



Rethinking markdown decisions in electronics retail

João Pedro Ramos Moucho

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Vera Lucia Miguéis Oliveira e Silva

Co-Supervisor: Prof. Eduardo Luís de Meireles e Oliveira

August 21, 2022

Abstract

In retail, inventory is often bought in advance, with long lead times, and based on poor initial demand predictions. Such situations may result in lower sales than expected and unsold stock, ultimately leading to a loss in revenue. In these circumstances, retailers often resort to price markdowns, i.e., permanent price reductions, to generate more demand for a product and clear its remaining stock before it becomes obsolete. To make these decisions, retailers often rely on rules of thumb or standardized approaches that lack any analytical foundation.

This dissertation arises in the context of a project with a large Portuguese retailer in the consumer electronics industry. It aims to improve their decision-making regarding markdown decisions and life cycle management of end-of-life products. Understanding how a price change will affect demand for a product is critical when deciding which price markdown should be applied. We call this relationship between price and demand Price Elasticity of Demand (PED).

Literature shows that a successful analytical approach to a markdown optimization (MDO) system comprises a sales forecasting model, an MDO algorithm and performance metrics to measure the system's effectiveness. With this in mind, we developed a solution encompassing a PED estimation model that predicts the impact that different price markdowns have on products' sales, followed by an MDO algorithm that recommends the optimal price for each product at each store.

The PED estimation model recurs to a supervised learning method that uses historical information to compute the expected sales for different discount levels. The optimal discount is suggested based on these predictions, according to the retailer's targets for margin and time until clearance.

These components are integrated into a Decision Support System (DSS), comprising an interface in which the retailer's managers have access to all the information needed to make the best possible decisions and where they validate the price suggestions.

This solution adds to the scarcity of studies regarding MDO concerning consumer electronics retail. Additionally, different from the classic revenue maximization problem applied to fixed selling seasons, we propose an intuitive approach to MDO that considers both clearance and margin objectives, directly defined by the retailer.

We ran a pilot test on four categories to validate the developed solution, resulting in higher stock clearance rates than in the previous year. Most price suggestions were accepted, and the retailer's commercial managers unanimously approved the developed solution.

Resumo

No comércio a retalho, o inventário é frequentemente comprado com grande antecedência, longos prazos de entrega e com base em fracas previsões de procura. Tais situações podem resultar em vendas mais baixas do que o esperado e em inventário não vendido, levando, consequentemente, a uma perda de receitas. Nestas circunstâncias, os retalhistas recorrem frequentemente a *markdowns*, ou seja, reduções de preço permanentes, para gerar um aumento na procura de um produto e limpar o stock remanescente antes que se torne obsoleto. Para tomar estas decisões, os retalhistas recorrem frequentemente a regras simples ou a abordagens *standard* que carecem de qualquer fundamento analítico.

Esta dissertação surge no contexto de um projeto com um grande retalhista português na indústria dos produtos eletrónicos. O seu objetivo é melhorar as suas decisões relativamente a *markdowns* e à gestão do ciclo de vida de artigos descontinuados. Ao decidir que redução de preço deve ser aplicada, é fundamental compreender de que forma essa alteração vai afetar a procura do produto. A esta relação entre preço e procura chamamos Elasticidade Preço-Procure (PED).

A literatura revista demonstra que uma abordagem analítica bem sucedida para um sistema de optimização de *markdowns* (MDO) inclui um modelo de previsão de vendas, um algoritmo de MDO e métricas de desempenho para medir a eficácia do sistema. Posto isto, desenvolvemos uma solução que consiste num modelo de estimativa da PED que prevê o impacto que diferentes descontos têm nas vendas dos produtos, seguido de um algoritmo de MDO que recomenda o preço ótimo para cada artigo em cada loja.

O modelo de estimativa da PED recorre a um algoritmo de aprendizagem supervisionada que utiliza informação histórica para calcular as vendas esperadas para diferentes patamares de desconto. O desconto ótimo é sugerido com base nestas previsões, de acordo com os objetivos pretendidos pelo retalhista para a margem e escoamento do stock.

Estes componentes são integrados num Sistema de Apoio à Decisão (DSS), que contém uma interface na qual os gestores do retalhista validam as sugestões de preço dos produtos e onde têm acesso a toda a informação necessária para tomar as melhores decisões possíveis.

Esta solução vem juntar-se à escassez de estudos de MDO relativos ao retalho de produtos eletrónicos. Além disso, ao contrário do problema clássico de maximização de receitas aplicado a janelas temporais fixas, propõe uma abordagem intuitiva a MDO que considera não só o objetivo de maximizar a margem como também o de acelerar o escoamento, cujos alvos são directamente definidos pelo retalhista.

Foi realizado um teste piloto em quatro categorias para validar a solução desenvolvida, resultando em taxas de escoamento mais elevadas do que as do período homólogo do ano anterior. A maioria das sugestões de preço propostas foram aceites, e os gestores comerciais do retalhista aprovaram de forma unânime a solução desenvolvida.

Acknowledgements

This dissertation concludes my Masters in Electrical and Computer Engineering and marks the end of a 5-year chapter of my life. It was not always an easy path, and it had some obstacles and less positive moments, but it contributed tremendously to my personal growth and gave me the necessary knowledge, skills and tools for my professional life.

This journey would not have been possible without the guidance and company of several people, who I sincerely thank:

To my supervisors, Professor Vera Miguéis and Professor Eduardo Oliveira, for accepting to counsel me through this final chapter and for all the wise advice, words of encouragement and confidence in my abilities.

To LTPlabs and all my colleagues for the warm welcome and for giving me the chance to be involved in such an interesting and challenging project.

To Nuno Pedro for working alongside me and guiding me on this project. Thank you for all your patience, availability and friendship.

To my brother and parents for all the support throughout this path, without which it would not have been possible.

To Catarina, for always being there, in the ups and downs, and for keeping me cheered up and motivated throughout this challenging journey.

Muito obrigado,

João Pedro Moucho

*“Tell me and I forget,
Teach me and I may remember,
Involve me and I learn”*

Benjamin Franklin

Contents

1	Introduction	1
1.1	Context	1
1.2	Purpose	2
1.3	Document Structure	3
2	Background Knowledge & Literature Review	5
2.1	Demand	5
2.1.1	Factors that influence demand	6
2.1.2	Price Elasticity of Demand	7
2.1.3	Cross-Price Elasticity of Demand	8
2.1.4	Sales lift	9
2.2	Forecasting	9
2.2.1	Types of forecasting methods	10
2.2.2	Time Series methods	10
2.2.3	Causal methods	11
2.2.4	Machine Learning	12
2.2.5	Performance Metrics	15
2.3	Markdown Optimization System	15
2.3.1	Modelling Assumptions in Markdown Pricing Problems	17
2.3.2	Related Work	18
2.3.3	Markdown Optimization Problem Formulation	22
3	Problem Statement	25
3.1	Retailer Overview	25
3.2	Current Process	26
3.3	Goals and Requirements	27
3.4	Proposed Solution	28
4	Methodology	31
4.1	CRISP-DM	31
4.2	Decision Support System Overview	32
4.3	Data Analysis	34
4.4	PED Estimation Model Development	36
4.4.1	Baseline Sales Forecasting	36
4.4.2	Feature Selection	37
4.4.3	Modelling Approach	38
4.5	Markdown Optimization	39
4.5.1	Price Markdown Response	40

4.5.2	Price Markdown Recommendation	41
4.6	The Stock Clearance Tool	46
5	Results	51
5.1	Model Performance	51
5.2	Pilot Testing	52
5.2.1	Price Recommendation	53
5.2.2	Stock Clearance Results	56
6	Conclusion	57
6.1	Summary	57
6.2	Main Contributions	58
6.3	Future Work	59
A	The Stock Clearance Tool	61
	References	65

List of Figures

2.1	Types of forecasting methods	10
2.2	Illustration of a Decision Tree	14
2.3	Illustration of a typical product life-cycle pattern	16
2.4	Illustration of the impact of a price markdown on sales variation	17
3.1	The retailer's product hierarchy	25
3.2	The process currently used by the retailer	27
3.3	Proposed solution for the new markdown pricing process	29
3.4	Representation of the PED Estimation Model	29
3.5	Optimal price suggestion according to the clearance and margin targets	30
4.1	Illustration of the CRISP-DM cycle	32
4.2	Decision Support System Diagram	33
4.3	Determining the limiting levels of the price markdown suggestion	43
4.4	Example of price suggestion in situation A	44
4.5	Example of price suggestion in situation B	44
4.6	Example of price suggestion in situation C	45
4.7	Stock Clearance Tool's home page	46
4.8	Parameter setting at the Stock Clearance Tool	47
4.9	SKU-level Configuration page	47
4.10	Model outputs at the Store-level configuration page	48
4.11	Model suggestion and What-If analysis at the Discount tool screen	49
5.1	Stock cleared (Latest campaign vs. Last year's homologous period campaign)	56
A.1	Product and store attributes at the Store-level Configuration page	61
A.2	Product attributes at the SKU-level Configuration page	62
A.3	Model outputs and price validation at the Stock Clearance Tool	63
A.4	Discount tool screen at the Stock Clearance Tool	64

List of Tables

2.1	Interpretation of the Price Elasticity of Demand coefficient	7
2.2	Summary of the analyzed studies	19
3.1	Products' provision type according to their stock rotation	26
4.1	Optimal markdown selection	43
5.1	Performance metric comparison	52
5.2	Number of SKUs, per category, included in the campaign	53
5.3	Number of SKUs, per provision type, included in the campaign	53
5.4	Percentage amount of SKUs with accepted price suggestions	54
5.5	Percentage amount of product-store combinations with accepted price suggestions	54
5.6	Average discount off, before and after the price validation	55
5.7	Price suggestion acceptance rates for the next stock clearance campaign	55

Abbreviations and Symbols

CRISP-DM	Cross Industry Standard Process for Data Mining
DT	Decision Tree
DSS	Decision Support System
GBM	Gradient Boosting Machines
LR	Linear Regression
MAD	Mean Absolute Deviation
MAPE	Mean Absolute Percentage Error
MDO	Markdown Optimization
ML	Machine Learning
MPE	Mean Percentage Error
MSE	Mean Square Error
PED	Price Elasticity of Demand
PLC	Product Life-Cycle
RF	Random Forests
RSME	Root Mean Square Error
SCT	Stock Clearance Tool
SKU	Stock Keeping Unit
WTP	Willingness to Pay
XED	Cross-Price Elasticity of Demand

Chapter 1

Introduction

This chapter introduces the reader to the central topics of the dissertation and to its main objectives. Section 1.1 explains the context under which the dissertation arises. Section 1.2 describes the problem and the primary goals of this work. Section 1.3 details the structure of the rest of the document.

1.1 Context

Effective product life-cycle (PLC) management is becoming increasingly necessary for retailers to remain competitive. As competition increases, sales margins decline, putting more pressure on daily operations. Therefore, PLC estimation and better decision-making regarding product pricing can be critical aspects of leveraging a retailer's performance.

The PLC corresponds to the sales pattern followed by a product throughout its life and is seen as a base for product control and planning. It is typically divided into four stages: introduction, growth, maturity and decline [1]. However, the conventional PLC pattern is hardly ever respected, and its stages are not always clearly identifiable.

The competitive nature of the retail sales world, the need to satisfy the ever-changing customer expectations and the constant technological advances result in many innovative solutions on the market and high price volatility. These particularly affect the high-tech field, where products with the latest technology are being brought into the market more quickly, ultimately leading to shorter PLCs [2]. Along with that, many high-tech companies face a situation of long lead times [3]. For that reason, retailers are often forced to purchase inventory months in advance and based on poor initial demand predictions, potentially leading to lower sales than expected and an inventory surplus, ultimately resulting in a loss in revenue.

In the attempt to reduce their losses, retailers apply price markdowns, which are permanent reductions in the price of a product, aiming to generate more demand. Effective price markdowns can mean a profit in the hundreds of millions for major retail chains. Thus, they can be the difference between a profitable and an unprofitable year [4]. For these reasons, pricing is one of the biggest challenges retailers face.

Pricing and revenue optimization need a consistent and reliable process of decision making, with constant monitoring and performance evaluation [5]. Until recently, retailers have primarily based their pricing decisions on subjective rules that they believe work reasonably [6]. However, in the last decades, the adoption of dynamic pricing strategies and the availability of advanced decision-support tools have been increasing across industries [7]. These tools can be highly beneficial in pricing decisions, namely in markdown pricing of end-of-life products. In the correct timing and magnitude, markdowns can generate high demand levels and minimize wasted opportunities of making a profit.

These solutions might be particularly beneficial for retail chains with multiple locations and channels, which require high inventory coordination between stores and have more decisions to be made. Many retailers understand that it would be advantageous for them to define clearance prices at the Store-level, to account for variations between stores. However, they often give up on doing so because of the complex and time-consuming nature of the decision [4]. A large number of stores, each with a large number of items, ultimately result in a tremendous amount of markdown decisions to be made. For this reason, a computer-based markdown pricing algorithm is required to automate markdown pricing processes and apply decisions according to each store's specific needs [4].

In this context, the project in this dissertation intends to apply these principles to a Portuguese retail chain in the electronics industry that lacks analytical support in PLC planning and markdown-related decisions. The dissertation focuses on the portion of the project that addresses the development of a Decision Support System (DSS) to help the retailer in the process of decision-making regarding clearance pricing.

1.2 Purpose

The main objective of this work is to improve the retailer's markdown pricing process for end-of-life products.

To achieve this, we propose a methodology to estimate the impact of price reductions on products' demand, i.e. the products' price elasticity of demand (PED), and suggest to the retailer the prices that should be applied.

The methodology includes the development of an innovative large-scale attribute-based PED estimation model that encompasses all the relevant business factors and is capable of defining products' expected markdown response for each price discount. The optimal price should then be selected and recommended to the retailer based on these predictions. The suggested price should consider the retailer's objective and markdown strategy. For each category, the retailer may define the intended margin and stock clearance targets. The price suggestion algorithm will consider these targets and find a balance between maximizing sales margins and clearing the inventory as fast as possible, in a trade-off between these two drivers.

The PED estimation model and the price markdown optimization algorithm will then be integrated into a DSS, which comprises an interface where the price suggestions will be presented

to the retailer for each product at each store. With the proposed solution, we expect to help the retailer improve its pricing processes and enable them to make better-supported decisions with added efficiency.

1.3 Document Structure

The dissertation is divided into five more chapters. Chapter 2 aims to fill the reader with core concepts that are important for the understanding of the dissertation and provides a literature review on markdown optimization. Chapter 3 describes the problem we focus on in this dissertation. We provide an overview of the retailer and the main issues in its current markdown pricing process, followed by a description of the requirements and the proposed solution. Chapter 4 details the implementation steps and the components of the developed solution. Chapter 5 presents and analyzes the results obtained from the developed solution. Finally, Chapter 6 concludes the document, summarizing the developed work and its main contributions and highlighting possible future improvements.

Chapter 2

Background Knowledge & Literature Review

This chapter describes the core concepts necessary for the understanding of this dissertation. Section 2.1 introduces the concept of demand and the factors that can impact its behaviour. It also describes the concept of price elasticity of demand, its derivatives and the sales lift. Section 2.2 provides an introduction to forecasting and its methods and presents commonly used metrics to evaluate forecasts' performance. Section 2.3 addresses the topic of markdown optimization and provides a review of relevant case studies that relate to the problem at hand.

2.1 Demand

Retailers are, by definition, intermediaries between suppliers and customers. They add value to both while extracting profit in the process. Retailers help suppliers by finding a market for their goods and help customers by enabling them to purchase the products they need and desire [8].

Therefore, a retailer's success is defined by how well it is able to fulfill the wants and needs of its potential customers. There has been an emergence in the adoption of analytical tools, with many retailers harnessing the power of computing to help project demand and satisfy it [8]. Demand Forecasting, often used interchangeably with Sales Forecasting, is, therefore, a crucial element in PLC planning and management.

Demand is described as the relationship between price and the quantity of a particular good or service that consumers are able and willing to purchase at that price [9]. The maximum value a consumer is willing to pay for a specific good or service is referred to as willingness to pay (WTP) [10].

As will be further described in Section 2.1.1, demand can be influenced by a wide variety of factors, making it highly unpredictable. Price is one of these factors. A rise in the price of a product most likely leads to a decrease in the quantity demanded. In contrast, a price reduction increases demand. To this inverse relationship between price and demand, we call Law of Demand

[11]. Due to this relationship, price is seen as a significant factor in shaping demand and is often integrated into Demand Forecasting models.

However, there are a few exceptions to the Law of Demand: Giffen goods and Veblen goods. Giffen goods are non-luxury products, typically a necessity, with few substitutes (e.g. bread, rice, wheat), whose demand rises when their price rises and vice-versa, defying the Law of Demand. Especially in poor communities, when there is an increase in price for a Giffen good, consumers have less to spend on other more expensive complementary goods. Thus, consumers become even more reliant on Giffen goods, which are basic necessities, increasing demand. Contrarily, Veblen goods are luxury products whose price is associated with high quality and high social status. Thus, higher prices make these products more desirable, increasing demand [12].

2.1.1 Factors that influence demand

Demand Forecasting can be performed at several aggregation levels (e.g. market-level, chain-level, product-level, store-level). In retail, operational decisions, such as pricing and inventory management, are directly related to demand forecasts and, therefore, these are performed at lower aggregation levels. The ability to accurately estimate demand for each product at each store is essential to a retail chain's growth and survival [13].

Multiple factors can affect demand and, consequently, the forecast. Identifying these factors and estimating their impact on demand is key for understanding and predicting demand behaviour. This section explores some of the most common factors known to influence demand, including [13]:

- **Product:** one of the most influential factors of a product's demand is, naturally, the product itself. That is, the product's features, characteristics, appearance and what it offers to the customer. For example, in [14], for a set of computer products, some of the identified key features that affect demand are the computer's processor, hard drive size and RAM. Therefore, a product's characteristics are naturally related to its demand, as they can be the deciding factor in a customer's choice. Additionally, brand awareness, online reviews, trends, sales history, stock levels and the stage of the PLC a product is at can also highly influence its demand.
- **Seasonality and Events:** many products are influenced by seasonality. Seasonality corresponds to a variation of demand that occurs at specific and regular time intervals and tends to repeat itself periodically, typically throughout a year, but also weekly and monthly. Seasonality can make forecasting more complex and less accurate and may vary from item to item. Therefore, its influence should not be neglected. Seasons, vacations, and seasonal promotions are examples of situations that impact demand. Holidays and the proximity of some sporadic occurrences such as special events and festivals can also influence demand behaviour. Some of these events may be location specific and affect stores differently.

- **Price:** as mentioned previously in this chapter, price is directly related to demand. In fact, demand highly depends on how customers react to price changes. This concept is called price elasticity of demand (PED) and will be further addressed in Section 2.1.2. For psychological reasons, retailers often use certain pricing rules or predetermined strategical prices, specifically designed to induce consumers into making a purchase (e.g. prices ending with ".99"). Price of substitute products, type of promotion, time since the last discount and the effects of marketing and competition are other factors that can influence demand.
- **Store:** several store-related aspects may also have an impact on demand. Factors such as location, local economy, and local competition can play a crucial role in product sales. Understanding the demographics in the area of a specific store is also vital for effective planning. The authors in [15] found that demographic variables, such as age distribution, level of education, income, family size and ethnicity, are much more influential than competition variables in estimating store-specific PED. The size of the store, its appearance, layout, product placement and the existence of substitute and complementary products in the store can all affect an item's sales.

2.1.2 Price Elasticity of Demand

Price Elasticity of Demand (PED) corresponds to the percentage change in quantity demanded produced by the respective percentage change in price [9], as shown in Equation 2.1.

$$\epsilon_p = \frac{\Delta_{\%} \text{ quantity}}{\Delta_{\%} \text{ price}} = \frac{\frac{\Delta \text{ quantity}}{\text{quantity}}}{\frac{\Delta \text{ price}}{\text{price}}} \quad (2.1)$$

Due to the Law of Demand, PED is typically negative. A negative change in price results in a positive change in quantity demanded, and vice-versa. As this relationship generally results in a negative value, it is common to use the absolute value of its coefficient.

The demand of a given product is said to be elastic if $|\epsilon_p| > 1$. Otherwise, it is inelastic. If $|\epsilon_p| = \infty$, then demand is perfectly elastic, if $|\epsilon_p| = 0$, it is perfectly inelastic and if $|\epsilon_p| = 1$, demand is unitary. The interpretation of this coefficient is summarized in Table 2.1.

Table 2.1: Interpretation of the Price Elasticity of Demand coefficient

Coefficient	Demand
$ \epsilon_p > 1$	Elastic demand
$0 < \epsilon_p < 1$	Inelastic demand
$ \epsilon_p = \infty$	Perfectly elastic demand
$ \epsilon_p = 0$	Perfectly inelastic demand
$ \epsilon_p = 1$	Unitarily elastic demand

The higher the value of PED, the more elastic demand is. Demand for goods with higher PED is more responsive to price changes. Thus, consumers might not buy a product if they consider

its price to have risen excessively. At the same time, price reductions might considerably increase demand. In contrast, inelastic goods allow for price raises, with little to no changes in demand [16].

Due to the high number of alternatives in the electronics industry, we can argue that price is a relevant factor that customers consider in their buying decisions. Thus, demand is elastic, making electronic products good candidates for price markdowns.

Therefore, measuring PED is crucial for understanding how price affects demand and making informed and accurate pricing decisions.

2.1.3 Cross-Price Elasticity of Demand

Price changes in one good can positively or negatively impact demand for other related goods. For two products that complement each other (e.g. smartphone and smartphone case), a price reduction in one can lead to a rise in the quantity demanded of the other. Conversely, for two products that are substitutes (e.g. two different laptops), a price fall for one can decrease demand for the other [9]. This is particularly important when introducing new products since they can reduce sales for other substitute products and translate into a situation of cannibalization between items.

Customers often do not have a unique product in mind but a set of needs to be satisfied. As such, many decisions are not made until the customer enters a store. For this reason, the price difference between two similar products can be a decisive factor in the customer's choice and, consequently, highly impact their demand.

Therefore, the cross-price elasticity of demand (XED) corresponds to the percentage change in the quantity demanded of good j caused by the percentage change in the price of good i , as shown in Equation 2.2.

$$\epsilon_{p_i \rightarrow j} = \frac{\Delta \% quantity_j}{\Delta \% price_i} = \frac{\frac{\Delta quantity_j}{quantity_j}}{\frac{\Delta price_i}{price_i}} \quad (2.2)$$

It is relevant to notice that just because price changes in product i influence demand for product j , the opposite may not necessarily apply, i.e., their demand might be affected at different magnitudes.

For substitute products, the value of XED is positive. If it is negative, then the products are complements. Therefore, this effect is a relevant factor to consider when pricing products since these decisions might produce unwanted effects on other related products.

However, it can be a complex variable to quantify and include in models. For example, in [17], the authors chose not to include cross-product dependencies in their price optimization model due to their high complexity. On the other hand, in [18], one of the used features to estimate PED for a style is its relative price compared to the average price of competing styles. Even though this is not a direct representation of XED, we can argue that it still captures cross-product interactions to a certain extent. We detail these approaches in Section 2.3.2.

2.1.4 Sales lift

Estimating the sales lift due to a specific campaign or marketing activity is a typical problem in marketing. It can be a good indicator of the success and profitability of an implemented price reduction and if the generated sales make up for the decrease in margins [19].

The lift is evaluated by comparing baseline and incremental sales, typically expressed in the number of units sold or in gross revenue. Baseline sales correspond to the sales the retailer would have achieved in the absence of a promotion or markdown. Incremental sales are the additional sales generated in the presence of a price reduction [19]. Sales lift is, then, given by Equation 2.3, and represents the increase in sales relative to the baseline.

$$\text{Sales Lift (\%)} = \frac{\text{Incremental Sales}}{\text{Baseline Sales}} \quad (2.3)$$

In theory, incremental sales are obtained by subtracting baseline sales from total sales. However, determining baseline sales can be tricky since there are several variables we have to consider, such as seasonality and competition. Usually, baseline sales are estimated recurring to forecasting models that analyze historical data, taking those variables into account. The models can be used to project baseline sales and, consequently, estimate incremental sales [19].

2.2 Forecasting

Forecasting is an usual statistical task in business, which serves as a guide to long-term strategic planning and aids in the process of decision-making at all company levels, securing daily operations. It uses all the available information, such as historical data and knowledge, to accurately predict the future [20].

In the context of retail, Sales Forecasting is a critical necessity and is among the most crucial components of supply chain management since it directly affects a company's overall performance [21]. In fact, PLC planning and markdown optimization (MDO) in retail rely heavily on these forecasts.

The accuracy of these predictions is highly dependent on the data and on the nature of what we are forecasting [22]. The predictability of an event depends on several aspects [20]:

- the understanding of the factors that contribute to it;
- the quantity of available data;
- the similarities between past and future;
- whether the forecasts can affect the thing that is being forecast.

There is a wide range of forecasting methods with different levels of complexity. The appropriate forecasting approach depends highly on the problem and the available data [20]. This section provides an introduction to forecasting and the types of forecasting methods.

2.2.1 Types of forecasting methods

As shown in Figure 2.1, and according to [20], there are two types of forecasting methods: qualitative and quantitative.

Qualitative (or subjective) methods are based on judgements and opinions, and they are often used for long-term forecasts and when there is no relevant data available. Examples include market research, managers' know-how and the Delphi method. These methods can be prone to bias, limitations and inconsistencies. Therefore, these approaches should be methodical and well-structured [20].

Quantitative forecasting is applied when past data is available and when there are reasons to believe that some of its patterns will continue into the future. The forecasts produced by these methods can later be adjusted by combining experts' judgement. Quantitative methods mostly use either time series data or causal data [20].

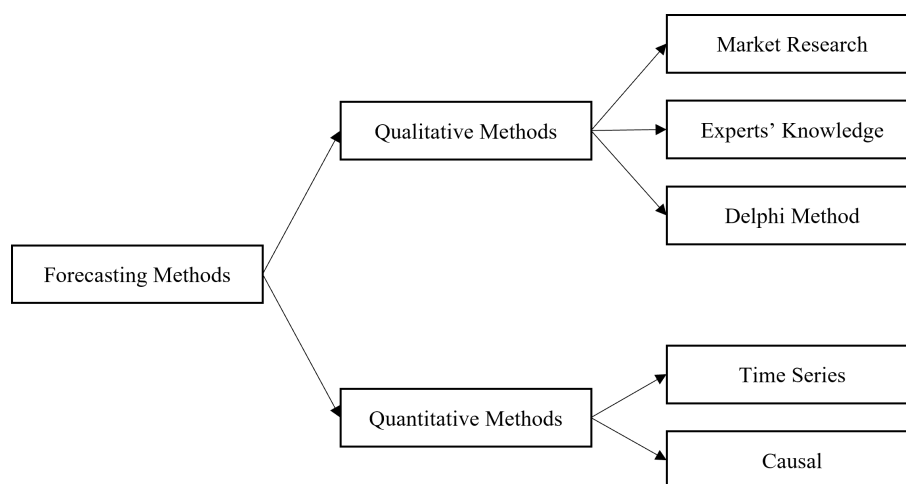


Figure 2.1: Types of forecasting methods, adapted from [20]

2.2.2 Time Series methods

Time series data corresponds to data collected over time that might follow a certain pattern or trend [23]. Therefore, time series methods use the historical data of the series to generate forecasts and predict the values of future observations. These methods exclusively use data about the variable that is being forecast, making no attempt to identify additional factors that may influence its behaviour. Examples of time series data include monthly rainfall, hourly electricity demand, and weekly retail sales [20]. Time series methods are normally used to predict baseline sales.

Time series data can have a variety of patterns that can be categorized into trend, seasonality and cycles. Decomposition procedures are often used to separate the time series into these components, allowing for the extraction of their individual effects and using them to make forecasts [20].

Examples of simple time series forecasting methods include the naive approach, the average method and the simple exponential smoothing.

In the average method, future values equal the average value of the observed data, assuming all observations have the same importance [20]:

$$\hat{y}_{T+h|T} = \frac{1}{T} \sum_{t=1}^T y_t \quad (2.4)$$

for $h = 1, 2, \dots$, where $y_1, \dots, y_T$ correspond to all the observations up until time T and $\hat{y}_{T+h|T}$ to the forecast.

Simple exponential smoothing methods use a smoothing factor α to control the importance the model will give to the most recent value, compared to previous ones [20]:

$$\hat{y}_{T+h|T} = \ell_t \quad (2.5)$$

$$\ell_t = \alpha y_t + (1 - \alpha) \ell_{t-1} \quad (2.6)$$

where ℓ_t is the smoothed value of the series at time t .

Some more advanced methods include, for example, the ARIMA models, which have been extended into other models to include factors such as seasonality (SARIMA) and other external variables (SARIMAX).

2.2.3 Causal methods

As the name implies, there is a causal relationship between variables in causal data. These include the variable to be forecast and a set of predictor variables. It focuses on cause-effect relationships, where a target variable depends on one or more independent variables [20]. For example, considering a dependent variable Y and N independent variables X , the resulting forecast would be dictated by a function of the predictor variables, as shown in Equation 2.7.

$$Y = f(X_1, X_2, X_3, \dots, X_N) \quad (2.7)$$

Through the use of predictor variables, forecasts take into account factors that would, otherwise, be ignored. As so, causal methods, also called explanatory methods, allow for the integration of several relevant features and try to explain the individual impact of each one of them in the target variable [20]. In retail forecasting, as explained in Section 2.1.1, the sales of a product can depend on its price changes, characteristics, competition and several other factors. Therefore, as we will see in Section 2.3.2, authors often choose to use causal methods to forecast the sales variation in the presence of price markdowns.

In this context of causal data and relating predictor and target variables, machine learning (ML) methods stand out, as we will detail in the following section.

2.2.4 Machine Learning

ML is a subfield of Artificial Intelligence that focuses on developing methods for extracting knowledge from data to accurately predict future outcomes [24].

These techniques can be a handy tool in the process of understanding the relationship between variables and making sense out of data. Through these algorithms, the model can learn from data and provide more accurate forecasts in shorter periods of time.

These algorithms can be categorized into different learning models: supervised learning, unsupervised learning and reinforcement learning.

- In **Supervised learning**, models use training data that comprises both the input variables and the corresponding target variable. These models can either perform classification or regression. In classification problems, input vectors can be assigned to a finite number of discrete target variables. In regression, the target variables are continuous, which usually is the case in Sales Forecasting [25]. This work calls for a supervised learning approach and, therefore, we will focus on supervised learning methods.
- In **Unsupervised learning**, there is no target variable, and its primary function is to discover patterns or similar instances within the data, for example, through clustering algorithms.
- In **Reinforcement Learning**, the algorithm is concerned with finding the actions that maximize a reward, learning through a process of trial and error.

2.2.4.1 Supervised Learning Algorithms

In Sales Forecasting, we are primarily interested in supervised learning algorithms, more specifically, in regression techniques.

In regression problems, the dataset is typically split into training, validation and test. The training data is used to fit the models, which are then tested on the validation set, containing data they have not seen up to that point, to validate if the models have become overfit to the training set [25]. The phenomenon of overfitting occurs when the model fits the training data too well and captures its noise, not being capable of performing accurately when confronted with new data [26]. The model with the best validation performance is then evaluated using the test dataset [25].

Some of the most common supervised learning algorithms are the linear regression (LR), decision trees (DT), random forests (RF) and gradient boosting machines (GBM), whose understanding is relevant to the understanding of the dissertation.

Linear Regression

LR is a commonly used method due to its simplicity and high explainability. It is the method used in [17] to estimate PED. LR models combine a set of independent predictor variables X to predict the output for a response variable Y , resulting in Equation 2.8 [27]:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} \quad (2.8)$$

where y_i is the i th response.

There are several possible methods of estimating the β parameters in the model, the most popular being least squares, by minimizing the squared distance between the observed response and the predicted values through Equation 2.9 [27]:

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^k \beta_j x_{ij} \right)^2 \quad (2.9)$$

One of the most significant advantages of LR is its linearity, making it an effortless and interpretable procedure. LR assumes four statistical assumptions: linearity, independence, homoscedasticity and normality [27]. The model's accuracy depends on how well the data meet these assumptions.

LR models can, however, be prone to overfitting. This phenomenon can be regulated and avoided by recurring to regularization techniques (e.g. lasso regression, ridge regression), which involve a penalty term being added to the error function to prevent the coefficients from reaching high values [25].

Decision Trees

DTs are one of the most popular approaches for supervised learning algorithms and can be used to perform both regression and classification. Starting at the root node at the top of the tree, a decision's result is determined by applying tests to the instance's attributes while navigating down the tree. Each node tests a single attribute, and the process progresses to the leaf node under the branch whose test matched the attribute's value. The final decision corresponds to the value of the terminating leaf (decision node) [28]. Figure 2.2 illustrates a DT, in this case performing classification, to predict if a particular email is spam or not, considering its attributes. In regression problems, the output is a continuous value.

Similar to LR, DTs are a straightforward and visual method that is easily interpretable. These can be important factors as they make the algorithm intuitive and easy to explain to the end-user of the solution.

However, like LR models, DTs are prone to overfitting. For this reason, these methods tend to be replaced by ensemble methods such as RFs and GBMs. In ensemble methods, a set of base

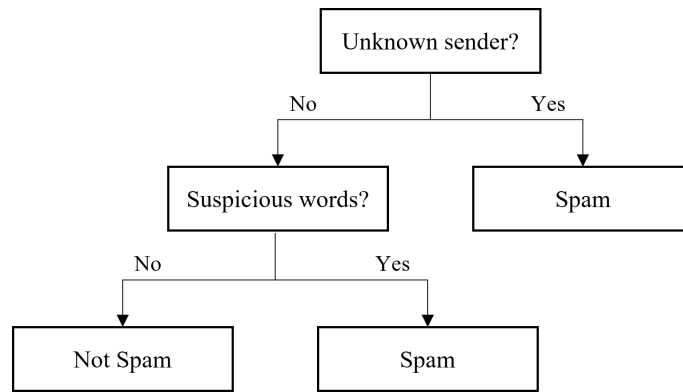


Figure 2.2: Illustration of a Decision Tree, adapted from [28]

models is constructed, and their individual decisions are then combined to generate new and more accurate predictions [29].

Random Forests

The RFs algorithm, introduced by Breiman in [30], is used for both classification and regression problems and uses an ensemble of de-correlated DTs, which together compose a "forest".

RFs algorithms combine bagging and random feature selection. In Bagging algorithms, data samples are randomly selected with replacement, i.e., a data point may get chosen more than once, increasing diversity and reducing variance. This was the chosen method by Ferreira et al. in [18], which we detail in Section 2.3.2.

In RFs, each node in a tree splits over a random subset of features. Consequently, individual trees are weaker predictors, but correlation with the other trees is lower, resulting in a more accurate and robust algorithm.

In classification problems, the output of the RF corresponds to the output selected by the majority of the trees. In regression problems, the final output corresponds to the average of the tree's outputs.

RFs correct the flaws in DTs, being capable of reducing overfitting and obtaining more accurate results, by sacrificing interpretability.

Gradient Boosting Machines

GBMs are also a tree-based method, developed by Friedman [31]. It is similar to RFs but, contrary to bagging methods, where weak learners are trained in parallel, in boosting methods, these are trained sequentially. The weak learners are built sequentially and try to reduce the error of the previous learners, minimizing a loss function.

Thus, DTs are trained on the pseudo-residuals of previous trees, giving more importance to the observations with larger errors. The pseudo-residuals correspond to the gradient of the loss

function that is being minimized [32]. Like RFs, GBMs have proved to be highly robust procedures for both regression and classification problems [31].

2.2.5 Performance Metrics

There are several methods to evaluate and measure a model's performance. These methods compare the predicted values and the actual values. In a forecasting setting, the difference between these values is called the forecast error [20].

Scale-dependent errors correspond to forecast errors that are on the same scale as the data. The most commonly used include the Mean Absolute Deviation (MAD), the Mean Square Error (MSE), and the Root Mean Square Error (RSME). However, these measures cannot be used to compare series that use different units [20].

Percentage errors, on the other hand, are unit-free and suitable for comparing forecast performances between different datasets [20]. Common metrics include the Mean Absolute Percentage Error (MAPE) and the Mean Percentage Error (MPE), also called Bias.

The following equations describe the mentioned metrics [20]:

$$MAD = \frac{1}{N} \sum_n^N |\hat{y}_n - y_n| \quad (2.10)$$

$$MSE = \frac{1}{N} \sum_n^N (\hat{y}_n - y_n)^2 \quad (2.11)$$

$$RMSE = \sqrt{MSE} \quad (2.12)$$

$$MAPE(\%) = \frac{100}{N} \sum_n^N \frac{|\hat{y}_n - y_n|}{y_n} \quad (2.13)$$

$$BIAS(\%) = \frac{100}{N} \sum_n^N \frac{\hat{y}_n - y_n}{y_n} \quad (2.14)$$

where $\hat{y}_1, \dots, \hat{y}_N$ correspond to the predicted values, $y_1, \dots, y_N$ to the actual values and N to the number of observations.

2.3 Markdown Optimization System

Pricing is a big challenge faced by retailers. Dynamic pricing is a strategy that applies prices to products in a variable and tactical way to react to demand and competition behaviour and maximize revenue [7]. There has been an emergence in the adoption of dynamic pricing strategies across all industries. This can be explained as a result of the increase in competition and availability of data and analytic tools [33]. For a long time, retailers have relied simply on their knowledge, judgments and simple rules of thumb to make pricing decisions. More recently, however, retailers started using analytic tools, and optimization software to aid in those decisions [5].

Retailers often face situations of excessive stock. There is rarely one single reason to explain excess inventory, as it is often caused by several factors [16]. It can be due to an inaccurate sales prediction, leading to lower than expected sales, to the introduction of a new product that negatively affects the sales of substitute products, or to several other reasons. In these situations, and when stock is higher than demand, retailers may choose to implement price reductions.

There are various mechanisms retailers use to reduce their products' prices. These can be classified into promotions and markdowns. The first corresponds to a temporary reduction in price, while the latter refers to a permanent price cut [5]. This work focuses on markdowns, specifically for end-of-life products, in the decline stage of the PLC.

As already explained, the PLC refers to the period a product stays in the market, from its introduction until its decline and consequent withdrawal. As illustrated in Figure 2.3, the cycle is typically divided into four stages: introduction, growth, maturity and decline [1]. Identifying the PLC stage a product is at each moment enables better planning and control over the cycle and helps make the correct strategical decisions, leading to cost savings and a boost in revenue.

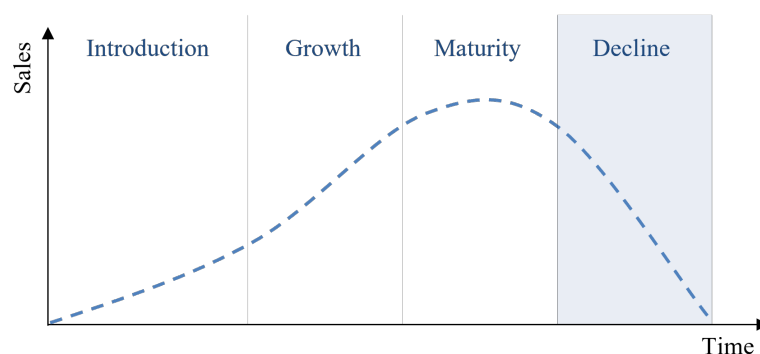


Figure 2.3: Illustration of a typical product life-cycle pattern

Technological products are characterized as having a short PLC, with an abrupt decline and a lack of maturity stage [3]. In fact, PLC curves were studied in [14] using a large dataset of computer products, and the authors concluded that half of the products had maturity stages less than 13% of the entire PLC length and that 20% of PLCs had no maturity stage at all.

Electronic products are frequently conceptualized with obsolescence in mind to ensure they do not outlast the following generation. As they are typically durable and pricy, the sales volume tends to be low. This problem is heightened for end-of-life products, which can have long periods without a single sale, making their behaviour even more difficult to predict.

Markdowns are typically applied to increase demand and profits, as consumers with lower WTP join the market potential. The increased demand enables retailers to clear the remaining inventory and make room for future items with possibly higher profitability. Once it is in decline and past its best-selling point, a product can have several markdowns until the end of its cycle. The first markdown defines the beginning of the clearance period. Figure 2.4 illustrates an example of the expected effect of a price markdown on sales.



Figure 2.4: Illustration of the impact of a price markdown on sales variation

The two main decisions retailers face are deciding the depth and the timing of the markdown. As explained in [7], both customer and retailer are confronted with a risky trade-off. The customer trades certainty about the price and availability of the item in the present against a perhaps lower price in the future, when the product may become unavailable. The retailer faces a trade-off between selling the articles at a price as high as possible and clearing the inventory at lower prices before the product becomes obsolete. Markdown pricing can, therefore, be viewed as a mechanism to distinguish between customers with contrasting price sensitivities [34].

Here lies the main difficulty in markdown decisions: setting the right price at the right moment. The retailer addressed in [16], due to a past failed strategy, ended up with a total of \$10 million in inventory surplus. Thus, it is critical to evaluate the impact of each strategy on sales and revenue before committing to one of them. In this context arises the relevance of MDO, whose main objective is finding the price that maximizes revenue while depleting the inventory.

According to Smith [35], a successful analytical approach to a markdown optimization (MDO) system comprises three components: a Sales Forecasting model, a clearance price optimization algorithm and financial performance metrics to measure the effectiveness of the system.

2.3.1 Modelling Assumptions in Markdown Pricing Problems

Dynamic pricing problems highly depend on the objective, industry and part of the cycle they are being applied to, as these factors influence retailers' decisions differently. Therefore, models' structure and applicability are affected by several assumptions and restrictions [7]. Three major characteristics of the market environment define the type of problem the retailer is facing [36]:

- **Myopic vs. strategic customers:** for myopic customers, WTP is the only deciding factor in their purchasing decisions. On the other hand, strategic customers consider an item's expected future price in their decisions and may wait for a price reduction before making a purchase. Despite being less realistic, myopic customer models are widely used in literature due to their higher mathematical tractability compared to the more complex strategic customer models [7].

- **Dependent vs. independent demand over time or finite vs. infinite population:** for durable goods, demand is typically dependent, and its population is treated as finite. Therefore, customers are removed from the market potential as they make purchases, as they will most likely not purchase the same product more than once. In contrast, in independent demand models, the population is considered infinite, as current sales do not impact future sales [7]. Demand and customer behaviour may vary highly between categories in the problem at hand. Although most electronic products may be classified as durable goods, stock levels, especially for end-of-life items, are very low compared to the population of potential customers. In such situations, because past sales represent such a small portion of this population, its impact is negligible, making infinite-population models a reasonable approximation. Again, despite not always being realistic, the population is often assumed infinite due to its higher analytical simplicity [37].
- **Replenishment vs. No Replenishment of Inventory:** whether the inventory can be replenished or not during the selling season. In the context of high replenishment lead times, especially for end-of-life items, where the retailer's objective is to clear the remaining amount of inventory, typically, there is no replenishment [36].

Talluri and van Ryzin [37] add another fundamental assumption in dynamic pricing models concerning the effects of competition. Monopoly models consider that demand only depends on the retailer's own prices and not on the prices of its competitors, assuming that historical price responses already incorporate the effects of competition. In oligopoly models, the expected competitive dynamics are explicitly modelled and computed, at a higher risk of modelling errors or erroneous assumptions. Perfect competition models are based on the assumption of an idealized market, where competing retailers offer identical commodities and, thus, a retailer is unable to influence market prices [7, 37].

As expected, due to its higher simplicity, the majority of dynamic pricing literature assumes monopoly situations with myopic customers, independent demand, and no inventory replenishment [7].

2.3.2 Related Work

Pricing decisions rely heavily on PED estimation, which is a necessary step toward an MDO system. Numerous methods can be used to estimate PED, and these are mainly based on historical sales and customer preference surveys [38]. This section will analyze some studies and approaches for PED estimation and MDO.

Despite the multiple existing studies on MDO, specific applications in consumer electronics retail are very scarce. Most relevant studies regard the fashion industry, whose products are characterized by even shorter PLCs and where PED estimation and MDO are common problems. In this industry, products are typically sold throughout a season, with a defined ending date by which all inventory should be depleted. For this reason, price optimization is a critical necessity, as wrong decisions can be even more costly. Additionally, as it has a defined time horizon, the

most common problem presented in these studies is price optimization to either maximize revenue in that period or clear all inventory. This has some differences compared to the problem in this dissertation, where there are no fixed selling seasons, and the clearance objective may differ from product to product, depending on the retailer's intentions.

However, as long as they follow the same assumptions, most of the approaches and conclusions based on these studies can be generalized and applied to other industries. We are also interested in studies regarding retailers with multiple stores, online retailers and omnichannel approaches. An overview of the reviewed studies is presented in Table 2.2.

Table 2.2: Summary of the analyzed studies

Study	Sector	PED Estimation	Novelty Aspects
Zaarour et al. [16]	Cosmetics Retail	Non-Linear Regression	Multiple stores; predetermined price set, clustering
Caro and Gallien [17]	Fashion Retail	Linear Regression	Finite set of prices; grouping of similar products; two-stage estimation
Ferreira et al. [18]	Online Retail	Regression Trees	ML methods; cross-product dependencies
Harsha et al. [39]	Retail (Omnichannel)	Not Specified	Cross-channel interactions
Smith and Agrawal [40]	Retail	Not Specified	Optimal inventory allocation; time and inventory dependant demand
Vakhutinsky et al. [41]	Fashion Retail	Log-Linear Regression	Store-level pricing; time and inventory dependant demand
Burman et al. [42]	Retail	Log-Log Regression	ML methods; large retail chain; weighted objective function
Loh et al. [43]	Online Retail	GBMs	ML methods; monotonic constraints

Zaarour et al. [16] present their work on MDO of discontinued cosmetic items, using weekly sales data from 54 weeks, sold by a large US retailer with over 6000 stores. The authors developed two models. The first corresponds to an optimal markdown strategy which uses multi-period non-linear programming to maximize revenue. The second uses linear programming to apply prices from a predetermined price set. Every Stock Keeping Unit (SKU) had a history of price fluctuations and several implemented discounts, which provided enough data for a reasonable estimation of PED for every product-store combination. However, the authors found that, for every SKU, there were significantly fewer weeks associated with low prices than weeks associated with higher prices. This kind of situation may also occur in this dissertation's project since the application

of higher discount levels is less common and often unnecessary and, thus, there are fewer data points for lower prices. Therefore, to better understand the price-demand relationship, the authors applied a k-means clustering algorithm to group the data, replacing each cluster with its centroid. The authors used nonlinear regression analysis to find the best fitting demand model for the data, experimenting with multiple functions, namely linear, quadratic and a power function ($V = \alpha P^\beta$), with the latter yielding the best fit. Parameter β corresponds to PED, V to sales volume, P to price and α is a positive constant. Compared to the current strategy, the results showed an improvement in profits per SKU, ranging from 8.6% to 22.6% and all inventory was depleted.

Caro and Gallien [17] implemented a multi-product price optimization system for fast-fashion retailer Zara. Until then, markdowns were determined by a manual and informal decision-making process and thus, the main objective of this work was to improve Zara's clearance pricing processes. Most products at Zara are characterized by short PLCs, and, due to the fixed-price policy during the regular season, there is a lack of price sensitivity data at the beginning of the clearance season. For this reason, PED estimation has no effect on the optimization of products' first markdown. This situation is contrary to what happens in the study by Zaarour et al. [16] and also to this dissertation, where there is no fixed-price policy. The authors implement a forecasting model, which is a particular instance of the model presented by Smith and Achabal in [4] with some changes. The model consists of an exponential form function, having as demand parameters past demand, the age of the item, the purchased quantity, the price discount and the effects of inventory levels. Due to the added complexity, the authors chose not to model cross-product interactions and the effect of competition and strategic customers. To overcome the lack of data, the authors developed a two-stage estimation procedure. In stage 1, coefficients for some regressors are calculated using data from the regular season and only later, as clearance sales data become available, the coefficient of the remaining explanatory variables, including PED, is estimated periodically via linear regression (LR). Contrary to what occurs in [16], the authors assume constant and identical PED for groups of similar products, as they find it gives better results than PED dependent on the price or the product. Along with the forecasting model, the authors built a price optimization model, limited to a finite number of prices (ending with "0.99"), with the new system indicating a 6% increase in clearance sales revenue.

In a study with online fast-fashion retailer Rue La La, Ferreira et al. [18] developed a multi-product price optimization model and incorporated it in a DSS. The authors face a similar challenge to the one in Zara: predicting demand for articles with no historical data. At Rue La La, products' prices are defined at the style level. A style corresponds to an aggregation of all sizes of identical items. To develop the demand prediction model, the authors first use sales data to estimate demand for styles sold in the past. Then, the obtained results are used to predict the sales of new styles. Even though PED is not directly estimated, three of the features in the model are related to price. Those are the price of the style, the percent discount off and the relative price of competing styles. Even though the latter is not a direct representation of PED, it is an indirect way of capturing product interaction effects. The authors chose to focus on models that are easily interpretable for managers, which is also a relevant aspect to consider in the project in this dis-

sertation. Regression trees with bagging outperformed other regression techniques, providing the best results. The demand prediction model is then used to determine the optimal pricing strategy in order to maximize revenue. The developed system was then tested in a field experiment and showed an expected increase in revenue of 9.7% for new styles.

Harsha et al. [39] studies the problem of price optimization in the context of cross-channel demand and supply interactions, with exogenous cross-channel fulfilment, within a large omnichannel retailer. The authors model omnichannel pricing as a dynamic stochastic program and propose two pricing policies that estimate how the inventory will be shared and utilized across stores, comparing the results. Following a pilot test at the retailer, the system resulted in a 13.7% increase in clearance sales revenue. Their approach was implemented commercially, integrating the IBM Commerce Markdown Price Solution.

Smith and Agrawal [40] formulates a MDO problem at a large retail chain and analyzes how inventory levels affect demand and optimal pricing. Their solution consists of three essential levers: pricing, inventory allocation and which stores to allocate the stock. The problem is solved in three steps. First, the optimal price is determined for each SKU. Then, considering the optimal pricing, the optimal inventory allocation is determined. Finally, the authors optimize what stores to stock, considering the optimal pricing and inventory allocation. The authors model demand as a function of exponential PED, a time-varying demand factor and, as was used in the Zara case study in [17], inventory level effects. Results show that the ideal combination of these three levers and considering the inventory effects can lead to substantial profit improvements.

Similar to [4, 17, 40], in a study on MDO by Vakhutinsky et al. [41], inventory levels' effects on demand are taken into account. The authors consider a SCAN*PRO demand model, which is assumed to be a multiplicative, separable function composed of three components: a time-dependent seasonality coefficient and price and inventory-dependent components. The best fit was achieved by assuming constant PED and having the inventory effect modelled as a power-law function. The seasonality coefficient was obtained from historical sales and PED was estimated via log-linear regression. The authors introduce two algorithms for revenue maximization: optimal continuous price control and markdown price control when price and time are discrete. The second model was tested against the current approach on weekly historical sales data of 50 000 store-level items from a fashion retail chain. Rather than producing financial results, the study compared regret, optimality and computation speed of both approaches.

Recent studies include [42], in which the authors use gradient boosting techniques and artificial neural networks to estimate demand for a large retail chain. They estimate the PED coefficients using the log-log model and assign them to the items. The items which have had minimal to no price changes are grouped with other similar items. The optimal markdown prices are obtained by maximizing an objective function of the overall yield, which is the weighted sum of sales units, revenue and margin. The weights are business inputs based on the retailer's objectives. The expected sales are calculated based on the demand predictions, on the PED estimations and on the decision variable, i.e., the optimal markdown price. They also consider business constraints: minimum and maximum markdown prices and percentage of inventory to be cleared for each

merchandise.

Another recent study [43] also uses GBMs to model demand. Similar to [18], PED is not directly estimated but the model uses price-related features, among others, as predictor variables. The authors apply monotonic constraints to the model so that demand does not decrease as the applied discounts increase. This constraint prevents the predictions from going against the Law of Demand.

Most of these studies contain the components of a successful MDO system mentioned by Smith in [35]. Indeed, the typical approach is to develop a model to predict demand, followed by a MDO algorithm and finishing by presenting financial performance metrics. It is also worth noting that many of these studies (e.g. [16, 17, 41]) consider already predetermined price levels to choose from when finding the optimal pricing.

There are some other interesting studies. For example, in [44] the authors use a multiplicative model to predict sales and find products' optimal price for an online retailer, also considering a set of discrete price levels. Namin et al., in [45], developed a demand and a markdown pricing model to simulate four different markdown strategies and find the optimal markdown timing and depth.

2.3.3 Markdown Optimization Problem Formulation

According to the studies reviewed in Section 2.3.2 and the assumptions made in Section 2.3.1, we can formulate a generic MDO problem. Most of these studies (e.g. [16, 41]) focus on maximizing revenue over a finite set of periods.

In [16], as we have seen in the previous section, sales volume is given by a power function ($V = \alpha P^\beta$). The authors determine revenue R by multiplying the price P of the product by the sales volume ($R = VP = \alpha P^{\beta+1}$). Therefore, the primary goal in the optimization problem is to find the unit price $p_{i,t}$ of each SKU i in each period t , that maximizes the total revenue R_i over all periods. A similar formulation is used in [41]. A second revenue maximization model in [16] considers a set of predetermined prices, where each $p_{i,t}$ must correspond to a specific value in set P_i .

Therefore, we can define the problem as follows, according to the following notation:.

Sets

T	Set of all periods
$V_i(p,t)$	Estimated sales volume for SKU i at price p and period t
P_i	Set of eligible prices for SKU i

Parameters

t_0	Starting period
t_f	Final period
t	An arbitrary period
$I_{i,0}$	Initial inventory of SKU i

Decision Variables

$p_{i,t}$ Optimal price for SKU i at period t , belonging to P_i

Mathematical formulation

$$\max R_i = \sum_{t=t_0}^{t_f} p_{i,t} V_i(p, t) \quad (2.15)$$

$$\sum_{t=t_0}^{t_f} V_i(p, t) \leq I_{i,0} \quad (2.16)$$

$$p_{i,t_j} \leq p_{i,t_i}, \forall t_i < t_j \in T \quad (2.17)$$

$$p_{i,t} > 0, \forall t \in T \quad (2.18)$$

$$I_{i,0} > 0 \quad (2.19)$$

The objective function 2.15 maximizes revenue R for SKU i over all periods. It is subject to a set of constraints: total sales volume must not be higher than the inventory (2.16); prices reductions in markdowns are permanent, i.e. price must not be higher than it was in previous periods (2.17); price must be higher than zero (2.18); the initial inventory value must be positive (2.19). The presented formulation is just a generic version of a revenue maximization problem, adapted from literature, that could be extended to more sophisticated approaches to include problem-specific factors. In our problem, however, revenue maximization is not the sole purpose of the MDO problem, as the retailer may have different and specific clearance and margin objectives in mind.

Chapter 3

Problem Statement

This chapter explains the problem at hand and presents the proposed solution. Section 3.1 provides an overview of the retailer that is the target of this work. Section 3.2 describes the current markdown pricing process used by the retailer, its current problems and opportunities for improvement. Section 3.3 presents and defines the main goals of the project, as well as its requirements. Section 3.4 gives an overview of the solution we propose to improve the retailer’s markdown pricing process.

3.1 Retailer Overview

The company addressed in this work is a large Portuguese retailer focused on consumer electronics. The retailer follows an omnichannel strategy, consisting of the integration of a network of hundreds of physical stores and an online channel.

The retailer’s product structure is composed by several *Business Units*, each containing several *Categories*, which, in turn, comprise different *Subcategories*. Each *Subcategory* has a set of *Base Units*, which splits into different *SKUs* at the bottom of the product hierarchy, as depicted in Figure 3.1.

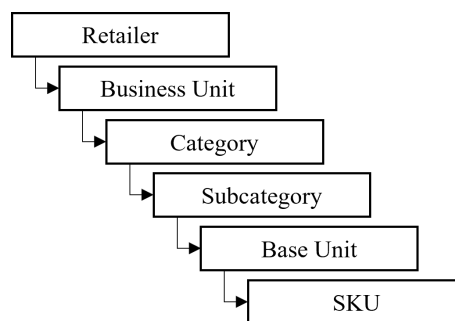


Figure 3.1: The retailer’s product hierarchy

As already explained, the project is focused on products in the decline phase of the PLC, particularly on products the retailer classifies as discontinued or provisioned. The latter can be

categorized into distinct provision types, which define how critical it is to clear the remaining inventory for that SKU. This classification is made according to a set of rules which compare the number of sales in the previous year with the amount of stock remaining at the present moment, as explained in Table 3.1.

Table 3.1: Products' provision type according to their stock rotation

Provision Type	Rule
A	Number of sales (last 12 months) < 10% stock
B	10% stock < Number of sales (last 12 months) < 50% stock
C	50% stock < Number of sales (last 12 months) < 100% stock
No Provision/Discontinued	100% stock < Number of sales (last 12 months)

A reduction in the stock value of these items is prematurely applied to encourage price reductions and clear stock faster. These rules apply only to products past their point of maturity, defined by the retailer as the moment a product reaches 2/3 of the total expected PLC sales and starts progressively registering an increasingly lower stock rotation. Otherwise, products that were recently introduced and, for that reason, have had low sales, would be classified as provisioned items.

The stock for end-of-life products has a significant impact on margins and the invested capital. These products are included in the stock clearance monthly campaigns, in which the manager of each category defines the product's prices at each store, expecting to minimize lost profits. The project, initially, will focus on four pilot categories that will be used to validate the developed solution. The approach will then be progressively extended to the remaining categories. To ensure that the solution can be applied to any product type, the selected pilot categories possess completely distinct characteristics regarding value, seasonality and diversity. These categories are: *Mobile Phones*, *Washing Machines*, *Food Preparation* and *Refrigerators*.

3.2 Current Process

The retailer follows a traditional standardized approach for pricing decisions based on a discount matrix, which they believe leads to reasonable outcomes. The discount matrix consists of a set of rules to decide which price discount (i.e. markdown) should be applied to each SKU at each store. These rules are based on the product's provision type, age, stock rotation, number of past campaigns and whether or not that SKU is a single piece in the store. The prices defined through the matrix are then approved or adjusted, one by one, by the managers of the categories, who have a vast experience in products' and stores' characteristics and their response to price reductions. This process is described in Figure 3.2.

Despite being a standardized process with logical rules, it has several downsides. The discount matrix is defined empirically, without any analytical base. The rules applied by the matrix consider only a minimal set of factors, neglecting other potentially influential aspects, such as seasonality. Furthermore, these factors make little effort to differentiate between products and stores, doing

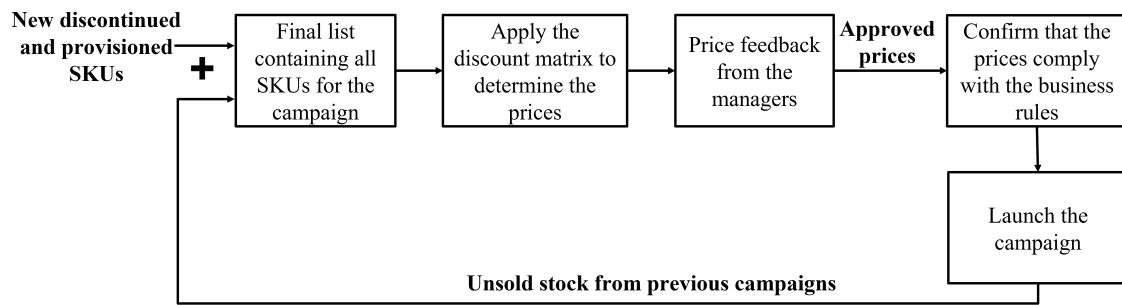


Figure 3.2: The process currently used by the retailer

a very poor job at capturing their individual nuances and responses to price changes. Also, the managers might have a specific margin or clearance objective in mind for their products, which is a factor the current process does not consider, making it challenging to define medium and long-term strategies. Considering this, the quality of the approach is dubious and solely based on experience, and the managers often need to adjust the chosen prices.

Additionally, the process is relatively inefficient. It relies on managers' price review and approval for hundreds of SKU-Store combinations. The process can be a very exhausting task for the managers as they need to make numerous individual decisions, which involve analyzing numerous metrics such as past sales and stock values.

Due to the high amount of time spent in the process, pricing decisions only occur once a month, which may be lower than the ideal frequency. To successfully clear the inventory and meet the desired stock clearance goals, there must be a quick reaction time and an appropriate selection of the markdown to be applied. Therefore, such a large retailer needs a more sustainable and efficient approach tailored to each store's and product's specific necessities.

3.3 Goals and Requirements

As introduced in Chapter 1, this dissertation arises in the context of a project with the retailer, whose ultimate goal is to improve markdown pricing decisions for end-of-life products.

We intend to do so by developing a methodology that predicts products' demand for several discrete discount levels and suggests the optimal price markdown (among those levels) to accelerate stock clearance and ensure the highest possible margins, according to the retailer's needs.

Additionally, there is a focus on making the decision process more efficient. We hope to improve the process's efficiency by incorporating the developed MDO system into an easy-to-use decision support tool. Here, managers will have access to all the information they need to make the decisions, including the predictions for all the discount levels, the suggested price and several metrics. The ultimate goal is that, as the managers become more confident in the system's suggestions, these decisions become as automated as possible, with progressively lower intervention

on the managers' side. Also, the process should shift to a weekly frequency, allowing for a faster reaction to the observed demand.

Given the mentioned goals, the DSS should integrate the following functional requirements, aligned with managers' needs and feedback:

- **Target Setting:** each manager should be able to set the desired targets for the SKUs in the respective category. They define, for the category they manage, the desired maximum number of weeks until all stock gets cleared and the minimum margin that must be ensured by the suggested prices. Additionally, the managers choose which of these targets is the main priority.
- **Strategy Definition:** each manager should be able to define if he either intends to apply, for a given SKU, the same price for all stores and channels (an omnichannel price) or if each store may have its own optimal price, adjusted to its specific needs. In other words, whether the decision should be made at the SKU-level or the SKU-Store-level.
- **Price Markdown Optimization:** according to the selected strategy and targets, suggest the optimal price markdown. Managers should be able to visualize, for the recommended price at each SKU-Store combination, the value of related metrics (e.g. expected sales, time until clearance, margins).
- **"What-If" Analysis:** besides the suggested price markdown, managers should be able to simulate how other possible discounts affect the related metrics. Based on this information and their business experience, if managers disagree with the suggested price, they should be able to change it directly on the support tool.

Additionally, the proposed DSS should respect certain characteristics that may shape how the solution and the functional requirements are fulfilled. As the managers are the system's end-user, we should prioritize finding a highly explainable solution, so they can clearly understand the results and suggestions. Also, the solution must apply to any category, as it will be extended to the remaining ones.

In the end, the most critical requirement is to ensure that the DSS is an intuitive and easy-to-use tool where the managers have all the input they need to make the best possible decisions.

3.4 Proposed Solution

Considering the issues raised in Section 3.2 and the main requirements in Section 3.3, we propose a new markdown pricing process, as depicted in Figure 3.3.

The new process possesses several advantages compared to the current one. It includes an advanced analytical model to define the price markdowns, allowing for a more agile and informed actuation in managing discontinued items and making it a faster and more effortless task for the managers.

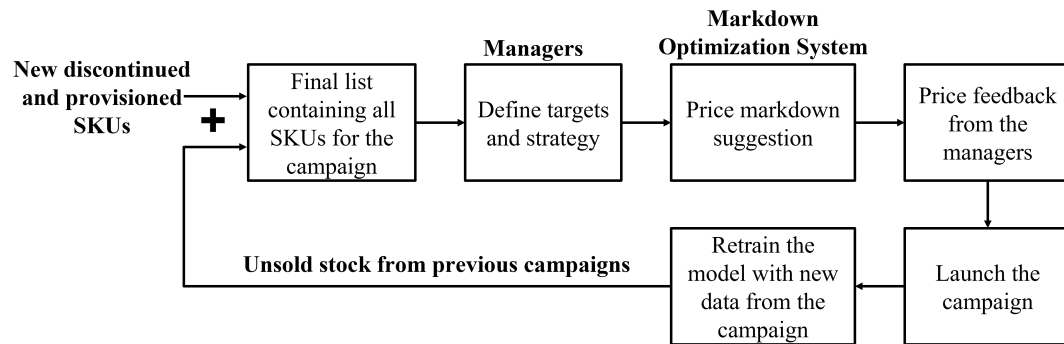


Figure 3.3: Proposed solution for the new markdown pricing process

All steps in the process will require less intervention. As the managers get more confident with the model's results, their validation becomes less and less necessary, shifting towards a more automated process.

As explained in Section 3.3 and aligned with the studies reviewed in Section 2.3.2, the MDO system is composed of a PED estimation model, built over a prediction of baseline sales, and an MDO algorithm.

- PED Estimation Model:** the primary purpose of the PED estimation model is to predict the increase in demand for a product caused by a given price reduction. These predictions are generated for a discrete set of discount levels (ranging between 5% and 95% off, in increments of 5%) for each product-store combination, using a supervised learning technique. The baseline purpose is to be the reference used by the model to estimate the expected sales increase. The baseline forecasts are, therefore, combined with the incremental sales estimations to predict sales for the next campaign, as shown in Figure 3.4. The model is not category dependent, and all categories use the same approach.

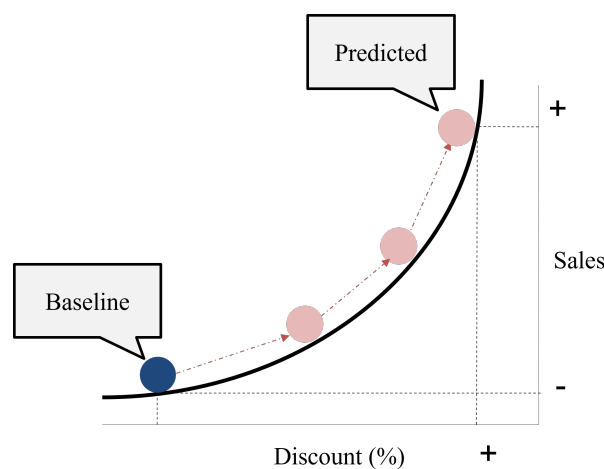


Figure 3.4: Representation of the PED Estimation Model

- **Markdown Optimization Algorithm:** through the PED estimation model results, it is possible to simulate the impact of each discrete price markdown on the expected sales and, consequently, on how long a product's stock will take to clear. Furthermore, we consider two simultaneous objectives when selecting the optimal markdown: maximizing margins and minimizing time until clearance. Considering this, and according to the targets defined by the managers, we compute the optimal price markdown (among the set of discount levels) for each product or product-store combination, depending on the selected strategy. Figure 3.5 shows an example of the discount suggestion according to the values of time until clearance and unit margin (which are merely an example and do not reflect any real instance).

Discount Level	Weeks Until Clearance	Unit Margin (%)
...	...	...
20%	7	19%
25%	4	11%
30%	3	4%
35%	2	-2%
...	...	...

Figure 3.5: Optimal price suggestion according to the clearance and margin targets

These components are integrated into the Stock Clearance Tool, which the managers use to validate the pricing decisions. The proposed solution is detailed in Chapter 4.

Chapter 4

Methodology

In this chapter, we detail the development and implementation of the proposed solution. Section 4.1 describes a general methodology commonly followed in data-driven projects. Section 4.2 provides an overview of the developed DSS and its components. Section 4.3 describes the available data and how it is prepared. Section 4.4 provides an overview of the PED estimation model and the steps taken during its development. Section 4.5 explains the reasoning behind the markdown optimization algorithm. In Section 4.6 we present the tool developed throughout the work, where the retailer’s managers make their markdown decisions.

4.1 CRISP-DM

The process of extracting valuable knowledge from data to help resolve business problems involves multiple complex stages. Therefore, there is a need for a standard methodology to assist in translating business problems into data-related tasks and guide these complex processes [46].

The Cross Industry Standard Process for Data Mining (CRISP-DM), illustrated in Figure 4.1, is a popular methodology that provides a comprehensive and structured approach for carrying out Data Mining and Data Science projects, making them easier to manage and implement [46]. We use this methodology as a guide for this project.

It provides an overview of the cycle of a data mining process, divided into six distinct phases which, ideally, should be performed respecting their specific order [46]. The six phases of the methodology are [46]:

- **Business Understanding:** the first phase of the cycle is focused on understanding, from a business perspective, the project’s main goals and requirements, defining the problem and designing a preliminary plan to accomplish these objectives. This phase relates to what was presented and discussed in Chapter 3.
- **Data Understanding:** the focus in this step is to collect and explore data and get familiar with it. Data Understanding is closely related to Business Understanding in the sense that, to formulate the problem, there must be an understanding of the data involved.

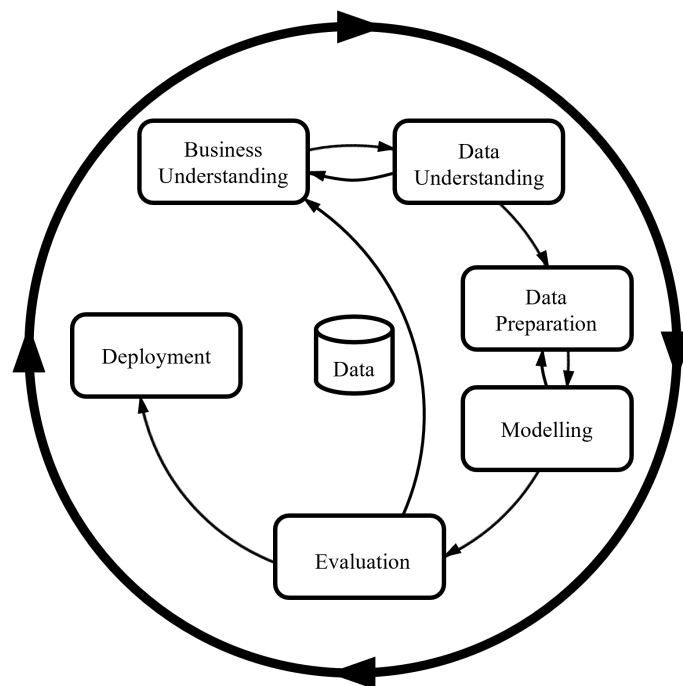


Figure 4.1: Illustration of the CRISP-DM cycle, adapted from [46]

- **Data Preparation:** this phase involves all the steps to create the final dataset from the raw data, namely data cleaning and transformation, attribute construction, and selection of aggregation levels.
- **Modelling:** selection and implementation of modelling techniques. This phase is strongly related to Data Preparation since the model's performance depends on the data.
- **Evaluation:** at this phase, the built models are evaluated and reviewed to ensure the business objectives were achieved.
- **Deployment:** finally, the final model is deployed, and the obtained results are arranged and presented to the client. Depending on the intended objective, the deliverable can range from a simple report to the development of software to be used by the client.

4.2 Decision Support System Overview

As shown in Figure 4.2, the developed solution is composed of several elements and relies on their successful implementation and integration.

The raw data, including the product, store and sales information, is retrieved from an Apache Hadoop cluster and stored in tables in Azure Databricks. This data migration between Hadoop and Databricks is made through a data ingestion process ensured by Azure Data Factory. The pricing data is loaded from an Apache Kafka data streaming topic that is also orchestrated in Azure Data Factory.

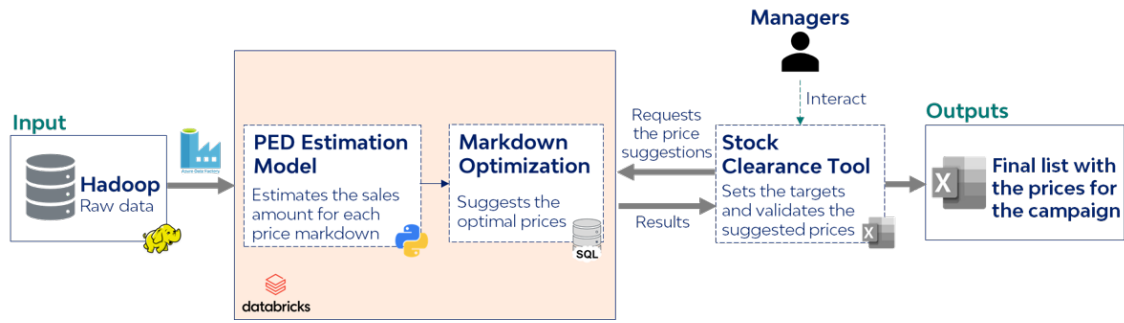


Figure 4.2: Decision Support System Diagram

- **Apache Hadoop** provides a framework for distributed storage and processing large amounts of data across computer clusters. It is used as the main data source in this work.
- **Azure Databricks** is a collaborative cloud-based data analytics platform used to process and transform large amounts of data and build ML models. It is highly flexible and developer friendly as it allows interchangeably using multiple programming languages (e.g. Python, SQL). All the tasks related to data preparation, model development and suggestion computation are performed on this platform.
- **Azure Data Factory** is an ETL tool for scale-out data ingestion, preparation and transformation. It centralizes the information from all the retailer's sources and allows scheduling jobs to be run at a specific day and time.

The PED estimation model, developed in Python, estimates the expected sales amount for every possible SKU-Store-Discount combination. These predictions are stored in a table, where each row reflects a different price markdown and its impact on sales for the related product and store.

From the sales estimations, the MDO algorithm proceeds to compute the optimal price markdown recommendation for every SKU-Store, according to the targets defined by the managers. From the table with the sales predictions, we construct a view which exclusively contains, for each SKU-Store combination, the row of the corresponding suggested markdown, obtained from the previous table. Views are dynamic virtual tables which contain the result of an SQL query. Therefore, a view's content corresponds to the exact data retrieved from other distinct tables or views in the database, according to the executed query's result. A view always displays updated information. Hence, if the data from the source tables changes, the view is recreated, and its data changes in conformity.

The final markdown suggestions depend on the clearance and margin targets defined by the managers, which are stored in a table and can be changed at any moment. Therefore, as the suggestions are stored in a view, every time a manager redefines the targets, the selected discount level on the view and, consequently, the price suggestion get automatically updated accordingly.

The Stock Clearance Tool (SCT) is the project's main deliverable and consists of an interface that the managers interact with to set the desired objectives and validate the price suggestions. The tool requests data from the views and tables and presents it in the interface. Through it, the managers can analyze the impact of different markdown scenarios on clearance and margin and, thus, validate the price suggestions before submitting their final pricing decisions.

4.3 Data Analysis

This section covers the steps of Data Understanding and Data Preparation of the CRISP-DM cycle. Data Understanding is closely tied to understanding the business and the problem, which we exposed in Chapter 3. Therefore, the proposed solution highly depends on the available data and its relevance to solving the problem.

We have access to data spread across several tables regarding multiple sources of information. These include products' attributes and hierarchy, location information, available inventory, campaign history and sales:

- **Product Data:** It is stored in a table where each row contains the SKU identifier, its description, brand and product hierarchy. It also contains information regarding the supplier, the original price of the product, its date of introduction and if the product is still active or not. Other attributes include the item's weight and dimensions and other product-specific characteristics (e.g. screen size in smartphones).
- **Location Data:** Contains information regarding all the different locations, which can correspond to either stores or warehouses.
- **Inventory Data:** Contains the data regarding the stock at each moment in each location, recorded daily for each SKU-Location combination.
- **Campaign Data:** Holds records of previous stock clearance campaigns since January 2021. These include, for each campaign and SKU-Store combination, the applied discount, the new price and the price a product had during the previous campaign. Besides the price, it contains the starting and ending dates of the campaign and information regarding products' provision type during that campaign and whether or not they were a single piece.
- **Sales Data:** Contains all transaction information regarding daily sales and returns of all products (not only the included in the stock clearance campaigns). As attributes, each sale is associated with a store, a product, its price and quantity sold and other attributes regarding the transaction information.

Due to the size of the retailer, which comprises many stores selling a large variety of products, some of these tables have millions of rows. Also, some of them contain hundreds of attributes, most of which are irrelevant to our task, making the data preparation a more challenging process.

Furthermore, the data is divided into several tables whose relationships are not always easy to link and are often inconsistent. Wrong values and missing data were common, and it was not always easy to interpret attributes and identify what they referred to.

We had to hold several meetings with different teams in the retailer to ensure that the variables were correctly understood and that we were working them correctly. For example, when calculating the available stock amounts, we were unsure of how to get their actual values since there were several attributes related to stock (e.g. stock in transit, unavailable stock, reserved stock, transferred stock). Also, we had no information on whether stock units were being displayed on the store shelves or if they were resting at the store's warehouse, which naturally has a significant impact on sales.

To summarize, we had large amounts of data, most of which were irrelevant to our end goal, spread across numerous tables, making it hard to use and analyze.

Therefore, we simplified and transformed the data, making it usable and putting it in the structure required for the model development. We joined the relevant attributes in a single dataset, cleaned the data and filtered it to only include the categories we are targeting, the items eligible for the stock clearance campaigns and the attributes that were present in all categories.

The selected attributes are mentioned in Section 4.4.2. We selected the features that, intuitively, made sense for our problem. Also, we performed a correlation analysis to only select variables with lower correlation. We excluded the rows with no value in the applied discount column (which we considered to be the most relevant variable). We removed the columns with undefined values in more than half of the rows. The same procedure was followed for the rows, where we removed the ones with undefined values for more than half of the attributes. The remaining undefined values were replaced by the most frequent value in the respective column.

We also calculated new fields, some of which were used as predictor variables in the model. These include, for instance, stock rotation, sales amount during the previous 60 days, time since the last purchase and the average price of the subcategory.

Through the price history, we also extracted the minimum price practised in the previous 30 days for each SKU at each store. Due to a recently introduced law in retail, instead of the discount relative to the original price, retailers must now display on products' labels the lowest price the product has had during the 30 days prior to the price reduction, alongside the current price. Therefore, the new price must be lower or equal to that minimum price, or customers would be highly unlikely to make a purchase. This rule is already inherent to price markdowns, but this price is an important field that we must extract so that our recommendations respect it.

As explained in Section 2.1.1, seasonality can highly influence a product's demand. Due to seasonality, markdowns could result in even fewer sales than in previous campaigns. These results could be misleading and lead to inaccurate predictions. Therefore, we calculated a factor to exclude the influence of seasonality in sales so that the model's predictions were independent of this effect.

We perform this operation at the subcategory level for each week of the year. Considering sales data from all products since 2017, we calculate each subcategory's yearly average of total

weekly sales. The result will work as the reference to find the relative sales increase or decrease in each week, compared to the yearly average. Then, for each week in the year, we average the total subcategory sales obtained in previous years during the corresponding week. The result is compared to the reference to calculate the seasonal factor, as shown in Equation 4.1, where w corresponds to the week in question and s to the subcategory.

$$\text{Seasonal Factor}_{w,s} = \frac{\text{Total Sales}_{w,s}}{\text{Yearly Average of Weekly Sales}_s} \quad (4.1)$$

For example, if, for a specific week, a subcategory has an average of 200 sales and its average weekly sales in a year is 100 sales per week, then the seasonal factor will be 2. Before the date enters the model, the sales amount of each SKU is divided by the seasonal factor of the corresponding subcategory and week to remove seasonality effects and put all data under the same circumstances. After the model predictions are computed, the resulting sales are multiplied by this factor to re-add the seasonality and obtain the final expected sales.

4.4 PED Estimation Model Development

Once the dataset with the relevant data is prepared, we focus on developing the PED estimation model, whose implementation steps are described throughout this section.

4.4.1 Baseline Sales Forecasting

The first step of the implementation is determining baseline sales. As explained in Section 2.1.4, baseline sales correspond to the sales amount that would be expected to be achieved if a product was under the influence of no promotional activities, i.e., when the product's price corresponds to the original price with which it was first introduced.

Before the start of every campaign, the baseline sales of that period are calculated through the average method, introduced in Section 2.2.2. For every SKU-Store combination, considering the seasonal factors, we average the historical daily sales during the 60 days prior to the first markdown, i.e., when the product was at its original price. The result is then multiplied by the campaign duration, in days, to obtain the total baseline sales during that period. The average method is frequently used to predict baseline sales and, despite being extremely simple, can lead to surprisingly effective results [20].

In our problem, the baseline works as a reference to find the increased sales and resulting sales lift caused by an implemented price markdown. Hence, the product's original price, i.e., the baseline, corresponds to a relative discount of 0%. Therefore, the percentage discount of a price markdown is relative to the original price and the corresponding generated sales lift is relative to the baseline.

4.4.2 Feature Selection

An essential step toward the model's implementation, and still part of the Data Preparation phase of the CRISP-DM cycle, is determining which predictor variables to include. In line with the steps described in Section 4.3, as our objective is to analyze the effects of price markdowns and how they impact the sales variation, we are only considering data regarding stores with stock and weeks in which there were price changes. Also, as we wanted our solution to be general and applicable to any category, we had to select features that were present in all categories. The manager's feedback regarding which factors could have the most influence on the final result, according to their experience, also helped us in this task.

We list and group the included predictor variables below:

- **Product Hierarchy:** variables of the category and subcategory;
- **Product Characteristics:** the product's brand, age and provision type, the number of previous campaigns it was included in and the time since it got discontinued;
- **Price Related:** these include the original price, the prices before and after the discount and the discount duration. Additionally, we also include variables for both the percentage price cut relative to the previous price and the relative discount compared to the original price. As mentioned in [17], customers are often more sensitive to the total markdown than they are to the reduction from the previous price;
- **Inventory and Store Related:** both global and local stock quantities, global and local stock rotation (at the SKU and subcategory levels) and a binary variable indicating if the article is a single piece at the corresponding store. Also, we include a variable regarding the space at the store used by the corresponding category;
- **Sales Related:** sales during the previous 60 days (at the SKU and subcategory levels) and time since the last sale;
- **Product Interaction:** as we discussed in Section 2.1.3, substitute and complementary products affect each other's demand. Similar to what was done in [18], we include a variable with the average price of the subcategory. Even though it is not a direct representation of XED, it can relate a product's price to the prices of similar items and capture product interaction to a certain degree. We also include as predictors the number of products of the subcategory in the store, their stock, and how many of them are under the effect of a price reduction.

As the target variable, we use the absolute incremental sales lift relative to the baseline, i.e., incremental sales.

4.4.3 Modelling Approach

With the baseline and the predictors set, we proceed to the development of the PED estimation model per se. As explained in Section 3.4, we develop a supervised learning model whose primary goal is to predict, for every product-store combination, the increase in demand caused by different price markdowns. The predicted incremental sales are then combined with the defined baseline to obtain the total sales prediction for a given markdown. We do not model PED explicitly but predict the sales generated by a price markdown considering multiple price-related features. Therefore, the estimated increase in sales can be interpreted as an indirect measure of a product's estimated PED.

One of the project's requirements is to enable the managers to analyze the impact of different price markdowns on sales, clearance and other related metrics. We consider a discrete set of discount levels, with relative discounts ranging between 5% and 95%, in increments of 5%. Sales are estimated for each of these possible markdowns, considering the seasonal factors explained in Section 4.3. These estimations are later used to find which of these possible discount levels results in the optimal markdown. They are presented to the categories' managers in an interface where they can analyze their impact individually and use their input to make a final pricing decision.

As seen in Section 2.3.2, demand is commonly modelled as a parametric function of price and other variables, using regression analysis to determine its coefficients, including PED (e.g. [16, 17]). Modelling demand as a parametric function makes it possible to formulate the MDO problem via linear programming, as done in Section 2.3.3, and thus find a more general solution applicable to any typical markdown pricing problem. However, our problem does not have a fixed selling season with a defined set of periods over which the prices are optimized, making it difficult to follow this approach.

On the other hand, Ferreira et al. [18] favoured non-parametric models. In their problem, the authors found that regression trees with bagging outperformed parametric regressors and claimed it to be an effective predictor of demand in terms of predictability and interpretability. We introduce bagging algorithms and RFs in Section 2.2.4.1. A benefit of non-parametric methods, which include ML techniques, is that they are more general because they do not require the specification of a parametric function relating features to demand. They can find non-monotonic relationships in the data and determine each feature's impact on demand.

Similar to [18], our problem calls for a more sophisticated approach, recurring to ML techniques. We also chose to focus on tree-based algorithms, but rather than regression trees with bagging or RFs, we opt to use a GBMs model, which we introduced in Section 2.2.4.1. GBMs are some of the most commonly used ML methods and they have shown to be highly customizable to the specific necessities of the application, showing an appreciable success in a wide range of domains [47]. Additionally, like the other tree-based methods, they do not require feature normalization. This method was used in [42, 43], which were the most recent studies we have analyzed in Section 2.3.2

However, Ferreira et al. [18] warns that non-parametric methods lead to a more difficult and

less tractable price optimization problem and may call for an approach tailored to the specific task at hand. We explain the approach to our problem in Section 4.5.

To prevent the predicted sales amount from decreasing as the applied markdown increases, we imposed a positive constraint on the features and, consequently, an increasing sales lift as a function of the price reduction. This same decision was applied in [43], where the authors also used GBMs and applied a monotonic constraint to the model.

We perform split validation to evaluate our model results, as will be further detailed in Section 5.1.

4.5 Markdown Optimization

Having implemented the PED estimation model, we now focus on the price markdown optimization (MDO).

In the studies reviewed in Section 2.3.2, most problems considered a fixed selling season and focused on maximizing revenue during that time horizon (e.g. [16, 17, 41]). Also, prices were typically computed in periodical reviews as more sales data became available throughout the season.

The retailer's clearance campaigns happen sequentially and continuously throughout the year, with products' prices being reviewed and set prior to the start of each campaign. However, our problem does not consider a fixed selling season by which we should have maximized revenue or cleared all inventory. Furthermore, the retailer may favour different markdown objectives depending on the category and product.

Ideally, the MDO problem could be solved by minimizing a cost function that accounts not only for the produced income through sales but also for the resulting potential cost of the decisions. For example, setting a price too low may leave potential gains on the table. On the other hand, setting a price too high may result in no sales and, consequently, no additional space for adding new products with possibly higher profitability. Some factors that impact the final profits include, among others, the opportunity cost, storage costs, cannibalization effects and expected revenue of newly introduced products. However, not only is it extremely complex to identify all the costs related to a pricing decision but also to quantify them correctly, ultimately leading to wrong decisions.

As introduced in Section 3.3, the managers should be able to set, for their category, the desired target for time until clearance and margin. These can be seen as business constraints to our optimal solution, similar to [42]. An option of incorporating both these objectives into a solution would be to enable the managers to attribute different weights to the objectives. This was also performed in [42], where, in the objective function, weights worked as business inputs by the retailer depending on their objectives regarding margin, sales units and revenue. However, finding the optimal weights would be a complex process, and the final output would be challenging to understand, which is what we want to avoid.

Instead, we focused on developing a solution with lower sophistication and higher explainability. For the managers to have confidence in the decisions being suggested, they need to understand the procedure and be capable of explaining why a given price markdown was or was not selected.

Also, one of the requirements of the project (see Section 3.3) was to enable the managers to perform a "what-if" analysis of the impact of the multiple price markdowns. That would not be possible if only the optimal scenario were returned.

The obtained results from the PED estimation model, described in Section 4.4, are used as input for the MDO algorithm. The sales predictions were defined for a set of discrete discount levels D relative to the product's original price, plus the baseline.

$$D = \{5\%, 10\%, 15\% \dots, 95\%\}$$

Considering this, we take a two-step approach. We first calculate the expected time until clearance for each price markdown and other relevant metrics and then proceed to suggest the optimal price, according to these estimations and the retailer's objectives.

4.5.1 Price Markdown Response

The predictions obtained through the PED estimation model regard the total expected sales, for each possible markdown, during the next campaign, which typically is approximately four weeks long. Using these predictions, we calculate for all discount levels in set D the resulting time until clearance, in weeks, for each product-store combination. To do that, we divide the available stock of the product p at store s by the respective expected weekly sales generated by a given markdown m (see Equation 4.2).

$$\text{Weeks until clearance}_{p,s,m} = \frac{\text{Available stock}_{p,s}}{\text{Weekly expected sales}_{p,s,m}} \quad (4.2)$$

During the first stages of pilot testing, we noticed, however, that we were not considering the stock in warehouses but only the in-store stock. As we are dealing with end-of-life items, most SKUs do not have stock in warehouses. Nevertheless, we were calculating an incorrect estimation of time until clearance for the SKUs that did. Furthermore, certain categories of more oversized products allocate only a single piece of each item to each store, while the remaining stock stays at a warehouse.

However, it is difficult to tell at which store the warehouse stock will be sold. To solve this problem, we developed a simple heuristic to fictionally distribute the warehouse stock between the online and physical stores and, therefore, obtain a likely sales distribution. The stock allocation of an SKU was restricted to the stores that still had not cleared it (i.e. still had at least one stock unit) and to the online store, which produces the highest sales volume out of any store. The heuristic to determine the destination of an SKU's stock units goes as follows:

1. Calculate the total sales of the respective subcategory in the previous six months at each eligible store;

2. For each store, divide the obtained subcategory sales amount by the number of stock units already allocated to that store;
3. Allocate a stock unit from the warehouse to the store with the highest sales-stock ratio;
4. Repeat steps 2 and 3 until there is no warehouse stock remaining.

As such, as stores with higher rotation are more likely to sell, they get more stock allocated to them. We then use the fictional stock at each store to calculate the time until clearance, according to Equation 4.2.

One of the project's requirements (see Section 3.3) was that managers could define if the optimal price is computed independently for each SKU-Store combination or if it should be computed at the SKU-level, i.e. each SKU has the same price at all stores.

At the SKU-level, the final time until clearance for an SKU corresponds to the time in the store that will take the longest to clear its stock, i.e. the maximum time until clearance among the stores. For example, if Store A takes five weeks to clear a given SKU and Store B takes seven weeks, then that SKU will take seven weeks to have its stock wholly cleared from all stores. Therefore, for decisions at the SKU-level, we consider the highest time until clearance among the stores in our price suggestion, typically leading to more aggressive suggestions than prices set independently for each store.

Besides the time until clearance, we also calculate the absolute and relative margins and the resulting final price produced by each discount.

The online store does not possess any actual stock and sells units from the physical stores and warehouses. Therefore, we did not directly suggest a price for SKUs at the online store. Instead, an item's price on the online channel is the same as the highest price among the stores. This criterion was approved by the managers, who reinforced that it also prevents customers from entering a store to buy an item, only to find out it is being sold at a lower price in the online channel.

4.5.2 Price Markdown Recommendation

This section explains the rationale behind the price markdown suggestion procedure.

The optimal price is suggested according to how the defined targets compare to the clearance and margin metrics, which were calculated for every discount level. Each manager sets the desired maximum number of weeks until clearance and the minimum relative margin that the selected price must ensure. The developed algorithm will then try to satisfy both these criteria with the price it suggests.

However, in some instances, more than one discount level is able to satisfy both criteria. Also, there may be situations where only one of the targets, or even none, can be met. For these reasons, the manager also selects the main priority: to maximize margins or clear the inventory as fast as possible. In such situations, the discount level will be selected according to that priority.

Initially, the chosen priority was the same for all the items in the category. However, during pilot testing, after managers' feedback, we understood that there should be a differentiation between SKUs. Therefore, the priority can now be set independently for each provision type. As explained in Section 3.1, articles with the C provision type and discontinued items, despite being in their end-of-life, may still be able to generate a decent amount of sales at higher prices. For that reason, in such items, the managers may prioritize either securing higher margins or accelerating clearance. Contrarily, provision types A and B are associated with items that have had little to no sales in previous months. These items are at a point at which they will not generate profit, and, therefore, the focus is solely on clearing their remaining stock as fast as possible. Thus, the priority for these types is predefined as clearance so that the algorithm suggests a more aggressive discount among the eligible options.

Additionally, single pieces for SKUs with no warehouse stock are also given clearance as the priority. Single pieces correspond to the last remaining piece of stock of a given article at a given store. These are often harder to sell and may even be damaged from being used as display pieces and, thus, need a more aggressive discount to be sold. As we will see in this section, the priority significantly impacts the chosen discount. Thus, this differentiation allows for a more accurate suggestion.

It is relevant to mention that, for each SKU, the corresponding number of weeks during which they have been part of previous clearance campaigns is subtracted from the defined clearance target. For example, a given SKU that has had eight weeks' worth of clearance campaigns gets that number of weeks deducted from the value defined by the manager, setting a new target for that particular item. This operation results in a more aggressive discount selection for these SKUs compared to others that got discontinued more recently. To avoid the suggestion of overly aggressive discounts (e.g. 95%), the resulting final target, after the deduction, is limited to a minimum of 4 weeks, which currently corresponds to the approximate duration of a campaign.

To more easily understand the explanation of the price suggestion procedure, let us consider, for the examples throughout this section, a pricing decision for an SKU entering its first stock clearance campaign and whose category's defined targets are:

- **Maximum Time Until Clearance:** 12 weeks
- **Minimum Margin:** 5%
- **Priority:** Maximize Margin

The first step of the procedure is to identify, based on the estimated metrics, the lowest discount level that satisfies the clearance target and the highest discount level that satisfies the minimum margin target. In other words, for the clearance objective, we determine the minimum discount level whose estimated time until clearance duration is lower than the target weeks. We then follow the same procedure for margin, where we find the maximum price discount allowed before the margin is lower than desired.

Figure 4.3 shows a representation of this step, considering the targets defined earlier in the example. The levels that satisfy the clearance condition (≤ 12 weeks) are depicted in blue, and the levels that satisfy the margin objective ($\geq 5\%$) are in green. In this case, the lowest eligible discount that respects the clearance target is the 25% level, and the 40% level is the highest eligible discount for margin before it becomes lower than desired. The margin and weeks until clearance shown in the following figures are merely examples and do not correspond to any actual instance.

	Discount Level	Weeks Until Clearance	Unit Margin (%)
	...	...	...
	15%	25	40%
Lowest Eligible Discount for Clearance →	20%	16	32%
	25% CDL	10	26%
	30%	8	17%
Highest Eligible Discount for Margin →	35%	6	10%
	40% MDL	4	5%
	45%	2	-4%
	...	...	...

Figure 4.3: Determining the limiting levels of the price markdown suggestion

For ease of comprehension, we henceforth refer to the lowest eligible discount level for the clearance target as CDL and to the highest eligible discount level for the margin target as MDL. Once these are determined, we proceed to select the final price markdown among the eligible options.

Considering this, we directly determine the price markdown according to the rules in Table 4.1. When priority is clearance, the selected price discount will correspond to the highest discount value of the two limiting levels (CDL and MDL). If priority is margin, the selected discount level will be the lowest among the two, resulting in a higher price and, consequently, a higher margin. The final suggestion always ends up being one of the limiting levels.

Table 4.1: Selecting which discount level to suggest, according to the priority

Priority	Suggested Markdown
Minimize time until clearance	$\max\{CDL, MDL\}$
Maximize margins	$\min\{CDL, MDL\}$

Let us consider a set of examples to better explain the reasoning behind these rules. Once CDL and MDL are determined, one of the following situations may occur while selecting which markdown to apply:

- A) Lowest Eligible Clearance Discount (CDL) = Highest Eligible Margin Discount (MDL):**
as seen in Figure 4.4, only one discount level satisfies both criteria simultaneously. Thus, in

this case, the priority has no influence and the single eligible level is selected as the optimal discount.

	Discount Level	Weeks Until Clearance	Unit Margin (%)
	...	...	...
	15%	25	32%
	20%	18	23%
	25%	14	15%
Suggestion →	30% CDL MDL	11	7%
	35%	8	1%
	40%	6	-6%
	45%	5	-12%
	...	...	...

Figure 4.4: Example of price suggestion in situation A

B) Lowest Eligible Clearance Discount (CDL) < Highest Eligible Margin Discount (MDL): more than one discount level satisfies both the clearance and the margin targets. Among the eligible levels that resulted from the intersection of the targets, the selected level corresponds either to the markdown that maximizes margins or the markdown that minimizes time until clearance, depending on the priority.

	Discount Level	Weeks Until Clearance	Unit Margin (%)
	...	...	...
	15%	25	40%
	20%	16	32%
Suggestion →	25% CDL	10	26%
	30%	8	17%
	35%	6	10%
	40% MDL	4	5%
	45%	2	-4%
	...	...	...

Figure 4.5: Example of price suggestion in situation B

In the situation depicted in Figure 4.5, four levels satisfy both criteria. According to the rules in Table 4.1, since the priority in our example is maximizing margins, we opt for the lowest discount among the eligible options, which corresponds to CDL.

C) Lowest Eligible Clearance Discount (CDL) > Highest Eligible Margin Discount (MDL): there is no intersection, i.e., no discount level satisfies both targets simultaneously, as depicted in Figure 4.6.

	Discount Level	Weeks Until Clearance	Unit Margin (%)
	...	...	...
	15%	37	18%
Suggestion →	20% MDL	32	10%
	25%	26	4%
	30%	18	-3%
	35%	14	-10%
	40% CDL	9	-18%
	45%	7	-24%
	...	...	...

Figure 4.6: Example of price suggestion in situation C

Since we considered margin the priority, the suggestion will correspond to the discount level that satisfies the margin target and is the closest to the clearance objective. Therefore, according to the rules in Table 4.1, the suggestion corresponds to MDL, which is the lowest discount among the two. This way, the selected discount satisfies the priority objective while being as close as possible to the unfulfilled target.

D) Priority target cannot be satisfied: in this rare situation, no discount level ensures that the priority target is met. If the priority is to clear the inventory, but the target weeks cannot be attained even by the most aggressive discount, the suggestion corresponds to the maximum possible price markdown (CDL = 95% off). If the priority is to maximize margins and all possible discounts result in lower than desired margins, then the model suggests that no discount is applied (MDL = 0%). However, this is a rare situation and likely caused by the definition of overly ambitious or unrealistic targets.

The suggested discount must respect the market rule inherent to markdowns: the new price must not be higher than the prices used in previous campaigns. Also, this ensures that the SKU's price is never higher than its lowest price in the previous 30 days, a recently introduced rule that retailers must follow. If the optimal price does not comply with this rule, the suggested markdown will correspond to the lowest possible discount that respects this restriction. Additionally, for operational matters, the prices for these items must end with the number 7.

The price suggestions, the alternative options and their respective impact on the related metrics can be analyzed in the developed tool, where the managers can validate the proposed prices.

4.6 The Stock Clearance Tool

The SCT is the project's main deliverable, and it is, essentially, an interface that incorporates the solution's components in a way that provides value and utility to the end-users, i.e. to the retailer's commercial managers. The construction of the tool corresponds to the Deployment stage of the CRISP-DM cycle. It was developed in Microsoft Excel, using VBA (Visual Basic for Applications), due to its simplicity of development and deployment and to the managers' familiarity with this software.

The tool was developed according to the managers' needs and feedback to ensure they have access to all the relevant information in the interface. It incorporates the model predictions for each discount level, the price suggestions and several relevant metrics to help make the decisions. Not only does it allow for better-supported decision-making, but the tool also makes the pricing process less exhausting and time-consuming, a factor the managers highly value.

From the tool's home screen (Figure 4.7), each manager can select their category and the campaign they are deciding on and also set the targets for that campaign. Once the desired parameters are set (through the menu shown in Figure 4.8), the managers update the data so that the tool returns the most recent results and suggestions, considering the newly defined targets. From this screen, the user can also navigate to the other screens.

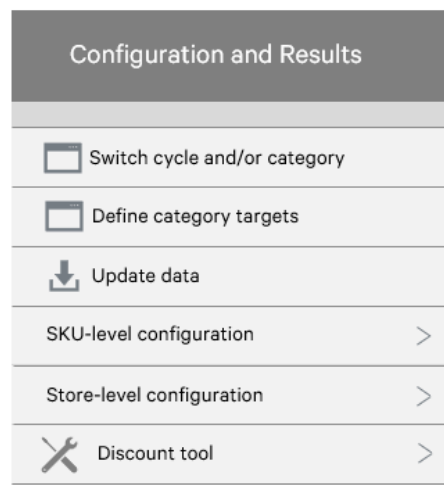


Figure 4.7: Stock Clearance Tool's home page

SKU-level Configuration

In the SKU-level Configuration screen, the manager has access to all the information regarding the products' hierarchy and attributes, the model outputs and several relevant metrics of all the SKUs in the campaign. These metrics include the stock quantity and value, rotation days, provision type, number of past campaigns, and several others. Each line on the screen, as shown in Figure 4.9 corresponds to a different SKU. Some values in the figures are hidden for confidentiality reasons.

Set category targets

Current criteria and target

Weeks until clearance:
18

Gross Margin (%):

Priority criteria:

Single Pieces:
Weeks until clearance

A & B:
Weeks until clearance

C:
Gross margin (%)

Descontinuados:
Gross margin (%)

Optimization criteria

Max Weeks until clearance:
18

Min Gross Margin (%):

Optimization priority

Descontinuados:
☐ Weeks until clearance
☒ Gross Margin (%)

C:
☐ Weeks until clearance
☒ Gross Margin (%)

Omnichannel

☒ Activate omnichannel prices for this category

Settings

☐ Only products with stock at the central warehouse

Number of weeks since clearance status

Apply

Figure 4.8: Parameter setting at the Stock Clearance Tool

Navigation			Product Attributes									
Store	Disc.	SKU	€/unit	units	units	€	units	€	Provision Type	No.	€	No.
			Unit Final Cost	Stock in Warehouse	Stock in Stores	Stock Value	Sales (last 30 days)	PVP Assortment		Stores w/ stock	Min Price (30 days)	Past Campaigns
		7898910	239,97 €	0	151	23 472,29 €	49	279,99 €	Descontinuado	96	239,97 €	3
		7898915	999,00 €	0	18	23 200,22 €	0	1 999,00 €	C	18	1 099,97 €	1
		7898916	1 899,00 €	2	11	20 159,37 €	4	2 749,00 €	Descontinuado	11	1 899,00 €	1
		7898917	1 999,00 €	0	7	17 004,82 €	1	3 599,00 €	Descontinuado	7	1 999,00 €	1
		7898918	2 599,00 €	0	4	16 195,08 €	1	5 999,00 €	Descontinuado	4	2 599,00 €	1
		7898919	2 899,00 €	0	6	15 605,88 €	0	3 999,00 €	B	6	2 899,00 €	0
		7898920	2 249,97 €	0	4	14 845,52 €	0	5 499,00 €	Descontinuado	4	2 249,97 €	2
		7898921	999,99 €	0	16	14 805,57 €	1	1 399,00 €	Descontinuado	15	999,99 €	0
		7898922	439,97 €	0	21	14 682,78 €	4	999,99 €	Descontinuado	20	439,97 €	2
		7898923	1 499,97 €	0	9	14 444,08 €	0	2 499,00 €	Descontinuado	9	1 499,97 €	1
		7898924	1 199,97 €	0	10	13 983,70 €	1	1 999,00 €	Descontinuado	10	1 199,97 €	1
		7898925	1 299,00 €	0	7	13 703,97 €	1	2 799,00 €	Descontinuado	7	1 299,00 €	1
		7898926	599,97 €	0	20	13 081,30 €	3	949,99 €	C	14	599,97 €	1
		7898927	566,97 €	0	15	12 585,30 €	3	1 199,00 €	Descontinuado	12	566,97 €	2

Figure 4.9: SKU-level Configuration page

In this screen, the manager can check and validate the price suggestions for each SKU and the resulting metrics for margin and time until clearance. At the SKU-level, the presented prices and discount percentages correspond to the respective average of all stores, weighted by stock quantity.

Suppose the manager disagrees with the suggested price for a given SKU. In that case, he can

manually choose a different discount level, immediately observing the impact of the change in the related metrics. Since these price changes are being made at the SKU-level, they get applied to every store with stock of the respective SKU.

This analysis can also be done for each store individually, by selecting the desired SKU. In the line of the respective SKU, there is a button that, when pressed, directs the user to the Store-level Configuration screen.

Store-level Configuration

This screen works similarly to the previous one. Here are presented all the stores which contain stock units of the selected SKU, where each line corresponds to a different store. Similar to the SKU-level Configuration screen, each store has its own attributes and metrics. It is equally possible to visualize the suggested decisions and their respective impact on clearance and margin, as shown in Figure 4.10.

Navigation		Store Hierarchy		Model Outputs						
Disc.	Location Code	Location Description	€	%	weeks	€/unit	€/unit	%	%	On Target?
			Discount PVP	% Discount PVP	Weeks until	Unit Margin w/ Prov.	Unit Margin wo/ Prov.	% Margin w/ Prov.	% Margin wo/ Prov.	
			230,97 €	30%	9					●
			230,97 €	30%	10					●
			230,97 €	30%	10					●
			230,97 €	30%	6					●
			230,97 €	30%	10					●
			230,97 €	30%	3					●
			230,97 €	30%	3					●
			230,97 €	30%	2					●
			230,97 €	30%	3					●
			230,97 €	30%	10					●
			230,97 €	30%	10					●
			230,97 €	30%	10					●
			230,97 €	30%	3					●
			230,97 €	30%	10					●
			230,97 €	30%	9					●

Figure 4.10: Model outputs at the Store-level configuration page

The suggested price markdowns may also be adjusted by the manager, if necessary. At the Store-level, manual adjustments to the suggested decisions can be made for each store independently, affecting only the price of the SKU at the store it concerns.

Discount Tool

The user can access the Discount Tool screen for a more detailed view, as shown in Figure 4.11. It gives the manager a view of all possible discount levels and their respective impact on the related metrics, highlighting the markdown suggestion. It is also possible to manually change the decision if intended.

This screen also presents, among others, graphs of the prices practised in previous campaigns and the number of sales in previous months, to aid the manager in the decision.

More figures regarding the SCT can be seen in Appendix A.

Discount table									Discount decision	
%	€	units	weeks	€/unit		%				
Discount level	PVP Discount	Weekly expected sales	Time until Clearance	Unit Gross Margin		Unit Gross Margin			Model suggestion	Manual decision
				w/ provision	wo/ provision	w/ provision	wo/ provision			
0%	549,97 €	1,03	7,74	11,00%	100,00%	11,00%	100,00%			
5%	522,47 €	1,05	7,64	11,00%	100,00%	11,00%	100,00%			
10%	494,97 €	1,08	7,42	11,00%	100,00%	11,00%	100,00%			
15%	467,47 €	1,12	7,12	11,00%	100,00%	11,00%	100,00%			
20%	439,97 €	1,18	6,76	11,00%	100,00%	11,00%	100,00%			
25%	412,47 €	1,26	6,36	11,00%	100,00%	11,00%	100,00%			
30%	384,97 €	1,35	5,92	11,00%	100,00%	11,00%	100,00%		<---	
35%	357,47 €	1,46	5,47	11,00%	100,00%	11,00%	100,00%			
40%	329,97 €	1,60	5,01	11,00%	100,00%	11,00%	100,00%			
45%	302,47 €	1,76	4,55	11,00%	100,00%	11,00%	100,00%		<---	
50%	274,97 €	1,95	4,10	11,00%	100,00%	11,00%	100,00%			
55%	247,47 €	2,18	3,67	11,00%	100,00%	11,00%	100,00%			
60%	219,97 €	2,45	3,26	11,00%	100,00%	11,00%	100,00%			
65%	192,47 €	2,78	2,88	11,00%	100,00%	11,00%	100,00%			
70%	164,97 €	3,17	2,52	11,00%	100,00%	11,00%	100,00%			
75%	137,47 €	3,64	2,20	11,00%	100,00%	11,00%	100,00%			
80%	109,97 €	4,21	1,90	11,00%	100,00%	11,00%	100,00%			
85%	82,47 €	4,90	1,63	11,00%	100,00%	11,00%	100,00%			
90%	54,97 €	5,73	1,40	11,00%	100,00%	11,00%	100,00%			
95%	27,47 €	6,75	1,18	11,00%	100,00%	11,00%	100,00%			

Figure 4.11: Model suggestion and What-If analysis at the Discount tool screen

Chapter 5

Results

This chapter presents the results and metrics obtained during the solution’s implementation and validation. Section 5.1 evaluates the performance of the PED estimation model. Section 5.2 reviews the acceptance of the price suggestions and presents the results obtained during the performed pilot test.

5.1 Model Performance

The project’s ultimate goal is to improve the retailer’s markdown decisions by recommending price markdowns. As described in Section 4.5, these suggestions depend on the margins and expected clearance rates related to each price markdown. Time until clearance is obtained from the predicted sales amount. Hence, the suggestions’ quality is highly dependent on the accuracy of these predictions. Before the deployment of the solution, and according to the CRISP-DM methodology, there is an evaluation phase in which the quality of the developed model is assessed. In this section, we evaluate the performance of the developed PED estimation model.

We used MAPE and BIAS as performance metrics, which we introduced in Section 2.2.5. Percentage errors are unit-free and suitable for comparing forecast performances between different datasets.

Additionally, MAPE’s ease of interpretation is a relevant factor in contexts which require the explanation of a forecast’s accuracy to non-experts in forecasting, which is the case in this work [13].

Unlike MAPE, which considers the absolute error, the BIAS (Equation 2.14) captures the model’s tendency to provide overly high or overly low forecasts.

As we are dealing with discontinued items, sales are typically low. For that reason, by following these metrics, even minor absolute errors can result in significant percentage variations, potentially leading to deceiving conclusions.

Considering this, we use another method, a simple naïve approach, to work as a reference to compare our model. We define this reference using a simple forecasting method, performed at the subcategory-level, consisting of the historical average of the stock clearance rate produced by each

price markdown, i.e., for each discount level. Despite being elementary, the average method can be surprisingly effective [20]. To calculate SKU-Store level expected sales, the stock of each product-store combination is multiplied by the clearance rate obtained for the respective subcategory by the naïve approach. By comparing these two methods, we can make clearer conclusions regarding the quality of our model’s results.

The dataset with previous campaign information, ranging from January 2021 to May 2022, was split into training and testing data, using the data since November 2021 to evaluate the results. The comparison of the metrics between our model and the naïve approach is presented in Table 5.1.

Table 5.1: Performance metric comparison

Category	GBMs		Naïve Method	
	MAPE	BIAS	MAPE	BIAS
Mobile Phones	30.2%	1.5%	40.0%	-17.3%
Washing Machines	25.8%	3.6%	50.6%	-50.0%
Refrigerators	25.5%	3.0%	51.9%	-50.1%
Food Preparation	30.4%	-5.4%	55.8%	-45.7%
Global	28.5 %	2.4%	43.0%	-25.1%

The developed model reached better results, as MAPE and BIAS present lower values. The naïve approach has a negative BIAS in all categories, meaning that the forecasts are constantly lower than the actual sales. Due to limitations in the available data, the used testing period is short. Promotional activity during some months of this period is typically high. As we do not apply the seasonal factor in the naïve approach, it results in a significant negative BIAS, underestimating sales. If campaign data over a more extended period were available, we would have favoured performing validation over an entire year.

On the other hand, our model’s BIAS is close to zero, meaning that it did not capture any general tendency to overestimate or underestimate sales. The MAPE results, despite being better than those obtained for the naïve approach, are still far from optimal.

5.2 Pilot Testing

Once a first version of the solution was deployed, we ran a pilot test on the four pilot categories aforementioned in Section 3.1 to validate the solution and find opportunities for improvement. The price suggestions for the campaign were validated by the respective managers through the developed tool. The main objectives were to evaluate how often the managers reject our price recommendations and to compare the operational results to previous years. Depending on how well the solution performs, the adoption of the DSS will progressively be extended to the remaining categories during the following campaigns.

During the week prior to the start of the campaign, we held meetings with the category managers, the end-users of the tool, to explain it and gather their feedback on its usability and results.

Along with the managers, we analyzed some products' elasticity curves and markdown suggestions to get their input on whether these were accurate or not, based on their experience, so we could make corrections to improve the solution.

5.2.1 Price Recommendation

While validating the price suggestions in the tool, the managers can reject them and define a different price. One of the goals of the pilot test is to evaluate the suggestion acceptance and understand how the price changes differ from the suggestion.

These metrics would enable us to spot tendencies (e.g. the price suggestions being too aggressive and the managers often having to define lower markdowns) and provide input to make corrections and adjust the model. Our focus was on increasing the suggestion acceptance rate, as it is an excellent indicator to assess managers' confidence in the system, which is the most relevant factor in the path towards a fully automated price markdown process with no human intervention.

Table 5.2 shows the amount of SKUs in the pilot categories that entered the first stock clearance campaign after the launch of the developed tool.

Table 5.2: Number of SKUs, per category, included in the campaign

Category	No. SKUs
Mobile Phones	142
Washing Machines	35
Refrigerators	38
Food Preparation	46

Table 5.3 shows the SKU distribution by provision type. As we can see, the number of SKUs included in the campaign can highly differ across categories due to their specific natures. Also, the vast majority of the items are non-provisioned SKUs.

Table 5.3: Number of SKUs, per provision type, included in the campaign

Provision Type	No. SKUs
A	17
B	13
C	10
Discontinued	221

As explained in Section 4.5.2, for this first campaign, the chosen priority (clearance or margin) was the same for all items in each category, independently of their provision type. However, non-provisioned items still possess a considerable amount of sales when compared to provisioned SKUs, where often only extremely aggressive markdowns had a chance at clearing the stock. Since most items are non-provisioned, all managers prioritized protecting margins when defining the targets in the tool. Therefore, the chosen discount would always correspond to the least aggressive among the eligible options. Table 5.4 contains the suggestion acceptance rates at the SKU-level by

category and provision type. Therefore, the values correspond to the percentage of SKUs whose price suggestions were left unchanged.

Table 5.4: Percentage amount of SKUs with accepted price suggestions

Provision Type	Category				
	Mobile phones	Washing Machines	Refrigerators	Food Preparation	Global
A	100%	100%	100%	100%	100%
B	100%	100%	100%	80%	92%
C	100%	100%	100%	60%	80%
Discontinued	80%	97%	96%	47%	79%
Global	81%	97%	97%	54%	81%

The managers have shown different postures regarding their suggestion acceptance. Some were more permissive and only changed the products' prices when they largely disagreed with the suggestion. Others were more doubtful about the tool and would alter prices even if they were barely off. The majority of price changes occurred for the discontinued items (with no provision) due to their higher value and stock amounts. Most prices for the provisioned SKUs have remained untouched since these are lower in stock quantity and value. Hence, the managers are more open to aggressivity in the markdowns. Nevertheless, the overall number of SKUs with changed prices was reduced for most categories.

However, analyzing the results at the SKU-Store level allows for different conclusions. Table 5.5 shows the percentage amount of product-store combinations whose price suggestion was accepted.

Table 5.5: Percentage amount of product-store combinations with accepted price suggestions

Provision Type	Category				
	Mobile phones	Washing Machines	Refrigerators	Food Preparation	Global
A	100%	100%	100%	100%	100%
B	100%	100%	100%	50%	76%
C	100%	100%	100%	64%	83%
Discontinued	37%	99%	74%	76%	53%
Global	38%	99%	75%	76%	54%

For certain categories, the results at this level of aggregation highly differ from the ones seen at the SKU-level. For example, in the *Mobile Phones* category, despite only having 19% of SKUs with the suggestion rejected, just 38% of SKU-Store combinations remained unchanged. This disparity is due to the fact that some SKUs are present in many stores while others have a low significance in the category's total inventory.

Therefore, the central realization to be drawn from this is that these price changes are mainly made at SKUs whose stock is still significant, mostly corresponding to non-provisioned items. Therefore, the managers tend to be more contained in the prices they give to these SKUs to protect margins and profit.

Table 5.6 compares our tool’s suggested discounts and the final discounts after the applied changes. The main takeaway is that, as expected, the managers opt for lower-than-suggested markdowns and try to protect margins, besides the fact that these are end-of-life items whose stock must be cleared. This suggests that we are being overly aggressive in our price recommendations.

Table 5.6: Average discount off, before and after the price validation

Category	Average Suggested Markdown	Average Final Markdown	Difference
Mobile Phones	39.66%	20.11%	-19.55%
Washing Machines	32.74%	30.00%	-2.74%
Refrigerators	41.47%	30.00%	-11.47%
Food Preparation	45.08%	32.60%	-12.48%
Global	39.95%	25.08%	-14.87%

We then applied several changes to our solution with the intent of giving more accurate suggestions and improving the acceptance rates. For the next campaign, we studied the average time SKUs take to clear their stock after they get discontinued. We concluded that the clearance target defined by most managers in the first pilot campaign was overly ambitious, which resulted in highly aggressive markdowns. Therefore, we proposed different targets to set in the tool. These were still ambitious objectives but reflected better the reality.

Additionally, as explained in Section 4.5.2, one of the changes to our solution was allowing the managers to now set priorities individually for each provision type. Provisioned and non-provisioned items could now have separate objectives. The focus with the provisioned items was typically on clearing them as fast as possible and, with the non-provisioned, on ensuring a minimum financial return. We also started including the in-warehouse stock, which we had been ignoring up until this point. To this effect, we considered a likely future sales distribution of these stock units across the stores (explained in Section 4.5.1) and used it, plus the in-store stock, to calculate products’ time until clearance.

This campaign also saw the inclusion of another pilot category. The managers have already reviewed the suggested prices for this next campaign, whose acceptance rates are shown in Table 5.7.

Table 5.7: Price suggestion acceptance rates for the next stock clearance campaign

Category	SKU-level	SKU-Store-level
Mobile Phones	98.1%	84.4%
Washing Machines	100%	100%
Refrigerators	100%	100%
Televisions	98.3%	99.6%
Food Preparation	100%	100%
Global	98.8%	93.8%

The implemented changes resulted in a clear improvement of the acceptance rates, with almost no rejected suggestions. The managers expressed satisfaction with the results and claimed that the

implemented solution also made the markdown pricing process more time-efficient.

5.2.2 Stock Clearance Results

This section analyses the results obtained during the first campaign, with the prices defined through the DSS.

Figure 5.1 shows the percentage of stock cleared in the first pilot testing campaign compared to last year's campaign in the same period. The results shown for this year's campaign refer only to the items whose price suggestions were accepted by the managers. As we can see, for these items, stock clearance was higher this year for all pilot categories than it was for last year's products.

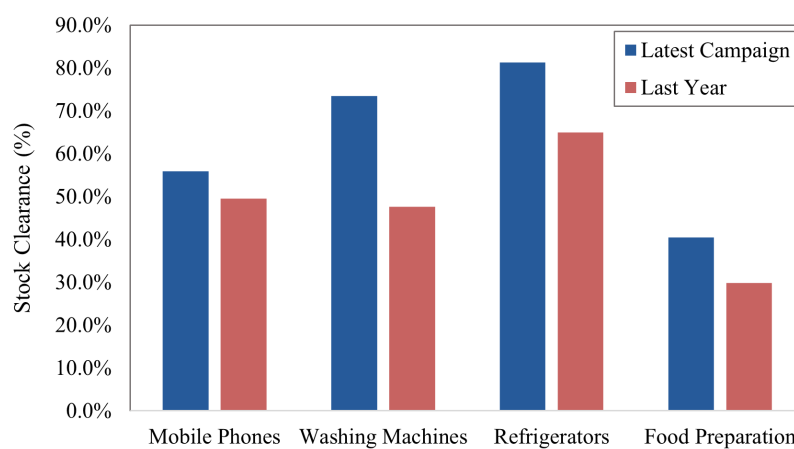


Figure 5.1: Stock cleared (Latest campaign vs. Last year's homologous period campaign)

Overall, now considering all items in this year's campaign, 6.6% more stock was cleared this year than in the same period last year. Also, the relative margin was 13,5% higher on average. However, despite being extremely positive results, these are not enough to conclude the effectiveness of our solution. The products included in both campaigns are entirely distinct, as they regard different years. Stock rotation and store distribution may also differ highly, and several external factors can also impact these results. Therefore, in the short term, it is not simple to assess clearly how the solution is performing. Nevertheless, these are promising results.

Chapter 6

Conclusion

This closing chapter provides an overview of the dissertation and its main conclusions. Section 6.1 summarizes the developed work. Section 6.2 mentions the main contributions. Section 6.3 closes the document with suggestions for future work.

6.1 Summary

This dissertation arises in the context of a project with a large Portuguese consumer electronics retailer, whose main objective is to improve their markdown decisions and their management regarding products in the decline stage of the PLC, reaching their end-of-life.

As we have seen, in retail, inventory is often bought in advance, with long lead times and based on poor initial demand predictions. This may result in lower sales than expected and unsold stock, ultimately leading to a loss in revenue. In such situations, retailers often resort to price markdowns, i.e., permanent price reductions, to generate more demand for the product and clear its remaining stock before it becomes obsolete.

The retailer in this project relied on a standardized approach for their markdown pricing decisions, consisting of a set of rules defined empirically and based on the managers' experience, with no analytical foundation. Furthermore, it was a rather inefficient process, as managers needed to review prices for the hundreds of product-store combinations.

Therefore, this project aimed to help the retailer with their markdown pricing processes by allowing them to make these decisions based on actual data and making the process more efficient and automated.

To this effect, we developed a DSS integrating multiple components, with the ultimate goal of optimizing price markdowns to meet the retailer's stock clearance and margin objectives. It comprises a PED estimation model and a price markdown recommendation algorithm. The primary function of the first is to compute the impact of different price markdowns on sales using ML techniques. With the predicted sales amounts for the different price markdowns, we suggest the optimal price to meet the retailer's objectives, finding a balance between clearing the stock as fast as intended and maximizing revenue.

These components are incorporated into an intuitive interface, where the managers validate the price suggestions and where they have access to all the information needed to make their decisions. Analogously, besides the suggestion, the managers can analyze what impact other price markdowns would have on clearance and associated metrics for every product at each store.

Finally, we ran a pilot test on four categories with distinct characteristics to validate the quality of the developed solution. This test consisted in using the developed tool to price the products for a stock clearance campaign. We gathered manager feedback regarding the usability of the tool, the price suggestions and the sales estimations, to work on improving our solution for the following campaigns and increase the suggestion acceptance. Our solution increased stock clearance and margin rates compared to the homologous period campaign last year. It was unanimously accepted as it provided value to the retailer, with the managers stating that it made the process more time efficient. The tool will continue on an experimental phase during the following campaigns, and the project will then be progressively extended to integrate the remaining categories of the retailer.

6.2 Main Contributions

This dissertation provided an in-depth literature review on markdown optimization (MDO) and related theoretical knowledge, including PED and forecasting. As mentioned, the literature on MDO in the consumer electronics industry seems scarce. Therefore we argue that the presented dissertation and case study may be an added contribution to help fill this gap.

Additionally, we developed a solution that uses PED estimations to compute different price markdown scenarios and suggest the optimal price depending on the retailer's priority and objectives. Most reviewed studies tackle revenue maximization problems but do not consider the costs of holding stock. However, in the context of end-of-life items with already low profitability, the retailer may favour clearing the stock faster, even if it means sacrificing margins. Higher clearance may translate into lower storage costs and additional space for future products with higher profitability, potentially leading to better results in the long term. Hence, we provide a different, intuitive and highly explainable approach to MDO considering both clearance and margin objectives, which the retailer's managers directly define according to their needs.

Furthermore, the literature mainly covers cases concerning a fixed time window, by the end of which revenue must be maximized, with the remaining stock being given, discarded or sold at a low salvage price to a third-party wholesaler or retailer. In this dissertation, however, we tackle a problem where we do not have a fixed season. Instead, we have more liberty to set the prices and protect margins, with unsold items having more opportunities to be sold during the following campaigns. Therefore, we follow a different approach that directly considers the retailer's interests. As a reference, instead of a fixed selling season, we consider managers' preferences for clearance and margin and suggest a balanced markdown that conjugates both criteria.

The developed interface, as the project's main deliverable, incorporates these components and deploys the solution in a way that is usable by the retailer's managers. The solution was tested

in a real-life situation and has shown to bring value to the retailer, contributing to making their markdown pricing processes more efficient and reliable.

6.3 Future Work

Considering the reviewed literature and the work developed, we present opportunities for further research and possible extensions and improvements of the implemented solution.

An interesting feature to be added to our price suggestion algorithm would be to evaluate the margin and clearance trade-off between contiguous discount levels, when selecting what price to suggest. In other words, to understand if the faster clearance rate ensured by a lower price would compensate for the reduction in margin relative to the previous discount level. For example, in instances in which two different prices have, according to our predictions, a minimal difference in clearance, the higher price would be the better option due to the added profitability, and vice-versa. We tried applying this principle in our suggestions through gradient techniques, but they proved ineffective. However, here lies an opportunity for additional flexibility and optimality for our solution.

One of the requests of the managers, following their feedback on our solution, was to not restrict the price selection to such a small set of discount levels. Instead, generating predictions for more discount levels or working with continuous price trajectories would be options for further improvement.

As we focused on developing a solution that could apply to any category, the selected predictor variables were limited to the ones available in all categories. However, if made available, including category-specific attributes (e.g. colour, screen size, processor) would surely increase the model's predictive power. Clustering similar products based on their characteristics and historical sales also seems to be a systematic approach in the reviewed literature. It could be particularly useful for predicting the behaviour of products that lack data regarding their response to price changes, such as recently discontinued products or provisioned items with no recent sales.

One aspect that seemed evident throughout the work is that many products had reached such a stage of obsolescence that they became a burden to the retailer, as they were almost impossible to sell, no matter the price. Therefore, extending our approach to the earlier stages of the PLC could be beneficial for extracting the maximum revenue out of every product before they become obsolete. It would require a continuous estimation of products' sales curves throughout their PLCs to detect underperforming items deviating from this projection. Then, for those items, preventive measures would be taken, such as lower price markdowns earlier in the cycle, to keep them from being discontinued and to extract the highest possible revenue from them.

Another frequent occurrence in retail is that certain stores take much longer than others to clear their stock due to being in smaller and less populated locations or other reasons. An improvement that the solution would benefit from is to optimally suggest a destination to transfer the stock units to, from stores with lower rotation to stores that historically sell faster and at higher prices. The solution would highly depend on the circumstances and the stock quantities, and would have to

consider the distance between stores, the costs of the operation and other factors that might not be easy to estimate.

Finally, we argue that an ideal MDO approach should consider not only the generated income but all the costs and effects inherent to a markdown decision. These would include, among others, storage and transportation costs, cannibalization and competition effects and the expected added profitability generated by the future products that would fill the space left by the cleared items. All these factors impact how quickly stock for an item should be cleared and, consequently, its price. We found no case study in the literature that could solve all these issues. In several of them, the authors explicitly state that they chose not to model these variables due to their high complexity. Some of these are incredibly complicated to estimate, and, therefore, a great deal of work could still be done in this direction.

Appendix A

The Stock Clearance Tool

SKU		Store		Discount		Load Data		SKU					
Navigation Store Hierarchy						Product Attributes							
Disc.	Location Code	Location Description	DOP	Insignia	Store Typology	€/unit	units	€	units	€	€	€	€
						Unit Final Cost	Stock Qty	Stock Value	Sales (last 30 days)	PVP Assortment	Last PVP	Provision Type	Provision Unit Value
	1339				GG		8	2 876,65 €	6	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	538				E		8	2 876,65 €	3	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	545				GG		8	2 876,65 €	5	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	1038				GG		7	2 517,07 €	4	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	1171				GG		7	2 517,07 €	2	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	424				E		6	2 170,34 €	0	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	541				E		6	2 157,49 €	0	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	588				E		6	2 157,49 €	0	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	1450				GG		6	2 157,49 €	0	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	535				GG		5	1 797,91 €	4	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	1036				GG		5	1 797,91 €	1	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	1856				E		5	1 797,91 €	1	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	866				E		5	1 797,91 €	1	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	534				E		5	1 797,91 €	1	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	1912				E		5	1 797,91 €	0	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	556				E		4	1 441,32 €	0	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	529				E		4	1 438,32 €	1	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	984				E		4	1 438,32 €	1	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	1091				E		4	1 438,32 €	0	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	593				E		4	1 438,32 €	1	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	599				E		4	1 438,32 €	3	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	1037				GG		4	1 438,32 €	1	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	39				E		3	1 078,74 €	3	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	537				E		3	1 078,74 €	1	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	5883				E		3	1 078,74 €	1	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	536				E		3	1 078,74 €	2	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	587				E		3	1 078,74 €	0	549,99 €	409,99 €	Descontinuado	0 € 349,99 €
	1034				E		3	1 078,74 €	1	549,99 €	409,99 €	Descontinuado	0 € 349,99 €

Figure A.1: Product and store attributes at the Store-level Configuration page



Figure A.3: Model outputs with the price suggestions and manual validation through price adjustments



Figure A.4: Discount tool screen at the Stock Clearance Tool

References

- [1] William E. Cox. Product life cycles as marketing models. *The Journal of Business*, 40(4):375–384, 1967.
- [2] Antti Saaksvuori and Anselmi Immonen. *Product Lifecycle Management*. Springer Books. Springer, 3rd edition, 01 2008.
- [3] Arie Goldman. Short product life cycles: implications for the marketing activities of small high-technology companies. *R & D Management*, 12(2):81–90, 1982.
- [4] Stephen A. Smith and Dale D. Achabal. Clearance pricing and inventory policies for retail chains. *Management Science*, 44(3):285–300, 1998.
- [5] R.L. Phillips. *Pricing and Revenue Optimization*. Stanford Business Books. Stanford University Press, 01 2005.
- [6] Michael Levy, Dhruv Grewal, Praveen K. Kopalle, and James D. Hess. Emerging trends in retail pricing practice: implications for research. *Journal of Retailing*, 80(3):xiii–xxi, 2004.
- [7] Steffen Christ. *Operationalizing Dynamic Pricing Models: Bayesian Demand Forecasting and Customer Choice Modeling for Low Cost Carriers*. Gabler Verlag, 1st edition, 2011.
- [8] Stefan Niemeier, Andrea Zocchi, and Marco Catena. *Reshaping retail: why technology is transforming the industry and how to win in the new consumer driven world*. John Wiley & Sons, 1st edition, 2013.
- [9] Steven A. Greenlaw and David Shapiro. *Principles of Microeconomics 2e*. Rice University, Houston, Texas, 2017.
- [10] Marine Le Gall-Ely. Definition, measurement and determinants of the consumer’s willingness to pay: A critical synthesis and avenues for further research. *Recherche et Applications en Marketing (English Edition)*, 24(2):91–112, 2009.
- [11] Alfred Marshall. *Principles of Economics*. Palgrave Macmillan London, 1890.
- [12] Robert T. Jensen and Nolan H. Miller. Giffen behavior: Theory and evidence. Working Paper 13243, National Bureau of Economic Research, July 2007.
- [13] Robert Fildes, Shaohui Ma, and Stephan Kolassa. Retail forecasting: Research and practice. *International Journal of Forecasting*, 2019.
- [14] Kejia Hu, Jason Acimovic, Francisco Erize, and Jan A. Van Mieghem. Forecasting product life cycle curves: Practical approach and empirical analysis. *SSRN Electronic Journal*, 2016.

- [15] Stephen J. Hoch, Byung-Do Kim, A. Montgomery, and Peter E. Rossi. Determinants of store-level price elasticity. *Journal of Marketing Research*, 32, 1995.
- [16] Nizar Zaarour, Emanuel Melachrinoudis, and Marius M. Solomon. Maximizing revenue of end of life items in retail stores. *European Journal of Operational Research*, 255(1):133–141, 2016.
- [17] Felipe Caro and Jérémie Gallien. Clearance pricing optimization for a fast-fashion retailer. *Operations Research*, 60(6):1404–1422, 2012.
- [18] Kris Ferreira, Bin Lee, and David Simchi-levi. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, 18, 11 2015.
- [19] P.W. Farris, N. Bendle, P.E. Pfeifer, and D. Reibstein. *Marketing Metrics: The Definitive Guide to Measuring Marketing Performance*. Pearson Education, 2nd edition, 2010.
- [20] Robin John Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 3rd edition, 2021.
- [21] Liu Yue, Ju Wangwei, Zhao Jianguo, Gao Junjun, Zheng Jiazhou, and Jiang aiping. Product life cycle based demand forecasting by using artificial bee colony algorithm optimized two-stage polynomial fitting. *Journal of Intelligent & Fuzzy Systems*, 31:825–836, 07 2016.
- [22] Michael Gilliland, Len Tashman, and Udo Sglavo. *Business Forecasting: Practical Problems and Solutions*. Wiley and SAS Business Series. Wiley, 1 edition, 2016.
- [23] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, and G.M. Ljung. *Time Series Analysis: Forecasting and Control*. Wiley Series in Probability and Statistics. Wiley, 5th edition, 2015.
- [24] F. Provost and T. Fawcett. *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O’Reilly Media, 1st edition, 2013.
- [25] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [26] Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995.
- [27] Xiaogang Su, Xin Yan, and Chih-Ling Tsai. Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(3):275–294, 2012.
- [28] John D. Kelleher and Brendan Tierney. *Data Science*. The MIT Press Essential Knowledge. The MIT Press, 2018.
- [29] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.
- [30] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [31] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2001.
- [32] Jerome H. Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

- [33] Xin Chen and David Simchi-Levi. Pricing and inventory management. *The Oxford handbook of pricing management*, 1:784–824, 2012.
- [34] Tudor Bodea and Mark Ferguson. *Segmentation, Revenue Management and Pricing Analytics*. Taylor & Francis, 2014.
- [35] Stephen A. Smith. Clearance Pricing in Retail Chains. In Narendra Agrawal and Stephen A. Smith, editors, *Retail Supply Chain Management*, International Series in Operations Research & Management Science, pages 387–408. Springer, Boston, MA, 2015.
- [36] Wedad Elmaghraby and Pınar Keskinocak. Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management Science*, 49, oct 2003.
- [37] Kalyan T. Talluri and Garrett J. Van Ryzin. *The Theory and Practice of Revenue Management*. International Series in Operations Research & Management Science. Springer, 1st edition, 2004.
- [38] I. Png. *Managerial Economics*. Blackwell Publishers, 2002.
- [39] Pavithra Harsha, Shivaram Subramanian, and Joline Uichanco. Dynamic pricing of omnichannel inventories. *Manufacturing & Service Operations Management*, 21(1):47–65, 2019.
- [40] Stephen Smith and Narendra Agrawal. Optimal markdown pricing and inventory allocation for retail chains with inventory dependent demand. *Manufacturing & Service Operations Management*, 19(2):290–304, 03 2017.
- [41] Andrew Vakhutinsky, Alex Kushkuley, and Manish Gupte. Markdown optimization with an inventory-depletion effect. *Journal of Revenue and Pricing Management*, 11(6):632–644, 2012.
- [42] Vibhati Burman, Rajesh Kumar Vashishtha, Srividhya Sethuraman, Ganesh Radhakrishnan, Prashanth Ganesan, Suresh Kumar V, and Sharadha Ramanan. Markdown pricing for a large scale retailer. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1817–1826, 2021.
- [43] Eleanor Loh, Jalaj Khandelwal, Brian Regan, and Duncan A Little. Promotheus: An end-to-end machine learning framework for optimizing markdown in online fashion e-commerce. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3447–3457, 2022.
- [44] Chengcheng Liu and Mátyás A Sustik. Elasticity based demand forecasting and price optimization for online retail. *ArXiv*, 06 2021.
- [45] Aidin Namin, Brian T. Ratchford, and Gonca P. Soysal. An empirical analysis of demand variations and markdown policies for fashion retailers. *Journal of Retailing and Consumer Services*, 38:126–136, 2017.
- [46] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, 1:29–40, 01 2000.
- [47] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in neuro-robotics*, 7:21, 2013.