $V_{w,t-1}$, the stock vector $S_{w,t}$, and the monthly objective $O_{w,t}$, along with the data collected by the industry, which is the distribution vector $X_{w,t}$.

It is important to mention that the format presenting the data is unusual because logically, a line should contain the data collected from the same monthly statement. It is unnecessary to divide the same monthly statement into six lines for each medicine purchased—two for sales data, two for stock data, and two for purchasing data—along with separate lines for quantity and value measurements, which introduces redundancies. These issues will be addressed through the application of data reduction and transformation techniques.

3.1.3 Dataset Issues

The received format of the dataset is unusable directly since it presents several problems. The quality of the data should be checked before applying machine learning to transform it into a format that is more easily and effectively processed[3]. Therefore, it is imperative to make necessary adjustments before starting machine learning[8]. In fact,

good data preparation is key to producing valid and reliable models[4]. Failing to do so may result in the well-known adage, "garbage in, garbage out," where the machine learns something wrong. Some detected issues of the dataset are represented as follows:

3.1.3.1 Dirty Data

Upon scanning the dataset, it becomes evident that there are a lot of missing values and noisy data, highlighting the necessity for **data cleaning**.

3.1.3.1.1 Missing Values In fact, the errors related to missing values are entirely normal. Sometimes, the wholesaler might not provide the monthly statement, or even if they do, the industry may fail to add it to the database. Additionally, the monthly statement is often added to the database without the distribution vector, or vice versa. This situation renders the sample, characterized by the objective and the distribution vector, unqualified for analysis. Such errors can be rectified using a data cleaning technique, which involves assigning default values to the missing entries, for example, or of course using the inventory dynamics formula that we will define later.

3.1.3.1.2 Noisy Data On the other hand, we found two types of errors in noisy data. The first type is represented by a zero value in the attribute of quantity measurement, indicating that the registered quantity is zero. However, in reality, it may not be zero since it is unreasonable for a purchasing operation to occur and its monthly statement to be full of zeros. Additionally, the data for the first 10 months or more of 2019 are almost all assigned to zero because the data recording platform was developed by Medicacom after that period.

The second type of noisy data results from the non-compliance with the inventory dynamics formula, often due to donation or sales of medicines occurring under the table (off the record) without proper traceability. These activities evidently bias the data, and these issues need to be addressed using data cleaning techniques. Additionally, we shall detect and resolve data value conflicts through outlier analysis.

3.1.3.2 Categorical Features

Before applying machine learning, we also need to convert the nominal attributes to numeric values. For example, the wholesaler and the medicine should be defined by identifiers. In fact, many machine learning algorithms, like regression, operate seamlessly with numerical inputs.

3.1.3.3 Redundant Features

To facilitate machine learning, we should eliminate redundant features, such as the extra measurements we have—both quantity and value—that essentially convey the same information. One measurement by quantity is entirely sufficient since the other measure is derived from it. These issues will be addressed through redundancy and correlation analysis, a technique of data integration.

3.1.3.4 Missing Features

Additionally, a sample should contain information about the objective, so the machine can learn who it was distributed to, as the startup didn't provide us with that data due to its confidentiality. However, this issue will be resolved by the budget constraint formula and feature engineering, a technique of data integration.

3.1.3.5 Data Complexity

The challenge posed by the monthly statement's division makes the analysis difficult. Thus, we shall pivot the stock values, the sale values, and the purchase values into separate columns, which will result in a reduced representation of the dataset that is much smaller in volume, yet closely maintains the integrity of the original data. Such issues will be addressed through numerosity reduction, a data reduction technique.

3.1.3.6 Heterogeneous Input Features

In order to facilitate the forecasting of the sales vector, which serves as the output feature of the machine learning model, it is preferable for the input features to be homogeneous with the output feature. Therefore, we will manually estimate the sales vector using estimation methods that involve reducing and transforming the dataset through a data transformation technique.

3.2 Data Preprocessing

In this section, we will look at the major steps involved in data preprocessing, namely, data cleaning, data integration, data reduction, and data transformation, in order to facilitate the forecasting of the sales vector.

3.2.1 Inventory Dynamics

Derived from the postulate of matter conservation, a medicine should not be disparate. Evidently, when a wholesaler purchases a medicine, this medicine should be stored in the inventory and then sold to a pharmacy. Based on that, it should pass through the distribution vector, the stock vector, and the sales vector. In other words, what the wholesaler purchases at the onset of the month, combined with what he has in his inventory at the beginning of the month, should be equal to what remains in the inventory at the end of the month, combined with what he sold during the month. In other terms, this could be expressed mathematically as follows:

$$x_{w,t}^i + s_{w,t}^i = s_{w,t+1}^i + v_{w,t}^i \quad (3.1)$$

3.2.2 Filtering Data

First of all, we filtered the data, keeping only the 32 pertinent wholesalers and the 154 medicines of interest, and eliminating the remaining ones.

3.2.3 Data Transformation

3.2.3.1 Converting Nominal to Numeric

Then, we converted the nominal values to numeric values. We assigned an identifier denoted as w to each wholesaler and replaced the wholesaler name column with the wholesaler identifier, where w ranges from 1 to 32. The correspondence between the wholesaler name and identifier was preserved in the wholesaler matrix, depicted in the right section of Figure 3.2.

Similarly, we replaced the medicine designation and code columns with an identifier i for each medicine, ranging from 1 to 154. The correspondence between the medicine name, code, and identifier was stored in the medicine matrix, as illustrated in the left section of Figure 3.2. Noteworthy is the fact that medicine identifiers were associated based on the sales tendency. For example, identifier $i = 1$ was assigned to the highest-selling medicine.

	A	B	C		A	B
1	DESIGNATION	CODE	ID Medicine	1	Wholesaler	ID Wholesaler
2			1	2		1
3			2	3		2
4			3	4		3
5			4	5		4
6			5	6		5
7			6	7		6
8			7	8		7
9			8	9		8
10			9	10		9
11			10	11		11
12			11	12		12
13			12	13		13
14			13	14		14
15			14	15		15
16			15	16		16
17			16	17		17
18			17	18		18
19			18	19		19
20			19	20		20
21			20	21		21
22			21	22		22
23			22			
24			23			

Figure 3.2: Screenshot of the wholesaler matrix (right) and the medicine matrix (left)

As for the 'DD/MM/YYYY' date format, we extracted the month $t = MM$, and calculated $\tilde{t}$ using the formula defined in 2.1.

3.2.3.2 Redundant Data Elimination

We then merged the measure and type of measure columns into one column, 'quantity', eliminating the measure of the value, as the quantity alone was entirely sufficient. The other measure could be derived from it by multiplying the quantity by the respective price.

We pivoted the type of operation into three columns: the stock vector, the sales vector, and the distribution vector, each encapsulating information from the quantity column.

We removed the lines with a "way of recording" column value equal to "mid month," as it indicated incomplete monthly statements, rendering the data unqualified for analysis.

Next, we removed the entire column, as it no longer served its purpose, featuring a uniform value of "onset of the month," for all lines.

3.2.4 Data Cleaning

Data cleaning routines attempt to fill in missing values, smooth out noise, while identifying outliers, and correct inconsistencies in the data. Our cleaning will be based on the inventory dynamics formula.

3.2.4.1 Handling Missing Values

Missing values in the stock vector, sales vector, and distribution vector columns were filled using the inventory dynamics formula defined in Equation (3.1), if and only if there was one missing value among the four quantities employed in the formula. Otherwise, remaining missing values were filled with zeros, as we assumed that these values referred to unoccurred purchase operations after consulting and understanding the data recording process.

3.2.4.2 Handling Noisy Data

Outliers were identified as values deviating by more than 10% from the inventory dynamics formula, and these samples were eliminated from the analysis.

3.2.5 Data Integration

3.2.5.1 Price Vector Integration

Subsequently, we added a new column containing the price of the respective medicine.

3.2.5.2 Monthly Objective Integration

We calculated the monthly objective using the budget constraint formula defined in Equation (2.2) and added the results in a new column.

3.2.5.3 Integrating the Sales Vector of the Last Month

Additionally, a new column containing the sales vector of the last month was appended, which would be useful for forecasting the sales vector.

3.2.6 Data Reformatting

Ultimately, the dataset obtained after applying these data preprocessing techniques is shown in Figure 3.3.

	A	B	C	D	E	F	G	H	I	J
1	ID Wholesaler	Generic Month	Month	ID Medicine	Stock Quantity	Sale Quantity	Distribution Quantity	UNIT PRICE	OBJECTIVE	Sale Quantity Last Month
2	1	13	1	2	26	1418	1800	2,789	112042	487
3	1	14	2	2	126	139	350	2,789	117993	1418
4	1	15	3	2	1301	567	1500	2,789	119292	139
5	1	16	4	2	787	1381	850	2,789	61981	567
6	1	17	5	2	103	499	0	2,789	7174	1381
7	1	18	6	2	0	920	600	2,789	97107	499
8	1	19	7	2	403	261	1200	2,789	96518	920
9	1	20	8	2	857	754	1000	2,789	90715	261
10	1	21	9	2	368	1055	600	2,789	96933	754
11	1	22	10	2	1044	575	1200	2,789	93401	1055
12	1	23	11	2	144	584	1500	2,789	109575	575
13	1	24	12	2	955	3050	2000	2,789	63914	584
14	1	25	1	2	2149	823	0	2,789	49878	3050
15	1	26	2	2	1616	495	0	2,789	57403	823
16	1	27	3	2	1049	635	0	2,789	57813	495
17	1	28	4	2	901	946	800	2,789	84592	635
18	1	29	5	2	806	616	500	2,789	68575	946
19	1	30	6	2	613	542	350	2,789	58060	616
20	1	31	7	2	0	643	0	2,789	32916	542
21	1	32	8	2	215	596	1000	2,789	93678	643

Figure 3.3: Screenshot of the dataset after applying preprocessing

3.3 Data Analysis

Although each feature of the obtained dataset, after applying preprocessing, carries significant and distinct information, applying the machine learning model fails to produce an accurate sales forecast vector. Specifically, our experimentation with the gradient boosting model revealed a mere 54.9% determination coefficient for the sales vector. This may be explained by the heterogeneity of the input features, as illustrated in Figure 3.4, highlighting the importance of each feature. Ideally, the input features should be homogeneous with the output feature. Therefore, there is a need for feature engineering, data analysis, and correlation analysis.

```
Entrée [20]: # Récupération des importances des features
feature_importances = model_gb.feature_importances_
print("In Forecasting the Sale Vector:")
# Affichage des importances associées à chaque feature (a, b, c)
for i, feature in enumerate(['ID Wholesaler', 'Generic Month', 'Month', 'ID Medicine', 'Stock Quantity', 'Sale Quantity', 'Distribution Quantity', 'UNIT PRICE', 'OBJECTIVE', 'Sale Quantity Last Month']):
    print(f" The Importance of {feature}: {feature_importances[i]*100}")
```

In Forecasting the Sale Vector:

The Importance of ID Wholesaler: 0.21989862617532593
The Importance of Generic Month: 1.4153147001056485
The Importance of Month: 1.3485890846942752
The Importance of ID Medicine: 8.468526710340086
The Importance of Stock Quantity: 25.771324746855107
The Importance of Distribution Quantity: 11.075767357041217
The Importance of UNIT PRICE: 0.09820434012508662
The Importance of OBJECTIVE: 0.2657528010149788
The Importance of Sale Quantity Last Month: 51.336621633648285

Figure 3.4: Importance of each feature for forecasting the sales vector

3.3.1 Correlation Analysis

Searching for the explanatory variables of the sales vector, we can evidently conclude from Figure 3.4 the importance of the sales vector of the last month, and the importance of both the stock vector and the distribution vector from Figure 3.5. This significance can be further augmented by applying a coefficient.

Although Figure 3.5 does not show a strong correlation between the sales vector and the objectives, the medicine, and the wholesaler, it is obvious and evident that there exists a correlation among these variables, and the key lies in rendering it homogeneous.

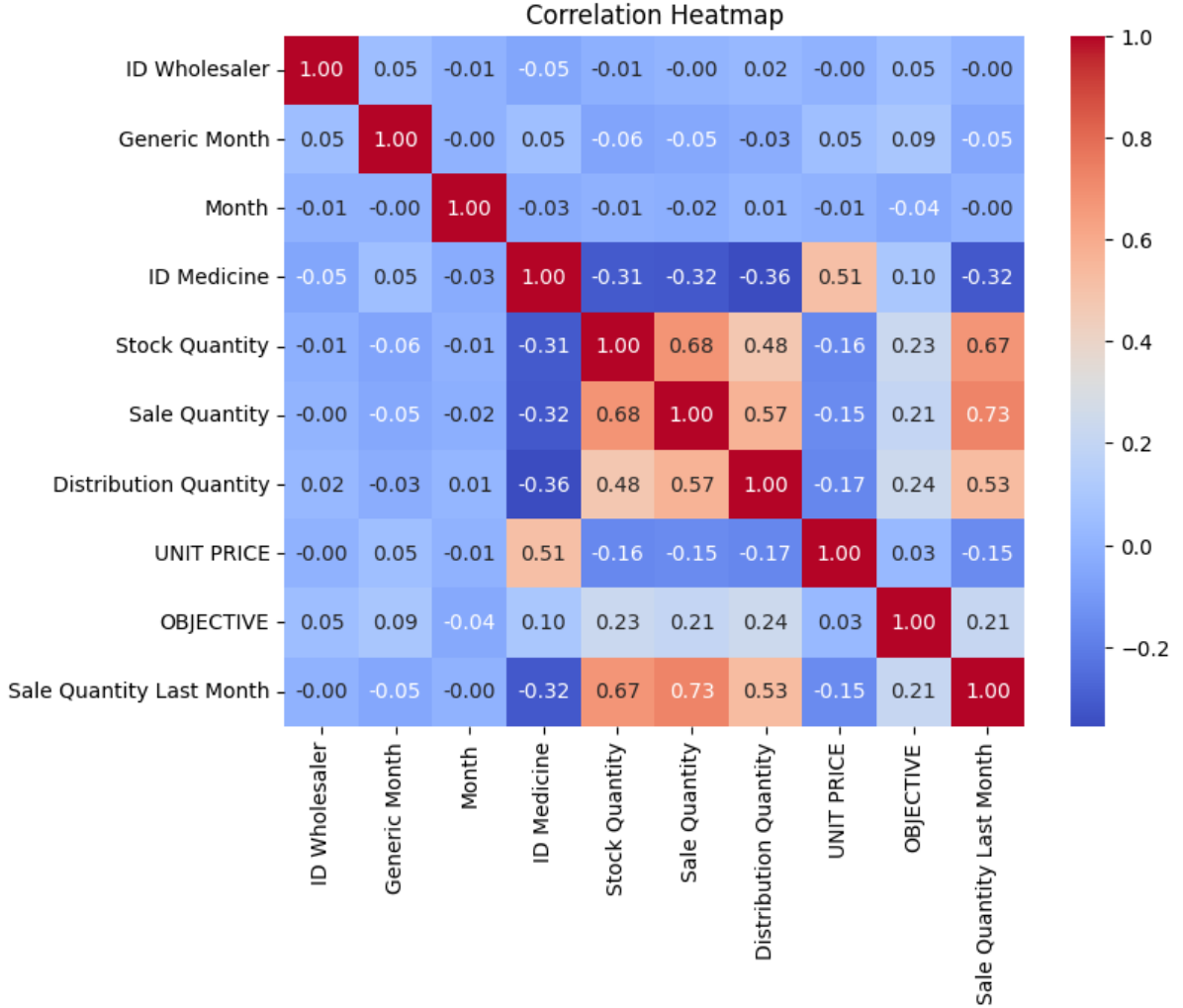


Figure 3.5: Correlation Heatmap

3.3.2 Feature Engineering

In this section, we will attempt to generate informative features that are homogeneous and highly correlated with the sales vector. We will utilize the large number of the existing features that are uncorrelated with the sales vector in order to uncover hidden patterns, improve model accuracy, and enhance the efficiency of machine learning algorithms. We propose four features that will encompass all the necessary information. Subsequently, we will eliminate any irrelevant features, if they exist.

$$\hat{v}_{w,t}^i = f(\hat{v}_{w,t}^{D,i}, \hat{v}_{w,t}^{T,i}, \hat{v}_{w,t}^{S,i}, \hat{v}_{w,t}^{G,i}) \quad (3.2)$$

3.3.2.1 Incrementation Matrix

Let $\kappa_{w,t} = (\kappa_{w,t}^1, \dots, \kappa_{w,t}^M) \in [0, 1]^M$ represent the incrementation vector, where each

element $\kappa_{w,t}^i$ signifies the average ratio between the sales vector of the current month t and that of the preceding month $t - 1$. In essence, multiplying this incrementation vector by the sales vector of the last month yields an approximate sales vector for the current month. Since our dataset spans the period between January 2019 and May 2023, $n_{max} = 3$. Formally, the incrementation matrix is denoted as $(\kappa_{w,t}^i)_{1 \leq i \leq M, 1 \leq w \leq 32, 1 \leq t \leq 12}$.

$$\kappa_{w,t}^i = \frac{1}{3} \cdot \sum_{n=1}^3 \frac{v_{w,\tilde{t}-12 \cdot n}^i}{v_{w,\tilde{t}-13 \cdot n}^i} \quad (3.3)$$

After applying this formula, we obtain the incrementation matrix shown in Figure 3.6.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	ID Medicine	Month	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2	ID Wholesaler	Month																		
3	1	1	5,31414	0,73592	1,09944	1,11687	0,57218	0,91409	1,20785	1,10236	1,14086	1,38751	1,62208	0,95726	0,85289	0,68145	1,18165	1,30036	0,82348	0,94
4	2	1	1,05798	1,45611	1,24189	1,67086	1,81763	2,1297	1,44587	1,85979	1,33559	1,46794	0,71506	0,52704	2,1336	1,79921	1,6093	1,89551	1,03261	0,96
5	3	1	1,80768	1,61611	1,49779	1,40471	1,35859	1,40851	1,55247	1,36199	1,26965	1,54292	1,34863	1,27439	1,26838	0,39103	1,41524	1,59031	1,13992	1,24
6	4	1	1,83976	2,46698	2,71744	2,25375	1,72965	2,18107	1,99776	1,94584	2,37301	2,47601	1,79931	1,49329	1,80295	0,33053	1,31838	2,96166	1,4374	1,83
7	5	1	1,86862	1,28263	1,27733	1,01046	1,46875	1,12843	1,1219	1,09649	1,02662	1,96238	1,08724	0,96039	0,88448	1,02479	1,23606	1,1082	0,93298	1,70
8	6	1	1,15491	1,32582	1,35304	1,32127	1,6283	1,79788	1,44685	1,42159	1,59405	1,13439	0,44248	1,70489	1,09595	0,14909	1,7566	1,76594	1,33374	2,42
9	7	1	1,84668	1,54009	1,28404	1,26932	1,05762	1,33044	1,2018	1,34632	1,20177	1,25839	1,2282	1,4187	1,13947	0,82413	1,0703	1,64401	1,05981	1,32
10	8	1	2,23405	0,80371	1,03387	1,51965	1,22135	1,84161	1,32942	1,50852	1,77692	1,0624	1,98293	1,94856	0,62484	1,6492	1,86168	1,68911	0,49464	1,95
11	9	1	1,70866	1,08053	1,13089	1,03284	1,21873	1,34513	1,08746	1,12527	0,64103	1,12391	1,08987	1,00746	1,08275	0,28368	1,25228	1,14632	0,98103	1,67
12	11	1	1,80768	0,96207	0,67314	0,87547	0,41812	0,64007	1,18039	0,78088	0,28617	0,79503	0,51883	0,5963	1,22381	1,1659	0,6087	1,16384	0,66094	0,88
13	12	1	1,80768	1,61611	1,49779	1,40471	1,35859	1,40851	1,55247	1,36199	1,26965	1,54292	1,34863	1,27439	1,26838	0,39103	1,41524	1,59031	1,13992	1,24
14	13	1	2,24018	1,04737	1,19925	1,0486	1,36623	0,87479	1,13557	1,06197	1,00379	1,02891	1,18256	1,13964	0,98378	1,35947	1,22549	1,19145	0,92323	1,95
15	14	1	1,80768	1,61611	1,49779	1,40471	1,35859	1,40851	1,55247	1,36199	1,26965	1,54292	1,34863	1,27439	1,26838	0,39103	1,41524	1,59031	1,13992	1,24
16	15	1	18,786	1,59006	1,5679	1,4527	1,19599	1,3983	1,18293	0,62034	0,69743	1,37378	1,52443	1,17478	1,14268	1,20603	1,02395	1,38523	1,50451	2,17
17	16	1	1,78774	1,51021	0,85789	1,46995	1,16396	1,4228	1,13252	1,12002	1,48202	1,20762	1,13781	1,25737	1,7016	1,15857	2,40695	1,8264	0,93746	1,81
18	17	1	1,80768	1,61611	1,49779	1,40471	1,35859	1,40851	1,55247	1,36199	1,26965	1,54292	1,34863	1,27439	1,26838	0,39103	1,41524	1,59031	1,13992	1,24
19	18	1	1,72542	1,18238	1,00835	0,80653	0,97406	0,8808	0,78186	0,84047	0,95097	1,08062	1,09803	1,18449	1,13137	1,11986	1,00943	1,06851	0,93959	0,95
20	19	1	1,80768	1,30209	1,32938	1,41104	2,05122	1,49667	1,38506	1,44689	1,44746	1,44073	1,43624	1,4044	1,51019	0,03049	1,76787	1,47909	1,1677	1,75
21	20	1	4,13918	1,07828	0,86797	0,97736	1,10335	1,08839	1,12828	1,07746	0,92665	0,98728	1,24137	1,03594	1,17555	1,03379	1,42591	1,06827	0,85562	1,32
22	21	1	1,80768	2,90686	2,19409	1,02665	1,18653	1,11581	1,30815	1,07899	0,88861	1,68715	1,48879	1,34048	1,13169	1,40264	0,60702	2,40678	0,62634	0,74
23	22	1	1,8725	1,46152	1,48019	0,88874	1,04283	0,87334	1,02319	1,34386	1,3078	0,97203	0,75481	1,01758	0,89647	1,43526	0,86272	1,00558	0,60701	0,17
24	23	1	1,80768	1,61611	1,49779	1,40471	1,35859	1,40851	1,55247	1,36199	1,26965	1,54292	1,34863	1,27439	1,26838	0,39103	1,41524	1,59031	1,13992	1,24
25	24	1	1,55707	1,28381	0,94847	0,90052	0,79782	1,0746	1,09367	0,79269	0,89797	0,99595	0,90651	0,64222	0,8929	0,92929	1,01267	0,98442	0,801	1,15

Figure 3.6: Screenshot of the incrementation matrix

3.3.2.2 Dynamically Predicted Sales Vector

In order to approximate the sales vector based on its correlation with the sales vector from the previous month, we introduce the dynamically predicted sales vector, denoted as $\hat{V}_{w,t}^D$, computed by multiplying the sales vector from the last month by the incrementation coefficient, as outlined below:

$$\hat{v}_{w,t}^{D,i} = \kappa_{w,t}^i \cdot v_{w,t-1}^i \quad (3.4)$$

3.3.2.3 Priority Matrix

Let $\rho_w = (\rho_w^1, \dots, \rho_w^M) \in [0, 1]^M$ represent the priority vector, where each element ρ_w^i signifies the ratio of the importance of medicine i to the wholesaler w , computed by dividing the quantity sold of all medicines by the quantity sold of medicine i by the wholesaler during the last 53 months recorded. Formally, the priority matrix is denoted as $(\rho_w^i)_{1 \leq i \leq M, 1 \leq w \leq 32}$.

$$\rho_w^i = \frac{\sum_{\tilde{t}=1}^{53} v_{w,\tilde{t}}^i}{\sum_{j=1}^{154} \sum_{\tilde{t}=1}^{53} v_{w,\tilde{t}}^j} \quad (3.5)$$

After applying this formula, we obtain the priority matrix shown in Figure 3.7.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ID Medicine / ID Wholesaler	1	2	3	4	5	6	7	8	9	10	11	12
2	1	0,116785411	0,038671841	0,029666333	0,035281403	0,032617879	0,024774519	0,023223769	0,027852247	0,030975693	0,030006474	0,021639187	0,01125391
3	2	0,129435813	0,036239735	0,029176821	0,039487366	0,044122744	0,016467397	0,033402834	0,030164756	0,025511638	0,037133516	0,019316578	0,00970877
4	3	0,130523635	0,040541605	0,043945067	0,032534956	0,035320762	0,035229257	0,033917692	0,028823938	0,018534758	0,023765769	0,022255943	0,03020667
5	4	0,116835848	0,03828723	0,033025292	0,037921646	0,030257384	0,023889398	0,029339085	0,029095569	0,024742635	0,0220875	0,016279663	0,02206515
6	5	0,158499139	0,041131878	0,039566839	0,031825806	0,035574419	0,0299348	0,027550169	0,023045421	0,021389538	0,019920686	0,023212412	0,02174288
7	6	0,105935722	0,045250487	0,044593204	0,034157666	0,044375084	0,034375786	0,02971545	0,025191659	0,022370429	0,024065389	0,017328238	0,02605390
8	7	0,094056286	0,035958487	0,032898302	0,038713628	0,032410381	0,025417599	0,027021517	0,029718198	0,021116252	0,02320884	0,016195072	0,02187245
9	8	0,115365545	0,058599955	0,035978565	0,043605902	0,028540246	0,025845482	0,031784357	0,024629759	0,032321385	0,029103067	0,026832151	0,0116467
10	9	0,16110346	0,036785375	0,037560997	0,031596414	0,040466432	0,033149927	0,025011434	0,024015825	0,018167157	0,019092713	0,022248625	0,0250655
11	10	0,062077595	0,062047443	0,04985936	0,027262193	0,029075653	0,031752766	0,028272303	0,021283378	0,023424207	0,013353263	0,017380779	0,02512136
12	11	0,062077595	0,062047443	0,04985936	0,027262193	0,029075653	0,031752766	0,028272303	0,021283378	0,023424207	0,013353263	0,017380779	0,02512136
13	12	0,085792148	0,045078272	0,038658559	0,038143153	0,037028806	0,040625611	0,032073348	0,031338856	0,019290663	0,02064774	0,017873693	0,02854585
14	13	0,108415908	0,044847931	0,044107745	0,037752637	0,024149482	0,027165835	0,028689146	0,028637055	0,028993949	0,020349651	0,019666484	0,01850395
15	14	0,062500621	0,079355368	0,073298783	0,042707963	0,021943947	0,032646761	0,027865456	0,031536354	0,027488037	0,022472334	0,019476818	0,02574396
16	15	0,152491077	0,034438341	0,034533809	0,030502775	0,033124586	0,035451443	0,023369739	0,025309025	0,018235835	0,022989292	0,020220717	0,02890261
17	16	0,165340947	0,063116756	0,033008994	0,034312597	0,016127207	0,028837335	0,024850417	0,017229805	0,034970253	0,026861765	0,025986191	0,01018878
18	17	0,06780104	0,041102855	0,037914583	0,043607515	0,037914583	0,028142961	0,032991432	0,031638571	0,023202576	0,026152445	0,030673472	0,01489005
19	18	0,074960457	0,072975764	0,059251525	0,036136732	0,017027678	0,031629792	0,030717885	0,02843309	0,024493095	0,014032075	0,018549844	0,0244807
20	19	0,062759963	0,05563632	0,053996736	0,046360992	0,022848158	0,032285058	0,032605464	0,031810514	0,034534547	0,022599345	0,024317564	0,02192567
21	20	0,070564826	0,055016349	0,048814759	0,043044375	0,030453214	0,032671648	0,032410776	0,028790795	0,030527127	0,023874383	0,022820665	0,01949096

Figure 3.7: Screenshot of the priority matrix

3.3.2.4 Traditionally Predicted Sales Vector

In order to approximate the sales vector based on its dependence on the objective, the wholesaler, and the medicine, we introduce the traditionally predicted sales vector, denoted as $\hat{V}_{w,t}^T$, computed by dividing the objective by the price of medicine i and multiplying it by the priority coefficient, as follows:

$$\hat{v}_{w,t}^{T,i} = O_{w,t} \cdot \rho_w^i \cdot \frac{1}{p^i} \quad (3.6)$$

3.3.2.5 Seasonal Trend Matrix

Let $\tau_t = (\tau_t^1, \dots, \tau_t^M) \in [0, 1]^M$ represent the seasonal trend vector, where each element τ_t^i signifies the percentage of the seasonal purchase trend of the i^{th} medicine during month t , independently of the wholesaler. This is computed by dividing the quantity purchased of all medicines at the onset of month t by all wholesalers, by the quantity purchased of medicine i by all wholesalers at the onset of month t , during the last 53 recorded months.

$$\tau_t^i = \frac{1}{4} \cdot \sum_{n=0}^3 \frac{\sum_{w=1}^{32} x_{w,\tilde{t}-12 \cdot n}^i}{\sum_{j=1}^{154} \sum_{w=1}^{32} x_{w,\tilde{t}-12 \cdot n}^j} = \sum_{w=1}^{32} \tau_{w,t}^i \quad (3.7)$$

where:

$$\tau_{w,t}^i = \frac{1}{4} \cdot \sum_{n=0}^3 \frac{x_{w,\tilde{t}-12 \cdot n}^i}{\sum_{j=1}^{154} x_{w,\tilde{t}-12 \cdot n}^j} \quad (3.8)$$

3.3.2.6 Sales Vector Predicted by Seasonal Trend

In order to approximate the sales vector based on its dependence on the objective, the seasonal patterns, and the medicine, we introduce the sales vector predicted by seasonal trend, denoted as $\hat{V}_{w,t}^S$, computed by dividing the objective by the price of medicine i and multiplying it by the seasonal trend coefficient, as follows:

$$\hat{v}_{w,t}^{S,i} = O_{w,t} \cdot \tau_t^i \cdot \frac{1}{p^i} \quad (3.9)$$

3.3.2.7 Global Trend Vector

Let $\tau = (\tau^1, \dots, \tau^M) \in [0, 1]^M$ represent the global trend vector, where each element τ^i signifies the percentage of the sales trend of the i^{th} medicine, independently of the wholesaler and the month. This ratio is computed by dividing the total quantity sold of all medicines across 53 recorded months by all wholesalers, by the quantity sold of medicine i across all recorded months by all wholesalers.

$$\tau^i = \frac{\sum_{\tilde{t}=1}^{53} \sum_{w=1}^{32} v_{w,\tilde{t}}^i}{\sum_{\tilde{t}=1}^{53} \sum_{j=1}^{154} \sum_{w=1}^{32} v_{w,\tilde{t}}^j} \quad (3.10)$$

3.3.2.8 Sales Vector Predicted by Global Trend

In order to approximate the sales vector based on its dependence on the objective, and the medicine, we introduce the sales vector predicted by global trend, denoted as $\hat{V}_{w,t}^G$, computed by dividing the objective by the price of medicine i and multiplying it by the respective global trend coefficient, as follows:

$$\hat{v}_{w,t}^{G,i} = O_{w,t} \cdot \tau^i \cdot \frac{1}{p^i} \quad (3.11)$$

3.3.3 Feature Selection

Consequently, we evaluate the proposed four features and identify the most relevant and significant ones, not only to improve the forecasting of the sales vector but also to reduce the dimensionality of the data, ultimately resulting in more accurate predictions.

```
Entrée [14]: # Récupération des importances des features
feature_importances = model_gb.feature_importances_
print("In Forecasting the Sale Vector:")
# Affichage des importances associées à chaque feature (α, β, γ)
for i, feature in enumerate(['Dynamically Predicted Sale Vector', 'Traditionally Predicted Sale Vector', 'Seasonally Predicted Sale Vector', 'Sales Vector Predicted by Global Trend']):
    print(f" The Importance of {feature}: {feature_importances[i]*100}")
```

In Forecasting the Sale Vector:
The Importance of Dynamically Predicted Sale Vector: 82.23227235545448
The Importance of Traditionally Predicted Sale Vector: 12.882075359389884
The Importance of Sales Vector Predicted by Seasonal Trend: 1.6578310038035837
The Importance of Sales Vector Predicted by Global Trend: 3.2278212813520444

Figure 3.8: Importance of each proposed feature for forecasting the sales vector

As Figure 3.8 illustrates, the key predictors for forecasting the sales vector are the dynamically predicted sales vector and the traditionally predicted sales vector. Thus, we keep only these two features and eliminate the rest, resulting the final dataset displayed in Figure 3.9. This dataset will be used for the machine learning model that will forecast the sales vector.

	A	B	C
	Dynamically Predicted Sale Vector	Traditionally Predicted Sale Vector	Sale Vector
1			
2	358	1554	1418
3	427	1636	139
4	102	1654	567
5	1028	859	1381
6	834	99	499
7	579	1346	920
8	514	1338	261
9	373	1258	754
10	780	1344	1055
11	762	1295	575
12	653	1519	584
13	856	886	3050
14	2245	692	823
15	248	796	495
16	364	802	635
17	1151	1173	946
18	572	951	616
19	715	805	542

Figure 3.9: Screenshot of the Final Dataset

Conclusion

In this chapter, we presented the dataset provided by Medicacom and explained the meaning of each attribute. Next, we identified the issues with the dataset, which arrived in a less-than-ideal state, and took strategic steps for data preprocessing. This involved employing techniques such as data cleaning, data transformation, data integration, and data reformatting. We established the inventory dynamics formula that guided our data cleaning process. Subsequently, we analyzed the data, uncovering hidden connections and identifying correlations between features, such as wholesaler behavior and seasonal patterns. We transformed the features into homogeneous ones using the sales vector, proposing four methods to approximate it. We evaluated these four features and identified the most relevant and significant ones to create our final dataset. This final dataset will be used in the next chapter which will detail the development process for the sales forecasting machine learning model.

Chapter 4

Sales Forecasting Model

Introduction

In this chapter, our focus centers on the development of a sales forecasting machine learning model capable of generating a forecast sales vector $\hat{V}_{w,t}$. First, we will specify its characteristics so it can be applicable to the final preprocessed dataset. Then, we will identify the type of supervised machine learning algorithm and demonstrate that our model should definitely be a regression algorithm. Next, we will limit and narrow down our choice to commonly used machine learning models for regression tasks, such as Random Forest Regressor, Gradient Boosting Regressor, Support Vector Regressor, and Neural Networks. We will then train each model on the preprocessed dataset and compare their evaluation metrics, such as the coefficient of determination R^2 , in order to choose the top-performing and best-fit model for our case. Afterward, we will present the chosen model and explain its algorithm. Following that, we will tune its hyperparameters by empirically testing the grid search to further enhance its performance, which we will discuss later. Once the creation of our sales forecasting machine learning model is complete, it will be prepared to determine the unknown components in the objective distribution problem statement. We will enumerate the steps followed to forecast the sales vector and approximate the threshold vector, the remaining two unknown vectors of the ODP. In fact, the work that we will explain in this chapter is **the key to completely stating the objective distribution problem** where all its components are known, allowing us to finally solve the ODP.

4.1 Model Specification

The final dataset obtained, as shown in Figure 3.9 and comprising 73,009 samples of reliable and representative data, will serve as the input for the sales forecasting model, where the sales vector $V_{w,t}$ functions as the dependent variable, while the dynamically predicted sales vector $\hat{V}_{w,t}^D$ and the traditionally predicted sales vector $\hat{V}_{w,t}^T$ act as predictors. In fact, the model should be a supervised one since the values of the outcome variable

are present in the dataset, and it should be constructed using the following approach:

$$\hat{V}_{w,t} = f(\hat{V}_{w,t}^D, \hat{V}_{w,t}^T) \quad (4.1)$$

We randomly divided the dataset into an 80% training set and a 20% test set. The samples in the training dataset will be utilized to fit the model parameters, while those in the test dataset will be used for an unbiased evaluation of the final model, which was trained on the training data.[15]

4.2 Model Selection

4.2.1 Regression

A regression model provides a function that describes the relationship between one or more independent variables and a dependent, or target variable.[13]

In a regression model, which is applicable to our case, the relationship between the dependent variable $V_{w,t}$ and the independent variables $\hat{V}_{w,t}^D, \hat{V}_{w,t}^T$ is represented by an equation of the form:

$$V_{w,t} = f(\hat{V}_{w,t}^D, \hat{V}_{w,t}^T) + \epsilon \quad (4.2)$$

Where:

- $V_{w,t}$ is the dependent variable (the quantity we want to predict).
- $\hat{V}_{w,t}^D, \hat{V}_{w,t}^T$ are independent variables (predictor variables).
- f represents the functional relationship between the dependent and independent variables.
- ϵ represents the error term, which accounts for the variability in $V_{w,t}$ that cannot be explained by the predictor variables.

The goal of a regression model is to estimate the parameters of the function f in such a way that the predicted values $\hat{V}_{w,t}$ are as close as possible to the actual values of $V_{w,t}$. The process of estimating these parameters typically involves minimizing the difference between the observed $V_{w,t}$ values and the predicted values using a suitable criterion, such as minimizing the sum of squared errors or maximizing the likelihood function.[12]

4.2.2 Possible Models

There are various machine learning models that can be applied to solve a regression problem. From this rich array of possibilities, we narrowed down our choice to commonly used machine learning models for regression tasks, including Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, AdaBoostRegressor, CatBoostRegressor, XGBoost Regressor, LightGBM, Support Vector Regressor, K-Nearest Neighbors Regressor, and Neural Networks. Each model is trained using the same training dataset and then tested with the test dataset. Next, we will select the best-performing model among them based on the best evaluation metrics.

4.2.3 Model Comparison

After training each model with the training set and testing it with the test set, we calculated the evaluation metrics for each model, including the mean absolute error (MAE), the root mean squared error ($RMSE$), and the coefficient of determination R^2 , as detailed in the table below:

Table 4.1: Model Comparison

Model	Evaluation Metric		
	MAE	$RMSE$	R^2
Linear Regression		193878.31	0.6958
Random Forest Regressor		159275.79	0.7501
Support Vector Regressor		271369.73	0.5423
K-Nearest Neighbors Regressor		172588.14	0.7148
Gradient Boosting Regressor	116.72	364.31	0.7917
AdaBoostRegressor	391.16	936.46	-0.3758
CatBoostRegressor	116.10	460.95	0.6666
XGBoost Regressor	116.52	416.90	0.7273
LightGBM	116.16	431.63	0.7077
Neural Networks (3 layers: 64, 32 and 1)	111.90	350.73	0.8060

The coefficient of determination, denoted as R^2 , is a statistical measure used to assess the proportion of the variance in the dependent variable that is predictable from the independent variables in a regression model. In simpler terms, it indicates how well the independent variables explain the variability of the dependent variable in a regression analysis.[2]

R^2 ranges from 0 to 1, where:

$$\begin{cases} R^2 = 0 : \text{The predictors do not explain any variability in the dependent variable.} \\ R^2 = 1 : \text{The predictors explain 100\% of the variability in the dependent variable.} \end{cases}$$

Therefore, the criterion for choosing the model is logically the one with the highest coefficient of determination R^2 . By applying this criterion, we find that the best model is the neural network with $R^2 = 80.60\%$, followed by the gradient boosting regressor with $R^2 = 79.17\%$. However, considering that the training phase in neural networks takes around 4 minutes, and in gradient boosting regressor takes around 5 seconds, and acknowledging that the coefficient of determination can be further enhanced by hyperparameter tuning, we have evidently chosen the gradient boosting regressor as the model for sales forecasting.

4.2.4 Gradient Boosting Regressor

In essence, gradient boosting regressor (GBR) is a popular machine learning algorithm used for both regression and classification tasks. It is an ensemble learning method that

combines the predictions of several base estimators (typically decision trees) to create a strong predictive model. Gradient boosting regressor builds trees sequentially, where each new tree aims to correct the errors made by the combined existing set of trees.[5]

The algorithm unfolds as follows:

1. **Initialization:** The algorithm starts with an initial prediction, often the mean of the target values for regression problems. This initial prediction is used as the first base estimator.
2. **Sequential Training:** It trains a series of weak learners (usually shallow decision trees) in a sequential manner. Each tree is trained to correct the errors of the combined predictions made by the existing ensemble of trees. The new tree focuses on the samples where the previous predictions were inaccurate.
3. **Weighted Combination:** The predictions from each new tree are combined with the predictions from the existing ensemble. Predictions from new trees are scaled down (using a learning rate) to prevent overfitting and to make the algorithm more robust.
4. **Iterative Process:** The process of adding new trees and adjusting the predictions is repeated for a specified number of iterations or until a certain level of accuracy is achieved.
5. **Final Prediction:** The final prediction is made by combining the predictions of all the trees in the ensemble.

Gradient boosting regressor has several hyperparameters that can be tuned to optimize its performance, such as the number of trees (`n_estimators`), the learning rate (`learning_rate`), the maximum depth of the trees (`max_depth`), and the minimum number of samples required to split an internal node (`min_samples_split`), among others.[5]

Advantages of gradient boosting regressor include its ability to handle complex relationships in the data and its resistance to overfitting, especially when proper hyperparameters are tuned.[5]

4.3 Hyperparameter Tuning

Hyperparameter tuning involves finding the best set of hyperparameters for a machine learning algorithm to optimize its performance. Hyperparameters are external configurations that are set prior to training a model and are not learned from the data, unlike model parameters. Proper tuning of hyperparameters can significantly improve a model's accuracy and generalization on unseen data, and can prevent overfitting (when the model performs well on training data but poorly on unseen data) and underfitting (when the model is too simplistic to capture the underlying patterns in the data).[6]

4.3.1 Empirical Test

To enhance the coefficient of determination R^2 , we empirically tried the grid search method in order to find the optimal hyperparameters. In this method, we specified a predefined set of hyperparameters each time, and the algorithm evaluated the model's performance for all possible combinations of these hyperparameters. While grid search is exhaustive, it was computationally expensive and took a considerable amount of time, requiring many minutes. Figure 4.1 illustrates an example of the grid search application and its empirical test results.

```
Entrée [ ]: # Identifying parameters
            parameters = {'subsample': np.arange(0.83, 0.86, 0.001)}
            # Model initialization
            gbt = GradientBoostingRegressor()
            # Gridsearch
            grid = GridSearchCV(gbt, parameters)
            # Coaching
            grid.fit(X_train, y_train)
            # results
            grid.best_estimator_

Out[233]: GradientBoostingRegressor(subsample=0.856)
```

Figure 4.1: Example of the grid search application

The final results of the empirical grid search test are as follows: `learning_rate=0.0955`, `max_depth=3`, `n_estimators=55`.

4.3.2 Model Performance

By applying the optimal hyperparameters we found, we observed an improvement in the coefficient of determination R^2 of the GBR, increasing from $R^2 = 79.17\%$ to $R^2 = 80.61\%$, as shown in the following figure:

```

Entrée [14]: from sklearn.ensemble import GradientBoostingRegressor
              from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

              # Création du modèle Gradient Boosting
              model_gb = GradientBoostingRegressor(learning_rate=0.0955, max_depth=3, n_estimators=55)

              # Entraînement du modèle sur l'ensemble d'entraînement
              model_gb.fit(X_train, y_train)

              # Prédiction sur l'ensemble de test
              y_pred_gb = model_gb.predict(X_test)

              # Calcul des métriques d'évaluation
              mae_gb = mean_absolute_error(y_test, y_pred_gb)
              rmse_gb = np.sqrt(mean_squared_error(y_test, y_pred_gb))
              r2_gb = r2_score(y_test, y_pred_gb)

              # Affichage des métriques
              print("Mean Absolute Error (MAE) with Gradient Boosting :", mae_gb)
              print("Root Mean Squared Error (RMSE) with Gradient Boosting :", rmse_gb)
              print("Coefficient of determination (R²) with Gradient Boosting :", r2_gb)

Mean Absolute Error (MAE) with Gradient Boosting : 114.50795459287681
Root Mean Squared Error (RMSE) with Gradient Boosting : 351.4920461602506
Coefficient of determination (R²) with Gradient Boosting : 0.8061730930408677

```

Figure 4.2: Gradient Boosting Regressor Performance

It is noteworthy that 80.61% of the variability in the dependent variable is explained by the independent variables included in the GBR model, which represents the best result we found among all the models we tested. The remaining 19.39% of the variability is unexplained and could be due to other factors not considered in the model or inherent randomness in the data

4.4 Finalizing the Objective Distribution Problem Statement

After creating a sales forecasting model, we can now finalize the objective distribution problem statement that we defined in Equation (2.8), wherein we had three unknown vectors: $\tau_{w,t}$, $\hat{V}_{w,t}$, and $\hat{H}_{w,t}$, in addition to the distribution vector $X_{w,t}^*$ that we are searching for. The first unknown vector, denoted as the trend vector $\tau_{w,t}$, was computed in Equation (3.8) when we analyzed historical sales data. As for the other two vectors, we will approximate them using the GBR model recently developed.

4.4.1 Sales Vector Forecasting

Forecasting the sales vector $\hat{V}_{w,t}$ involves four simple steps, as follows:

1. **Coefficients Upload:** Upload the incrementation matrix $(\kappa_{w,t}^i)_{1 \leq i \leq M, 1 \leq w \leq 32, 1 \leq t \leq 12}$, the priority matrix $(\rho_w^i)_{1 \leq i \leq 154, 1 \leq w \leq 32}$, and the price vector $(p^i)_{1 \leq i \leq 154}$.
2. **Data Input:** Enter two key pieces of information, specifically the sales vector of the last month $V_{w,t-1}$ and the current objective $O_{w,t}$.
3. **Input Preprocessing:** Compute the dynamically predicted sales vector $\hat{V}_{w,t}^D$, and

the traditionally predicted sales vector $\hat{V}_{w,t}^T$ using the following formulas:

$$\begin{cases} \hat{v}_{w,t}^{D,i} = \kappa_{w,t}^i \cdot v_{w,t-1}^i \\ \hat{v}_{w,t}^{T,i} = O_{w,t} \cdot \rho_w^i \cdot \frac{1}{p^i} \end{cases} \quad (4.3)$$

4. **Gradient Boosting Regressor Application:** Execute the GBR model on the dynamically predicted sales vector $\hat{V}_{w,t}^D$ and the traditionally predicted sales vector $\hat{V}_{w,t}^T$, resulting in the forecast sales vector, $\hat{V}_{w,t}$, which is obtained as follows:¹

$$\hat{V}_{w,t} = \lfloor f_{GBR}(\hat{v}_{w,t}^{D,i}, \hat{v}_{w,t}^{T,i}) \rfloor \quad (4.4)$$

4.4.2 Threshold Vector Approximation

Approximating the threshold vector $\hat{H}_{w,t}$ involves three simple steps, following the forecasting of the sales vector process:

5. **Input Preprocessing:** Normally, matrices are uploaded, the objective is entered, and the sales vector is forecasted. Compute the dynamically predicted sales vector of the next month $\hat{V}_{w,t+1}^D$ and the traditionally predicted sales vector of the next month $\hat{V}_{w,t+1}^T$ using the following formulas:

$$\begin{cases} \hat{v}_{w,t+1}^{D,i} = \kappa_{w,t+1}^i \cdot \hat{v}_{w,t}^i \\ \hat{v}_{w,t+1}^{T,i} = O_{w,t} \cdot \rho_w^i \cdot \frac{1}{p^i} = \hat{v}_{w,t}^{T,i} \end{cases} \quad (4.5)$$

6. **Forecasting the Sales Vector of the Next Month:** Execute the GBR model on the dynamically predicted sales vector of the next month $\hat{V}_{w,t+1}^D$ and the traditionally predicted sales vector $\hat{V}_{w,t+1}^T$, resulting in the forecast sales vector of the next month, $\hat{V}_{w,t+1}$, which is obtained as follows:

$$\hat{V}_{w,t+1} = \lfloor f_{GBR}(\hat{v}_{w,t+1}^{D,i}, \hat{v}_{w,t+1}^{T,i}) \rfloor \quad (4.6)$$

7. **Approximating the Threshold Vector:** Each element $\hat{h}_{w,t}^i$ of the approximate threshold vector signifies the approximate maximum quantity of the i^{th} medicine beyond which, if the stored quantity of the i^{th} medicine in the inventory of wholesaler w , at the onset of month t , surpasses it, this medicine will remain dormant for more than 2 months. Hence, the threshold vector $\hat{H}_{w,t}$ is approximated using the following formula:

$$\hat{H}_{w,t} = \hat{V}_{w,t} + (1 + \mu) \cdot \hat{V}_{w,t+1} \quad (4.7)$$

where μ denotes a safety margin, typically of the order of 5%, to compensate for the errors introduced by the predictions of the GBR model.

¹ $\lfloor \cdot \rfloor$ denotes the entire part

4.4.3 Complete Problem Statement

In this stage, we know all the vectors employed in the objective distribution problem, except, of course, the distribution vector $X_{w,t}^*$ that we are searching for. One way to completely state the objective distribution problem is as follows, where all its components are known:

Find the optimal distribution vector $X_{w,t}^*$ in order to:

$$\text{maximize } \sum_{i=1}^M (x_{w,t}^i \cdot p^i) \cdot \tau_{w,t}^i \quad (4.8)$$

subject to constraints:

$$\begin{cases} \sum_{i=1}^M x_{w,t}^i \cdot p^i = O_{w,t} \\ s_{w,t}^i + x_{w,t}^i \geq \hat{v}_{w,t}^i, \quad \forall i \in \llbracket 1; M \rrbracket \\ s_{w,t}^i + x_{w,t}^i \leq \hat{h}_{w,t}^i, \quad \forall i \in \llbracket 1; M \rrbracket \\ x_{w,t}^i \in \mathbb{N}, \quad \forall i \in \llbracket 1; M \rrbracket \end{cases} \quad (4.9)$$

where:

- the trend vector $(\tau_{w,t}^i)_{1 \leq i \leq M}$ is computed using Equation (3.8)
- the sales vector $(\hat{v}_{w,t}^i)_{1 \leq i \leq M}$ is forecasted using Equation (4.4)
- the threshold vector $(\hat{h}_{w,t}^i)_{1 \leq i \leq M}$ is approximated using Equation (4.7)
- the objective $O_{w,t}$ and the stock vector $(s_{w,t}^i)_{1 \leq i \leq M}$ are provided by the industry
- the price vector $(p^i)_{1 \leq i \leq M}$ is sourced from the database

Conclusion

In this chapter, we developed a machine learning model for sales forecasting, capable of generating a forecast sales vector $\hat{V}_{w,t}$. We started by specifying its characteristics to suit the final preprocessed dataset. Then we identified its type as a regression model. Next, we narrowed down the choice of the model to commonly used machine learning models for regression tasks. We tested each one on our dataset and compared their results along with their evaluation metrics, such as the coefficient of determination R^2 . Consequently, we selected the Gradient Boosting Regressor as the top-performing and best-fit model for our case. Subsequently, we defined it and presented its algorithm. Afterward, we fine-tuned its hyperparameters through empirical testing using grid search to further enhance its performance. We then discussed the results, deeming them acceptable. Upon completion of our sales forecasting machine learning model, it became ready to determine the unknown components in the ODP statement. We enumerated the steps followed to forecast the sales vector and approximate the threshold vector, the remaining two unknown vectors of the ODP. Finally, we fully stated the objective distribution problem where all its components are known, paving the way for the next chapter to finally solve the ODP.

Chapter 5

Solving the Objective Distribution Problem

Introduction

In this chapter, we will focus on solving the objective distribution problem using an operational research approach. To achieve this, we will identify the type of the ODP as a resource allocation problem. We will then define the bounded knapsack problem (BKP), which is a resource allocation problem applicable to our context, with numerous algorithms capable of solving it. Next, we will convert the ODP to a BKP by making the necessary changes and introducing the intermediate variables for the translation. To establish a clear correspondence between the variables of the ODP and the BKP, we construct an analogy table. With this groundwork in place, we will state the objective distribution problem in the framework of a bounded knapsack problem, defining all its components. Subsequently, we will solve the BKP using exact methods such as dynamic programming and approximate methods such as metaheuristic algorithms. We will compare these methods. By doing so, we will **finally solve the objective distribution problem and respond to the wholesaler and the industry's needs** by providing them with the optimal distribution vector $X_{w,t}^*$ that prevents understocking and overstocking and guarantees the best objective distribution.

5.1 Bounded Knapsack Problem

The knapsack problem is the following problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine which items to include in the collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.[7]

There are several variations of the problem, one of which is the Bounded Knapsack Problem. In this version, a knapsack of capacity C^1 should be filled by use of n given item

¹To avoid any confusion between the quantities defined in the ODP problem and those defined in the BKP problem, each entity defined in BKP shall be denoted as a letter unused by the ODP.

types, where type j has profit or value ϑ_j , weight ω_j , and a bound m_j on the availability. The problem is to select a number χ_j ($0 < \chi_j < m_j$) of each item type such that the profit sum of the included items is maximized without having the weight sum to exceed C . [1]

Formally, the Bounded Knapsack Problem can be defined as follows:

1. **Input:**

- n items, where each item j has a weight ω_j , a value ϑ_j , and a maximum quantity m_j .
- Knapsack capacity C

2. **Objective:** Maximize the total value of items in the knapsack without exceeding the total weight capacity.

3. **Constraints:** The total weight of selected items cannot exceed the knapsack capacity, and for each item, the quantity selected cannot exceed the maximum quantity available.

The BKP may thus be defined as the following optimization problem:

$$\begin{aligned}
 & \text{maximize} && \sum_{j=1}^n \vartheta_j \cdot \chi_j \\
 & \text{subject to:} && \sum_{j=1}^n \omega_j \cdot \chi_j \leq C, \\
 & && \chi_j \in \{0, 1, \dots, m_j\}, \quad j = 1, 2, \dots, n
 \end{aligned} \tag{5.1}$$

where all coefficients are positive integers.

Solving the BKP can be challenging, especially when the input size is large, as it is considered **NP**-hard. There are dynamic programming-based approaches that can be used to solve this problem, which involve constructing a table to store intermediate solutions and iteratively filling it based on subproblems. [1]

5.2 Converting the ODP to a BKP

Our ODP is of type resource allocation problem. To address this, we aim to transform it into a well-established resource allocation problem for which solutions already exist. In order to facilitate the resolution of the ODP, we will convert it to a BKP, as the BKP already contains numerous algorithms capable of solving it.

5.2.1 Simplification by the Understocking Constraint

Given the problem statement in Equation (4.8), the first step of the resolution becomes evident through the understocking constraint. In fact, the minimal distribution vector $X_{w,t}^{min}$ should precisely be equal² to the forecast sales vector $(\hat{v}_{w,t}^i)_{1 \leq i \leq M}$ so that the wholesaler could respond to the demand, as defined by the understocking condition. Thus, the optimal distribution vector $X_{w,t}^*$ is determined as follows:

$$X_{w,t}^* = X_{w,t}^{min} + X_{w,t}^+ \quad (5.2)$$

where $X_{w,t}^+ = (x_{w,t}^{+,1}, \dots, x_{w,t}^{+,M}) \in \mathbb{N}^M$ denotes the complementary distribution vector that we will search for using the BKP, and where:

$$x_{w,t}^{min,i} = \hat{v}_{w,t}^i, \text{ for } i = 1, 2, \dots, M \quad (5.3)$$

5.2.2 Modification on the Overstocking Constraint

By analogy, using the overstocking constraint, we define the maximum distribution vector $X_{w,t}^{max}$ as follows:

$$x_{w,t}^{max,i} = \hat{h}_{w,t}^i - s_{w,t}^i, \text{ for } i = 1, 2, \dots, M \quad (5.4)$$

This leads to the following constraint:

$$x_{w,t}^{min,i} \leq x_{w,t}^{*,i} \leq x_{w,t}^{max,i}, \text{ for } i = 1, 2, \dots, M \quad (5.5)$$

or still can be expressed as:

$$\hat{v}_{w,t}^i \leq x_{w,t}^{*,i} \leq \hat{h}_{w,t}^i - s_{w,t}^i, \text{ for } i = 1, 2, \dots, M \quad (5.6)$$

or another constraint pertains to the complementary distribution vector:

$$0 \leq x_{w,t}^{+,i} \leq \hat{h}_{w,t}^i - s_{w,t}^i - \hat{v}_{w,t}^i, \text{ for } i = 1, 2, \dots, M \quad (5.7)$$

5.2.3 Bound Vector

Let $\beta_{w,t} = (\beta_{w,t}^1, \dots, \beta_{w,t}^M) \in \mathbb{Z}^M$ represent the bound vector, where each element $\beta_{w,t}^i$ signifies the maximum quantity of the i^{th} medicine that the complementary distribution vector can attain. This can be expressed as follows:

$$\beta_{w,t}^i = \hat{h}_{w,t}^i - s_{w,t}^i - \hat{v}_{w,t}^i, \text{ for } i = 1, 2, \dots, M \quad (5.8)$$

²If : $\sum_{i=1}^M x_{w,t}^{min,i} \cdot p^i \leq O_{w,t}$

5.2.4 Analogy Between ODP and BKP

Once the problem statement is simplified through the understocking constraint, we can easily transform the objective distribution problem into a bounded knapsack problem using the following table:

Table 5.1: Analogy Between ODP and BKP

Bounded Knapsack Problem		Objective Distribution Problem	
<i>Meaning</i>	<i>Symbol</i>	<i>Meaning</i>	<i>Symbol</i>
Item	j	Medicine	i
Number of Items	n	Number of Medicines	M
Knapsack Capacity	C	Residual Objective	$O_{w,t}^{res} = O_{w,t} - \sum_{i=1}^M x_{w,t}^{min,i} \cdot p^i$
Weight	ω_j	Price	p^i
Value	ϑ_j	Value Vector	$\nu_{w,t}^i = p^i \cdot \tau_{w,t}^i$
Bound	m_j	Bound Vector	$\beta_{w,t}^i = \hat{h}_{w,t}^i - s_{w,t}^i - \hat{v}_{w,t}^i$
Number of copies	χ_j	Comp. Distribution Vector	$x_{w,t}^{+,i} = x_{w,t}^{*,i} - \hat{v}_{w,t}^i$

5.2.5 ODP Statement in the Framework of a BKP

Having established the analogy table, we are now able to state the objective distribution problem in the format of a bounded knapsack problem. One way to concretize that is as follows:

Find the complementary distribution vector $X_{w,t}^+$ that:

$$\begin{aligned}
 & \text{maximize} && \sum_{i=1}^M (p^i \cdot \tau_{w,t}^i) \cdot x_{w,t}^{+,i} \\
 & \text{subject to:} && \sum_{i=1}^M p^i \cdot x_{w,t}^{+,i} \leq O_{w,t}^{res}, \\
 & && x_{w,t}^{+,i} \in \{0, 1, \dots, \beta_{w,t}^i\}, \quad i = 1, 2, \dots, M
 \end{aligned} \tag{5.9}$$

where:

- the trend vector $(\tau_{w,t}^i)_{1 \leq i \leq M}$ is computed using Equation (3.8)
- the bound vector $(\beta_{w,t}^i)_{1 \leq i \leq M}$ is computed using Equation (5.8)
- the sales vector³ $(\hat{v}_{w,t}^i)_{1 \leq i \leq M}$ is forecasted using Equation (4.4)
- the threshold vector⁴ $(\hat{h}_{w,t}^i)_{1 \leq i \leq M}$ is approximated using Equation (4.7)
- the stock vector⁵ $(s_{w,t}^i)_{1 \leq i \leq M}$ is provided by the industry
- the residual objective is computed as follows: $O_{w,t}^{res} = O_{w,t} - \sum_{i=1}^M \hat{v}_{w,t}^i \cdot p^i$
- the price vector $(p^i)_{1 \leq i \leq M}$ is sourced from the database

³used in calculation of the bound vector

⁴used in calculation of the bound vector

⁵used in calculation of the bound vector

And after solving the BKP by finding the complementary distribution vector $X_{w,t}^+$, we will conclude using the following formula, which is derived from Equation (5.2), the optimal distribution vector $X_{w,t}^*$:

$$X_{w,t}^* = X_{w,t}^+ + \hat{V}_{w,t} \quad (5.10)$$

5.3 Solving the Bounded Knapsack Problem

Various methods are known for solving the BKP, ranging from exact methods that provide optimal solutions to heuristic approaches that offer approximate solutions. Among these methods are both exact and approximate techniques. The most well-known method for solving the BKP is the dynamic programming approach, which provides an exact solution.

5.3.1 Dynamic Programming

Dynamic programming breaks down the problem into smaller subproblems, solving them recursively and storing the solutions in a table to avoid redundant computations. Specifically, Dynamic programming algorithms for the BKP involve the construction of a multidimensional table, where each cell represents a subproblem of the original problem. The table is systematically filled based on the relationships between subproblems, allowing for the computation of optimal solutions iteratively. By leveraging the principle of optimality, these algorithms avoid redundant computations and significantly improve the efficiency of the solution process.[4] Although this algorithm provides an optimal solution, it is simple to code, as shown below:

```

1 def knapsack_bounded(values, weights, bounds, k, i=0, lookup=None):
2     if lookup is None:
3         lookup = {}
4
5     if (i, k) in lookup:
6         return lookup[(i, k)]
7
8     if i == len(values):
9         return 0, [0] * len(values)
10
11     max_value = 0
12     max_counts = [0] * len(values)
13
14     for count in range(bounds[i] + 1):
15         if weights[i] * count <= k:
16             value, counts = knapsack_bounded(values, weights, bounds, k
17 - weights[i] * count, i + 1, lookup)
18             current_value = count * values[i] + value
19             if current_value > max_value:
20                 max_value = current_value
21                 max_counts = [0] * len(values)
22                 max_counts[i] = count
23                 max_counts[i + 1:] = counts[i + 1:]
24
25     lookup[(i, k)] = max_value, max_counts

```

```
25 return max_value, max_counts
```

Listing 5.1: Knapsack Bounded Algorithm

This algorithm, like other exact algorithms, is designed to provide optimal solutions by exhaustively exploring all possible combinations of items within the weight constraint. These exact methods guarantee finding the best solution, but they can be computationally expensive, especially for large instances of the problem.[1]

In other words, executing this dynamic programming algorithm on our case, with a time complexity of $O(O_{w,t}^{res} \cdot M \cdot (\beta_{w,t}^i)_{max})$, blocked our laptop for more than an hour. Therefore, we need a metaheuristic method.

In fact, the choice of method depends on the problem size, the required level of optimality, and the computational resources available. Exact methods are preferred for small problem sizes, while metaheuristic methods are preferred for larger problem sizes or when an approximate solution is acceptable. Also, a set of relevant methods depends on the selected formulation of the KP.[1]

5.3.2 Metaheuristic Algorithm

Metaheuristic approaches offer approximate solutions by employing smart strategies to navigate the solution space more efficiently. Metaheuristics are particularly valuable when dealing with large-scale instances of the BKP, where finding an optimal solution within a reasonable time frame becomes challenging. Although metaheuristic methods do not guarantee optimality, they often yield satisfactory solutions in practice.[10]

Drawing inspiration from the dynamic programming algorithm, we have developed a formula as shown below, that approximately calculates the complementary distribution vector $X_{w,t}^+$:

This methodology, accumulated after empirical tests, has been proven to effectively calculate the complementary distribution vector $X_{w,t}^+$.

Once the BKP is solved by finding the complementary distribution vector $X_{w,t}^+$, we can easily conclude, using Equation (5.2), the optimal distribution vector $X_{w,t}^*$, the solution of the ODP.

Conclusion

In this chapter, we solved the objective distribution problem using an operational research approach. First, we identified the type of the ODP as a resource allocation problem and then chose a well-known resource allocation model applicable to our context, with various predefined algorithms to solve it. We defined the bounded knapsack problem. Next, we converted the ODP to a BKP by implementing essential modifications and introducing intermediate variables for the translational process. To establish a clear correspondence between the variables of the ODP and the BKP, we constructed an analogy table. Then, we stated the objective distribution problem in the framework of a bounded knapsack problem. Afterward, we solved the BKP using exact methods such as dynamic programming and approximate methods such as metaheuristic algorithms. We chose the metaheuristic algorithm for its rapidity and acceptance of results. Ultimately, we arrived at the optimal solution for the objective distribution problem, proposing the optimal distribution vector $X_{w,t}^*$ and fulfilling the precise requirements of the wholesaler and the industry. The next chapter will detail how the solution was implemented in the notebook and provide instructions for using the implemented solution.

Chapter 6

Implementation and Validation

Introduction

In this chapter, we will explain the process to be followed for using the implemented solution and clarify the input and output from the user's perspective. We will define the input data and the specific format to follow explicitly. Then, we will provide detailed instructions for using the notebook. This step aims to **make our work easily usable by Medicacom**. Afterward, we will test the model using a randomly selected sample. Subsequently, we will comment on the generated results solely in terms of the comparison between the forecast sales vector and the sales vector, as the optimal distribution vector is a decision made by the wholesaler.

6.1 Notebook and Matrices

As shown in the sequence diagram (Figure 2.2), the industry inputs the wholesaler's name, the month t , the sales vector from the last month $V_{w,t-1}$, the stock vector $S_{w,t}$, and the objective $O_{w,t}$. They attend to receive an optimal distribution vector and, optionally, the forecast sales vector, as specified in the project requirements. This is how we build our model, and it's also how the notebook functions: it takes the input variables and produces the output variables. We implemented our solution in a Python notebook on Google Colab and shared it with Medicacom, alongside the necessary matrices for the notebook to function.

Within the model integration process, specific matrices are necessitated, which we used throughout the resolution, namely the wholesaler matrix, the medicine matrix, the price vector, the preprocessed dataset, the trend matrix, the priority matrix, and the incrementation matrix. These matrices are essential for the notebook to function, and we have calculated and formatted them appropriately in CSV files, ready to be uploaded to the notebook.

It is noteworthy that the notebook functions each time one sample is taken to generate only one optimal distribution vector. A sample by definition consists of a wholesaler

and a month. This means that every time we select a particular wholesaler and a specific month, we input their corresponding data to ultimately receive the outcomes.

6.2 Instructions for Using the Notebook

To ensure the proper functioning of the notebook and the model, the following steps must be carefully followed¹:

1. Select a sample, a specific wholesaler w , and a particular month t , where an optimal objective distribution of medicines needs to be determined, meeting demand requirements while avoiding both overstock and understock situations.
2. Create an Excel file to enter the sample coordinates following this format:
 - Include a column titled “Medicine Code”, housing the medicine codes
 - Integrate a column named “Sales vector of the last month”, $V_{w,t-1}$, capturing the quantities of medicines sold by wholesaler w in the previous month $t - 1$
 - Encompass a column labeled “Stock vector of the current month”, $S_{w,t}$, representing the quantities of medicines present in the inventory of wholesaler w at the onset of month t

Save this Excel file in CSV format with semicolons as separators.

3. Import the seven constant files, namely wholesalerMatrix, medicineMatrix, preprocessedDataset, priorityMatrix, incrementationMatrix, trendMatrix and priceVector, along with the sample file.
4. In the “INPUT: Sample Coordinates” section, update the variables: wholesaler_name, month t , objective $O_{w,t}$, and sample_path, as illustrated in Figure 6.1.
5. Click “Run”, followed by “Restart and run everything.”
6. In the downloaded Excel file, detailed estimations for each product will be provided, indicating both the forecast quantity to be sold by the wholesaler w during month t and the recommended quantity for purchase.

Throughout the implementation process, error exceptions have been integrated to assist users in identifying the specific cause of errors and pinpointing the step where the protocol is not being respected.

¹It’s noteworthy that this notebook is provided to Medicacom data scientists and not to the industry. For the industry, it should be user-friendly.

INPUT: Sample Coordinates

```

Entrée [2]: '''
Voici la liste des grossistes :

...

Entrée [3]: # Choix du grossiste
wholesaler_name = "          " # Remplacez ceci par le nom du grossiste choisi

# Choix du mois (de 1 à 12)
month = 7 # Remplacez ceci par le mois choisi

# Mettre à jour l'objectif de ce mois en Dinar
objective = 100000 # en TND

# Chemin du fichier CSV contenant les ventes du dernier mois et le stock au début de ce mois de l'échantillon
# Il est impératif de suivre ce format : Code Produit, Vente dernier mois, Stock au début de ce mois
sample_path = "/content/ECHANTILLON G1 M7.csv" # Remplacez ceci par le chemin du fichier de l'échantillon

```

Figure 6.1: Input of the Sample Coordinates

6.3 Testing the Implemented Solution

In order to test and validate the implemented solution, we chose a sample of the wholesaler ' ' in the month of 'July 2022'.

6.3.1 Input

As Figure 6.1 illustrates, we inputted the data as follows:

```

wholesaler_name = " "
month = 7
objective = 100000 # en TND
sample_path = "/content/ECHANTILLON G1 M7.csv"

```

And we created a CSV file, shown in Figure 6.2, containing the “Sales vector of the last month” $V_{w,t-1}$ and the “Stock vector of the current month” $S_{w,t}$.

	A	B	C
1	Stock vector of the current month	Sales vector of the last month	Medicine Code
2	763	171	
3	67	20	
4	69	23	
5	273	107	
6	4	50	
7	62	85	
8	365	380	
9	32	73	
10	197	245	
11	73	38	
12	168	93	
13	547	2414	
14	0	0	
15	26	676	
16	757	878	
17	398	1250	
18	39	67	
19	8	-8	

Figure 6.2: Screenshot of the Input Data of the Sample

6.3.2 Output

After executing the model on the input data of the selected sample, which lasts approximately about 10 seconds, we obtain the model output in an Excel file downloaded systematically, as represented in Figure 6.3.

In Figure 6.3, the optimal distribution vector $X_{w,t}^*$ is referred to as the “Quantity to purchase proposed”, while the forecast sales vector $\hat{V}_{w,t}$ is labeled as the “Quantity to sell this month”.

	A	B	C	D
	Medicine Designation	Medicine Code	Quantity to sell this month	Quantity to purchase proposed
1				
2			3375	7481
3			786	383
4			832	1257
5			686	489
6			723	1626
7			81	201
8			174	102
9			597	451
10			201	179
11			557	389
12			441	837
13			205	129
14			293	350
15			469	180
16			55	7
17			235	314
18			205	129
19			39	7
20			209	338
21			186	264
22			78	188
23			462	693
24			206	34
25			145	34
26			163	99
27			90	3

Figure 6.3: Screenshot of the Model Output

6.3.3 Discussion

To gain insight into the performance and realism of the output, we compare the forecast sales vector $\hat{V}_{w,t}$, referred to in Figure 6.3 as the “Quantity to sell this month” with the actual sales vector $V_{w,t}$.

In Figure 6.4, we present the forecast sales vector and the actual sales vector together to visualize the difference. We calculated the coefficient of determination, R^2 , between the forecast sales vector and the actual sales vector, which was 78.53%. This value is entirely acceptable and sufficient for making decisions regarding the distribution vector.

	A	B	C	D	
1	Medicine Designation	Medicine Code	Quantity to sell this month	Sales vector	Quantity to p
2			3375	2499	
3			786	652	
4			832	609	
5			686	564	
6			723	773	
7			81	362	
8			174	352	
9			597	316	
10			201	195	
11			557	518	
12			441	430	
13			205	99	
14			293	283	
15			469	262	
16			55	47	
17			235	212	
18			205	128	
19			39	124	
20			209	87	
21			186	147	
22			78	163	
23			462	0	
24			206	312	
25			145	91	

Figure 6.4: Comparing the forecast sales vector and the sales vector

Conclusion

In this chapter, we have explained the process to be followed for using the implemented solution and have clarified the input and output from the user's perspective. We defined the input data and the specific format to be followed explicitly. Then, we provided detailed instructions for using the notebook. Afterward, we tested the model using a randomly selected sample. Subsequently, we commented on the generated results solely in terms of the comparison between the forecast sales vector and the sales vector, as the optimal distribution vector is a decision made by the wholesaler. We concluded that the result is entirely acceptable and sufficient for making decisions regarding the distribution vector, as our main goal is to propose an optimal distribution vector, not to forecast the sales vector.

General Conclusion and Perspectives

Due to demand fluctuations, inventory management intricacies, and resource allocation dilemmas being major challenges in the healthcare sector, various studies have been conducted, including those utilizing statistics, estimation, and machine learning approaches. In our report, we employ these approaches alongside our accumulated knowledge in data analysis, data preprocessing, operational research, and optimization, to follow a strategic plan, initially formulating the objective distribution problem mathematically and subsequently solving it using a metaheuristic approach by projecting it onto another resource allocation problem—the bounded knapsack problem. This process involves creating a sales forecast machine learning model, specifically the gradient boosting regressor, which allowed us to completely define the objective distribution problem.

In simpler terms, we have successfully developed a model that takes the monthly statement of a specific wholesaler and a particular month as input, then proposes the optimal objective distribution vector to the wholesaler, preventing both understocking and overstocking and ensuring the best objective distribution to meet pharmacy demands.

A significant portion of this engineering internship was dedicated to mathematically reformulating the problem and preprocessing the dataset, driven by our belief in the engineer’s fundamental task of modeling before solving and our adherence to the postulate of ‘Garbage In, Garbage Out.’ Beyond enhancing our expertise, this internship allowed us to refine our scientific rigor, critical thinking, open-mindedness, and adaptability—qualities essential for an engineer.

The next stage will be to integrate the developed model, implemented in a Python notebook, into a user-friendly platform. This will enable industry professionals to use it without the need for expertise in data science.

Furthermore, future research will focus on integrating the upcoming new database into the training set of the gradient boosting regressor to enhance its predictions further, since the current dataset comprises recorded medicine purchasing operations spanning the period between January 2019 and May 2023, which is evidently biased due to the effects of Covid-19.

References

- [1] *Bounded Knapsack Problem*. Accessed: 13/07/2023. URL: <https://ceur-ws.org/Vol-3403/paper21.pdf>.
- [2] *Coefficient of Determination*. Accessed: 17/10/2023. URL: https://en.wikipedia.org/wiki/Coefficient_of_determination.
- [3] *Data Mining*. Accessed: 15/10/2023. URL: <https://urlz.fr/ogWC>.
- [4] *Dynamic Programming*. Accessed: 13/07/2023. URL: <https://itnext.io/identify-knapsack-pattern-solve-with-dynamic-programming-b673bed3324f?gi=e4c03ce5b332>.
- [5] *Gradient Boosting*. Accessed: 14/08/2023. URL: https://en.wikipedia.org/wiki/Gradient_boosting.
- [6] *Hyperparameter Tuning*. Accessed: 14/08/2023. URL: https://en.wikipedia.org/wiki/Hyperparameter_optimization.
- [7] *Knapsack Problem*. Accessed: 13/07/2023. URL: https://en.wikipedia.org/wiki/Knapsack_problem.
- [8] *Machine Learning and Data Preparation*. Accessed: 05/07/2023. URL: <https://www.yumpu.com/fr/document/read/67482157/apprendre-le-machine-learning-en-une-semaine-guillaume-saint-cirgue-z-liborg/37>.
- [9] *Mediacom Presentation*. Accessed: 21/09/2023. URL: www.medicacom.tn/en/#presentation.
- [10] *Metaheuristic Approaches*. Accessed: 13/07/2023. URL: <https://urlz.fr/ojl8>.
- [11] *Overstocking Understocking*. Accessed: 12/10/2023. URL: <https://zupan.ai/blog/what-is-overstocking>.
- [12] *Regression Analysis*. Accessed: 17/10/2023. URL: https://en.wikipedia.org/wiki/Regression_analysis.
- [13] *Regression Model*. Accessed: 17/10/2023. URL: <https://www.ims1.com/blog/what-is-regression-model>.
- [14] *Sequence Diagram*. Accessed: 13/10/2023. URL: <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/>.
- [15] *Training and Test Data Sets*. Accessed: 17/10/2023. URL: https://en.wikipedia.org/wiki/Training,_validation,_and_test_data_sets.
- [16] *Understanding Understocking*. Accessed: 12/10/2023. URL: <https://zupan.ai/blog/what-is-understocking>.