FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Data-driven Insights for Grocery Retailers: Developing a Serverless Tool for Business Analysis

Enzo Facca Pegorin

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Beatriz Brito Oliveira

July 30, 2023

Abstract

This master's thesis focuses on developing a software for Deloitte Technologies Portugal that aims to provide comprehensive insights for companies in the grocery retail sector and support decision-making. The project is motivated by the increasing demand for data-driven decision-making in the retail sector and the limitations of existing product comparison platforms. Prior to the development, research has been conducted on related topics. It includes a review of current commercial platforms with similar capabilities available for the consumer, followed by a study of relevant articles on product comparison, discussing the alternative methods employed in different markets. Next, we have researched collecting and analyzing data from various e-commerce platforms using web-scraping techniques, which implies technical and ethical challenges. Finally, it explores how we could aggregate value to our analyses by applying pricing and recommendation algorithms to our data. The project's main objectives were to develop an algorithm for data acquisition and parsing, establish accurate product correspondences by matching the products' European Article Number, and distribute the collected data through a dashboard and an Application Programming Interface. We discussed the challenges in formatting the data for our usability and the benefits of our choice for computing infrastructure. By using serverless technologies and cloud services provided by Amazon Web Services, we leveraged scalability, security, cost-efficiency, and high performance for our computing, networking, and storage resources. The project successfully demonstrated its capabilities in acquiring highly available and precise data, enabling clients to identify trends and drive improvements in their strategies. The showcased analyses revealed the maturity of the acquired data and anticipated possible use cases for our users.

Keywords: Grocery, e-Commerce, Web-scraping, Product comparison, Serverless technologies, Amazon Web Services.

Resumo

Esta tese de mestrado centra-se no desenvolvimento de um *software* para a Deloitte Technologies Portugal que tem como objetivo fornecer uma visão abrangente para as empresas do sector do retalho alimentar e apoiar a tomada de decisões. O projeto é motivado pela crescente procura de dados que suportem decisões no sector do retalho e pelas limitações das plataformas de comparação de produtos existentes. Antes do desenvolvimento, foi efectuada investigação sobre temas relacionados. Incluindo uma análise das actuais plataformas comerciais com capacidades semelhantes disponíveis para o consumidor, seguida de um estudo de artigos relevantes sobre comparação de produtos, discutindo os métodos alternativos utilizados em diferentes mercados. Em seguida, investigámos a recolha e análise de dados de várias plataformas de comércio eletrónico utilizando técnicas de coleta de dados, o que implica desafios técnicos e éticos. Por último, explorámos a forma como poderíamos agregar valor às nossas análises, aplicando algoritmos de preços e de recomendação aos nossos dados. Os principais objectivos do projeto consistiram em desenvolver um algoritmo para a aquisição e análise de dados, estabelecer correspondências precisas de produtos através da correspondência do Número de Artigo Europeu dos produtos e distribuir os dados recolhidos através de um *dashboard* e de uma Interface de Programação de Aplicação. Discutimos os desafios na formatação dos dados para a nossa usabilidade e os benefícios da nossa escolha para a infraestrutura de computação. Ao utilizar tecnologias sem servidor e serviços em nuvem fornecidos pela Amazon Web Services, tirámos partido da escalabilidade, da segurança, da eficiência em termos de custos e do elevado desempenho dos nossos recursos de computação, rede e armazenamento. O projeto demonstrou com êxito as suas capacidades de aquisição de dados altamente disponíveis e precisos, permitindo aos clientes identificar tendências e impulsionar melhorias nas suas estratégias. As análises apresentadas revelaram a maturidade dos dados adquiridos e anteciparam possíveis casos de utilização para os nossos utilizadores.

Palavras-chave: Comércio eletrónico alimentar, Recolha de dados, Comparação de produtos, Tecnologias sem servidor, Amazon Web Services.

Acknowledgments

To my family, Natália, Luiz, Cláudia, and Alice, for encouraging all my steps since always, no matter what.

To my friends from FEUP, Costa, Vitão, Jijo, and Rui, for being by my side through this journey.

To the professors from FEUP, especially my supervisor Beatriz, for the inspiration during these five years.

To the colleagues from Deloitte, especially the Retail Match team, Vasco, Paulo, Bastos, José, and Francisco, for the support during this work

"If we have data, let's look at data.
If all we have are opinions, let's go with mine."

Jim Barksdale

Contents

List of Figures

List of Tables

Abbreviations and Symbols

ADCLUS	Aditive Clustering
AI	Artificial Inteligence
API	Application Programming Interface
AWS	Amazon Web Services
B2B	Business to Business
B2C	Business to Consumer
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CSS	Cascading Style Sheets
DB	Database
DOM	Document Object Model
EAN	European Article Number
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
IAM	Identity and Access Management
IP	Internet Protocol
LRFMP	Lenght Recency Frequency Monetary and Periodicity
MAB	Multi Armed Bandit
ML	Machine Learning
PP-OOS	Post Purchase Out-Of-Stock
POC	Proof of Concept
RFM	Recency Frequency Monetary
RDS	Relational Database Service
UI	User Interface
URL	Uniform Resource Locator
XML	Extensible Markup Language
XPATH	XML Path Language
WTP	Willingness to Pay

Chapter 1

Introduction

This chapter provides a background for the topics discussed in this thesis, including the growth of digital retailing, the impact of the COVID-19 pandemic on the grocery retail market, and the concept of product comparison platforms. It highlights how digitization changed the market globally and in Portugal and discusses the digital competition among retailers in the segment. Furthermore, the chapter outlines the motivation, scope, and limitations of the research, as well as the specific objectives to be pursued. It also provides an overview of the structure and organization of the following chapters.

1.1 Background

During the COVID-19 pandemic, the online shopping market has grown in the face of people's sudden need to avoid leaving home during confinement. This increase in demand has required retail chains to reallocate their efforts and expand their portfolio of digital services. After the end of the restrictive measures, the changes in habits that occurred during this period remain, as customers have improved their perception of previous concerns.

The increasing relevance of eCommerce dynamically changes the concepts that guide retailers' strategic decisions. The online consumer has access to multiple sources to be compared simultaneously, and by shopping from behind a screen instead of walking through aisles some of the marketing values have mutated (Jílková and Králová, 2021).

Grocery Retail and Digitalization

The food retail sector has one primary trait that differentiates it from other retailers, resilience. By serving the entire population with items of recurrent purchase, the sector shows consistent trends that others like Vehicles, Technology, or Entertainment do not (Dong and Byrne, 2022). However, tendencies inside the sector tend to shift constantly, offering the customer the opportunity to choose between large chains or small local shops, options for processed and fast versus organic and conscious products, local or imported, quality or price, and finally, the latest dilemma, physical versus online.

The democratization of smartphones and internet access gave birth to the concept of omnichannel, where the same products offered in physical stores became available on digital platforms. This change mainly impacted the general merchandising sector, where the logistics of non-perishable items are simplified, facilitating deliveries to the customer. Food retailers also participated in this movement but faced very different challenges, for example, delivering products outside dense urban zones and the suspicions of the quality of the arriving fresh products (Dannenberg et al., 2020) (Abbu et al., 2021). Although revenues in the grocery retail sector from 2010 to 2019 averaged a positive growth of 1,31% per year (Dong and Byrne, 2022), the profit margin is decreasing, and big players are pushing these margins even lower (Halzack, 2015).

Nevertheless, a phenomenon of global impact changed the headings of the digital food retail sector, the COVID-19 pandemic that led the world to lockdown in early 2020. During the first months of strict restrictions, online sales jumped 300%, and retailers reassigned more workers to handle online orders and offered more products online (Redman, 2020). Figure 1.1 shows how the share of online sales was boosted during the pandemic and maintained its rise after the most critical phases. These numbers result from people using online shopping to respect social distancing and from developing habits of cooking at home instead of ordering or eating at restaurants (Grand View Research, 2020). Naturally, these trends normalized as the effects of the pandemic washed away. However, the market changed permanently. Customers have developed new consuming habits, are now more confident in the service, and are willing to continue using digital solutions (Mercatus, 2020). Companies have structured their services to meet the demand, providing solutions like click-and-collect, same-day deliveries, and subscription plans. As indicated in Figure 1.2, it is estimated that by 2025 online shopping will be responsible for 21,5% of all groceries bought in the US, increasing 60% of the expected value calculated before the pandemic for the same period (Mercatus, 2020), additionally, the average value spent per customer on online grocery should increase 78%, forming a $ 243 billion market in the US at 2025 (Yuen, 2022).

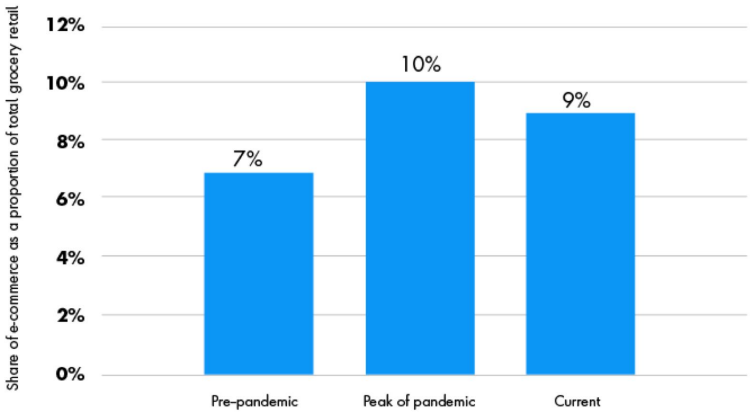


Figure 1.1: Global development of e-commerce shares of grocery stores before and after the COVID-19 pandemic (Coppola, 2021).

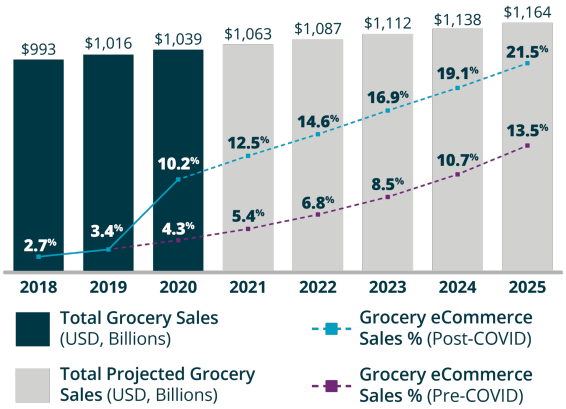


Figure 1.2: Physical and eCommerce grocery sales in the United States (Mercatus, 2020).

Portuguese Grocery Retail

The Portuguese food retail market was valued in 2022 at € 19,7 billion, ranking 12^{th} in Europe. Its more than 13.000 (World, 2022) stores can be identified in many formats ranging from small conveniences to hypermarkets, but the larger ones combine most of the representation. Around 70% of the sales come from hyper or supermarkets. The market share for the segment is held by six significant players: Sonae (Continente), Jeronimo Martins (Pingo Doce), Lidl, Intermarche, Auchan, and Dia (Minipreço) sharing 26,8%, 22,9%, 11,3%, 8,8%, 5,6%, and 3,9% respectively (Medina, 2021). The first two being Portuguese companies means that national businesses are responsible for around 50% of the sector. The market is profoundly marked by two main strategies present in around 46% of the segment sales (Paupério, 2020): Discounts and Loyalty Programs. These strategies are often combined as improved discounts for members or exclusive offers. Accumulated values can also be spent on partners like gas stations or pharmacies. As a result, half of the Portuguese shoppers search actively for promotions when shopping even though they are not willing to switch stores for them, and one-third admit that they choose brands based on promotions (Paupério, 2020). Another strong trend highlighted in large retailers' web-sites is the presence of private-label products competing in most categories. A report from the International Private Label Yearbook shows that at the country level, the market share of retailers' own-brand lines is 43% (Jorge, 2020).

As mentioned previously, digitalization is redirecting the future of food retailing, and the Portuguese market is not oblivious to the change. Boosted by the pandemic, general online sales rose 44% (Medina, 2021) as the habit became popular among all age groups, geographical areas, and educational backgrounds, expanding a niche previously occupied by younger generations with great economic power. Portuguese companies are taking advantage of this new habit and consolidating their presence in the segment. Except for Lidl and Dia, the remaining four larger players offer online platforms with the complete product listing available in physical stores, with an option to choose between delivery at home for a fee or in-store pickup without added costs. In the second quarter of 2019, *Continente online* led the market with a share of 59,5% and a penetration rate of

9,7%. *Auchan online* coming second with a share of 22% and penetration of 3,3%. *Intermarché online* places third with a share of 5,4% and a penetration of 3,7% (Stevens, 2020).

Digital Competition

In the context of such digital advances in the segment, the relationship between retailers and consumers is getting more energetic. Every retail chain with an online presence hosts massive amounts of information in its systems. The capability of simultaneously accessing content from multiple retailers yields a powerful tool for those who acquire it. However, the abundance of available information poses challenges for retailers in analyzing and extracting meaningful insights. With such a vast volume of data, it becomes increasingly difficult to identify and filter out the genuinely informative data that can drive pricing or marketing decisions (Bumblauskas et al., 2017). This information overload can hinder the decision-making process, as retailers struggle to discern the critical factors influencing their strategies, a phenom called *data binge*. Therefore, it is crucial that after acquiring large sets of information, the capability to analyze and obtain relevant insights remains, usually through consolidated dashboards, graphs, and tables, instead of looking at raw collections.

Nevertheless, combining publicly available information with retailers' internal data can result in a significant advantage. By leveraging external and internal sources, companies gain a comprehensive understanding of the products, brands, and categories on the market. This benefits the decision process in policies like demand forecasting, pricing optimization, personalized marketing, assortment planning, product segmentation, and supply chain efficiency (Akter and Wamba, 2016).

Retailers who fail to consider this vast source of information may find themselves at a competitive disadvantage. Those who harness the power of data analytics and information systems are more likely to stay ahead of the competition. However, multiple competitors utilizing these resources without regulation could lead to a controlled market environment (Tian et al., 2021). Fortunately, ample resources are available to guide retailers in utilizing information systems. Industry research, market reports, and data analytics platforms offer valuable insights and best practices.

Simultaneously, customers are directly impacted by the informed strategies of the retailers they shop from (Carolan, 2018). This fierce competition could lead to better conditions for the seller, but the consumer may be blinded by such strategies and without control over their decisions. However, the customer also benefits from the abundance of information available. They have the ability to choose between the most compelling online stores effortlessly, access comparison platforms, and find detailed product descriptions and reviews, empowering them to make informed decisions.

In this context, a product comparison platform is a big-data-sourced tool that can guide retailers to position themselves based on competition or help customers become better informed of the retailers' practices. On the platform, information can be manipulated to filter only relevant

fields for the user and avoid unnecessary information. By selecting similar products from different retailers or different products from the same retailer, one can compare features like pricing, packaging, promotions, and nutrition facts, improving its perception of the market.

1.2 Motivation

This dissertation has been proposed by the Products, Services, Utilities, and Resources branch at Deloitte Portugal. In a larger scope, the project called RetailMatch offers a Software as a Service that acquires, manipulates, stores, and distributes public information from grocery retailing websites. The algorithm is capable of differentiating particular features and using that information to match equivalent items from different sellers.

Our solution is developed to be used by retail chains or companies related to the retail market, to provide comprehensive insights about the products, brands and prices listed in each retailer. This information will help clients to improve their marketing decisions and pricing strategies. However, the project in its full scale is complex and costly to maintain, therefore the necessity to develop a POC (Proof of Concept) contemplating 3 Portuguese chains to be offered as a demonstration for prospecting clients, with selected data and improved features designed to be highly available and hold precise information.

Other platforms for product comparison where the user can search for the desired product, view price history or compare offers from different retailers already exists for a long time, however, RetailMatch's uniqueness is achieved through its granular and historical data. Most solutions on the market do not offer their users access to low level information about products like nutritional facts or country of origin. They also do not hold information for long periods, making it harder to perform historical analysis on products.

1.3 Scope and Limitations

This work aims to investigate and develop a solution for the use of information about products listed on grocery e-commerce platforms for product comparison and strategic decision-making by retailers. The study aims to explore the potential benefits and challenges of using this data source and to provide insights for retailers seeking to improve their competitiveness and profitability.

The scope of this research includes collecting and analyzing data from various grocery e-commerce platforms using web-scraping techniques. The study focuses on being exhaustively comprehensive of the different food products or non-food products sold on grocery e-commerce platforms. The collected data will be used to identify product characteristics, such as brand, price, and packaging, and use them to match correspondent products from different origins. It will also explore the potential use of this data for further comparison of products in the context of pricing strategies or product recommendations.

The research is limited by several factors, including the availability and quality of data obtained from grocery e-commerce platforms. Despite being collected daily, the data may need to be

updated or may represent only some of the population of products sold on the platforms. The study will also be limited by the capabilities of the web-scraping tools and techniques used to collect the data, which may not be able to capture certain types of information or may introduce errors into the data. Another limitation of this research is that it does not consider other data sources, such as data from physical stores, online marketplaces, or third-party data providers. The study will also not consider the impact of external factors, such as changes in consumer preferences, economic conditions, or regulatory changes. Finally, the project could benefit from other information sources, namely information about consumers' activity on the retailers' web-sites or information owned by the retailer about the products that are not published on the web-site. This sort of data could improve recommendation algorithms or strategic insights.

Despite these limitations, this research aims to contribute to understanding the potential benefits and challenges of using data from grocery e-commerce platforms for product comparison, pricing insights, and strategic decision-making. The findings of this study will be of interest to retailers, policymakers, and researchers in the fields of marketing, e-commerce, and data analytics.

1.4 Objectives

The main goal of our project is to develop an algorithm that not only successfully completes all the required steps but also ensures the scalability and efficiency of the full-scale implementation. Firstly, our code must proficiently ingest data from the designated web-sites, ensuring that all relevant information is captured accurately and reliably. We must adapt the process to the limitations of different sources.

Once the data is acquired, the next step is parsing and formatting it to adhere to the desired database structure. Proper parsing is essential to extract meaningful information from the raw data, while appropriate formatting ensures consistency and compatibility within the database. During parse we expect to alter the raw data as minimum as possible, while performing necessary corrections.

One of the key challenges lies in establishing accurate correspondences between products during the matching stage. This involves identifying similar products from different sources, ensuring consistency in naming conventions, and effectively linking related information through the use of the products EANs (European Article Number).

Furthermore, the information we collect must be presentable through an dashboard and an API (Application Programming Interface). The dashboard will function as an predefined overview of the data as a whole, while the API is the opening for the user to adapt the information to their own need without coding or altering the algorithm.

To ensure optimal performance and scalability, we prioritize the use of serverless technologies whenever possible. Leveraging serverless architectures offers numerous benefits, including automatic scaling, reduced operational complexity, and cost efficiency.

In addition to completing the code development, it is essential to define other quantifiable objectives to evaluate the project's success. These objectives encompass aspects such as the quality

and quantity of the acquired data, the effectiveness of our running infrastructure in handling data processing and storage, and the potential analyses that can be performed using the information collected. By setting clear objectives, we can assess the project's outcomes and measure its impact on retail e-commerce and related domains.

1.5 Structure

In addition to the introduction chapter, this dissertation contains 4 more chapters.

In Chapter 1, is provided an introduction to the research topic, presenting the background and motivation behind the study. The chapter also defines the scope and limitations of the project and outlines the objectives to be achieved.

Chapter 2 presents a comprehensive literature review of relevant topics related to the project. It provides an overview of existing product comparison platforms and also discusses existing solutions in the Portuguese market. Following, it delves into the concepts of data mining, web developing and cloud computing. Furthermore, it explores various applications of these technologies, focusing on product recommendation and pricing.

Chapter 3 describes the methodology employed in the research. It starts by introducing the concepts of cloud computing and infrastructure. The chapter then discusses the data structure and the considerations for ensuring data and infrastructure security. It further elaborates on the process of data ingestion and parsing, as well as the challenges related to matching and conflicts. The chapter also addresses the critical aspect of data quality and concludes with a discussion on deployment strategies.

Chapter 4 presents the results and analysis derived from the implemented solution. It focuses on data visualization techniques employed to present the acquired data effectively. The chapter also explores applications built upon the collected data and discusses the performance of the infrastructure used.

The final, Chapter 5, summarizes the key findings and contributions of the research. It discusses the implications of the study and its significance in the field. Additionally, it identifies areas for future work and further research. The chapter concludes the thesis, highlighting the achievements and potential avenues for future exploration.

Chapter 2

Literature Review

The literature review chapter in this thesis presents a comprehensive analysis of the existing research and knowledge pertaining to various aspects of grocery platforms, product comparison, web-scraping, and use cases of our project. By critically examining the relevant literature, this chapter establishes a solid foundation for the subsequent discussions and analysis, providing valuable insights into the current state of the field and identifying research gaps and opportunities. Through a systematic review of the literature, this chapter aims to enhance our understanding of the subject matter and contribute to the existing body of knowledge in the field of e-commerce and data-driven decision-making.

2.1 Overview of Grocery e-Commerce Platforms

In an article reporting the creation of a platform for product comparison through web-scraping, the author introduces his work by providing an overview of common design patterns found in grocery web-sites. The work mentions the search bar and the category list, clearly defining the usability for each. It also includes details found on product pages. Lastly, the author describes pages that should be avoided for not containing any contribution to the target data (Xie, 2016).

These patterns tend to repeat in retailing web-sites of most genres, and the same happens in Portugal, where similar structures are present in all of the chains we considered. In all five analyzed web-sites, the user is welcomed to the main page, where they usually find highlighted products, current deals, options to log in or select an online store, and most importantly, the search bar and the category tree.

Both the search bar and category tree serve the goal of leading to the product page, although the first display instant results. In contrast, the second requires the user to navigate to the category holding the desired product. The category tree is more appealing to users who want to compare competing products or browse the web-site like walking through the store aisles. When a product is selected, the product page will be loaded. There, the user has access to the product image, name, a short description, packaging contents, brand, and other fields that may vary depending on the

product or the retailer, for example, the country of origin, the nutritional facts, or the indication of a biologic or vegan product.

Like a regular customer, the objective of our scraper is to reach the product pages. However, we seek to do so in the fastest and most effective way, and for that is crucial to be familiarized with the web-site's structure. Since most web-sites do not display a page with all the listed products together, which would be ideal for our scraper to move from one to another until the end, our approach is based on using the division of every main category to have a list of URL (Uniform Resource Locator) from the products to be scraped, and then move to accessing each product page and extracting the information.

In the development of retail web-sites is common to see the application of technologies like HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), XPath (XML Path Language), DOM (Document Object Model), and XML (Extensible Markup Language). Even though we do not need to deep into how a web-site should be developed for this subject, it is relevant to understand the priciples since it is our responsibility to indicate to the parser what path should be followed and what fields are relevant or not.

- HTML (Hypertext Markup Language) is used in creating and structuring web-pages. It provides a set of tags and elements that define the content and layout of a web-page. HTML tags specify the structure of text, images, links, headings, paragraphs, and other elements. It acts as the backbone of a web-site, providing the foundation for displaying content and enabling the integration of other technologies (Raggett et al., 1998).

- CSS (Cascading Style Sheets) is a styling language used to control the visual presentation of HTML elements on a web-page. It allows developers to define styles such as colors, fonts, layout, and positioning. CSS separates the design and layout from the content of a web-page, enabling consistent and flexible styling across multiple pages. With CSS, designers can create visually appealing and responsive web-sites, ensuring a consistent user experience across different devices (World Wide Web Consortium, 2010).

- XML (eXtensible Markup Language) is a crucial technology for retail e-commerce web-sites, playing a significant role in data exchange, data feeds, customization, configuration, and web services. It enables seamless integration and communication between systems by providing a standardized format for transmitting structured data such as product catalogs, inventory information, and customer data. XML-based data feeds facilitate the import and synchronization of product information across platforms, while XML configuration files allow for the customization of web-site layouts and themes. Additionally, XML is commonly used in web services and APIs, enabling structured data exchange between systems, enhancing interoperability and facilitating seamless integration within the e-commerce ecosystem (Bray et al., 2008).

- XPath (XML Path Language) is a query language used to navigate and select elements in an XML document. XPath is commonly used in web development to extract specific

data from HTML documents. It provides a syntax for traversing the HTML structure and locating elements based on their attributes, tags, or other properties. XPath is handy for web-scraping, data extraction, and automation tasks, allowing developers to target specific elements on a web-page and extract the desired information (Robie et al., 2014).

- DOM (Document Object Model) represents the structure of an HTML or XML document as a tree-like structure. It provides a programming interface for dynamically accessing and manipulating a web-page's elements and content. With the DOM, developers can use JavaScript or other scripting languages to interact with the web-page, modify its structure, update content, handle events, and create interactive functionality. The DOM enables dynamic rendering and real-time updates, making it a vital technology for building interactive and responsive web-sites (Robie and Texcel Research, 1998).

Finally, appendix A contains screen captures of the main and product page of the web-sites for five major chains in the Portuguese market. This allows identifying the mentioned patterns like the category filter on the main page or the product information in the product page.

2.2 Product Comparison Platforms

In recent years, the availability of vast amounts of data on the internet has presented new opportunities for businesses to gain insights and make informed decisions. For grocery retailers, this wealth of online data offers valuable information about product offerings, prices, and other relevant attributes. To effectively compete in the dynamic retail market, retailers must stay updated on competitor prices, track product contents, and compare their offerings with those of other retailers.

The primary objective of this literature review is to identify and examine the different approaches and methodologies employed by existing tools in the domain. By understanding the range of available solutions, their features, and their limitations, we can gain insights into the best practices and potential areas for improvement in developing our data acquisition and analysis.

The first article to be reviewed shares its objectives and goals with our project, even though on a different market. The author aims to construct a web application that can offer data on the current retail prices for items connected to medical supplies. The application serves as a platform for users to access and compare the highest retail prices of these goods, providing valuable market price information and raising awareness about price manipulation. The article emphasizes the use of web-scraping, precisely the HTML method, for efficient retrieval of product data from marketplaces and credible government sources (Nugroho, 2022).

Like this work, many other researchers propose their solution to acquiring product information and matching the items through their characteristics. However, techniques and approaches diverge depending on the challenges of different markets and regions.

Another paper reports the design of "Upoma", a web-site to display the price comparison of products from nine popular eCommerce in the Bangladeshi market (Alam et al., 2020). A Python

code using Scrapy is responsible for scraping and ingesting the product information from all retailers and loading it into a database, which will be consulted by the web-site when a user searches for a product. However, the authors face a significant challenge regarding product matching. Web content in Bangladesh can be seen in 3 different languages, and since the matching algorithm uses textual similarity of the product name, this is a barrier when the same item is listed in different languages. Through the transliteration of text in English and Bangla, they claim the algorithm has a 93% success rate in corresponding the matched results to the user-inserted query.

Also on the same subject, Xie (2016) makes a pervasive review of the techniques applied to web-scraping products on the internet, especially to how the web-site is structured and the programming of the scraper considering that. The author reviews multiple web-scrapers categories and divides them according to their strategies. Next, it is explained how the algorithm is developed to match products. The product information like name, brand, description, packaging size, and weight are extracted from the web-page, and the text content is tokenized in English and Chinese. The tokens are then vectorized and weighted according to a formula described in the thesis, finally allowing to make connections between words and, ultimately, matching products. This technique is proven to work, however, there is a considerable effort into corresponding text that is arbitrarily defined by each retailer, and the situation is even more challenging when two languages are involved.

To end the review on proposed product matching tools, two articles share a common aspect that diverges from the previous but are as pertinent to be considered. Asawa et al. (2022) and Khatter et al. (2022) propose to tackle the same research goal: developing a web-site with a UI (User Interface) where users can search for a product and have access to listed prices for that product on different online retailers. Like the previous, they also use web-scraping, but with a different approach. Instead of repeatedly scraping and ingesting data from all retailers, they execute the scraping commands based on the user inputs to search for a product. A UI presents the user with a search bar where one can input keywords. The algorithm then inserts these keywords as a search field in the retailer's web-page URL, as if a user searched for the product directly on the web-site. Asawa et al. (2022) defined their algorithm to initially search for exact matches for the searched keyword on the web-site. If none is found, then proceed to find the cheapest closer match. Again, if none is found, the web-site will present a warning message. This technique presents some drawbacks. First, inputting the same keywords on different web-sites is not guaranteed to return the same products. What leads to the next, if an input is too vague, like in the example given by the author of one of the articles where they search for "hp notebook", this could lead to numerous results that certainly will diverge from one retailer to another. Finally, another disadvantage of this technique compared to the others is that no historical data is saved to a database, therefore not allowing a price tendency graph or other historical analysis.

In conclusion, comparison platforms are highly valued by consumers, but developing a platform that informs customers about prevailing product prices poses numerous challenges. The initial hurdle lies in acquiring the necessary data, as it is typically not readily available from third-party sources. To overcome this, web-scraping has emerged as the preferred solution. However,

the subsequent challenge involves interpreting product information accurately to facilitate matches and enable meaningful comparisons. Text analysis presents particular difficulties due to various external factors, making a solution based on unique product identifiers more preferable. Another point of divergence among researchers pertains to data storage: whether it should be stored in a database or queried on demand. In our specific use case, the former option seems more advantageous. While all authors agree on the importance of providing a user interface for product searches and results, the specific features of the user interface may differ across implementations.

Solutions in the Portuguese Market

In the Portuguese market, some platforms provide product comparison and price analysis. While some offer a wide variety of products and outlets for the customer to compare and choose from, others specialize in supermarket chains and grocery retailing.

Moving from the wider to the closer spectrum, the first solution of its kind available for Portuguese consumers is called "KuantoKusta". This platform's development started in 2004, destined to be the first price comparison tool for the Portuguese market (Think-BIG by SAPO, 2022). Currently, the platform is thriving due to its marketplace capabilities that connect researching a product price to buying it in partner stores. Another function that popularized the web-site is a collection of three tools to guide its user to the best offer. One can navigate through categories or insert text on a search bar to find a product that, upon selection, will be displayed on a page with its description and the following analysis: the best prices, which shows the price drop for a given product compared to the average of the lowest prices over the last 30 days; price alerts, where the user sets how much he or she is willing to spend on a given product and is notified when this amount is reached; price history up to 3 months, which tells the consumer how the price has evolved for a given product. Although it is comprehensive on the available categories like electronics, toys, pets, house, bricolage, automobile, or sports, it lacks information on grocery products. Supermarkets are not among the stores considered in the comparison tool, so users can not use it for food shopping.

Like the previous, "Zwame" is a price comparison platform specialized in electronics and technology items. Similarly to "KuantoKusta", the user can search for a product name or navigate the web-site to a product page where price history and current prices on different stores are displayed.

Before moving to grocery-specialized solutions, "DECO" is an independent consumer rights organization that promotes independent product testing, hosts a purchase complaint board, and advocates for consumer rights through the law. The company seeks to help consumers avoid being caught in misleading promotions by offering on its web-site a simple tool where the user can insert a product URL or its name and the store it is being sold and will be provided with prices over the last 3 for that product on that store or current prices on other stores. Complementary information on the product is not provided, and the only action the user can take after searching for a product is to proceed to the seller's web-site. However, the comparison does not include grocery products

or supermarket web-sites, and even on other common categories like electronics, some products or stores could not be found through the search-bar.

Next, "SuperSave" is an app developed by recently graduated engineers that focuses on grocery retailing. The backstage functioning of the app is not disclosed. However, its functionalities are straightforward to the user (Maciel, 2022). On the mobile app, the user has access to 5 large super in the country, and searching for a product returns its image, price comparison from multiple chains, and a chart with its history. The main goal of the developers is to help shoppers save during grocery shopping, offering an app that allows the user to search products, compare prices or wait for the best period, and add to a shopping list that links directly to be purchased in the retailers' web-sites.

"Lisie" is another app with the same goal as the previous: allow the customer to create a shopping list with the desired grocery items and compare their prices on different chains to select the cheapest (Gonçalves, 2023). The app presents product information, price history, and comparison between retailers, like "SuperSave".

Although these solutions do not disclose their data sources and therefore, we are not able to conclude what information they have access to but choose not to disclose, it is possible to note that through the platform, the user is unable to perform other analyses that do not depend on the price, for example, selecting specific categories, packaging content, nutritional facts or geographical localization of the store

In conclusion, the Portuguese market is not poorly served with price comparison platforms, specially for broader products. A common trait between all reviewed cases is offering the linkage to the product page on the seller web-site, indicating a possible source of capitalization to referral bonus for the platform owner. Another common feature is the lack of information about the product description, packaging content, nutritional information, country of origin and etc on the platform.

Although the customer does have alternatives for product comparison, none of them offer analysis other then the product price, and none of them are extensive on product information, even the grocery oriented. Our solution aims to offer both features for a competitive advantage, besides the ambition to be the leader in product matching efficiency and quality. A relevant difference between these solutions and ours, is that they are B2C (Business to Consumer) oriented (the consumer is their target), meanwhile ours is destined to B2B (Business to Business), what means another company that would then publish or not the information to the customer.

2.3 Data Mining and web-scraping

Navigating through a retailing web-site, a user has access to countless amounts of information about the company and its products, and in the current period of information and digitization, the collection of such data can be used for high-value applications. The situation begins with how to collect and save significant amounts of data that are available primarily only for visualization. If only a handful of information was to be copied, a user could manually annotate the desired data.

However, a more efficient and automatic process is necessary for operating on the full scale of a retailing web-site, which is a precedent for web-scraping.

A web-scraper is software capable of navigating the web-crawler through the code of a URL, either automatically or according to a determined path, and extracting the information on selected fields of the code as if a user was copying the information from a URL page and pasting to a spreadsheet but in a fast-paced and precise way (Lawson, 2015). The web-crawler is a bot that navigates the web-site making requests as a regular user, but capable of automatically acquiring the current page's results and proceeding to the next at a much faster pace than a regular user. However, this could lead to an abuse of the provider infrastructure, which will be given more depth in the following paragraphs.

In this situation, many companies offer APIs for accessing their databases regularly, making formatted requests to the interface instead of mining the code behind a web-page. This formalizes access to information on the web-site and makes it easier to receive the information in the desired fields since it will be structured in a standard way and followed by documentation. However, this option is only sometimes available. The API provider may charge for requests or control what information can be accessed, so web-scraping remains an alternative in this situation (Mitchell, 2018).

Although many companies choose to facilitate the collection of available information, others may choose to block web-scrapers. As the bots behave differently from human users, it is possible to block IP's (Internet Protocol) that make excessive requests, present constant patterns, or access from specific known locations. The CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a well-known barrier for impeding bots' access (Moradi and Keyvanpour, 2015), and spider-trap is another effective way of stopping the process by making the crawler crash.

Techniques and Tools

web-pages like the ones available in grocery retailers' platforms consist of HTML code to structure objects like images and texts. The code formatting is destined to be the most human-friendly possible, which brings a downside when it is necessary for a machine to interpret and navigate the code.

There are many different techniques for web-scraping. One can simply copy and paste information, use HTTP requests to receive the desired fields, use DOM parsing to visualize items in a tree, computer vision analyzers, off-the-shelf web-scrapers, or HTML parsing (Saurkar et al., 2018). Although all of the mentioned have their positive and negative aspects, the one with more appealing characteristics to this project was the last.

Next is deciding which web-crawler framework is best suited for the project. A crawler framework mainly contains schedulers, URL servers, downloaders, spiders, and an engine. This packaging makes it easier for the developer to deploy a scraper, as it only needs to care about the logical part of the crawler, such as the extraction of page information and the generation of the subsequent request. In this article, Yang and Thiengburanathum (2020) review the performance and features

of ten open-source crawlers by collecting the results of many studies on the subject. The studies simulated HTML requests to evaluate how the frameworks would perform regarding robustness, efficiency, speed, and scalability. Among Python and Java frameworks like Scrapy, Pyspider, Portia, Heritrix, and others 6, the results showed that Scrapy is the only Python framework to offer all of the considered features except a graphical UI. It has a free license, Parallel multi-thread, supports Proxies, APIs, and HTTP (Hypertext Transfer Protocol) requests, and accepts cookies and CSS/XPath selectors. An overview of the functioning of scrapy can be seen in figure 2.1.

These results support the decision to use Scrapy in the project. As Python is the common language for the whole project development, it is required that the selected scraper is available in the language, therefore leading to the selection of Scrapy, since it outperforms its competitors in speed and efficiency, despite having a steep learning curve. During the ingest steps of the project, the Scrapy parser receives the response of an HTML request and then uses specified paths inside the document to retrieve the desired information. The elements in these paths are specified by CSS selectors or Xpaths.

However, as mentioned before, web-scraping is not necessary when an API is available, so for the retailers with that option, Scrapy would be replaced by simply parsing the results of API requests.

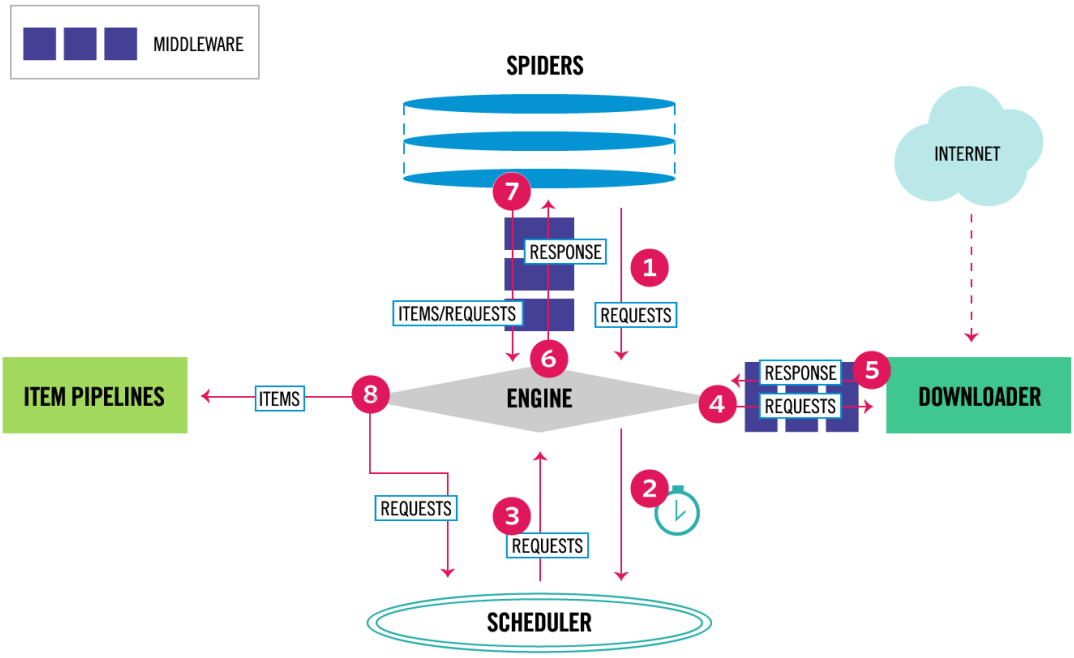


Figure 2.1: Scrapy Framework Architecture (Scrapy Developers, 2023)

Computing Infrastructure

This kind of task can be very time and resource-consuming, as it requires constant access to the internet for the page requests while also requiring processing of the collected data. Logically the efficiency of the task is related to the hosting computing power and connection quality, but the demand for these resources is variable and hard to predict. With that perspective, cloud computing presents an appealing solution (Khder, 2021).

Chaulagain et al. (2017) researched how AWS (Amazon Cloud Services) cloud infrastructure can be applied to web-scraping and concluded that benefits like unlimited storage, elastic and flexible resource allocation, parallel processing, and cost efficiency match the task needs. They also stated that AWS and competing offerings are adequate, offering computing and storage services prepared to handle the necessary loads. Such AWS services are the ones selected for the execution of this project.

Legal and Ethics

Web-scraping is a powerful tool that can provide valuable data for various purposes, but it is essential to consider this technique's ethical and legal implications. One primary concern is the potential violation of copyright and intellectual property laws, as web-scraping can involve copying and republishing content from web-sites without permission. Additionally, web-scraping can raise privacy concerns if personal information is collected and used without consent.

In a review of the legality of web-scraping, Krotov and Silva (2018) highlights the main aspects to be considered about the subject. The paper begins by mentioning that web-sites can prohibit programmatic access by including a "terms of use" policy on their site. However, violating these terms may only necessarily result in legal consequences if the user explicitly agrees to comply with them. The text then discusses copyright infringement related to scraping and republishing copyrighted material, including limitations under the "fair use" principle. Next, the authors explore the purpose of web-scraping and how illegal or fraudulent use of data obtained through web-scraping can lead to legal consequences. For example, knowingly accessing confidential and protected data. Additionally, if web-scraping damages a web-site or server, the person responsible can also be prosecuted. The ethics of web-scraping are then discussed, focusing on potential harm to web-site owners and customers. One potential harm is compromising individual privacy by unintentionally revealing the identity of web-site participants. Using data without consent can also violate research subjects' rights. Additionally, organizations have the right to maintain certain aspects of their operations confidential, and web-scraping can unintentionally reveal trade secrets or confidential information. Finally, web-scraping can lead to financial losses for web-site owners by creating data products that compete with the original owner's offerings and diminishing the value of the web-site's advertisements. Yet, there are also positive ethical aspects of the technique. By allowing retailers to obtain pricing information and other data about their competitors, web-scraping can help level the playing field and promote fair competition. This can ultimately benefit consumers by ensuring that prices are competitive and products are of high quality.

However, some steps can be taken to address these ethical and legal issues. Firstly, obtaining permission from web-site owners before scraping their data is essential. Many web-sites have policies that prohibit or restrict web-scraping, so it is important to respect these guidelines. Secondly, it is crucial to ensure that any personal information collected is appropriately anonymized and not used for any malicious purposes. Thirdly, it is essential to ensure that any data collected is accurate, reliable, up-to-date, and used for legitimate purposes (Krotov et al., 2020).

Web-scraping *vs.* other data-sources in food retail

The information gathered in this project is obtainable through more than just web-scraping and similar sources. Other alternatives could provide some of the information we require to perform our analysis. However, considering other methods, a list of pros and cons arises. Food price and availability indexes could be acquired through results of public studies regularly conducted on most markets. Meanwhile, the retailer could provide other more specific data about products and transactions, sometimes at a cost. These methods are not desire-full for our application since having access to manipulated and filtered data implies thrusting the quality and having no access to extensions of this data (Cavallo, 2013).

Hillen (2019) delves into the potential of web-scraping as a valuable method for gathering data in the realm of food price research, with a focus on the advantages and limitations associated with this approach. The author highlights the advantages of web-scraping, including its cost-effectiveness, as open-source software can be used without significant expenses. Additionally, web-scraping enables researchers to collect data at various frequencies, facilitating more detailed analysis of price dynamics and the application of statistical methods (Edelman, 2012), instead of relying on the public release of reports from third parties. The method also grants access to a broader range of product details on the retailer's web-site, such as package size, brand, and quality differentiation, which may enhance the depth and accuracy of food price analysis. Furthermore, web-scraping allows researchers to customize their data sets based on specific needs and preferences, ensuring transparency in data collection and eliminating the risk of omitted variables or undisclosed aggregations. However, it also mentions several limitations associated with web-scraping. Collecting historical data may pose a challenge, as web-scraping primarily focuses on real-time data collection, and therefore it is necessary to store the collected data for historical analysis. The sheer abundance of data available through web-scraping may tempt researchers to gather more data than necessary, potentially leading to exploratory data mining rather than hypothesis-driven research (Massimino, 2016). Additionally, web-scraping typically provides public product information only, lacking transaction data on consumer behavior, such as clicks or purchases.

Cases of web-scraping grocery web-sites

The next two articles share the same objective and utilize equivalent techniques for achieving it. They are focused on web-scraping for grocery products, but not for price comparison. Although

the authors do not propose to make correspondences between products from different retailers, they discuss on many challenges of the process of data acquisition and manipulation.

The first article explores the development of the FLIP: Food Label Information Program, Canadian branded food composition database developed by employing web-scraping techniques to gather data from seven major Canadian e-grocery retailer web-sites (Ahmed et al., 2022). Addressing the challenges of creating and maintaining national food composition databases, particularly in the dynamic packaged food and beverage sector, the authors emphasize the need for a database that offers brand-specific and up-to-date nutrition information. The study sheds light on the advantages of leveraging automated data collection methods to enhance the database's coverage and temporal relevance. By scraping food labeling information such as nutritional composition, pricing, product images, ingredients, and brand names, FLIP provides a comprehensive understanding of the Canadian food supply. The authors highlight the significance of this approach, as it allows for a detailed evaluation of the market and enables the identification of fraudulent practices or price manipulation by retailers. This research not only contributes to the development of a robust food composition database but also showcases the potential of automated techniques, such as web-scraping and AI (Artificial Inteligence) / ML (Machine Learning) -based optical character recognition, in collecting vast amounts of data from e-grocery retailers.

The other highlighted article on the subject introduces foodDB. A tool for analyzing nutritional facts in the food and drink marketplace. The study's objective is to address the limitations of traditional methods in creating food composition tables and highlight the research potential of foodDB (Harrington et al., 2019).The authors describe foodDB as a terabyte-scale, weekly updated database that utilizes big data techniques for data collection, processing, storage, and analysis. The database collects comprehensive data on food and drink products available in major UK supermarkets, including product name, price, serving size, promotion details, nutrient composition, ingredients, dietary information, brand, manufacturer, and more. The methodology employed in this study involves scraping HTML requests for automated extraction of nutrition and availability data from supermarket web-sites using Python code. The collected data provide insights into the relationship between nutritional quality and marketing of branded foods, as well as timely observations of product reformulation and changes in the food marketplace. They emphasize the flexibility of the developed code, which can leverage loose coupling and object-oriented practices to adapt to changes in individual supermarket web-sites and integrate new data sources. With its comprehensive sampling and granularity, foodDB offers a robust solution for monitoring and analyzing the food and drink marketplace, enabling researchers to uncover valuable insights regarding the dynamics of nutrition and food marketing.

Conclusion

In summary, web-scraping can be a valuable tool for obtaining data and insights for various purposes on a much larger scale. However, it is necessary to consider the ethical and legal implications of using this technique. By obtaining permission, ensuring data accuracy and reliability,

and complying with applicable laws and regulations, web-scraping can be a valuable and ethical practice.

2.4 Applications

Other insights could be obtained through the information gathered by the project that go beyond simply the quantification or direct comparison of entries in the database. More complex analysis could permit even more strategical insights to be obtained with the same data.

Two potential applications for the collected data could be contemplated to extend its utility. First, it is considered applying the data to product recommendation, where retrieved characteristics allow segmentation of products into categories, and later output substitute or complementary alternatives for an item based on similarities. To do that, it is also important to segment both the product and the consumer, matching consumer preferences and product characteristics. The second function is pricing strategies. The provided information allows multiple values to be compared across the same or different products, categories, retailers, and regions, providing insightful information for strategic decisions.

Product Recommendation

Recommendation Systems are intelligent systems that learn about the products a customer has previously interacted with in order to make customized recommendations for undiscovered goods that are likely to catch their interest. These systems can have a positive impact on a retailer's sales by inducing the consumer to add to the basket complementary items that will enhance the experience with the focal product, also it can recommend a replacement item of a higher value that could be of more interest for the buyer and will increase the chance of completing a purchase, or even proposing alternatives for a situation where a customer faces an out-of-stock product. In the following chapters, when mentioning "focal product" it is about the item a customer initially intended to buy and is the center of further recommendations. To successfully understand how recommendations can impact the final result of a sale, it is important to perceive how different products interact and how different types of clients can impact the success of a recommendation.

It is important to note that our project only extracts information regarding listed products, therefore we do not have access to any kind of purchasing history, sales numbers or customer clicks on web-sites. However, our tool is destined to possibly be used by retail chains and other companies, and those do have access to this kind of information and that is why it is relevant to be considered as a possible application.

Recommendation Algorithms

To begin the discussion on algorithms it is relevant to discern two types of recommendations, *complementary* and *substitute* (Shocker et al., 2004). The first occurs when the system recommends products that are different from the focal but complement it as a companion item or service, for

example, when a mobile phone is a focal product, complementary recommendations are chargers, headphones, or cases. The second can be identified as another product that is similar to the focal, again using a mobile phone as an example, substitute recommendations are phones from other brands, with different prices or some different specifications. According to economic theory, complements raise market demand for the focal product because they make it more likely that customers would find additional benefits, in contrast, substitutes lower demand for the focal product due to competition .

When researching how both types of recommendations on different stages of an online purchase impact the willingness to pay, Zhang and Bockstedt (2020) concluded that customers in the final stages of a purchase, i.e. when a decision on the final product has been made, may find substitute recommendations "redundant and unnecessary", meanwhile a customer which is on early stages of the purchase, i.e. reviewing products before a final decision, may find complementary recommendations "disruptive" as aspects of the focal product like price or specifications are more relevant at that stage than accessories or extra services for it. They also realized that for both types of recommendations, the price of the recommended prices could impact the WTP (Willingness to Pay) of the consumer, as visualizing more expensive suggested products could increase the WTP on the focal product.

Continuing on the duality of recommendation systems, Yu et al. (2019) researched available recommendation algorithms for both types, and concluded that while substitute recommenders have numerous popular contributions, there still exist research gaps and opportunities for complementary recommenders. They attribute this gap to the fact that finding complements for products require more work because they are typically weakly related by multiple implicit principles, unlike substitute products that may be precisely specified by interchangeability or similarity. The work begins by establishing the relationship of complementary items, stating that frequent co-purchases are the most accurate data source and therefore defining four forms of interaction:

- Simultaneously purchased item
- Simultaneously purchased categories
- Consecutively purchased item
- Consecutively purchased categories

However, those definitions are insufficient, and two more concepts are introduced:

- **Asymmetry:** purchase of the complement usually occurs as a result of that of the source item rather than the other way around, i.e. a customer purchases a charger for their phone, but not a phone for their charger.
- **Transient:** the item consecutively purchased after the source item can be identified as a complement only when the time interval complies with a certain scenario-specific threshold, i.e. the products bought concurrently with the focal product have a higher chance of being complementary than after a period has passed.

Next, the paper reviews available solutions for complementary recommendation algorithms. The first method adopted was unsupervised learning algorithms, where the use of association rules made it simpler to apply. However, association rules are vulnerable to parameter selection which often leads to a large number of irrelevant connections and bad results and also performs poorly when deployed in large datasets with complex connections. For that, other solutions like the use of Frequent Patterns were proposed to accelerate searches in connections. Another alternative inside unsupervised learning is the use of deep learning, especially the popular model (Baskets And Browsing to Vector) BB2Vec, that joins basket data and browsing history and can have good performance even with smaller datasets of co-purchases.

Supervised learning algorithms were the next step, dealing with the complexity of relations by introducing human-defined labels. Multiple models have been created under this category, for example, the *Visual model*, which creates relationships based on visual aspects of the product image; The *Textual Model*, which leverages the fact that textual descriptions and titles are easier to classify than images; The *Multi-modal Input Model*, that combines both of the previously mentioned in a neural network; The *Co-occurrence Embedding Model*, that instead of using feature-based models, which requires higher storage and computing power, creates relations based on the co-occurrence of listings, a technique adopted in unsupervised models; And finally the *Sequence Model*, proposing a Contextual Recurrent Neural Network, that utilizes contextual features like timestamp, events, and sequence of viewed items to predict whether a substitute or complementary recommendation is preferred for the situation.

Finally, another relevant topic for product recommendation is dealing with PP-OOS (Post Purchase Out-Of-Stock), a situation that one-quarter of customers have experienced in online grocery shopping (Mintel, 2021). PP-OOS happens when the consumer has already completed the purchase and is no longer active in the process, just waiting for the delivery, but in the time-space between finalizing the purchase and the retailer collecting the items for delivery, one item is no longer in stock. The solution for that is replacing the missing item with another, but the decision on what will the replacement be can have a direct influence on the overall experience of the consumer, as one-third of the customers who had replaced items were unhappy (Mintel, 2021).

In their research, the authors began by categorizing the substitution into two segments: matching the brand of the unavailable item, but changing the flavor or matching the flavor but changing the brand, where they also evaluated the impact of changing the brand for a national-brand (NB) or a private-label(PL). PLs have a very strong presence in the Portuguese market, where the expressive adoption of customers mainly for the lower price brings chains to offering extensive product lines, especially when those come from the same origin as the NB ones. Furthermore, it has been researched the effect of offering a product present in the shopping history of the customer. Before moving to their conclusions, it is important to explain two concepts used in the text. A category is Vertical when belonging products can be ordered according to quality or performance, usually labeled from standard to premium. In contrast, a category is Horizontal when the belonging products are ordered by abstract features, like flavor.

First, they concluded that substituting the same flavor is appropriate in a horizontal category,

but in a vertical category, the substitution should reflect the same brand to increase acceptance. Second, when matching flavor, in a horizontal category a private-label substitution is preferred to a national-brand one, as the first is considered a copy of the leading national-brand and therefore should present the most resemblance, but the opposite is true in a vertical category. And lastly, offering a replacement that customers have already purchased considerably raises customer satisfaction in both categories. In other words, familiar products lessen the uncertainty associated with substitution.

Consumer Segmentation

The techniques mentioned above base the recommendations on item characteristics and customer actions but lack information on customer characteristics. For that, a whole range of complementary solutions seeks to identify profiles among the clients for better results in recommendation algorithms.

The first work to be looked into makes a bridge from the previous section to this one. Jin Park and Nyeong Chang (2009) divides the already mentioned recommendation systems into two categories: content and collaborative filtering, where the first one suggests an item similar to the focal, and the second suggests an item that users with similar profiles selected. The problem sits at the intersection of both, where methods based on products disregard groups of customers, and vice-versa. As a solution, the authors propose a model that analyzes product features combined with information on collective behavior similar to the customer in sight. The suggested model creates profiles first by defining an individual profile based on purchase history, clicks, and basket insertions, where for every step, product characteristics are taken into consideration and then searching for a group of customers with similar actions to be analyzed. Supported by experiments in a simulated context, the authors claim that the model not only recommends items that the customer has more interest in but also can recommend a more extensive set of items

Also to improve existing techniques, Peker et al. (2017) reformulates a common approach to clustering methods for segmentation. Clustering consists of grouping objects - in clusters - based on feature similarity, and in the context of data science, one of the most used algorithms is the K-means (MacQueen, 1965), which assigns an object to a cluster based on how closely it is located to the cluster centroid. The authors establish their work in extending the commonly used RFM model (Hughes, 1996) for defining features for k-means clustering computation. The new proposed LRFMP feature model includes Recency (R), Frequency (F) and Monetary (M) features from the previous, but adds Length (L) and Periodicity (P).

- **Length:** this feature is the time interval, in days, between the customer's first and last visits. It shows customer loyalty and the higher the length is, the more loyal a customer is.

- **Recency:** it indicates how up-to-date the interaction of a customer with the company is, and gives information about the repeat purchase tendency.

- **Frequency:** is the number of purchases or visits within a certain period, and it is an indication of customer loyalty.

- **Monetary:** is the total amount spent or the average amount spent per visit during a certain period and measures the contribution of the customer to the revenue of a company.

- **Periodicity:** reflects whether customers visit the stores regularly, defined by the standard deviation of the customer's inter-visit times.

The work presents five grocery retail customer profiles at its conclusion: "high-contribution loyal customers, "low-contribution loyal customers, "uncertain customers", "high-spending lost customers" and "low-spending lost customers". These clearly defined categories aid businesses in characterizing and profiling various customer segments. A benefit like this allows a retail company's management to effectively modify its strategy and effectively spend its resources.

Another relevant article to the field results from a review of implementations for marketing analytics methodologies made by France and Ghose (2019). In the chapter destined to *Customer Segmentation and Grouping* the authors look into clustering techniques to create market partitions and next explain the problems of those and how they are solved. Two techniques of greater importance are reviewed. First, hierarchical clustering (Johnson, 1967) consists of building a tree of clusters, by either starting from a single node and expanding until every item has its own cluster, or instead starting from multiple nodes and combining until only one cluster is left. The different strategies produce a variety of performance results. The next approach is partitioning clustering, where the items are assigned to a pre-defined number of clusters based on a determined rule, for example, minimization of the euclidean distance. A well-known example of this technique is the "k-means", which was already previously mentioned. The application of this method includes a series of challenges, for example, defining the number of clusters, how to manipulate unbalanced data, how to deal with objects with dimensional correlation, and finally, how to define the cluster limits. These variations cause the method to be susceptible to minor changes in the data-set, and therefore should be followed by valuation metrics.

The article follows by stating that clustering alone is insufficient for marketing applications, as it only splits information based on features, but does not tell how to integrate those features into marketing decisions. The solution for this situation comes from a method called Managerial Intuition, where the segmentation strategy is often implemented using two methods (Green et al., 1977). The first type is "a-priori" segmentation, where a cluster-defining descriptor, such as a preferred brand or brand category, is present. The second method is post-hoc segmentation, which involves doing analytical segmentation on a variety of demographic, behavioral, or psychographic traits. Another approach to Managerial Intuition, by Wind (1978) points out that the choice of segmentation bases should vary upon the application. For instance, appropriate segmentation criteria for studies on product positioning include product usage, preference, and benefits; pricing decisions, price sensitivity, and deal proneness; and for advertising criteria, benefits sought, media usage, and psychological traits.

Furthermore, the ability to anticipate marketing response using just demographic segmentation is constrained (Dhalla and Mahatoo, 1976). Psychographic tests that have been adjusted for use in marketing are intended to examine underlying psychosocial characteristics and have a limited ability to forecast the purchase of particular goods. Consumers can be grouped based on their buying habits using behavioral data, such as brand loyalty, but these data cannot tell the difference between a customer who purchases a product because it is the only one available and a customer who has high utility for the product.

France and Ghose (2019) continues addressing other barriers to clustering methods. First is the problem with the assumption taken by default on most techniques that any object lays in only one cluster, but in the real world that is not always true. One possible solution is the adoption of the ADCLUS (**AD**itive **CLUS**tering) Model (Shepard and Arabie, 1979), which implements a similarity matrix for visualizing objects in multiple clusters. The second addressed problem is the targeting of objects that sits at the edges of clusters, as the techniques for that clusters could be less effective on them. For that, it is proposed Fuzzy Clustering reshaped in many versions, for example, the c-means.

The last article on the subject for this literature review is important because it inserts many of the previously defined concepts in the context of the Portuguese market. The dissertation by Campos (2021) reports the analysis of 237 answers in a survey about Portuguese consumer habits to segment the respondents and define marketing strategies for each group. The chosen method for segmenting the results is hierarchical clustering, using Minimum Euclidean Distance for assigning objects to clusters, in a way that the number of the cluster does not need to be defined *a-priori*. Both hierarchical clustering and euclidean distance concepts have been described in the previous paragraphs of this chapter. The features considered were: age, income, frequency of purchase of supermarket products in an online store in the last six months, the average value of each supermarket purchase, and frequency of purchase of products from specific categories. These features characterize the RFM method, also mentioned previously.

Pricing

Pricing analysis can also be performed with the product information retrieved from retailers' websites. Our data allows generating pricing insights for one company based on a comparison of products, categories, and chains. By accessing the information on published product prices, it is possible to compare differences between one focal product on different chains, in different regions, or compared to concurrent products. This type of information can lead to many opportunities for any retailer, like automatic price changes in commerce or support the decision-making process for pricing strategy on any retailer.

Aparicio et al. (2021) documents the findings of pricing studies based on data collected from many leading grocery retailers in the United States. The data included variations in location, timing, and frequency, to allow investigation of the following subjects. They begin by studying price differentiation, i.e., when a focal product price varies between locations of the same retailer

or retailers in the same location. They presented some interesting conclusions, first is that non-uniform prices are more present in online retailers, second, private label products prices are much more uniform than other brands, lastly, price dispersion increases in perishable items and also is three to five times larger at competing chains in the same location than in same chains at different locations. Another factor that influences prices for delivery products is shipping costs, as their research show orders for delivery in closer locations present no variations, meanwhile an extra 10 miles of distance can impact 0.14% of the price. Finally, price differences are directly related to region demographics, as zip codes with higher income are assigned higher prices. On the quantification of price changes, the authors concluded that price changes are much more frequent but represent smaller changes, allowing for the expansion of the price grid.

Another pricing feature introduced by technology is price matching, where retailers change their prices to equalize or beat their competitors. They found that retailers often price-match each other's price for the same product and delivery zip code, in particular, approximately 83% of the matching events take place on prices that are below the median price. However, competitors are not synchronized, i.e., a retailer in a specific location is not more likely to change its prices if a competitor in the same location does.

As pointed out by Aparicio and Misra (2023) when proposed to study the employment of AI for pricing tactics and their impacts, they presented a list with three relevant mentions:

1. **Dynamic Pricing:** Brick-and-mortar retailers have been updating price tags on shelves over time for many years. With digitalization, price tags no longer need to be manually changed by an employee; instead, the price displayed on an eCommerce web-page can change automatically upon a script according to a set of rules. Airlines were one of the first to adopt this tactic, automating prices based on search and demand, buyer location, or proximity to departure. More recent examples can also be found in Uber or similar mobility apps, which use models to rise their prices when demand is high and lower when demand falls, this rewards drivers that relocate to demand-full zones where clients are willing to pay more for the service. Other companies to adopt this strategy are Airbnb, car rentals, and American retail giants like Amazon and Walmart, which use algorithms to match each other prices.

2. **Price discrimination:** This tactic also benefits from simplified price changes, but the subject moves from the environment to the customer. Firms can create dedicated location-based prices for individuals or groups of clients. It is estimated that pricing based on demographic purchasing power, can increase profits up to 19% compared to uniform rules. However, this technique raises concerns about indirectly creating age, race, sexual orientation, or any other social discrimination when defining prices for specific groups, and studies show that this can happen.

3. **Demand learning:** Useful for answering questions like "If my product was cheaper, would it sell more?", "Would it be profitable selling it for that lower price?". So to solve the "How much should my product cost?" question, retailers have been combining economic theories

with manual price variation to find an optimal value. Yet, ML algorithms can automatically find those values through dynamic experimentation with reinforcement learning, like MAB (Multi-Armed-Bandit) or Q-learning, allowing to balance price experimentation and learning during the process, showing that combining reinforcement learning with economic theory results in higher profits than standard methods.

Chapter 3

Methodology

Undertaking a project like this involves many necessary functionalities working together in different stages. Figure 3.1 provides an end to end vision of the steps of the algorithm, highlighting the technology used in each. Figure B.1 (in the appendix) represents a closer approach to the states in the step-function that acquires and manipulates the data from each retailer.

Everything begins with the acquisition of raw data in the step we call "ingest". We run web-scraping algorithms tailored to the features of each web-site, capable of extracting the information we will work along the other steps. The ingest phase required efforts to retrieve all the possible content of the pages while obeying the ethics of web-scraping.

Next, we must parse all the information from multiple web-site to a single standard. As each seller use their preferred formats and contents, we must anticipate inputs from each source to have always the same output. This stage is directly related to the Data Quality, where we constantly search for improvements in the data already ingested into our database, sometimes to improve parsing code for the future, other times to correct the already parsed data.

Once we have standardized fields in the tables of our database, we can finally proceed to match products across retailers. Matching depends on the quality of the results of the previous steps, but when everything runs smoothly, we can use the EAN (European Article Number) of the collected products to make matches among different stores.

After the extraction and manipulation phases are complete, the last step is publishing the data for our users. That is were the dashboard and API are introduced. These are the offered solutions for making our data available without requiring users to deal directly with our databases and algorithms.

The code for our algorithm is mostly written in Python, our databases use PostgreSQL, and our front-end contains Java Script and the React Framework. The code is hosted at an version control repository, to allow stability checks and to improve collaboration.

The main role of our project is played by AWS. The cloud provider is responsible for multiple computing, storage, networking, authentication, and business intelligence resources we use along every step. The benefits and compromises of using such services are mentioned along the chapter.

This chapter provides a comprehensive overview of the development process undertaken in the project. It begins examining the use of cloud computing and infrastructure, detailing how the project leverages cloud-based services for scalability, security, and performance. The next section focuses on data structure, explaining how the product data from grocery retailers' web-sites is organized in database tables. The following section is about how ensure the security of our cloud resources and our data. The chapter further explores the data ingest and parse process, highlighting the web-scraping techniques and technologies employed to automate data acquisition. It also addresses the challenges of matching and resolving conflicts between products from different retailers. Data quality is extensively discussed, emphasizing the measures taken to ensure accuracy and consistency of the acquired data. The chapter finalizes with a description of the deployment phase, outlining the implementation and integration of the developed tool in front-end solutions and the access to our data through an API (Application Programming Interface).

3.1 Cloud Computing and Infrastructure

The project achieves numerous benefits by leveraging the power of AWS cloud services. Firstly, it significantly improves security by employing AWS's robust infrastructure and implementing industry best practices. Secondly, it offers unparalleled scalability, allowing the project to rapidly scale its capacity horizontally or vertically to meet future demands. Thirdly, it ensures high availability through AWS's globally distributed infrastructure, reducing the risk of downtime. Lastly, the cloud-based infrastructure reduces costs associated with owning and managing physical infrastructure while providing granular control over expenditure on each service utilized in the project (Mukherjee, 2019).

A serverless architecture means the cloud provider manages the servers and computing power on behalf of their clients by allocating machine resources as needed. The capacity planning, configuration, management, maintenance, fault tolerance, or scaling of containers, virtual machines, or physical servers is not a concern for developers. The results of serverless computing are volatile and are stored after being processed in brief bursts. There are no allocated computing resources when an app is not in use, allowing for only the actual amount of resources used by an application being used to determine pricing (Li et al., 2023).

In the list of cloud resources employed in our project, multiple services are considered serverless.

- **AWS Lambda:** is a serverless, event-driven compute service that lets you run code for virtually any type of application or back-end service without provisioning or managing servers. We write desired functions in code, like Python, and assign it to the Lambda, which will execute it when triggered. Lambdas can connect to other services, for example, to S3 Buckets to save and load data. We used Lambdas to execute all ingestion, parsing and matching process and also as the computing power for the back-office.

- **AWS Step Functions:** is a visual workflow service that functions like a state machine, where developers can chain the deployment of multiple services in sequence or in parallel. This service permits passing information directly from one service to the other, and can be scheduled to launch at defined times. We primarily used this service to trigger the sequence of AWS Lambda functions. On Appendix B it is possible to visualize an capture of how the step function relates all the necessary lambdas for the whole process.

- **AWS S3:** is a storage service for any type of data. It offers virtually unlimited space, and can vary its pricing and performance according to the needs of the user. Data is stored in "Buckets", a denomination used for the container that will hold objects and files. We used this service to store .json files created during the ingestion and parsing process, and also to host the static files required for the functioning of the front-end web-site.

- **AWS RDS (Relational Database Service):** is a managed relational database service that can adapt to many database engines. We use PostgreSQL for our project. Being a fully managed service means the responsibility for maintaining and providing the service, for example updating the database engine is responsibility of AWS, and not ours.

- **AWS IAM (Identity and Access Management):** as the project is developed in an enterprise context, its important to centralize the distribution of access and permission to higher roles, providing only the minimum necessary to the rest of the developers. IAM allows this functionality, simplifying the management of developing accounts.

- **AWS Cognito:** add user sign-up and sign-in features and control access to our web. Amazon Cognito provides an identity store that scales to millions of users, supports social and enterprise identity federation, and offers advanced security features to protect consumers. We use it to distribute access to our front-end web-site.

- **AWS Quicksight:** is a serverless architecture for the deployment of Business Intelligence Dashboards. We publish dashboards powered by Quicksight on our front-end. The dashboards can access our S3 buckets to retrieve the data it will deploy.

- **AWS Athena:** is an interactive query service that makes it simple to analyze data directly in Amazon S3 using standard SQL. This service is responsible for bridging the required assortments showed on Quicksight and the raw storage of data on a data-lake in a S3 bucket.

- **AWS Glue:** is an ETL(Extract Transform Load) service, which means it is responsible for reaching the data stored in our RDS database, transforming it as required and loading to the S3 buckets which will be used by Athena and Quicksight.

- **Networking Services:** two points create the necessity for utilizing many networking services. First we are dealing with possibly sensitive information, which is private and should not be accessed without authorization. And Secondly, using multiple cloud services means they are mostly connected through the internet, and therefore must be correctly configured

to not be wrongly exposed. AWS offers all the necessary solutions, therefore our project leverages services like NAT gateways, sub-nets, VPCs (Virtual Private Cloud is a service that launches AWS resources in a isolated virtual network), and route tables. We also utilize the API Gateway, a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. More detailed explanation of the necessity of API Gateaway is explained in the following section.

- **Other Services:** many other services have supporting roles for our development, for example Cloud Formation allows creating, deploying and managing many of the previously mentioned resources through coding, making it very simple to replicate the infrastructure if necessary. Another supporting resource with vital importance in the developing stages is Cloud Watch: a monitoring and management service that provides data and actionable insights for services in AWS. Lambdas can be linked to this resource, allowing for easy access to the written logs created by the runtimes.

Figure 3.1 showcases the infrastructure architecture of the project, explaining how the AWS Resources are present in every step through different services.

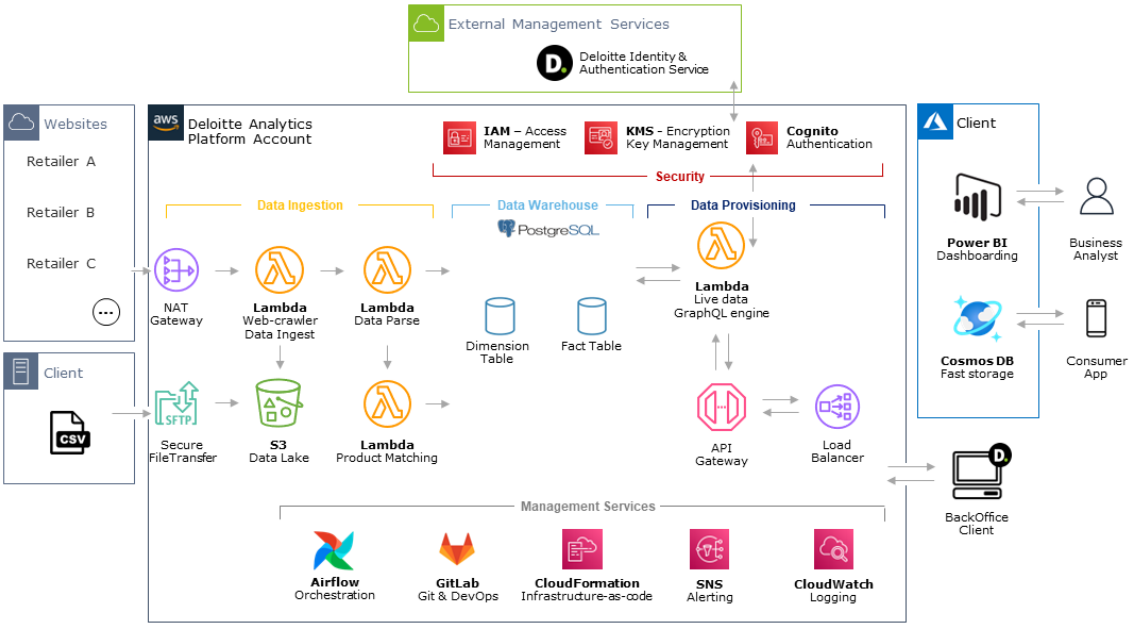


Figure 3.1: Infrastructure Solution Architecture

3.2 Data Structure

The database in our project is divided into fact and dimension tables. By definition, the first contains measurements, metrics, and facts about process, while the second contains descriptive attributes to be used as query constraining.

The first destination of newly ingested data is the raw_retailer table. Each retailer has its table, namely *raw_retailerA*, *raw_retailerB*, and *raw_retailerC*, but for the sake of representation, they are identical in structure, and therefore the proprieties apply to all three. The entries in these tables are assigned a numerical id and the timestamp from the date they were added. The table contains all data retrieved about a product, like its name, brand, product code, content, price, image, URL, EAN, category, promotions, and more. Multiple entries of the same product are possible, diverging only in id and timestamp, however entries are deleted after an interval of certain days has passed in order to maintain a viable size of the tables. The information on these three tables is the source for populating all the other tables.

Once the retailer's raw tables are filled with products, the remaining tables can be populated. These have no constraint or dependency on the raw tables, they receive their data through another code that copies data from one table to another.

Beginning with the *product_brand*, this table hosts the information on brands with products on the raw tables. This table assigns an id to each brand that can be used in other steps of the analysis. Also adds tokens and the timestamp of the most recent update or insert of this brand.

Next, the *product_category*, is populated with a hand-defined category tree with different levels designed to be mapped to the actual categories that are obtained from retailers. This process is achieved through the use of tokens. For every entry of a product from a new category path, this path and the retailer name will be merged into a token, and this token will be inserted in an array in the token column of the correct row of the *product_category* table so that the next product with the same category path can follow this trail to the same standardized category. Although requiring some manual effort during the initial population of the table, after a few ingest cycles there are sufficient entries for the table to be representative in future ingestion, allowing for the products to be automatically allocated to the mapped category. The field *parent_id* is a foreign key to its own id, that holds the relation between different levels of the tree and the field *haschildren* is True if the category has other categories beneath it or equals False if the category is end level. A category with products can not also be a parent of other categories, so a category with product is called "end level".

Continuing, the *store* table holds the information about the digital stores that originated the online access to the product. This table is important for contemplating products with geopricing or limited availability depending on the location. The rows of this table are for example the retailer name, the store address, its coordinates and its identification.

Moving to the *product_ref*, this table is responsible for holding matched products. The matching process is based on the product's EAN, where if a product has an EAN that is not present in any other entry of the table, it becomes a new row, importing the brand id and the category id from other tables. However, if the EAN is already assigned to an entry, then this entry will have just the respective retailer product code updated, as the other information will remain for the product from the initial retailer. A problematic situation can emerge if an EAN is present in multiple rows, which would cause the unique product code constraint to be violated. This situation is referred as conflicts, and is further investigated in the following sections.

Finally, the *product_match* is the conjunction of information from all the other tables. It is responsible for merging the information about matched products in *ref* to pricing information from the raw tables.

Figure 3.2 represents the relationship between the tables in a class diagram.

3.3 Data and Infrastructure Security

Although all the information collected for our project is obtained through scraping and requests for publicly available data on the internet, it is essential to acknowledge the significant intellectual and financial investment required to obtain and sort this data. Therefore, protecting the aggregated value of this data from unauthorized access is of utmost importance.

To ensure secure access, our cloud infrastructure follows the principle of granting minimum necessary privileges to users. This involves multiple layers of authentication and authorization mechanisms. Users are required to have a secure password, utilize a public and private key pair, and employ multi-factor authentication. Access is only granted to the specific resources that are essential for each developer's role and responsibilities.

In addition to robust authentication, access to the resources also depends on the user being connected to the authorized network either locally or through a VPN. This ensures that access is restricted to authorized individuals within the trusted network. Furthermore, IP whitelisting is enforced to control and limit access to our database content, allowing only approved IP addresses to interact with the data.

To protect our cloud services from unauthorized public access, we have implemented a comprehensive set of security measures. These services are deployed within a VPC, which establishes a private network that connects to the internet through a regulated Internet Gateway. Within the VPC, we carefully define and configure subnets as public or private based on the resources' specific requirements. Resources that necessitate internet connectivity are placed in public subnets, while those that do not require external access are isolated within private subnets. Additionally, we employ Security Groups, which act as virtual firewalls, to control inbound and outbound traffic for associated resources, further bolstering the security of our infrastructure.

When it comes to offering access to external users, we rely on AWS Cognito, a fully managed service, to create authorized login credentials. This allows us to authenticate and authorize external users, granting them access to interact with our API and access our dashboard. However, it is important to note that the actual databases and core algorithm remain securely protected and inaccessible to external users.

Finally, since our architecture is serverless-based, all hosting servers are deployed in secure AWS facilities. By leveraging AWS's robust infrastructure, we benefit from increased physical security and reliability. Additionally, to mitigate the impact of potential data center outages caused by geographical incidents, we strategically distribute our resources across different regions, ensuring redundancy and continuity of service.

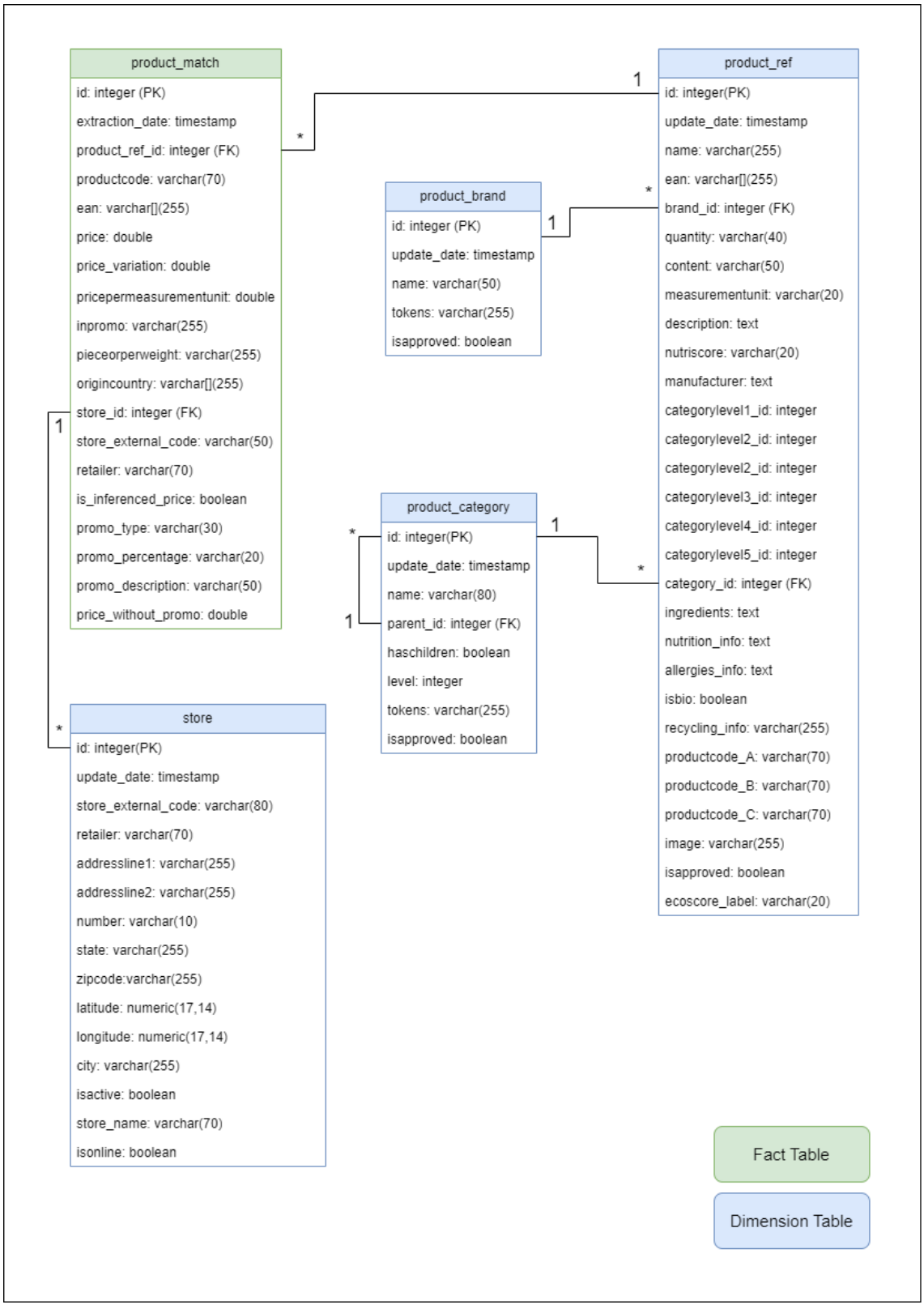


Figure 3.2: Database Class Diagram

Through the implementation of these comprehensive security measures and utilizing serverless technologies, we strive to safeguard the integrity and confidentiality of our collected data while providing authorized users with secure access to the necessary resources.

3.4 Data Ingestion and Parsing

The acquisition of data for the project originates from grocery retailing web-sites, which tend to follow similar design patterns. Each web-site categorizes its products into main categories, such as Fruits, Drinks, Bakery, and more. Within these larger divisions, further subdivisions exist, allowing customers to navigate through specific product types. For example, within the Drinks category, one can discover subcategories like Waters, Beers, Wines, Juices, Sodas, and more.

To ensure accurate data collection, it is essential to identify and exclude irrelevant pages that may not contain relevant product information. These pages could include particular selection tabs, temporary promotions, or informational pages. Considering these factors, an automated parser can be designed to avoid such pages during the subsequent data collection and processing steps. This approach ensures that only valuable and pertinent product information is extracted for analysis.

Due to the vast number of products available on each retailer's web-site, manual data acquisition becomes impractical and time-consuming. To overcome this challenge, the project utilizes web-scraping tools to automate the data collection process. Among the available options, Scrapy, an open-source web-scraping framework written in Python, emerges as the preferred software tool. Scrapy enables users to define the structure of the web-sites quickly. It utilizes a set of rules and selectors to navigate through web-pages by sending requests to the server and parsing the HTML content to extract relevant data.

The web-scraping ingest process is programmed in an daily schedule, running on specific hours during the morning, as the web-sites have less less users during that time of the day. Divided into two crucial steps, firstly the algorithm collects the URLs of the product pages from the retailer's web-site. Secondly, it gathers detailed information from each product's web-page.

The initial step involves annotating the URLs for the web-site's main categories of products and providing them as a list to the scraper. On most web-sites this is information is hard-coded in the HTML of the main page, so we can simply instruct the scraper to extract all the structure. However, some web-sites choose to hide this information from automatic bots by not publishing it in the HTML structure and not including in the network response, so the only alternative is manually navigating through the web-site and creating ourselves the list that will be given ahead, luckily, most web-sites have only a few tenths of categories, making it viable to hand-annotated. Subsequently, the scraper automatically traverses the list, extracting the URLs of individual products within the given categories. The results, containing the individual product URLs, are then saved in .json files for further processing.

Once the product URLs are obtained, the next scraper comes into play. This scraper takes the file with all the single product URLs as input and iterates through the list, scraping relevant information from each product's page. The desired information includes the product name, listed

price, EAN, internal product code (numeric identifier for that product, different to the barcode and defined by the retailer), brand, image URL, quantity, content, measurement unit, description, promotion status, promotion price (if applicable), origin country, organic or vegan labels, nutritional information, ingredients, and the category path within the product's category tree.

As explained before, not every situation requires web-scraping techniques. These can be replaced by API requests when the web-site host provides the service. In this case, the ingest step can be simplified, replacing the scraper with a Python algorithm that makes precise API calls. One of the benefits of such interface is having access to documentation explaining how the requests should be formatted to reach the desired data and how the response will be built, facilitating the following parsing process since the desired information will be in exact fields. When using an API to make requests for data, one can be sure not to violate any ethical or legal rule that other techniques may interfere with.

With the successful completion of the ingestion process, the algorithm proceeds to the parsing stage. During this step, the objective is to make minimal alterations to the collected data, focusing only on the formatting changes required to comply with the predetermined column formats in the database. The parsing stage may involve actions such as removing special characters, standardizing numeric and text formats, correcting measurement units, replacing aliases, and other similar adjustments. Ensuring consistent and standardized data formatting makes subsequent analysis and manipulation more efficient and reliable.

After the parsing stage, the algorithm finalizes by uploading all the processed information into the corresponding raw table in the database. This table serves as the repository for the acquired data, allowing for easy access and further manipulation or extraction as required by the project. The data is now readily available for various analytical tasks, providing valuable insights into the grocery retail domain and supporting informed decision-making processes.

3.5 Matching and Conflicts

Before data is moved from the raw table to the other tables where matching will occur, it is crucial to conduct a comprehensive final check for potential "conflicts". Solving these conflicts play a significant role in ensuring the accuracy and integrity of the data. To understand conflicts, it is necessary to revisit the process of matching products from different retailers.

The matching process is based in the utilization of the product EAN that has been recently ingested and is now present in the raw_retailer table. An EAN is a 13-digit barcode that offers numerous benefits for sellers in the retail industry. It serves as a unique identifier for products, enabling efficient supply chain operations. EANs facilitate automated checkout processes, reducing human errors and improving transaction speed. Additionally, EANs enhance product visibility and interoperability across different platforms, making it easier for sellers to list and sell their products across various sales channels.

Since the EAN is unique to each product across all retailers, the purpose of its utilization is searching for correspondences within the entries of the product_ref table and confirming matches

or creating new entries. The algorithm seeks a positive match for an already seen EAN in the product_ref table entries. If such a match is found, it highlights that the product already exists. Consequently, the algorithm proceeds to copy the relevant information from the raw table to the ref table, ensuring that the product details are up to date and consistent across the system.

On the other hand, if a positive match is not found for the EAN in any of the entries within the product_ref table, it suggests that the product is entirely new and hasn't been previously processed. In this case, the algorithm creates a new entry for the product. This allows the system to accommodate new products and facilitates future matching processes.

However, unfortunately we can not guarantee that the information we extract from the internet is 100 % correct. Sellers may wrongly list products with altered EAN, so there is a potential issue that arises when the same EAN is present in two or more entries within the product_ref table but for different products. The constraints in place do not explicitly prohibit such occurrences, which can lead to conflicts during the matching process. For instance, if the algorithm attempts to insert the details of a product with a duplicated EAN into two different rows in the ref table, it can interfere with constraints such as the requirement for product codes to be unique.

To address this challenge, a dedicated function is implemented to thoroughly check each recently ingested product for potential conflicts. The function assesses whether the product's EAN is or will be associated with multiple entries in the ref table, indicating a conflict situation. If a conflict is identified, the corresponding product ID is flagged to prevent matching during subsequent processes and it is not added to the table. Additionally, the flagged product ID is inserted as an entry in the product_conflict table, enabling further analysis and resolution.

The product_conflict table contains the items that require manual allocation and resolution by an operator who is responsible for deciding the best course of action. These conflicts are typically complex scenarios that cannot be resolved automatically due to the need for human judgment. By highlighting these conflicts, the system ensures that they receive the necessary attention and expertise for appropriate resolution.

Once a solution has been determined by the operator, the conflicted item is marked as a "cleared conflict" within the system. This designation indicates that a resolution has been found and implemented. Consequently, during the next cycle of data ingestion, the item is reintroduced to the rest of the tables, allowing it to proceed through subsequent matching processes.

This systematic approach to identifying and managing conflicts promotes data accuracy, integrity, and reliability within the system. By involving human operators in the resolution process, the system benefits from their expertise and decision-making abilities, ensuring optimal outcomes and minimizing the risk of data inconsistencies.

3.6 Data Quality

A single grocery retailer can list for sale over 40.000 products throughout dozens of categories, each with another dozen of sub-categories, where every partition of products may have different traits, as the relevant features for an apple under the Fruits category may not coincide with the

features of a room deodorizer with apple scent under the cleaning products category. As if inside a single retailer there were not enough standards, the proportion increases when products from different retailers are supposed to match each other in one single standard.

Data quality consists of reviewing the data present in the different tables of the DB (Database) after ingesting and parsing the multiple origins and looking for fixable problems of incompatibility between sources. Most issues can be solved by adjusting the previous steps of the algorithm so the next intakes are correct, however, other situations may require manual correction item by item of the DB so the next correspondent ingestion follows the correct path, as they are caused by intrinsic characteristics of the product and cannot be altered by the previous steps.

Here are some examples of data quality issues and how they where solved:

- Some retailers would use different abbreviations for the same measurement units and, in other cases, use descriptive words. This required a unification of patterns and replacing the units in all entries. For example: "lt" is replaced by "l"; "k" or "kilograms" by "kg"; different scales like "ml", "cl" and "l" all need to be unified; and words like "pieces", "rolls" or "cans" would become simply "u" for "units";

- On products sold in packs, for example 6 packs of 4 units may be listed with quantity equal 1 and content equal 24 units or may be listed with quantity equal 4 and content equal 6 units. This would need to be standardized for correct matching

- Because of situations like the mentioned, sellers can use either the weight or the number of units in a product to base the price per measurement unit, with could lead to diverging values between retailers.

- Some retailers would have the brand included in the product name, when that should be in two different columns of the DB, or the opposite, when a product line is identified as Brand and moved away from the name

- Some retailers would group the EAN of a pack of products and the EAN of a single product in the same corresponding internal product code, while other retailers could have that information in different codes;

- Retailers often diverge on category tree structure, leading to issues when assigning products to their categories;

The first four circumstances in the given example are resolved by updating the parsing algorithm to consider these scenarios, and it will not happen again for new products that join. Nevertheless, situations like the next two cannot be corrected by code.

The last mentioned example deserves an extension to its solving. As described in the data structure section, an ingested product must be placed on one of the categories in the standard tree based on the category from the retailer it originated, however after analysing the category placement of the products in the database, it was noticed that for several categories, unrelated products

were together. The cause was in the detail that the category name "Dry Meat" for example, was present in the butchery category but also in the dog food one and those were merged. Solving this situation required two approaches: first, to rectify the more than 13.000 existing products in the wrong categories, they had to be manually reallocated to the correct categories. However, the problem would continue to exist for unprecedented items, which led to a complete overhaul of the code responsible for placing products in categories. Revisiting the code structure for matching a product to a category, the initial solution consisted of placing the product in the standard category that had a matching name with the final category from its retailer. However, the new solution was improved to consider not only the final sub-category but all of the previous ones in the retailer category tree, in a way that even though the final would be "Dry meat" the complete path would differ in "Butchery - Dry Meat" and "Animal Food - Dry Meat" and result in two different destinations on the standard tree.

In conclusion, data quality depends on continuous monitoring of general data in the database to identify new problems and correct the ones already identified, sometimes by altering the code and others by altering the actual data.

3.7 Deployment

The Retail Match project aims to provide clients with easy access to the retrieved information without requiring them to interact directly with complex data tables or lines of code. To achieve this, two solutions have been developed as interfaces between users and the underlying data: a web-site with predefined charts and tables focused on data-quality and an alternative interface that can be accessed from the web-site, an interactive AWS QuickSight dashboard, focused on displaying information related to the products.

The web-site and dashboard presented as front-end solutions are just two possible applications of the collected data. This solutions are powered by an API, which is implemented by the back-end system and provides access to a data-lake containing all the historic information collected. This API allows users to retrieve data based on their specific needs and preferences, enabling them to develop their own customized front-end solutions.

For instance, if our project's direct user is a retail chain, it would be valuable for them to perform business analyses against their competitors or gain general market insights, keeping the information to themselves. On the other hand, another user might be interested in acquiring the information to create a price comparison tool between retailers and commercialize the access to the tool to other users. In this scenario, the API would be capable of handling requests with different levels of abstraction, catering to the diverse requirements of various users.

By granting users access to the API, they have the flexibility to shape the available data according to their unique use cases. This empowers them to extract valuable insights, conduct in-depth analyses, and generate custom reports aligned with their specific business objectives.

This section, referred to as the "Back-office," encompasses both the back-end and front-end components of this interface, developed in a hybrid of coding in Python and JavaScript and deploying on AWS Services. More details of each division will be explained ahead.

On figure 3.3 we can have a closer view at how the different services provided by AWS play their role in the functioning of the back-office.

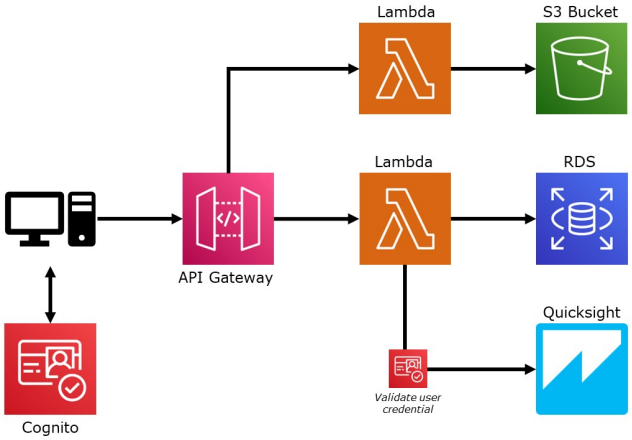


Figure 3.3: Back-office back-end and front-end architecture

Back-end

The back-end plays a crucial role in handling user requests, processing data, and retrieving the desired information. Upon receiving requests from the front end, the back-end algorithm parses the URL, extracting the specified fields and filters.

The development relies on FastAPI to handle incoming requests and redirect them to the appropriate data processing functions. The framework leverages its routing capabilities to match the received routes and execute the corresponding code logic. We defined possible subjects that can be queried through the routes in the URL. The user can specify to view the details about: Catalog Issues (General overview of the whole data-set), Products, Brands, Categories or Stores.

Additionally, the URL can contain user-selected filters from the front-end UI, further refining the data query. These filters impact the resulting query to AWS RDS, allowing users to customize the data displayed based on their specific requirements. For example, selecting issue type, category level, issue priority, searching for a brand or product name, and others are possible.

Once the desired route and filters have been determined, the back-end queries the AWS RDS database, retrieving the relevant data for the specific request. The retrieved data is then processed and transformed to generate the charts and tables required for the front-end display.

The programmatic functionalities of the Back-end are deployed as resources of AWS Lambdas. This architecture allows for efficient routing and scalability as the functions can dynamically scale in or out based on demand.

Front-end

As explained before, the versatility of the API allows for multiple applications of the acquired data. Currently the project has two ready to use front-end interfaces for its users aggregated into an web-site, but users are free to develop their our solutions with the provided data and tools.

The front-end of the Retail Match Back-office is developed using ReactJS, a popular JavaScript library for building user interfaces. It provides a component-based architecture that allows for modular and reusable code, making it ideal for creating a dynamic and interactive user interface. The front-end interface enables users to interact with the Back-end by selecting filters and specifying their preferences.

After login in, the user is presented with the Quality side of the Front-end, where the objective is to rank the issues along products, categories, brands, and stores based on priority. It is also presented a counter of issues by type and category, allowing an overall view of the data quality situation of the project. Navigating through the UI, it is possible to access different pages to view specific results. From the main page, the user can head to other pages, like the product catalog, or take an inside look at the issues related to categories, brands, or stores.

The user can switch between the quality interface and the AWS Quicksight dashboard at the bar displayed on the upper part of the web-site. This second side of the front-end brings to the user an analytic view of the data, displaying tables and charts extracted from AWS Athena. The information presented on the dashboard is, for example, the total of products ingested, store location on a map, price changes for a single product, products per retailer, brand or category, and many others.

Access to the web-site is secured using AWS Cognito, an identity management service. Cognito provides user authentication and authorization functionalities, ensuring only authorized users can access the Retail Match Back-office and its features. This security layer adds extra protection to the sensitive grocery product data being presented.

The static files for the front-end interface, including the JavaScript code and associated assets, are hosted in an Amazon S3 bucket. These files are accessed through an API Gateway, which acts as a proxy to the S3 bucket, delivering the necessary resources to the users' browsers. This architecture allows for efficiency and scalability.

3.8 Contributions

As mentioned in previous chapters, this work is based on a larger-scale version of the same project. Therefore not every step in the process had to be created from the beginning. The result is a combination of adapted and original features. The full-scale project will be referenced as Retail Match in this section for textual comprehension.

Starting with the cloud infrastructure, we have used the existing infrastructure from Retail Match as a guide to what would be required for our project. All the mentioned resources, like Step-functions, Lambdas, RDS database, and S3 storage, use Retail Match as an example but

are developed independently and with authentic algorithms, requiring adjustments to work in a completely unrelated stage and to offer original features.

The data structure on our tables also follows the guidelines from Retail Match. Since the objectives and methods of the project are essentially the same, there was no reason to change the relationship between tables of the definition of tables, mainly because the idea of this project is to have a smaller but resembling version of the full-scale. Again, the tables had their definition inspired by Retail Match but are entirely unrelated and independent.

Moving to the Security section, there were no precedents in this subject since functionalities like access for external users or connecting to the internet are not available in Retail Match. This required a completely new deployment of functionalities.

Data ingestion and Parsing were developed specifically for our use case. The data source and the available fields change from one stage to another, making it unviable to reuse coding. However, we share techniques like web-scraping, API requests, and parsing algorithms to achieve similar goals on both projects.

Again, the Matching and Conflicts steps use adaptations from the code deployed in Retail Match. Firstly, EANs are available to any retailer in Europe, so we can use the matching algorithm as long as it is present. Secondly, conflicts are linked to the data structure and the use of EANs, so we also could benefit from a variation of the original code adjusted to our scenario.

Advancing to Data Quality, even though some issues were noted in Retail Match and their solving could be anticipated for our project, the information sourced from retailers has a meaningful variation from one chain to another. This signifies that most problems present in other stages may not happen in ours, and the opposite is also true, requiring a specific development to solve the data quality issues in our project. The constant mutation in the scraped web-sites requires regular effort to maintain our code updated.

Finally, deploying the back-office solutions in our project required a conjunction of new and adapted algorithms. We developed an original networking infrastructure and the linkage with the database. However, we also adapted other existing implementations from Retail Match to our necessities, like the routing for the API, the UI for the front-end, and the dashboard.

Chapter 4

Results and Analysis

The goal of this project is to provide market insights derived from the inspection of data from food retailers collected via web scraping. All are hosted using serverless architecture. We logically split the results presentation into two parts. We first assess the data's utility in terms of both the quantity and quality. The infrastructure's performance in completing the required tasks is then presented.

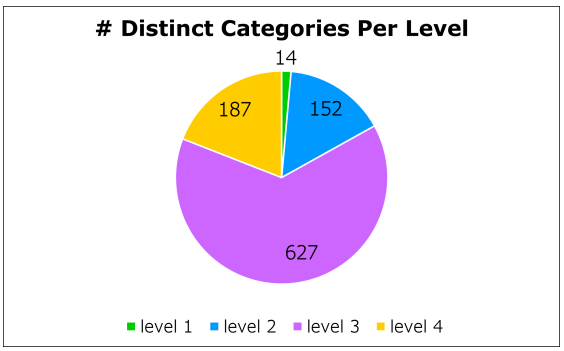
4.1 Collected Data

Results Presentation

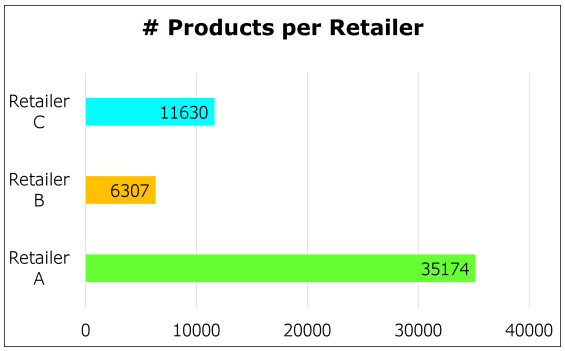
Our algorithm generates data based on items, categories, brands, stores, or retailers. The examination of the components of each part is necessary in order to assess the value of this data to our clients. However, because not all data is qualitative, some must be dealt as quantitative. Figure 4.1 is a compilation of the data displayed in the graphs to which our clients have access via the dashboard.

To summarize the panorama of the project after it has a mature data-set, here are some valuable results:

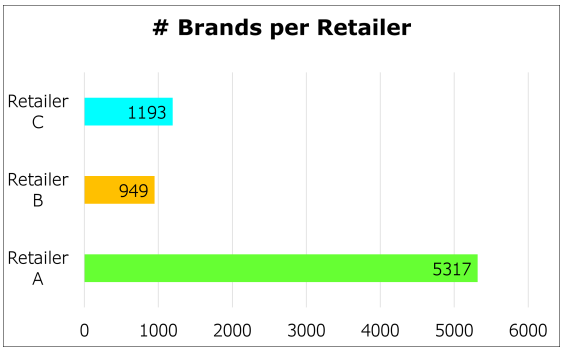
- Daily, around 53.111 products are ingested across the 3 retailers. Retailer A providing around 35.174, Retailer B providing around 6.307, and Retailer C around 11.630.
- Over 44.534 unique products are listed in our database.
- From those 53.111, 4.315 are products with presence in two of the three retailers, and 2.333 are matched from all three retailers. Consequently, 7.239 products represent matches. That means 12,52% of the products are matched.
- 5.817 Distinct brands we collected from retailers and added to our entries.
- The 980 categories are distributed according to four levels: the highest level, 1, has 14 categories. Next, level 2 has 152; level 3 has 627; And finally, level 4 has 187 categories.



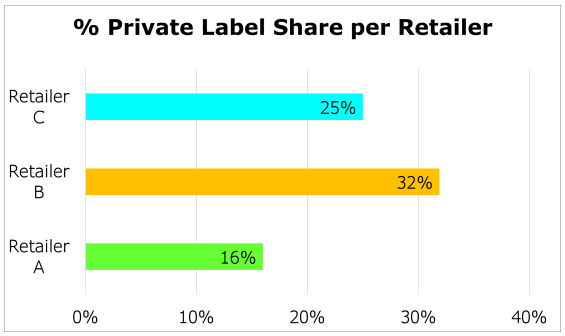
(a) Distribution of Categories per Level



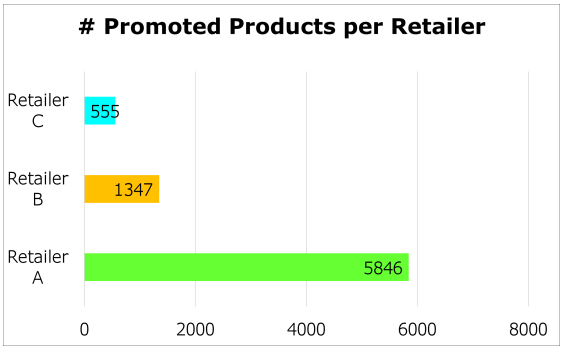
(b) Number of Individual Products per Retailer



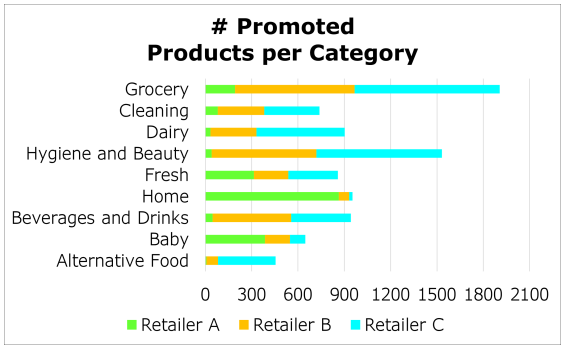
(c) Number of Individual Brands per Retailer



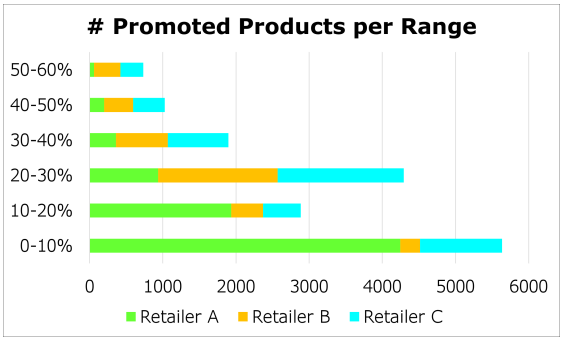
(d) Percentage of Private Labels in a Retailer Catalog



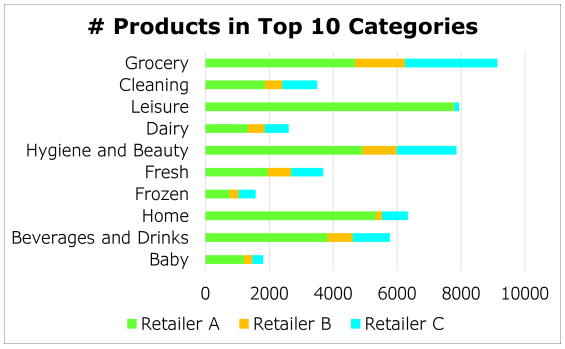
(e) Number of Individual Promoted Products per Retailer



(f) Number of Individual Products per Category and Retailer



(g) Distribution of Discount per Amount



(h) Categories with Largest Composition of Products

Figure 4.1: Information Extracted from Dashboard

- Currently, the three retailers together have 7.748 products with active promotions. The discounts range from 1 to 60% of the original listed price, with an average of 17% discounts over the original listing.
- So far, 36 products were assigned to the product_conflict table, what means that 0,07 % of the ingested products resulted in conflicts that could not be resolved automatically by code and await for an operator to resume ingestion.
- At the moment, the project has been running with stability for 28 days, totaling 1.526.865 ingestion entries (entries for the same product may happen in different days and for different retailers).

Table 4.1 shows the occurrence rate for some relevant fields in our product_ref table that are not obligatory.

Field	Occurrence Rate
Quantity	99,7 %
Content	99,9 %
Measurement Unit	99,8 %
Description	8,6 %
Ingredients	34,8 %
Image	100 %
Nutriscore	0,1 %

Table 4.1: Occurrence Rate for non obligatory fields in *product_ref* table.

Results Discussion

The overview of the data produced by our project indicates that it serves the purpose of correctly representing the selected retail chains individually and in groups. We can effectively work with retailers from different scales and adapt our algorithm to unify formats from different origins in one standard. The acquired data is representative and prepared for analysis between different sources.

We can correctly identify other subjects related to retail, like brand presence, category proportion, and active promotions. Through our historic aggregation, we constantly improve our dataset to remain relevant and expressive as time passes.

One of the most significant issues we face during product matching is dealing with faulty EANs, as those are the foundation of our matching algorithm; however, as shown by the results, we can automatically solve most of the occurrences leaving only a hand-full of cases to be solved manually without interfering with the project functioning.

The distribution of products across categories shown in Figure 4.1f indicates that some retailers are more present in some categories, but the most common categories are balanced accordingly to

the retailer's total of products. Figure 4.1a indicates that the placement of categories across the levels follows a "tree" formatting, as the higher levels are less present than the end levels. However, ideally, the products would only be distributed across three levels. Therefore, the categories at level 4 signal that new entries are being created and should be included in future revisions of the standard mapping.

The Figures 4.1g, 4.1f, and 4.1e about promoted products allows us to identify different strategies across each retailer. Some are more oriented to fewer great discounts, whereas others may offer more promotions but with a smaller discount. The distribution of promotions across categories does not always follow the proportion of total products in each category by retailer. For example, Retailer A is the leader in grocery products, but the last in promoted Grocery Products, and the same happens for Cleaning, Dairy, Hygiene and Beauty, and others. We can also note that some of the categories with the most promoted products are not among the larger ones; for example, Alternative Food is not among the top 10 largest categories but is one of the top categories with promoted products thanks to retailer C, what indicates a particular strategy of the chain.

Looking into the brands we collected, the same pattern of products per retailer repeats in brands per retailer. However, the private labels do not follow this pattern. As Figure 4.1d indicates, the smaller retailer, B, has the largest proportion of private labels in its catalog, while A has the least, even though it has almost five times more products. This is another symptom of the different approaches used by each retailer.

In summary, these are some of the possible insights to be extracted. The presented graphs are general manipulations designed to demonstrate the combined capabilities of our data and dashboard. Collecting historic data precisely as it is ingested allows users to shape the dashboard to their needs.

4.2 Computing Performance

Results Presentation

- To process the 35.174 products for Retailer A the whole process takes 87 minutes, averaging 6,73 products per second.
- To process the 6.307 products for Retailer B the whole process takes 15 minutes, averaging 7,00 products per second.
- To process the 11.630 products for Retailer C the whole process takes 37 minutes, averaging 5,24 products per second.
- In total, to process the 53.111 products it takes 139 minutes, however each retailer runs in parallel. Averaging all three retailers, algorithm scores 6,32 products per second.
- Since its stable implementation, the process has a success rate of 89% (from the 322 combined runs so far, only 38 failed to run).

Table 4.2 contains the billed memory for each lambda invoked during Ingest, Parse, Conflict Solving and Matching. Map Concurrency is how many times each lambda has to cycle iteratively in order to contemplate all the necessary products in periods of maximum fifteen minutes, as the service imposes.

Retailer Name	Ingest	Parse	Conflicts	Match	Map Concurrency
Retailer A	813 MB	128 MB	121 MB	96 MB	10
Retailer B	149 MB	150 MB	120 MB	95 MB	9
Retailer C	183 MB	220 MB	120 MB	95 MB	3

Table 4.2: Billed memory per lambda function for each retailer.

Results Discussion

The results obtained from the infrastructure developed for our project highlight the advantages of deploying our resources in AWS serverless computing.

By leveraging serverless technologies, we have gained access to high-performance hardware that efficiently handles the required tasks. This has resulted in faster processing times and improved overall performance.

Additionally, the scalability of serverless computing has played a crucial role in our project's success. The ability to dynamically adjust our computing resources based on the specific needs of each process has allowed us to optimize resource allocation and effectively manage workload fluctuations. Whether it's a high-demand period or a low-activity phase, we can easily scale up or down to ensure efficient resource utilization and cost-effectiveness.

Another significant advantage of our infrastructure is the high availability it offers. The informative logs provided by AWS enable us to closely monitor the performance of our algorithm and promptly address any issues or failures that may occur. This has resulted in minimal downtime and a seamless operation of our system, ensuring a reliable experience for our users.

Furthermore, with AWS taking care of essential security measures and regular maintenance tasks, we can focus on other critical aspects of our project without compromising on security or system stability.

In summary, our project has benefited from a robust and reliable infrastructure deployed on AWS. We have witnessed a significant reduction in downtime, effective utilization of computing and storage resources, simplified maintenance procedures, and consistently high performance across all operations. By harnessing the power of serverless computing and leveraging the extensive capabilities offered by AWS, we have successfully created an infrastructure that meets our project's requirements and ensures optimal results.

4.3 Use Cases

To illustrate how users of our project could make analysis and obtain insights from our data, we assembled a compilation of real use cases we have encountered in the data we possess.

Every product, brand or retailer mentioned in this section has been anonymized to preserve the identity of the involved entities.

1. Historical data collection allows identifying situations like the one shown in Table 4.3. For that specific product (EAN: 560xxxxxxx279; product_ref ID: 1046), we can notice that retailers A and B launched a promotion for the same product at the same price that began and ended in the same period. The product in the point has been anonymized to preserve the identity of the involved brands. With our current data, we could identify 2.581 possible situations like this. This occurrence is more common in categories related to Health and Beauty, House and Cleaning, and Baby.

2. Another viable topic for analysis is Geopricing, which is characterized by the variation in the price of a product according to the geographic location of the stores of the same retailer. In the example in Table 4.4, we selected five products that vary in price depending on the stores they are sold. These stores from Retailer C are situated more than 200 kilometers apart. We can see that the variation is not temporary, as it is maintained for several days. Like these selected products, our database indicates other 1.131 products from these two stores from Retailer C with the same type of variation.

3. Digital platforms allow dynamic changes in product prices. We can use the aggregated information about product prices to track how it variates over time. As time passes, this information increases its potential by making it possible to complete long-term analyses and predictions for upcoming trends. Figure 4.2 indicates price variation for a few selected products in Retailer A's Fresh Fruits and Vegetables section of one store. The category shows price variation of over 60% for its products. From the time the algorithm has collected information, it is already possible to notice price trends. As more data is collected, it will be possible to compare the prices in different seasons and see its impact on fresh product prices.

4. This kind of historical information supports verifying price rises before an expected sale, like Black Friday, which would be an illegal practice in most countries, and there is significant media attention to such cases during that period. Unfortunately, we have not collected data for that period, but this capability will be employed when the time comes.

   Also, we can analyze products that have been in promotion for very long periods. This could give consumers a false impression that they are benefiting from an occasional discount and buying something impulsively, although the discount could be considered a regular price for that product. These analyses also benefit from more extended time series, but we can indicate the possibility of performing such research with the data collected so far.

Date	Product Code	Price	In Promo	Price w/o Promotion	Store Code	Retailer
14-06-2023	2xxxx32	2.99	False	Null	4xxxx9	A
14-06-2023	5xxxx71	2.99	False	Null	4x1	B
13-06-2023	2xxxx32	2.99	False	Null	4xxxx9	A
13-06-2023	5xxxx71	2.99	False	Null	4x1	B
12-06-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
12-06-2023	5xxxx71	1.94	True	2.99	4x1	B
11-06-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
11-06-2023	5xxxx71	1.94	True	2.99	4x1	B
10-06-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
10-06-2023	5xxxx71	1.94	True	2.99	4x1	B
09-06-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
09-06-2023	5xxxx71	1.94	True	2.99	4x1	B
08-06-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
08-06-2023	5xxxx71	1.94	True	2.99	4x1	B
07-06-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
07-06-2023	5xxxx71	1.94	True	2.99	4x1	B
06-06-2023	2xxxx32	2.99	True	Null	4xxxx9	A
06-06-2023	5xxxx71	1.94	True	2.99	4x1	B
05-06-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
05-06-2023	5xxxx71	1.94	True	2.99	4x1	B
04-06-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
04-06-2023	5xxxx71	1.94	True	2.99	4x1	B
03-06-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
03-06-2023	5xxxx71	1.94	True	2.99	4x1	B
02-06-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
02-06-2023	5xxxx71	1.94	True	2.99	4x1	B
01-06-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
01-06-2023	5xxxx71	1.94	True	2.99	4x1	B
31-05-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
31-05-2023	5xxxx71	1.94	True	2.99	4x1	B
30-05-2023	5xxxx71	1.94	True	2.99	4x1	B
30-05-2023	2xxxx32	1.94	True	2.99	4xxxx9	A
29-05-2023	2xxxx32	2.99	False	Null	4xxxx9	A
29-05-2023	5xxxx71	2.99	False	Null	4x1	B
28-05-2023	2xxxx32	2.99	False	Null	4xxxx9	A
28-05-2023	5xxxx71	2.99	False	Null	4x1	B

Table 4.3: Price Combination for the same product in retailer A and B.

Date	Ref ID	Product Code	EAN	Price	InPromo	Store Code	Retailer
26-06-2023	387	2xxx9	289xxxxxxxxx0	0.77	False	2xx0	C
26-06-2023	387	2xxx9	289xxxxxxxxx0	0.57	False	7xx6	C
25-06-2023	387	2xxx9	289xxxxxxxxx0	0.77	False	2xx0	C
25-06-2023	387	2xxx9	289xxxxxxxxx0	0.57	False	7xx6	C
24-06-2023	387	2xxx9	289xxxxxxxxx0	0.77	False	2xx0	C
24-06-2023	387	2xxx9	289xxxxxxxxx0	0.57	False	7xx6	C
26-06-2023	124518	5xxx2	843xxxxxxxxx7	2.99	False	2xx0	C
26-06-2023	124518	5xxx2	843xxxxxxxxx7	1.99	False	7xx6	C
25-06-2023	124518	5xxx2	843xxxxxxxxx7	2.99	False	2xx0	C
25-06-2023	124518	5xxx2	843xxxxxxxxx7	1.99	False	7xx6	C
24-06-2023	124518	5xxx2	843xxxxxxxxx7	2.99	False	2xx0	C
24-06-2023	124518	5xxx2	843xxxxxxxxx7	1.99	False	7xx6	C
23-06-2023	130784	5xxx6	560xxxxxxxxx8	31.99	False	2xx0	C
23-06-2023	130784	5xxx6	560xxxxxxxxx8	24.99	False	7xx6	C
22-06-2023	130784	5xxx6	560xxxxxxxxx8	31.99	False	2xx0	C
22-06-2023	130784	5xxx6	560xxxxxxxxx8	24.99	False	7xx6	C
21-06-2023	130784	5xxx6	560xxxxxxxxx8	31.99	False	2xx0	C
21-06-2023	130784	5xxx6	560xxxxxxxxx8	24.99	False	7xx6	C
10-06-2023	124517	6xx8	872xxxxxxxxx9	4.99	False	2xx0	C
10-06-2023	124517	6xx8	872xxxxxxxxx9	3.29	False	7xx6	C
09-06-2023	124517	6xx8	872xxxxxxxxx9	4.99	False	2xx0	C
09-06-2023	124517	6xx8	872xxxxxxxxx9	3.29	False	7xx6	C
08-06-2023	124517	6xx8	872xxxxxxxxx9	4.99	False	2xx0	C
08-06-2023	124517	6xx8	872xxxxxxxxx9	3.29	False	7xx6	C
24-06-2023	107013	2xxx0	869xxxxxxxxx2	12.0	False	2xx0	C
24-06-2023	107013	2xxx0	869xxxxxxxxx2	9.99	False	7xx6	C
23-06-2023	107013	2xxx0	869xxxxxxxxx2	12.0	False	2xx0	C
23-06-2023	107013	2xxx0	869xxxxxxxxx2	9.99	False	7xx6	C
22-06-2023	107013	2xxx0	869xxxxxxxxx2	12.0	False	2xx0	C
22-06-2023	107013	2xxx0	869xxxxxxxxx2	9.99	False	7xx6	C

Table 4.4: Price variation in Retailer C's stores.

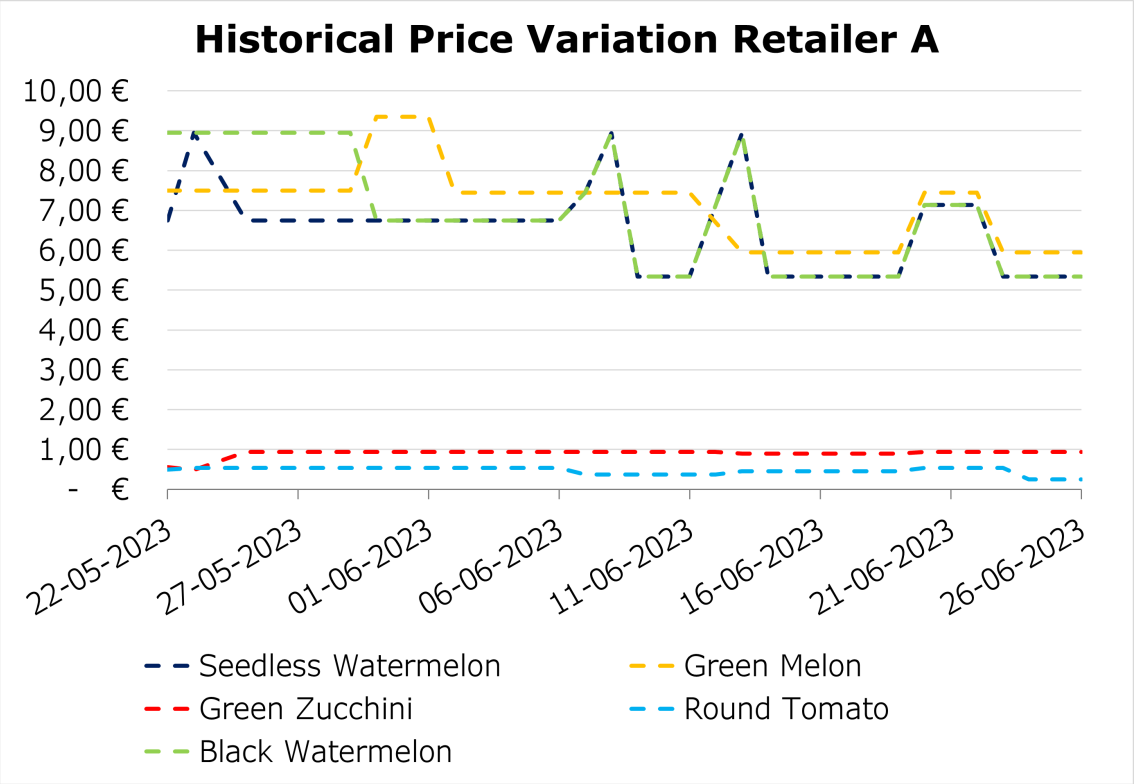


Figure 4.2: Historical price variation Retailer A

Table 4.5 displays the categories with more products in this situation, indicating that they are more frequent in specific categories than others. The results point to 10.005 products in this situation. It is important to note that as the time series increases, we can better filter these results to separate products constantly in promotion from others in long promotions. If, instead of 30 days, we could analyze for 90 days, there would probably be fewer results. Also, no exact metric defines very long and constant promotions, as it would be the analyst's responsibility.

Category Name	Category ID	Number of Products
Yogurts	6045	368
Makeup	6426	363
Masks and conditioners	6150	290
Shampoos	6089	277
Biscuits, cookies, and cakes	5746	242
Laundry detergent	6098	234
Table wines	6059	210
Cheeses	6057	189
Oral hygiene	6043	184
Clothing care	6308	170

Table 4.5: Top 10 categories with most products in promotion for more than 30 followed days.

5. Inflation is a relevant topic moment, and there is great interest in researching its impact on retailers and consumers (Barros, 2023). The price analysis allows tracking price rises and comparing with the inflation rate. By looking at a product's price *versus* the content of its pack, we can identify cases where the brand or retailer reduced the net content to maintain the price or where the price was increased beyond inflation but without increasing the content proportionally.

6. Finally, there are multiple other insights that could be obtained by working with the data. Other ideas are still being considered for future implementation. Machine Learning models could be used to predict future prices for products. Other ML models could also be used to relate products in clusters according to the set of information we have about the products. There can be developed matching capabilities between complementary or substitute products to allow the algorithm to work like a recommendation system. With a redesigned UI, a client could use use the data to create a B2C oriented platform for product comparison.

Chapter 5

Conclusion and Future Work

This chapter summarizes the objectives, research, and methodology. The results are shortly discussed, and next, we revisit the main difficulties encountered during development. Finally, it outlines the research contributions and suggests future directions for the project, offering potential areas for further development and improvement in the field.

5.1 Summary

In recent years the democratization of the internet and smartphones led to a permanent change in our society. Habits changed in many ways, and this transition impacted the retail sector. Especially after the COVID-19 pandemic, the preference for shopping online instead of in a physical store has increased. The food retail sector is one of the joiners in this transition. Retail chains brought competition to online markets, matching each other prices and creating dynamic strategies, and offering delivery options for all their products. However, the consumer is also better informed and has more mobility to choose between sellers.

In this work, we proposed to develop a software to inform and guide decisions for companies in the retail sector. By accessing data about products, brands, and retailers, the client with access to our software will be able to notice patterns between its competitors, anticipate trends and drive improvements to its strategies.

This project completed its purpose by performing with highly available and precise data. We function in a minor adaptation to prototype a simplified version of a full-scale deployment, but already showcasing our capabilities and displaying powerful insights.

We use data mining from the web-sites where retail chains sell their products to obtain our data. Then we parse and upload this data to a relational database. With the collected data, we use each product's EAN to match and compare the products in three major Portuguese chains. Nevertheless, before the data can be published in dashboards or accessible through the API we provide to our clients, there has to be an effort in data quality to guarantee standards and unify contents from all retailers.

All these functionalities are achieved through the utilization of cloud services provided by AWS. We chose to deploy all the computing, storage, networking, and authentication tasks in the serverless managed environment provided by Amazon. This resulted in a project with high performance, high availability, scalable when necessary, and the required security of our data.

Finally, after running the project with stable results for a month, we are satisfied with the quality and quantity of the collected data. It allows our clients to make the necessary strict analyses and is representative of the robust algorithm we developed. The cloud infrastructure responds to the necessities of our project, creating virtually no limitations to its computing potential.

5.2 Main Difficulties

Throughout the many development steps of the project, we faced multiple difficulties. Even though we solved all of them, some required more effort than others.

In the web-scraping stage, we faced severe barriers to obtaining information from constantly changing web-sites. We developed an algorithm that adapts to most changes, eventually requiring minor updates. Another adversity was caused by dealing with the excess of requests to the web-site that would eventually block our requests. This signaled that we were abusing the hosting infrastructure, so we should make more controlled and efficient requests to reduce that stress.

Another series of obstacles occurred once we began populating our tables with data. First, we had to define a standard category tree that would conciliate the formats from all retailers. This involved manually corresponding hundreds of categories. Also, during the first ingestion of the data on our tables, we had to identify divergences from the standard we defined for each field and make corrections to our code and tables, which sometimes could be a manual and time-consuming job.

Finally, the deployment on AWS brought multiple benefits but also some challenges. Since everything was hosted on the cloud, we had to carefully design what services should be accessible to the public internet or private. This proved challenging, especially when a private resource required access to public internet content. Another problem was that during the debug stages of our Python code, the time necessary to deploy to the cloud services and run the whole job was longer than simply running a local test version. Some algorithms could not run locally because we had to test how the cloud resources would interfere with their functioning, and these long wait times were impactful.

5.3 Main Contributions

The main contribution of this project is the successful development of a comprehensive software solution that fulfills all the required capabilities. This achievement provides Deloitte with a valuable asset that has the potential to attract new clients, not only for our team but also for other teams with similar objectives. The code and cloud infrastructure we have developed are valuable examples for other internal projects requiring similar functionalities. This demonstrative version of the

software also acts as a testing stage for features that may be implemented in the full-scale project, ensuring its effectiveness and seamless integration. Overall, this software solution represents a significant milestone for our team and Deloitte, offering great potential for growth and providing a solid foundation for future success.

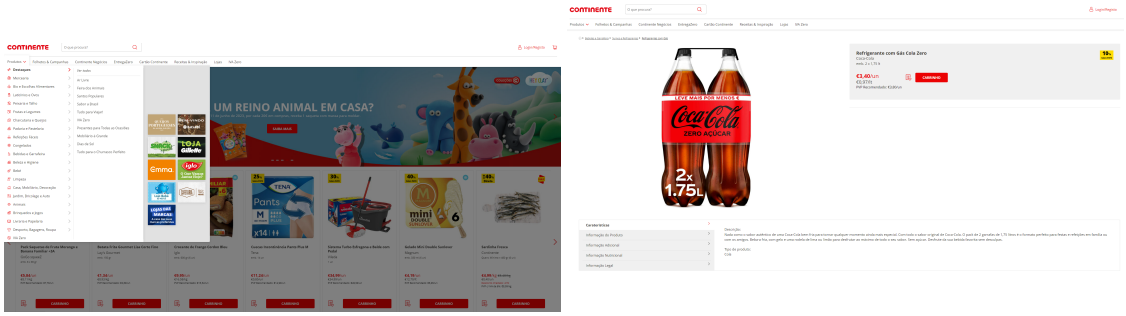
5.4 Future Work

Upon accomplishing the proposed objective, several potential improvements have been identified throughout the development process, which are yet to be implemented in the future. Firstly, extracting additional fields from product pages could enhance the data stored in our tables. However, due to the time-consuming nature of this task and the added complexity it introduces to the algorithm, we have decided to focus on extracting only the most relevant fields. Secondly, there is room for optimizing the scraping process to make it more efficient regarding resource consumption and reducing the impact on the target web-sites. However, this optimization has been deferred to a later code revision once the core functionalities have been thoroughly refined.

Another feature on our roadmap is the capability to rectify fields of previous entries if new ingestion of the same products updates those fields. While this decision could enhance data quality, it requires careful analysis of its impact, as it may compromise historical data. Finally, our dataset accumulates valuable historical data as time progresses, enabling more precise analysis. In the future, we are prepared to expand our sources by adding more retailers to our data ingestion process. Moreover, we are open to replicating our deployment model in other retail markets or even different countries, thereby extending the reach and applicability of our solution.

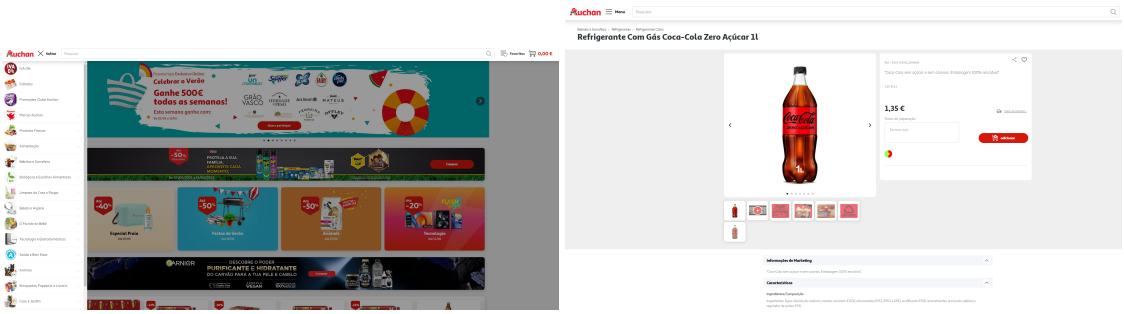
Appendix A

Captures from Retailers Web Pages



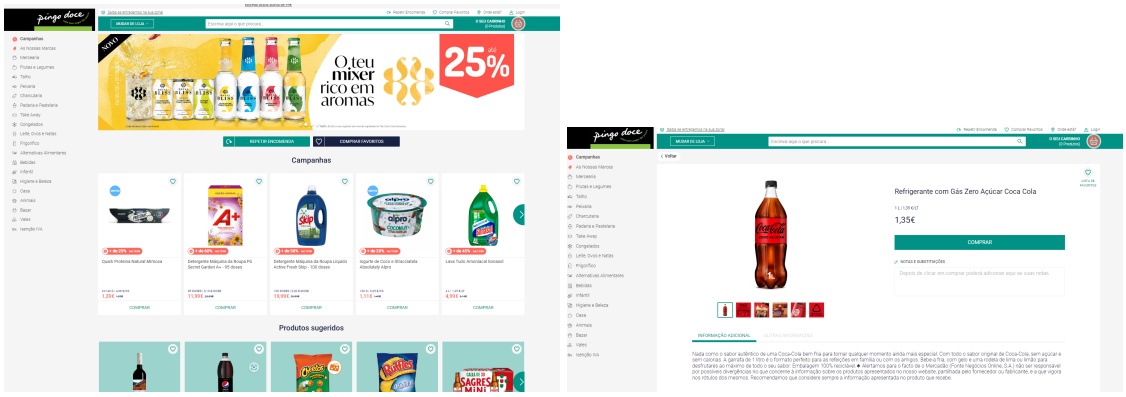
(a) (b)

Figure A.1: Main and Product page at Continente's Website (a) and b) respectively)



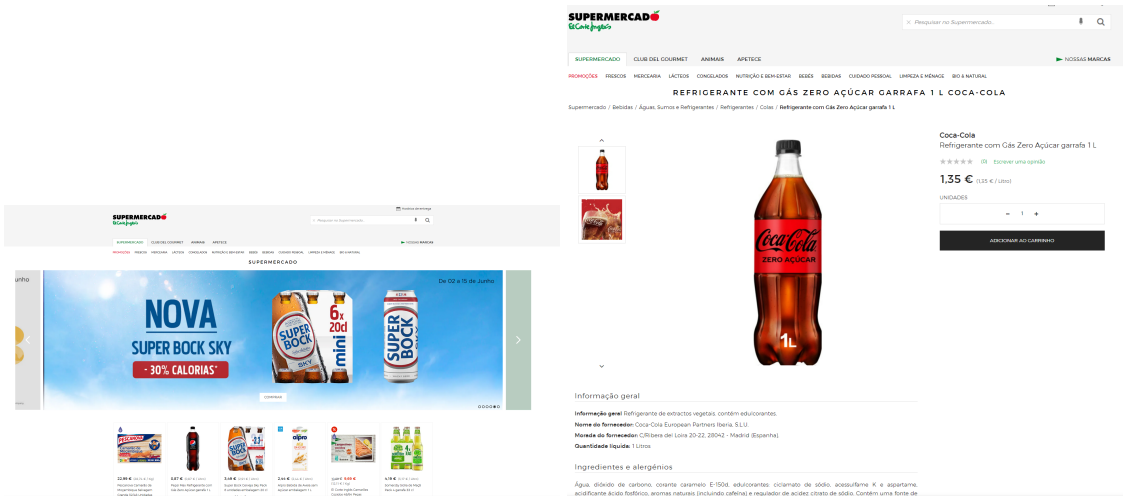
(a) (b)

Figure A.2: Main and Product page at Auchan's Website (a) and b) respectively)



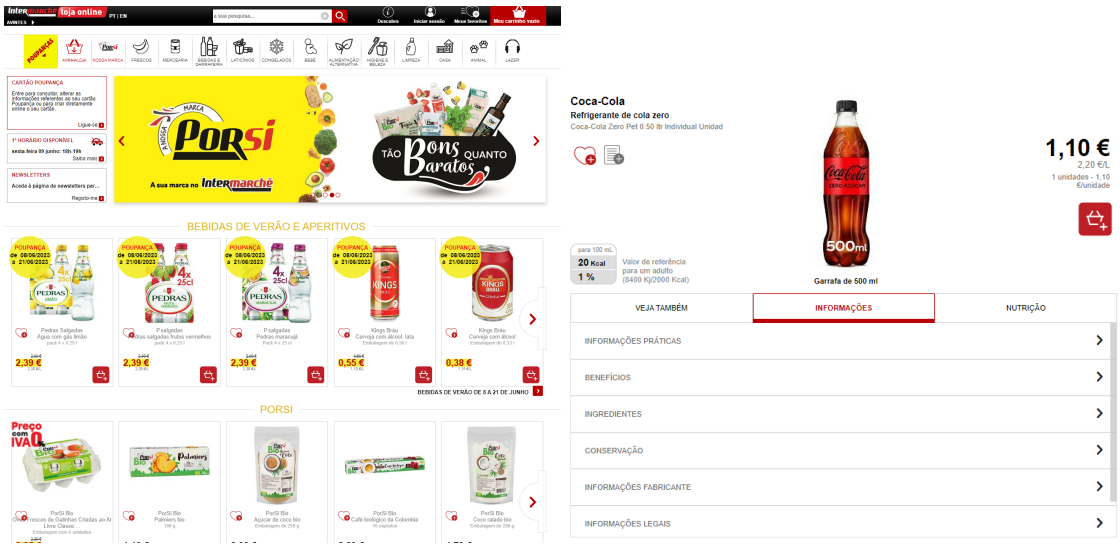
(a) (b)

Figure A.3: Main and Product page at Mercadão's Website (a) and b) respectively)



(a) (b)

Figure A.4: Main and Product page at Supercor's Website (a) and b) respectively)



(a) (b)

Figure A.5: Main and Product page at Intermarche's Website (a) and b) respectively)

Appendix B

AWS Step-Function

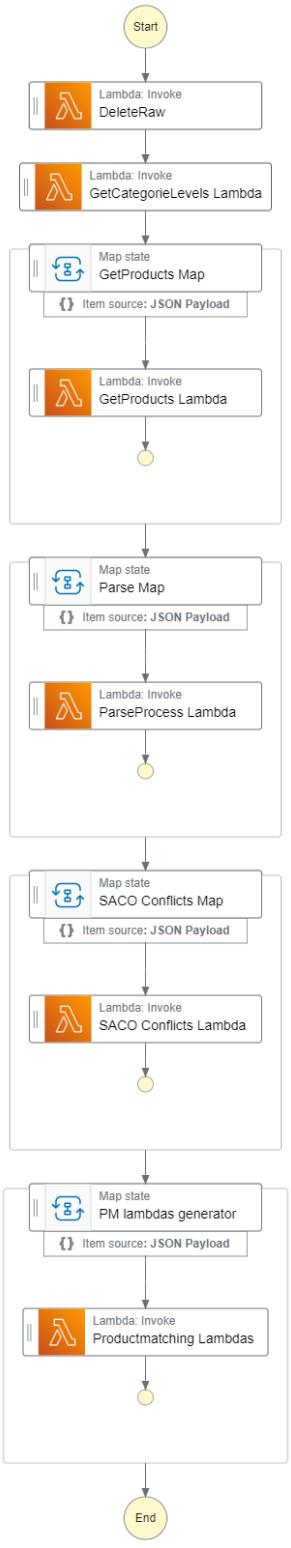


Figure B.1: AWS Step-Function

References

Abbu, H. R., Fleischmann, D., and Gopalakrishna, P. (2021). The digital transformation of the grocery business-driven by consumers, powered by technology, and accelerated by the covid-19 pandemic. In *Trends and Applications in Information Systems and Technologies: Volume 3 9*, pages 329–339. Springer.

Ahmed, M., Schermel, A., Lee, J., Weippert, M., Franco-Arellano, B., and L'Abbé, M. (2022). Development of the food label information program: A comprehensive canadian branded food composition database. *Frontiers in Nutrition*, 8:1319.

Akter, S. and Wamba, S. F. (2016). Big data analytics in e-commerce: a systematic review and agenda for future research. *Electronic Markets*, 26:173–194.

Alam, A., Anjum, A. A., Tasin, F. S., Reyad, M. R., Sinthee, S. A., and Hossain, N. (2020). Upoma: A dynamic online price comparison tool for bangladeshi e-commerce websites. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 194–197. IEEE.

Aparicio, D., Metzman, Z., and Rigobon, R. (2021). The pricing strategies of online grocery retailers. Technical report, National Bureau of Economic Research.

Aparicio, D. and Misra, K. (2023). Artificial intelligence and pricing. *Artificial Intelligence in Marketing*, 20:103–124.

Asawa, A., Dabre, S., Rahise, S., Bansode, M., Talele, K. T., and Chimurkar, P. (2022). Co-mart-a daily necessity price comparison application. In *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pages 1076–1080. IEEE.

Barros, R. (2023). Conheça o impacto da inflação no seu carrinho de compras. *O Público*. Accessed: 2023-06-27.

Bray, T., Paoli, J., Sperberg-McQueen, C. M., Eve Maler, E., and Yergeau, F. (2008). *Extensible Markup Language*. W3C. Accessed: 2023-06-28.

Bumblauskas, D., Nold, H., Bumblauskas, P., and Igou, A. (2017). Big data analytics: transforming data to action. *Business Process Management Journal*, 23(3):703–720.

Campos, A. C. V. D. (2021). Caracterização da segmentação dos consumidores do retalho alimentar online como adaptar a comunicação ao público-alvo? Master's thesis, Universidade Católica Portuguesa.

Carolan, M. (2018). Big data and food retail: Nudging out citizens by creating dependent consumers. *Geoforum*, 90:142–150.

Cavallo, A. (2013). Online and official price indexes: Measuring argentina's inflation. *Journal of Monetary Economics*, 60(2):152–165.

Chaulagain, R. S., Pandey, S., Basnet, S. R., and Shakya, S. (2017). Cloud based web scraping for big data applications. In *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 138–143. IEEE.

Coppola, D. (2021). Global development of e-commerce shares of grocery stores before and after the coronavirus (covid-19) pandemic. Technical report, Statista. Accessed: 2023-06-23.

Dannenberg, P., Fuchs, M., Riedler, T., and Wiedemann, C. (2020). Digital transition by covid-19 pandemic? the german food online retail. *Tijdschrift voor economische en sociale geografie*, 111(3):543–560.

Dhalla, N. K. and Mahatoo, W. H. (1976). Expanding the scope of segmentation research: Segmentation research must cover more of the total marketing problem if it is to be operational and profitable. *Journal of Marketing*, 40(2):34–41.

Dong, X. and Byrne, A. (2022). Retail trends. Technical report, U.S. Department of Agriculture. Accessed: 2023-06-23.

Edelman, B. (2012). Using internet data for economic research. *Journal of Economic Perspectives*, 26(2):189–206.

France, S. L. and Ghose, S. (2019). Marketing analytics: Methods, practice, implementation, and links to other fields. *Expert Systems with Applications*, 119:456–475.

Gonçalves, R. (2023). Filipe simões jorge (lisie.app): Estamos à procura de investidores e a definir o modelo de negócio. *Hipersuper*. Accessed: 2023-06-23.

Grand View Research (2020). Food & grocery retail market size report. Technical report, Grand View Research. Accessed: 2023-06-23.

Green, P. E., Carroll, J. D., and Carmone, F. J. (1977). Design considerations in attitude measurement. *Moving ahead with attitude research*, pages 9–18.

Halzack, S. (2015). The staggering challenges of the online grocery business. *The Washington Post*. Accessed: 2023-06-23.

Harrington, R. A., Adhikari, V., Rayner, M., and Scarborough, P. (2019). Nutrient composition databases in the age of big data: fooddb, a comprehensive, real-time database infrastructure. *BMJ open*, 9(6):e026652.

Hillen, J. (2019). Web scraping for food price research. *British Food Journal*, 121(12):3350–3361.

Hughes, A. M. (1996). Boosting response with rfm. *Marketing Tools*, 3(3):4–10.

Jílková, P. and Králová, P. (2021). Digital consumer behaviour and ecommerce trends during the covid-19 crisis. *International Advances in Economic Research*, 27(1):83–85.

Jin Park, Y. and Nyeong Chang, K. (2009). Individual and group behavior-based customer profile model for personalized product recommendation. *Expert Systems with Applications*, 36:1932–1939.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.

Jorge, V. (2020). Quota de mdd em portugal acima da média europeia. *Distribuição Hoje*. Accessed: 2023-06-23.

Khatter, H., Sharma, A., Kushwaha, A. K., et al. (2022). Web scraping based product comparison model for e-commerce websites. In *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, pages 1–6. IEEE.

Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing & Its Applications*, 13(3).

Krotov, V., Johnson, L., and Silva, L. (2020). Tutorial: Legality and ethics of web scraping. *Communications of the Association for Information Systems*.

Krotov, V. and Silva, L. (2018). Legality and ethics of web scraping. *Emergent Research Forum*.

Lawson, R. (2015). *Web scraping with Python: scrape data from any website with the power of Python*. Community experience distilled. Packt Publishing.

Li, Y., Lin, Y., Wang, Y., Ye, K., and Xu, C. (2023). Serverless computing: State-of-the-art, challenges and opportunities. *IEEE Transactions on Services Computing*, 16(2):1522–1539.

Maciel, J. (2022). Supersave, a app portuguesa que compara os preços para ajudar a contornar a inflação. *O Público*. Accessed: 2023-06-23.

MacQueen, J. (1965). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, page 281.

Massimino, B. (2016). Accessing online data: Web-crawling and information-scraping techniques to automate the assembly of research data. *Journal of Business Logistics*, 37(1):34–42.

Medina, A. (2021). The portuguese food retail sector. Technical report, U.S. Department of Agriculture. Accessed: 2023-06-23.

Mercatus (2020). The evolution of the grocery customer. Technical report, Mercatus. Accessed: 2023-06-23.

Mintel (2021). Uk online grocery retailing market report 2021. Technical report, Mintel. Accessed: 2023-06-23.

Mitchell, R. E. (2018). *Web Scraping with Python: Collecting More Data from the Modern Web*. O'Reilly Media, second edition edition.

Moradi, M. and Keyvanpour, M. (2015). Captcha and its alternatives: A review. *Security and Communication Networks*, 8(12):2135–2156.

Mukherjee, S. (2019). Benefits of aws in modern cloud. *arXiv*.

Nugroho, M. D. (2022). The highest retail price checker for web based medical equipment and necessities. *Engineering, MAthematics and Computer Science (EMACS) Journal*, 4(2):61–65.

Paupério, M. F. A. (2020). Avaliação do impacto de ações promocionais nas vendas no contexto do retalho alimentar. Master's thesis, Universidade do Porto.

Peker, S., Kocyigit, A., and Eren, P. E. (2017). Lrfmp model for customer segmentation in the grocery retail industry: a case study. *Marketing Intelligence & Planning*, 35(4):544–559.

Raggett, D., Le Hors, A., and Jacobs, I. (1998). *HTML 4.0 Specification*. W3C. Accessed: 2023-06-28.

Redman, R. (2020). Fmi: Online grocery sales jumped 300% early in pandemic. Technical report, Food Industry Association. Accessed: 2023-06-23.

Robie, J., Chamberlin, D., Dyck, M., and Snelson, J. (2014). *XML Path Language (XPath) 3.0*. W3C. Accessed: 2023-06-28.

Robie, J. and Texcel Research (1998). *What is the Document Object Model?* W3C. Accessed: 2023-06-28.

Saurkar, A. V., Pathare, K. G., and Gode, S. A. (2018). An overview on web scraping techniques and tools. *International Journal on Future Revolution in Computer Science & Communication Engineering*, 4(4):363–367.

Scrapy Developers (2023). Architecture overview. Accessed: 2023-06-23.

Shepard, R. N. and Arabie, P. (1979). Additive clustering: Representation of similarities as combinations of discrete overlapping properties. *Psychological Review*, 86(2):87.

Shocker, A. D., Bayus, B. L., and Kim, N. (2004). Product complements and substitutes in the real world: The relevance of "other products". *Journal of Marketing*, 68(1):28–40.

Stevens, F. C. (2020). A relação do marketing de relacionamento na lealdade do consumidor online da auchan retail portugal no grande porto. Master's thesis, Universidade Católica Portuguesa.

Think-BIG by SAPO (2022). Kuantokusta: o primeiro comparador de preços português em constante evolução. *SAPO*. Accessed: 2023-06-23.

Tian, H.-p., Tan, Q.-y., Yao, M.-j., and Liu, C.-x. (2021). The value of commitment: Should weaker retailer follow the price of dominate rival? In *2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 1230–1234. IEEE.

Wind, Y. (1978). Issues and advances in segmentation research. *Journal of Marketing Research*, 15:317–337.

World, I. (2022). Supermarkets & grocery stores in portugal. Technical report, Ibis World. Accessed: 2023-06-23.

World Wide Web Consortium (2010). *What is CSS?* W3C. Accessed: 2023-06-28.

Xie, J. (2016). *Automating price matching on e-commerce websites using natural language processing*. PhD thesis, Massey University.

Yang, D. and Thiengburanathum, P. (2020). A comparison of open source web crawlers for e-commerce websites. In *2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, pages 200–205. IEEE.

Yu, H., Litchfield, L., Kernreiter, T., Jolly, S., and Hempstalk, K. (2019). Complementary recommendations: A brief survey. In *2019 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS)*, pages 73–78. IEEE.

Yuen, M. (2022). Digital grocery will be a 243 billion market in the us by 2025: Here are the stats and trends you need to know. Technical report, Insider Intelligence. Accessed: 2023-06-23.

Zhang, M. and Bockstedt, J. (2020). Complements and substitutes in online product recommendations: The differential effects on consumers' willingness to pay. *Information and Management*, 57.