U.PORTO

FACULDADE DE ECONOMIA
FEP UNIVERSIDADE DO PORTO

An Intelligent Recommendation System for Campaigns in the Retail Business

João Manuel Conde Marçal Ferreira Pereira

Dissertation

Master in Data Analytics

Supervised by

Professor Pedro Campos
Professor Bruno Veloso

2023

Acknowledgments

The process of developing this work proved to be more difficult than I had anticipated, and I would like to express my gratitude to my family, friends, and everyone at the office (B2F) who helped me along the way.

In addition, I would like to deeply thank my supervisor, Professor Pedro Campos, and co-supervisor, Professor Bruno Veloso, for their prompt assistance in both technical and report development areas.

Finally, I would like to thank the retail group for providing the necessary information for developing this solution.

Abstract

In this age of abundant information, the retail sector must establish systems for properly evaluating data patterns and trends. These strategies are critical in assisting retailers making well-informed selections about which products and how to include them in promotional campaigns. Recommender systems are intended to provide data-driven insights to merchants, allowing them to streamline the typically arduous processes involved in campaign development. Aside from automating campaign creation, these systems are designed to recommend product combinations that correspond to consumer preferences, making them a great resource for merchants looking to improve and automate marketing initiatives.

As a result, this work focuses on the development of a recommender system to automate the process of creating and simulating marketing campaigns in the retail industry using data mining and machine learning techniques. The system is composed of four different models divided into two flows: the first (Products Discovery) simulates a basket of products to include in the campaign based on prior trends, data mining methodologies, and basket analysis, recommending a balanced selection of popular and profitable products, and the second (Campaign Generation) provides an estimated price for each product and simulates the daily outcome of the campaign through each of the retailer stores, providing insights such as profits, costs, revenues and sale quantities. The system is additionally supplemented by a web app for campaign generation and user interaction with the system, as well as multiple dashboards for analyzing simulations and evaluating output metrics.

Evaluations and results shown that all models performed well, producing satisfactory results in terms of recommended products, estimated prices, and forecasted campaigns. Furthermore, the retailer's ability to personalize both campaign generation and dashboard analysis provides agility and improves decision-making capabilities. As a result, the recommender system's ease of use and precise analytics reduce the time and effort spent on the campaign development process, giving retailers more time to focus on details and specific insights.

Keywords: Recommender System, Retail Sector; Campaigns; Data Mining; Machine Learning; Basket Analysis; Web App; Dashboards

Resumo

Nesta era de abundante informação, o setor de retalho deve estabelecer sistemas para avaliar adequadamente os padrões e tendências dos dados. Estas estratégias são cruciais para ajudar os retalhistas a fazer seleções informadas sobre quais os produtos e como os incluir em campanhas promocionais. Os sistemas de recomendação têm como objetivo fornecer informações baseadas em histórico de vendas aos comerciantes, permitindo-lhes simplificar os processos normalmente árduos envolvidos no desenvolvimento de campanhas. Além de automatizar a criação de campanhas, estes sistemas são projetados para recomendar combinações de produtos que correspondam às preferências dos consumidores, tornando-se um ótimo recurso para comerciantes que procuram melhorar e automatizar iniciativas de marketing.

Como resultado, este trabalho concentra-se no desenvolvimento de um sistema de recomendação para automatizar o processo de criação e simulação de campanhas de marketing na indústria de retalho através do uso de técnicas de data mining e machine learning. O sistema é composto por quatro modelos diferentes divididos em dois fluxos: o primeiro (Descoberta de Produtos) simula um cesto de produtos a incluir na campanha com base em tendências de vendas, metodologias de data mining e basket analysis, recomendando uma seleção equilibrada de produtos populares e lucrativos, e o segundo (Geração de Campanhas) fornece um preço estimado para cada produto e simula o resultado diário da campanha em cada uma das lojas do retalhista, fornecendo informações como lucros, custos, receitas e quantidades vendidas. O sistema é complementado por um aplicativo web para a geração de campanhas e interação do utilizador com o sistema, bem como vários painéis analíticos para analisar simulações e analisar métricas de apreciação.

As avaliações e resultados mostram que todos os modelos tiveram um bom desempenho, produzindo resultados satisfatórios em termos de produtos recomendados, preços estimados e campanhas simuladas. Além disso, a capacidade de o retalhista personalizar tanto a geração de campanhas quanto a análise dos painéis analíticos proporciona agilidade e melhora as capacidades de tomada de decisão. Como resultado, a facilidade de uso do sistema de recomendação e as análises que fornece reduzem o tempo e esforço gastos no processo de desenvolvimento de campanhas, permitindo aos retalhistas concentrarem-se nos detalhes.

Palavras-chave: Sistema de Recomendação, Setor do Retalho; Campanhas; Data Mining; Machine Learning; Basket Analysis; Aplicativo Web; Painéis Analíticos

List of Abbreviations and Acronyms

DA	Data Analytics
CRM	Customer Relationship Management
BA	Basket Analysis
ML	Machine Learning
BI	Business Intelligence
AI	Artificial Intelligence
ROI	Return on Investment
ML	Hyperparameter tuning
CV	Cross-Validation
ARIMA	Auto-Regressive Integrated Moving Average
LSTM	Long Short Term Memory
VAR	Vector Autoregression
DDL	Data Definition Language
ETL	Extract, Transform and Load
DW	Data Warehouse
SA	Staging Area
UI	User Interface
SQL	Structured Query Language
PBI	Microsoft Power BI
FP-Growth	Frequent Pattern Growth
ROI	Return on Investment
OHE	One Hot Encoding
RMSE	Root Mean Squared Error
R2	R-Squared
MAE	Mean Absolute Error
SHAP	Shapley Additive Explanations
KPI	Key Performance Indicators

Contents

1. Introduction 1

1.1. Motivation 1

1.2. Retail Group 2

1.3. Problem Definition 2

1.4. Contents 3

2. Literature Review 5

2.1. Data Analytics in Retail 5

2.2. Campaigns 6

2.3. Artificial Intelligence approaches 7

2.4. Forecasting Campaigns 8

2.4.1. Regression models 9

2.4.2. Time series 10

2.5. Market basket analysis 11

2.6. Contributed Work 12

3. An Intelligent Recommender System for Retail Campaigns - Data and Models 15

3.1. Data source 15

3.2. Data structure and content 16

3.3. Models Flow 17

3.4. Products Discovery flow 18

3.4.1. Best Products model 19

3.4.2. Basket Analysis model 25

3.5. Campaign Generation flow 28

3.5.1. Data Rearrangement 29

3.5.2. Price Estimation model 31

3.5.3. Forecasting model 35

4. Evaluation and Results 41

4.1. Models Evaluation 41

4.1.1. Evaluation Protocol 41

4.1.2. Products Discovery Evaluation 42

4.1.3. Campaign Generation Evaluation 45

4.2. Reporting 52

5. Conclusions & Future Work 57

References 60

Annex 64

List of Figures

Figure 1 - Overall architecture of the recommender system 18

Figure 2 - Product category analytics 20

Figure 3 - Weights per top categories 21

Figure 4 - Products per category display 22

Figure 5 - Best Products filtering example 24

Figure 6 - Products Discovery architecture 25

Figure 7 - One Hot Encoding representation 26

Figure 8 - Basket Analysis output filtering 27

Figure 9 - Basket Analysis architecture 28

Figure 10 - Snapshot of materialized aggregated data from sales 30

Figure 11 - Price Estimation C Day Difference 32

Figure 12 - Average C Day Difference applied to a specific product 34

Figure 13 - Price Estimation architecture 35

Figure 14 - Forecasting model architecture 38

Figure 15 - Snapshot of the Products Discovery evaluation process 44

Figure 16 - Price Estimation model's evaluation (dashboard) 46

Figure 17 - SHAP analysis 49

Figure 18 - Forecasting model's evaluation (dashboard) 50

Figure 19 - Cockpit dashboard overview 53

Figure 20 - Scatter chart overview 54

Figure 21 - Cross filtering 55

List of Tables

Table 1 - Main research papers overview & comparison 14

Table 2 - Data source tables required 16

Table 3 - Definition of labels present in the best categories analytics 20

Table 4 - Definition of labels present in the best categories weights 21

Table 5 - Definition of labels present in the product ranking analysis 23

Table 6 - Definition of labels present in the basket analysis input dataset 26

Table 7 - Definition of labels present in the aggregated data from sales 30

Table 8 - Content of Price Estimation C Day Difference analysis 33

Table 9 - Definition of labels present in the data for training the regression model 37

Table 10 - System's flow 39

Table 11 - Definition of labels present in the Products Discovery evaluation process 44

Table 12 – Forecasting model cross-validation results 47

Table 13 - Forecasting model evaluation measures 48

1. Introduction

This chapter will summarize the project's motivation, the retail group which the study takes place, the problem definition, and the general substance of the thesis. The present study uses AI and data analytics to demonstrate how the retail business can be leveraged to generate more automated and successful marketing campaigns.

1.1. Motivation

The retail market is fiercely competitive, with companies continuously looking for innovative methods to boost sales and gain a competitive advantage. Effective marketing campaigns are one of the most used methods for retailers to achieve this (Kallier Tar & A Wiid, 2021). On the other hand, creating appellative marketing initiatives may be complex since it necessitates a thorough grasp of customer behaviour and industry trends (Kelley, 2020).

Data Analytics (DA) and Artificial Intelligence (AI) have seen significant growth in the retail business in recent years, and these technologies have the potential to revolutionize how merchants create and execute marketing activities. One of the primary advantages of employing AI in retail is its capacity to evaluate enormous volumes of data swiftly and efficiently. This enables merchants to observe patterns and trends that would be difficult or impossible to spot manually (Cao, 2021). An AI-based recommendation system, for example, may assess client purchase history, patterns, and demographic data to provide customized marketing recommendations, which can lead to improved conversion rates and sales (Kelley, 2020). Basket analysis and price forecasting may provide valuable insights into customer behaviour and market trends, helping firms to create more targeted and profitable campaigns.

Not only is DA useful in marketing and assessing business indicators, but it is also helpful in logistics, enabling the effective storage, analysis, and comprehension of vast volumes of data created by integrating online and physical shop orders. As (Lalou et al., 2020) stated in their research, DA and AI can help third-party logistics operators who oversee receiving, storing, and delivering items to retail outlets with demand forecasting and decision-making. These operators may enhance inventory management and human resource planning by applying data analytics, resulting in more efficient and sustainable supply chain management. This is especially critical in today's fast-changing economy when demand uncertainty is high, and supply chains must deal with demand fluctuations.

This study aims to look at how data mining, analytics and AI may improve the efficacy of marketing efforts in the retail business, by lowering retailer efforts and increase sales. The research will focus on using data analytics, heuristics, basket analysis and forecasting to establish an intelligent recommendation system for retail marketing.

1.2. Retail Group

The current research and the previously described analyses will be created utilizing real data from a genuine retail business in Portugal.

The Portuguese group has been in the retail industry for more than 60 years and is a market leader in wholesale food and non-food goods. The firm has evolved and brought consumers throughout the years owing to its appealing quality-price ratio, selling thousands of products daily in different stores. The retail company even has its own branded items, garnered customers' trust and increased their devotion to the retail group.

Three hypermarket consumer purchase data from this Portuguese retailer will be used in the present work, in which an average of 28 million products or packs of products are sold each year to around 400 loyal clients.

1.3. Problem Definition

With the introduction of technology and the explosion of data availability, the retail business has seen tremendous changes in recent years. This has resulted in an excess of data for merchants to examine and make sense of, but the sheer volume of data can make it difficult for managers to operate their firms successfully. Understanding client habits, which is critical for making informed decisions and optimizing business operations is one of the retail managers' most demanding challenges (Venuturumilli et al., 2016).

Retail managers must rely on human data analysis, which is time-consuming and prone to mistakes, in the absence of innovative technologies such as AI and data analytics. Furthermore, manual data analysis can only give limited insight into client behaviour and purchase trends. As a result, managers may need more information to make judgments.

The application of AI and DA in the retail business, on the other hand, can change how managers understand and make choices about their consumers (Lekhwar et al., 2019). These tools help managers quickly and correctly evaluate vast volumes of data, giving them a thorough insight into client behaviour and purchase trends. This can help them better judge which things to invest in, how to price them, and how to sell them.

Furthermore, these technologies can assist retail managers in identifying patterns and trends that would be difficult or impossible to find through manual examination. This can assist them in identifying new opportunities and developing strategies for capitalizing on them. Managers, for example, may evaluate which products are most popular, when times of day or week are busiest, and which promotions are most successful by evaluating client purchase data (Kallier, 2017).

In this manner, the goal of this work is to develop an automated campaign generator to assess retailers on the difficult process of creating a campaign, by recommending the most efficient and profitable products to include and their prices, and then forecasting the daily results.

Initially, the user informs the recommender of how many products are meant to be included in the campaign, and the model then suggests products that should be included based on the outputs of a product rule-based model and a basket analysis model. In addition, the system calculates a projected price for each product based on a variety of parameters, including seasonality, supplier purchasing costs and product price history in previous campaigns. Finally, the system generates an output consisting of daily forecasts of each product's selling volumes across all stores. This will provide the retailer (as a system user) with an overview of the campaign's profitability, costs, revenues, and quantities sold, enabling for more effective decision making.

The outputs of the recommender system should not be followed blindly without careful examination and investigation. As the name implies, this solution is intended to recommend a more efficient basket of products to the retailer, exhibiting the results of each product during the campaign duration, and the retailer may then choose to use the entire output as a next campaign, or even select only a subset of products based on their success.

1.4. Contents

This work aims to create an intelligent recommendation system for retail campaigns that will employ data analytics, basket analysis and campaign results forecasting to automate the process of carrying out campaigns and marketing activities. The system will assess customer behaviour and market trends and generate recommendations for effective marketing.

In Chapter 2, the literature review will look at existing research on data analytics and AI in the retail industry, with a focus on automated campaigns using price forecasting systems and market basket analysis, as well as an overview of specific forecasting approaches such as regression models and time series. Following that, Chapter 3 introduces the data that will be used as well as its structure, and covers all the models produced in the solution, as well as how they communicate with one another to generate the recommender's outputs. In Chapter 4, multiple methodologies are employed to analyze these models, and the overall results are displayed on analytic dashboards. Finally, Chapter 5 will explain the work's final conclusions, as well as some future improvements to enhance the system's capabilities and precision.

2. Literature Review

This review aims to assess the present level of research in the retail industry on data analytics, campaigns, and artificial intelligence. This review will apply these principles to campaign results forecasting and market basket analysis.

Section 2.1 will investigate the application of data analytics in the retail sector, emphasizing its potential for improving decision-making and consumer experience. The function of campaigns in the retail business, including targeted marketing and tailored promotions, will be examined in Section 2.2.

Section 2.3 will dive into artificial intelligence applications in retail, including sales forecasting and market basket analysis. Sales forecasting entails using machine learning algorithms to project future sales based on past data, whereas market basket research entails analyzing consumer purchase data to find potential cross-selling opportunities.

Overall, this literature review aims to provide a complete overview of the present state of research in these domains and to highlight prospective topics for future research in the retail industry's application of data analytics, marketing, and artificial intelligence.

2.1. Data Analytics in Retail

Data analytics (DA) has recently become a more crucial tool for retailers. They can discover important client base segments using these methods and then modify their marketing and sales strategies better to suit the requirements and preferences of these segments. Providing individualized and pertinent items and services can aid companies in boosting client loyalty and revenue (Prasad & Venkatesham, 2021).

DA may give organizations a competitive edge in the fiercely competitive retail sector by assisting them in identifying new opportunities and streamlining existing operations (Lekhwar et al., 2019).

Retailers can employ these techniques, for instance, to segment their clientele, personalize marketing initiatives and spot cross-selling opportunities. Retailers can also enhance the customer experience by evaluating data on client interactions and purchase histories, for instance, by making recommendations or providing individualized discounts (Ricci et al., 2015).

Furthermore, the Customer Relationship Management (CRM) area is crucial for data analytics in the retail sector. Retailers can better understand their consumers' requirements, tastes, and behaviour by evaluating customer data, and they can then utilize this knowledge to focus their marketing and sales efforts. Enhancing client satisfaction and loyalty can increase sales and profitability (Lekhwar et al., 2019).

Additionally, by incorporating intelligence into the pricing calculation, DA can enhance and provide relevant outcomes in the product selling value. Pricing optimization uses data and analytics to determine an item's or service's best price based on variables including demand, cost, competition, and other outside considerations. Online retailers can improve sales by setting prices that appeal to customers through price optimization, and they can also maximize profits by ensuring that prices are set at a level that allows for an acceptable margin (Ferreira et al., 2016).

2.2. Campaigns

Campaigns are promotional efforts performed by retailers to attract customers and enhance sales (Yee et al., 2010). These promotions might be bargains, loyalty programs, or one-of-a-kind events. In the digital era, merchants increasingly use email marketing, social media, and other online methods to engage with customers and spread the word about their initiatives (Kallier, 2017).

The capacity to segment and target customer groups is a vital component of retail marketing operations. Data analysis and CRM systems may assist in gaining a comprehensive insight into consumer demographics, interests, and behaviour (Lekhwar et al., 2019).

However, successful retail campaigns must also be well-planned and implemented (Yee et al., 2010). This includes developing specific goals and objectives, establishing the amount of money and resources required, and monitoring and evaluating the campaign's outcomes. Focusing on analytics, often known as data-driven decision-making, might be incredibly advantageous (Davenport & Harris, 2007).

Basket Analysis (BA) (which will be approached in topic 2.5) and predictive models for categorization are also widely utilized to automate the campaign generation process. In the paper "Applying Instant Business Intelligence in Marketing Campaign Automation" by (Yee et al., 2010), an automated Business Intelligence system was developed to assist retailers in defining the most effective campaigns, increase sales and provide a more pleasant shopping experience for users.

In this solution, a BA model was used to understand the most popular things purchased jointly, so that clients may consider purchasing additional products that were secondary to them at the time. In addition, Machine Learning (ML) models such as Naive Bayes and

Decision trees were utilized in a categorization prediction model to aid merchants in client segmentation, resulting in an automated direct marketing solution (Yee et al., 2010).

Moreover, information technology helps firms meet customer trends in competitive business contexts (Lekhwar et al., 2019). Organizations can only prosper if they can predict client trends properly since client behaviours and expectations are intimately connected to the satisfaction percentage of customer-oriented presentations. However, it is critical to maintain organization, keep data updated, and build solid solutions that promote scalability and trust in the results. Organizations must keep sales data available and update it regularly to accurately evaluate clients' purchasing patterns.

To recap, retail campaigns are an important tool for attracting new consumers and retaining existing ones, and they are effective when a mix of targeted marketing, skilled planning and execution, and data-driven decision-making is employed.

2.3. Artificial Intelligence approaches

Artificial intelligence (AI) is a fast-evolving area with the potential to dramatically affect a wide range of businesses and facets of our everyday life. AI can give businesses a competitive advantage by allowing them to make better and faster choices, automate jobs, and improve consumer experiences (Davenport, 2018).

According to artificial intelligence scientist Stuart Russell, AI can radically transform human civilization’s character by revolutionizing how we live and work (Russell, 2019). The author argues in his book "Human Compatible: Artificial Intelligence and the Problem of Control" that the key to ensuring that AI evolves in a way that benefits humanity is to connect it with human values and aims.

Furthermore, because the positive impact of applying these solutions is enormous, they may be actively utilized in the business sector. In his book "How to Put the Artificial Intelligence Revolution to Work," Thomas Davenport claims that AI solutions improve decision-making by analyzing large amounts of data and providing insights that can help businesses make better decisions. The automation of various tasks that are repetitive, time-consuming, or require a high level of accuracy; improves customer experiences by enhancing client interactions and gives enhanced competitiveness by obtaining an advantage over competitors through the ability to make better and faster choices, automate activities, and improve customer experiences. (Davenport, 2018).

AI, like other industries, has transformed the retail industry. The growth of e-commerce platforms, the rising demand for client profiling and technological advancements forced the automation and computerization of various formerly manual procedures. The response to this revolution was centered on extensive data collection and analysis of customer actions and trends (Hunt & Rolf, 2022).

Despite the benefits, some retailers are still determining these new approaches. According to McKinsey (Chui et al., 2017), roughly 42 per cent of retail owners admitted being unsure about the benefits of AI, particularly in terms of business cases and return on investment (ROI) and so, it is critical to comprehend how AI solutions might benefit this industry.

AI solutions in the retail sector can improve consumer interactions at many stages of the customer experience, such as search, suggestions, and after-sales assistance. They can also be used to increase the efficiency of real and virtual store administration and optimize merchandising. Moreover, by enhancing demand forecasting and automating ordering and warehouse procedures, they can be used to optimize supply chain management (Cao, 2021). In addition, AI can be utilized in marketing management to dynamically modify prices and make smart marketing decisions, increasing sales and profits.

2.4. Forecasting Campaigns

Estimating campaign results is critical in assisting organizations in making educated decisions and attaining their objectives. It can help organizations set realistic expectations and decide whether to change their strategies (Ahmad et al., 2016). Campaign outcomes can vary widely amongst firms, and companies may be willing to commit large sums of money to achieve their goals. In this case, prediction of results can help both businesses and their customers. Analyzing past data from the same sector and local competitors can produce realistic estimates of results. This can be performed using data mining techniques, which may include gathering information on a variety of aspects influencing results, such as target audience, marketing channels, and product/service quality. Businesses may utilize the expected results to alter their tactics and make educated decisions (Khaydukova et al., 2015).

In terms of the technological principles underlying result forecasting, regression models or time series approaches are commonly used, as discussed in the sections that follow.

2.4.1. Regression models

It is prudent to introduce the notion of Machine Learning (ML) before digging into regression models. Machine learning aims to extract knowledge from data (Rebala et al., 2019). ML is also known as predictive analytics or statistical learning, and it is a research topic that combines statistics, artificial intelligence, and computer science (Müller & Guido, 2017b).

There are two main learning models in ML: supervised and unsupervised. The supervised models focus on predicting an outcome based on data entries with answers, whereas the unsupervised models rely on patterns and similarities rather than answers (Rebala et al., 2019). Regression models are supervised and forecasted based on existing data (with responses), with the addition of working with numerical features. Classification models, on the other hand, despite also being supervised, are used to forecast categorical data (Janiesch et al., 2021). As a result, regression models can be used to forecast prices since they predict numerical values based on existing data and features.

However, selecting the right model to produce forecasts is a critical step during the development process. Various existing models for each type of ML technique differ greatly; therefore, it is critical to carefully select the proper model that will achieve the best performance for the input data, features, and existing constraints (Rebala et al., 2019).

Techniques such as cross-validation (CV) and hyperparameter tuning are often employed during this phase to determine the optimum method and settings for the input dataset, as described by (Lucas et al., 2020) in his study. CV evaluates an ML model's performance by training it on a section of the dataset, testing it on a different portion, and then repeating this procedure numerous times with different splits of the data to obtain an average performance. This can be done with various input models to determine which is more likely to produce better results (Ziegel, 2003). On the other hand, hyperparameter tuning is the process of adjusting a machine learning model's hyperparameters (or configurations) to improve its performance on a given dataset (Atkinson et al., 2020). Prior to training, hyperparameters are configuration options for a machine learning model. They are not learned from the training data and are frequently set using heuristics or a process known as hyperparameter optimization. Different machine learning models have different sets of hyperparameters and selecting the proper values for these hyperparameters can substantially impact the model's performance (Geron, 2017).

One noteworthy use of regression models in forecasting comes from (Ferreira et al., 2016), who solved a multi-product pricing optimization model with reference price impacts using regression trees with bagging and an efficient method built by the researchers. The researchers conducted a field experiment to test the price decision support tool and observed that it resulted in a 9.7% gain in income with a 90% confidence interval, emphasizing how these models may aid in price forecasting and, as a result, enhanced revenue.

2.4.2. Time series

Time series analysis is defined by (Rob J Hyndman & George, 2014) as a set of techniques used to evaluate and forecast data gathered over time. These methodologies indicate that data patterns and trends can be used to estimate future values. This methodology employs decomposition, smoothing, and exponential smoothing to reduce noise and discover underlying patterns in the data. It also includes more sophisticated approaches, such as autoregressive integrated moving average (ARIMA) models, which may be used to provide more accurate forecasts (Brockwell & Davis, 2002).

Even though these models are primarily used to estimate numerical values based on historical data, efficiently assessing trends, cycles, and seasonal components, the ARIMA model is often utilized in forecasting stock prices or economic indicators (Rhanoui et al., 2019).

In (Chouksey, 2018) publication, one application of this approach was detailed for forecasting stock prices. In his study, three time series models were employed to make predictions: ARIMA, PROPHET, and KERAS with LSTM (Long Short-Term Memory) to compare their performance. The results showed that, while the LSTM model looked to be more capable of predicting stock values in the near term, the PROPHET and ARIMA models provided a more consistent output.

Another application of time series models for forecasting product prices in the retail sector comes from (Ahmad et al., 2016), where the researcher independently utilized a basic Autoregression model and a Vector Autoregression (VAR) model with just pricing data from local competitors as input, as well as wholesale pricing regions and delayed wholesale prices (prices from the previous day of model training) as exogenous input. The VAR model is a statistical model used to evaluate multivariate time series data. It varies from the ARIMA model in that VAR represents the linear interaction of several variables with each other, whereas ARIMA models the reliance of a single variable on its previous values and past errors. Overall, his work (Ahmad et al., 2016) stated that the results revealed that incorporating delayed wholesale pricing resulted in a substantially larger improvement in prediction accuracy, ranging from almost 12% to 40% when compared to not using wholesale prices. The total improvement obtained by integrating rivals and delayed wholesale prices ranged between 5 percent and 50 percent.

2.5. Market basket analysis

Understanding client purchasing patterns, namely, what products are typically purchased together, is an intriguing yet critical component of data analytics in the retail sector. Knowing this may help shop owners determine which product packs should appear together in promotions or campaigns or even aid in reorganizing actual stores (Grau, 2017).

The Apriori algorithm is the most often used method for mining association rules from a transactional database that meet the user-specified minimum support and confidence levels. This strategy is widely utilized in numerous areas, including banking, telecommunications, marketing, commerce, and web analysis, and several adaptations have been developed in recent years (Chen et al., 2005). The Frequent Pattern Growth algorithm (FP-Growth) is another extensively used approach for identifying itemsets in large datasets without using the Apriori algorithm's generating and testing processes. This approach lowers time loss by first compressing the database into a tree structure known as the FP tree, which includes the itemsets' association information. The database is then partitioned into conditional data structures, each of which relates to a frequent object, and these databases are mined independently. The method works by continually reducing the difficulty of discovering huge common itemset models into the problem of locating minors and merging suffixes. It gives strong selectivity by employing somewhat repeating items as a suffix, lowering search costs dramatically (Sagin & Ayvaz, 2018).

Furthermore, finding association rules is one important step in developing market basket analysis solutions. Association rules are a data mining technique that is used to detect patterns or links between distinct objects in a dataset. These criteria can be used to detect links between different types of information or to identify goods that are frequently purchased together (Sagin & Ayvaz, 2018). One other important concept for understanding this process are the frequent itemsets, which are sets of items that frequently appear together. The Apriori and FP-Growth algorithms are commonly used to identify frequent itemsets (Grau, 2017).

Nevertheless, it is crucial to comprehend associations and frequent itemsets, and to evaluate them using some criteria. As explained in (Pradana et al., 2022) research, there are some measures that are also used to compare and analyze the strength of the associations between itemsets, such as:

- **Support**: quantifies the frequency with which a given itemset appears in transactional entries in the database (a group of items). It is obtained by dividing the number of transactions which contain the itemset by the total number of transactions.

$$Supp(A \rightarrow B) = \frac{Transactions\ containing\ A\ and\ B}{Total\ of\ transactions} \qquad [2.1]$$

- **Confidence**: is a measure of an association rule's correctness defined as the ratio of transactional records containing both itemset A and itemset B to transactional records containing just A. It is used to represent the possibility that a consumer who buys item set A will also buy item set B.

$$Conf(A \rightarrow B) = \frac{Transactions\ containing\ A\ and\ B}{Transactions\ containing\ A} \qquad [2.2]$$

- **Lift**: is an indicator of the strength of association between two itemsets derived as the ratio of observed support for itemsets A and B to predicted support if A and B were independent. It is used to show how probable itemset B is to be purchased when itemset A is purchased.

$$Lift(A \rightarrow B) = \frac{Conf(A \rightarrow B)}{Expected\ Confidence}, \qquad [2.3]$$

$$in\ which\ Expected\ Confidence = Supp(B)$$

Despite this, there are other more specific indicators such as leverage, which measures the influence of B on the occurrence of A, and conviction which represents the degree of dependency of the association rule by measure of the degree of unexpectedness of an association rule, defined as the ratio of the expected confidence of the opposite rule to the confidence of the original rule (Han et al., 2012).

2.6. Contributed Work

This section describes the primary publications and studies that formed the basis for the current research. The key objectives, models and technologies, data, and assessment metrics of each of these investigations are summarized in Table *1*. The purpose of this section is to provide an overview of the present state-of-the-art used, as well as to highlight the strengths and weaknesses of previous research efforts.

Paper Name/Reference	Task/Objective	Models/Technologies Used	Data	Evaluation Measures
Prasad, J. P., & Venkatesham, T. (2021)	Understanding the impact of big data analytics on the retail sector	Big data	Retail customer data	Effectiveness of data-driven decision making in the retail sector
Hunt & Rolf (2022)	Discuss the implications of AI and automation in retail for consumers, retail organizations and workers	AI, Automation and Robotics	Results from 5 focus groups with European trade unionists working in retail	Focus group conversations and transcript analysis yielded insights that were compared to public data, academic research, and media stories
Cao (2021)	Investigate the benefits of AI for retailers	Grounded theory multiple-case analysis	54 representative retailers' adoptions and implementations of AI between 2008 and 2018	Data- and solution-centric perspectives, as well as the concept of value creation logics
Ahmad et al. (2016)	Predict the retail prices of products at every outlet in each city	Four vector autoregression models	Historical retail pricing of the goods at a target outlet and competitor outlets, as well as the product's anticipated wholesale price	Outperforms a simple autoregression approach in experiments carried out using data obtained from outlets in five North American cities
Khaydukova et al. (2015)	To investigate the effect of total product quality on black tea retail pricing	Potentiometric electronic tongue and Partial Least	Black tea samples purchased in retail stores in	Mean relative errors of about 15% for Spain’s tea bags and 25% for loose-packed tea from Russia for the prediction of retail price

		Squares (PLS) regression	Spain and Russia	using PLS regression models
AKGÜL et al. (2018)	Examining the concept of business intelligence and its infrastructure for benefit in various sectors, with emphasis on the retail sector	Business Intelligence	Enterprise data	Emphasized the benefits of business intelligence applications and its contributions to the applied sectors
Grau, G. R. (2017)	Analyze the market basket of clients in a Spanish retail organization to discover things that are purchased together	BigML, machine learning techniques, analytics tools	Customer purchase data from a retail company in Spain	The project's performance is measured by the results presented at the conclusion, which should reveal which things are purchased together at the store
Ferreira, K. J., Lee, B. H. A., & Simchi-Levi, D. (2016)	Optimizing pricing decisions for an online retailer	Machine learning techniques and a multiproduct price optimization algorithm	Historical lost sales and demand of new products	Increase in revenue by approximately 9.7% with a 90% confidence interval of [2.3%, 17.8%]
Kallier, S. M. (2017)	To determine the influence of RTM campaigns of retailers on consumer purchase behavior in South Africa	Personalized real-time marketing, multi-channel engagement	Data was collected from consumers of retail stores	Quantitative approach, influence on consumer purchase behavior

Table 1 - Main research papers overview & comparison

3. An Intelligent Recommender System for Retail Campaigns - Data and Models

This chapter provides an overview of the data that was utilized in the study, as well as the models that make part of the Recommender System for Campaigns in the Retail Business.

In addition to providing the data source, the information regarding the content of the dataset will also be explained, serving as the basis for the remainder of the study, giving the background knowledge required for understanding and interpreting the results.

Also, in this chapter provides an overview of the recommender system models and workflow, which is composed by three major components: Models Flow, Products Discovery, and Campaign Generation. To begin, the Models Flow section will approach at the solution's architecture, outlining how the many components connect with one another to achieve a single output: the automated campaign.

The Products Discovery component identifies the most profitable product categories as well as the association between commonly purchased products. Then, the Campaign Generation component will cover how the estimated prices of products and the daily campaign results predictions based on previous data are used to generate campaign results. The User Interface, which will be also mentioned, enables campaign settings to be configured and recommendations to be viewed (section B of the Annex).

The tools used to develop these models as well as the entire architecture of the system are described in the section A of the Annex.

Finally, this chapter will go over the specifics of each component and how they interact to form an effective recommender system for retail campaigns.

3.1. Data source

As previously stated in 1.2, the data for this study came from a Portuguese retailer. As is customary in the retail sector, most of the information that enters a company's database originates from sales to the end customer, which when done on the retailer's premises are often done via Point of Sales (POS) equipment. With the evolution of technology, POS devices became more autonomous and able to acquire more information from each sale. These devices, also known as cash registers, oversee reading all the things that the consumer wishes to buy, creating invoices, and then entering all the information into a database. This process occurs every time a client makes a purchase, independently of the store, since all the information is gathered onto a central database.

In the current case, the retailer saves the sales data in a Structured Query Language (SQL) database, which exists in their local servers. This database contains information about the whole business of the retailer, including product inventory, stocks, sales, client information, data regarding purchases to suppliers, and even workers history, having data from 2017 to the middle of 2022.

3.2. Data structure and content

Because the database's real structure is composed of several tables referring to various sectors of the business and just a small quantity of data will be required for this study, procedures to organize and structure the information needed the data will be required. Data Definition Language (DDL) operations will be established to do this, by creating visualizations that will strictly contain the information needed. DDL refers to actions that change, delete, or generate metadata rather than altering existing data (Amornchewin, 2018).

Table *2* shows which tables will be used as a foundation for the DDL operations that will be pursued further.

Table name	Description
DIM_Produtos	Contains information of each existing product and its category.
DIM_Campanhas	Includes structural information of previous campaigns, such as temporal insights (when it started and ended) and the type of campaign.
FACT_Campanhas	Contains which products were in each campaign.
FACT_Encomendas	Has the data regarding purchases from the retailer to suppliers, including the products, quantity, and purchase price.
FACT_Vendas	Possesses all sales data from the retailer, including products sold, quantities, and closing price of sale.
DIM_Loja	Contains structural data from the retailer stores.

Table 2 - Data source tables required

As seen, the table names have a prefix of FACT or DIM, which may aid in determining which type of table it is, either by knowing if the table contains facts (FACT) or dimensions (DIM).

According to one of the most widely read books in this field, “The Data Warehouse Toolkit” by (Kimball & Ross, 2013), in Business Intelligence (BI) workflows, raw data enters unprocessed and unstructured, and during the Extract, Transform, and Load (ETL) process, all information is treated and organized before being allocated to a Data Warehouse (DW). Initially, in the Extract (E) phase, data is extracted from the source and placed in a Staging Area (SA) to be worked on. The information will next be rearranged and treated according to the business rules in the Transform (T) step. Finally, the transformed data is placed in a DW during the Transform (T) phase. Tables in the DW are pre-defined as facts (FACT) or dimension (DIM), depending on whether they contain numerical data (such as sales information, inventories, number of people, etc.) or categorical information (such as product details or workers information for instance), respectively.

The retailer's database is the output of a BI workflow, and so, it is defined as a DW, guaranteeing quality and organization in the data.

3.3. Models Flow

A campaign generator system requires several pieces of information and computations to produce an output, which should include the following key components: the products to be included in the campaign, an approximate price for each product, how many quantities of each product will be sold on each day of the campaign, and finally, the total profit that this campaign will have.

To achieve all these aspects, several analyses and processes must be carried out during the campaign creation process, since factors such as the real cost of each product and the amount of stock on hand may visibly impact the campaign result. There are also user inputs that are critical to the campaign's design, such as how many days it should last or how many products it should feature. As a result, these flows, which handle numerous distinct components in campaign development, must execute quickly and interact with one another.

Therefore, the solution, which is created and staged in Python, includes the models required to achieve the simulated outcomes, which are: Best Products, Basket Analysis, Price Estimation, and Forecasting Model, in that sequence. Each of them runs in flow and requires some type of user input which is read from the system's UI (described in the section B of the Annex), whether it's the number of products to explore, the price estimation behavior, or the campaign duration. Because of their similarities and roles in the recommender system's architecture, these models are grouped into two flows: the Products Discovery flow, which is composed of the Best Products and Basket analysis models, wherein the product basket is generated, and the Campaign Generation flow which evolves the Price Estimation and

Forecasting models, where the previously discovered basket of products is associated with prices, and then the actual campaign is forecasted.

Furthermore, all the models interact in some manner with the primary SQL Server database, providing significant information about existing items, costs, margins, stocks, and sales. All this information is necessary to compute the outputs of each model.

Finally, the produced campaign details, including daily anticipated results for each product, must be stored in the database so that they may be reviewed and compared to the actual campaign results as they occur.

The diagram below depicts how each model communicates with each other, with the UI, and with the database in the appropriate phase sequence.

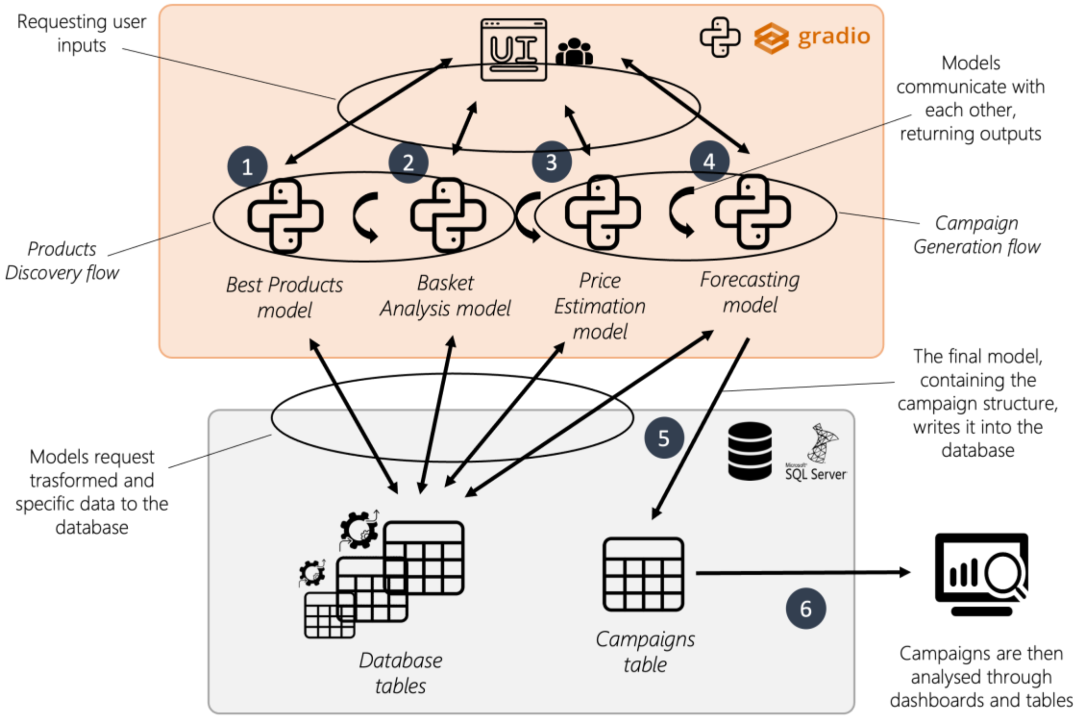

Requesting user inputs
UI
gradio
Models communicate with each other, returning outputs
1
2
3
4
Products Discovery flow
Campaign Generation flow
Best Products model
Basket Analysis model
Price Estimation model
Forecasting model
The final model, containing the campaign structure, writes it into the database
Models request trasformed and specific data to the database
5
SQL Server
Database tables
Campaigns table
6
Campaigns are then analysed through dashboards and tables

Figure 1 - Overall architecture of the recommender system

3.4. Products Discovery flow

One of the most crucial parts of developing a retail campaign is to define its content. Carefully choosing products, having a diverse basket of possibilities across categories, and selecting the ones that are more profitable are all important considerations.

Retailers may boost their sales and revenue by identifying the greatest products. Furthermore, having a diverse product basket with multiple categories can assist retailers in attracting a diverse range of customers. Also, understanding which products are the most profitable can assist merchants in optimizing their marketing strategies and promotions to maximize the Return on Investment (ROI).

The Products Discovery flow, accountable for generating the campaign basket, encompasses the Best Products and the Basket Analysis model. The first section's logic is entirely computed using SQL Server and selects an initial set of products to include in the campaign's basket. This first batch of products was picked with the goal of raising client interest and revenues for the merchant in mind.

The Basket Analysis section, which is executed using Python, is then supposed to enrich the basket with products that are more likely to be sold, while considering the already selected set of products by the Best Products section.

In the parts that follow, the importance of product discovery and how it might affect a retailer's overall success will be covered, as well as the path for identifying the best products and assessing the basket.

3.4.1. Best Products model

The Best Products for a retailer to include in a campaign are those that will improve sales, consumer interest, and profitability. So, when deciding which products to include, several factors should be considered, including seasonality and customer trends, actual stock in warehouses, product popularity in previous campaigns, and profits (Kallier Tar & A Wiid, 2021). Profit is one of the most essential topics, which may be defined as (for product *x*):

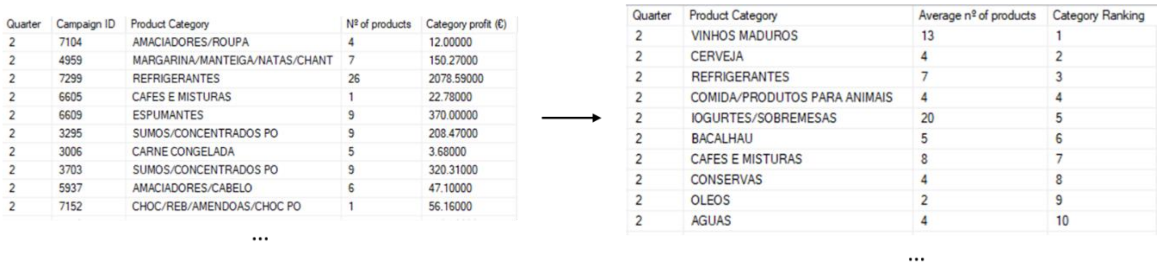
$$Profit\ (x) = (Unit\ price\ (x) - Unit\ \mathrm{cost}\ (x))\ \times\ Sale\ quantity\ (x)\quad [3.1]$$

Furthermore, focusing solely on the most profitable products and their trends may not be the best strategy. As previously mentioned, having a diverse basket is an important consideration, and to do that, the product category should also be investigated.

To do so, the first step is to research previous campaigns, analysing the products included, their respective categories, and the success of each product in such campaigns.

However, the success of some products may vary seasonally, and this factor has a greater impact in the retail industry.

Consuming cod fish, for example, is more likely to occur during the winter in Portugal, although watermelon may be more popular during the summer. To deal with this, a seasonal component should be added to the analysis.



Quarter	Campaign ID	Product Category	Nº of products	Category profit (€)
2	7104	AMACIADORES/ROUPA	4	12.00000
2	4959	MARGARINA/MANTEIGA/NATAS/CHANT	7	150.27000
2	7299	REFRIGERANTES	26	2078.59000
2	6605	CAFES E MISTURAS	1	22.78000
2	6609	ESPUMANTES	9	370.00000
2	3295	SUMOS/CONCENTRADOS PO	9	208.47000
2	3006	CARNE CONGELADA	5	3.68000
2	3703	SUMOS/CONCENTRADOS PO	9	320.31000
2	5937	AMACIADORES/CABELO	6	47.10000
2	7152	CHOC/REB/AMENDOAS/CHOC PO	1	56.16000

...

Quarter	Product Category	Average nº of products	Category Ranking
2	VINHOS MADUROS	13	1
2	CERVEJA	4	2
2	REFRIGERANTES	7	3
2	COMIDA/PRODUTOS PARA ANIMAIS	4	4
2	IOGURTES/SOBREMESAS	20	5
2	BACALHAU	5	6
2	CAFES E MISTURAS	8	7
2	CONSERVAS	4	8
2	OLEOS	2	9
2	AGUAS	4	10

...

Figure 2 - Product category analytics

As shown in the Figure *2*, the first analysis (table on the left) shows the number of goods in each category in each campaign, as well as the quarter in which the campaign happened, and the seasonal component is thus ensured.

The output of the above figure is composed by the following labels:

Label	Definition	Data Type
	Table on the left	
Campaign ID	Unique identifier of each campaign that ever happened.	Integer
Quarter	Quarter of the year in which the campaign happened.	Integer
Product Category	An aggregator of products based on their use and type.	String
Nº of products	Number of products in the respective category and campaign.	Integer
Category profit (€)	Represents the profit achieved by the category, in the respective campaign.	Float (currency)
Purpose	Obtain the number of products and profit in every existing campaign for each existing category.	
	Table on the right	
Average nº of products	Average number of products in the respective category, for every past campaign, per each quarter.	Integer
Category ranking	Ranking of categories by the profit achieved in past campaigns, per quarter.	Integer
Purpose	Obtain most profitable categories in each quarter, based on previous campaigns.	

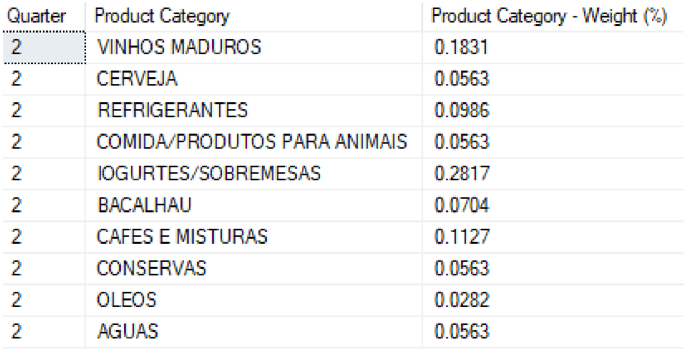
Table 3 - Definition of labels present in the best categories analytics

With that information, the second analysis (table on the right) can be performed, which involves ranking each category based on its profit. The categories that earned the most profit in prior campaigns can thus be discovered and ranked, together with the average quantity of products within each category.

Following that, by ordering the category ranking ascending and getting, say, the top ten categories, you can filter the results and get the most profitable categories based on previous campaigns, customer trends, and seasonality. Then, a method for generalizing the number of products based on the top campaigns must be discovered, which can be accomplished by obtaining a category weight. Each category weight can now be computed using the calculation below (for category *c*, and top *x* categories *t*):

$$Weight\ (c) = \frac{Average\ n^{o}\ of\ products\ (c)}{\sum Average\ n^{o}\ of\ products\ (t)} \quad [3.2]$$

Further to the calculation of the corresponding weight for each top category, the succeeding output is obtained:



Quarter	Product Category	Product Category - Weight (%)
2	VINHOS MADUROS	0.1831
2	CERVEJA	0.0563
2	REFRIGERANTES	0.0986
2	COMIDA/PRODUTOS PARA ANIMAIS	0.0563
2	IOGURTES/SOBREMESAS	0.2817
2	BACALHAU	0.0704
2	CAFES E MISTURAS	0.1127
2	CONSERVAS	0.0563
2	OLEOS	0.0282
2	AGUAS	0.0563

Figure 3 - Weights per top categories

This output is composed by the following labels:

Label	Definition	Data Type
Quarter	Quarter of the year in which the campaign happened.	Integer
Product Category	An aggregator of products based on their use and type.	String
Product Category – Weight (%)	The weight of the corresponding category, in the total of the top *x* categories.	Float (percentage)
Purpose	Obtain the most profitable categories and their weights, to further know the number of products to choose per category.	

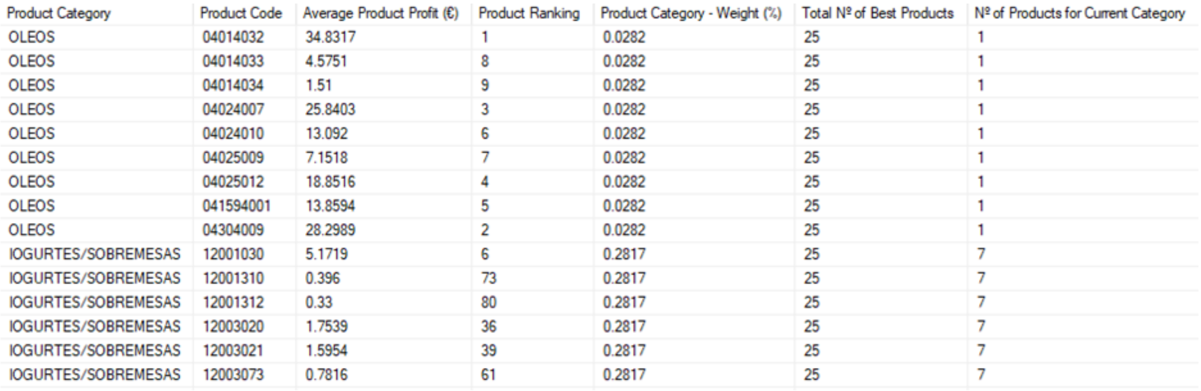
Table 4 - Definition of labels present in the best categories weights

The next step is to use the estimated weights to acquire the previously indicated top items. To accomplish so, an overview of existing items per category, as well as their average profit in previous categories, must be computed. This data must account for previously indicated factors such as seasonality and existing stock per product. As a result, only data from campaigns in the same quarter as the one in which the process is being run is used, and items with insufficient stock for the campaign duration defined by the user (labelled as such) are eliminated, so they are not even suggested in the next phases.

This allows one to comprehend the success of these items as well as how they are dispersed in terms of profitability inside their respective categories. With this information, it is also possible to calculate the respective product ranking, which will order products by their profit inside each category.

To finish the process and collect the best items to include in the campaign, one piece of information must be obtained: the exact number of best products to include. This amount is set by the system's user and can range from five to fifty items.

This information is depicted in the following figure, along with an example of the actual calculations:



Product Category	Product Code	Average Product Profit (€)	Product Ranking	Product Category - Weight (%)	Total Nº of Best Products	Nº of Products for Current Category
OLEOS	04014032	34.8317	1	0.0282	25	1
OLEOS	04014033	4.5751	8	0.0282	25	1
OLEOS	04014034	1.51	9	0.0282	25	1
OLEOS	04024007	25.8403	3	0.0282	25	1
OLEOS	04024010	13.092	6	0.0282	25	1
OLEOS	04025009	7.1518	7	0.0282	25	1
OLEOS	04025012	18.8516	4	0.0282	25	1
OLEOS	041594001	13.8594	5	0.0282	25	1
OLEOS	04304009	28.2989	2	0.0282	25	1
IOGURTES/SOBREMESAS	12001030	5.1719	6	0.2817	25	7
IOGURTES/SOBREMESAS	12001310	0.396	73	0.2817	25	7
IOGURTES/SOBREMESAS	12001312	0.33	80	0.2817	25	7
IOGURTES/SOBREMESAS	12003020	1.7539	36	0.2817	25	7
IOGURTES/SOBREMESAS	12003021	1.5954	39	0.2817	25	7
IOGURTES/SOBREMESAS	12003073	0.7816	61	0.2817	25	7

…

Figure 4 - Products per category display

As can be seen, the number of products to include in each category is computed using the previously calculated category weights as well as the overall number of best products to include (user input). As a result, this number may be defined as follows:

$$N^{\underline{o}}\ of\ best\ products\ per\ category = round(Category\ Weight\ (\%)\ \times Total\ N^{\underline{o}}\ of\ Best\ Products, 0) \quad [3.3]$$

The labels displayed in Figure *4* can be defined as:

Label	Definition	Data Type
Product Code	Unique identifier of each product.	String
Average Product Profit (€)	The average profit achieved with each product in past campaigns, per category, in the quarter of analysis.	Float (currency)
Product Ranking	Ranking of products by the profit achieved in past campaigns, per category, in the quarter of analysis.	Integer
Total Nº of Best Products	The number of best products to be included in the campaign, chosen by the user.	Integer
Nº of Products for Current Category	The number of products for each category to include in the campaign, calculated using the formula [3.3]	Integer
Purpose	Obtain product rankings per category based on profitability, using data from previous campaigns in the same quarter of analysis and filtering out products with insufficient stock.	

Table 5 - Definition of labels present in the product ranking analysis

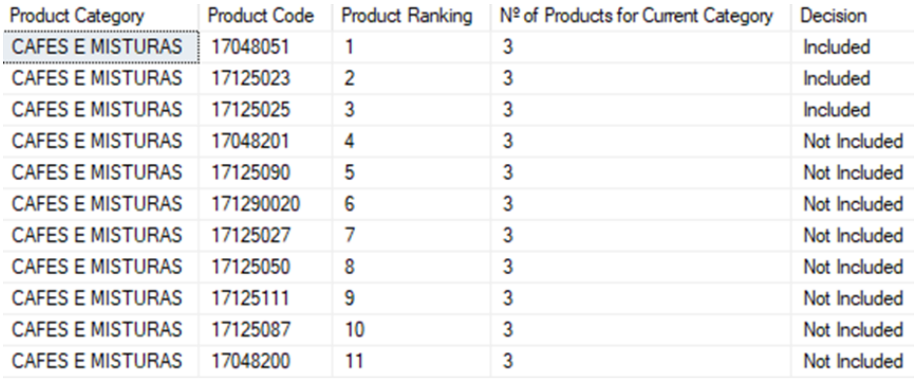
It is worth noting that the first five columns of this table (shown in Figure 4) are combined into a single final table that contains the average profit and ranking of each product per category for the current quarter and is updated once every day in the database. The concept of materialization is to store data from a previous analysis into a table, and this is used when the analysis takes too long due to its complexity or volumetry of information, and when it is written into a table, the results are displayed more quickly, avoiding the need to pre-compute all the information. This implies that all the computations required to arrive at the best items based on the sales table (which has tremendous volumetry) do not have to be performed in every single iteration. So, every day, this table is updated with the current day's sales data, and it is then used in each execution along with the total number of best goods (defined by the user).

Finally, one more rule must be applied in order to find the final basket of best products, which can be accomplished by filtering the results presented in Figure 4 to only get records in which the *Product Ranking* is lower or equal to the *Nº of Products for Current Category*, allowing the most profitable products in each category to be obtained based on the previously calculated number of products per category (for instance, if some category has three products, then the three most profitable products from that category, based on past campaigns in the same quarter, will be the chosen ones).

$$Decision\colon \begin{cases} Product\ Ranking \leq Nº\ of\ Products\ for\ Current\ Category, Included \\ Product\ Ranking > Nº\ of\ Products\ for\ Current\ Category, Not\ Included \end{cases}$$

[3.4]

The rule present in the equation [3.4] represents the filtering that is made on the results. As an example, Figure *5* demonstrates this constraint for a single category (*CAFES E MISTURAS*), which got a total of three products based on earlier category weights computation and total number of top products to include:



Product Category	Product Code	Product Ranking	Nº of Products for Current Category	Decision
CAFES E MISTURAS	17048051	1	3	Included
CAFES E MISTURAS	17125023	2	3	Included
CAFES E MISTURAS	17125025	3	3	Included
CAFES E MISTURAS	17048201	4	3	Not Included
CAFES E MISTURAS	17125090	5	3	Not Included
CAFES E MISTURAS	171290020	6	3	Not Included
CAFES E MISTURAS	17125027	7	3	Not Included
CAFES E MISTURAS	17125050	8	3	Not Included
CAFES E MISTURAS	17125111	9	3	Not Included
CAFES E MISTURAS	17125087	10	3	Not Included
CAFES E MISTURAS	17048200	11	3	Not Included

…

Figure 5 - Best Products filtering example

In this manner, the output which is passed to the Best Products model would consist of a collection of products, the total number of which was determined by the user, composing the most lucrative products per category, where the number of products per category reflects the behaviour applied to successful previous campaigns. The architecture behind the Products Discovery process is demonstrated in Figure *6*, containing the steps involved and the communication between systems.

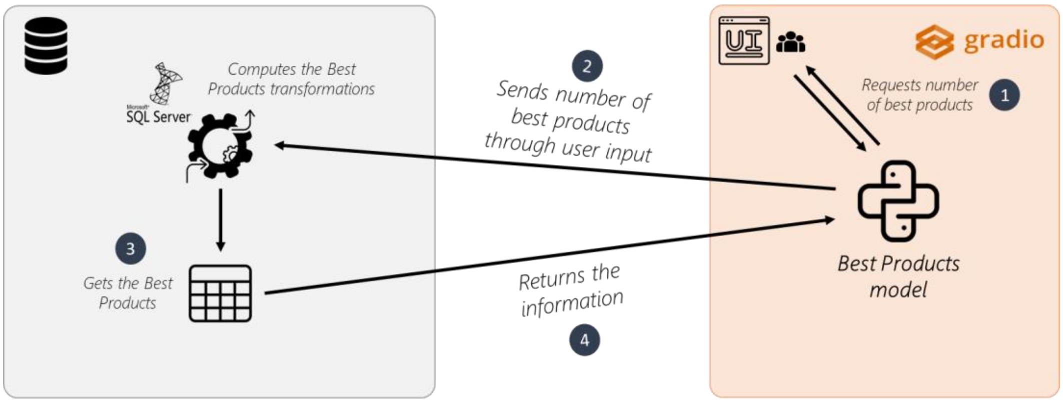

Computes the Best Products transformations
SQL Server
2
Sends number of best products through user input
gradio
Requests number of best products
1
3
Gets the Best Products
Best Products model
Returns the information
4

Figure 6 - Products Discovery architecture

3.4.2. Basket Analysis model

As previously approached in section 2.5, the Basket Analysis process is a technique used to identify patterns and associations between items frequently purchased together.

Regarding the type of data in hands, the Apriori method was the one to use to calculate product associations, due to its effectiveness in mining association rules from transactional databases, and it is also preferable due to its scalability and incremental mining feature, allowing fast results in large datasets.

To begin, data is required to calculate relationships between items, and these associations can only be identified when every product purchased, every transaction that has ever occurred is analysed and as expected, the more data utilized for calculating associations, the more accurate the findings will be. So, as the first step, the model will need to go through all the sales data. Because sales data from a retailer contains a large amount of information, and this process will only serve to enrich the actual basket of products discovered in the previous section, the basket analysis model will only work with invoices that include the products computed in the Best Products section, this way computing associations with less data and producing more accurate results.

To begin, this model will take the list of best products as an initial input and calculate associations using data from every invoice that has featured that product, enriching the final basket with products that normally go well with the ones previously discovered.

After knowing the best products, a request to the database is made to get the actual sales data. Table *6* contains the definitions of the dataset that serves as the second input to the model.

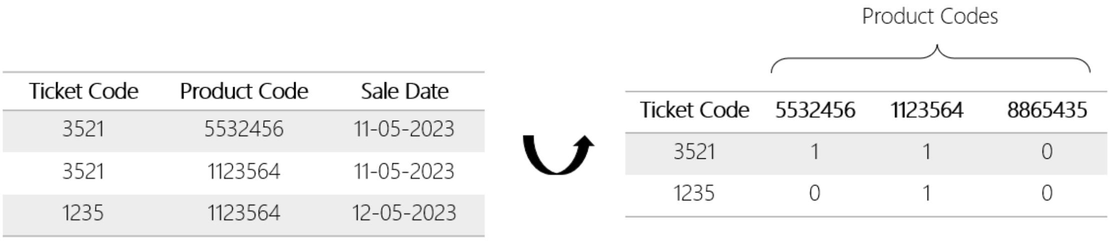
Label	Definition	Data Type
Product Code	Unique identifier of each product.	String
Ticket Code	Unique identifier of the invoice in which the product was purchased.	Integer
Sale Date	Date in which the invoice was produced.	Date
Purpose	Obtain the list of sales in which the best products existed.	

Table 6 - Definition of labels present in the basket analysis input dataset

The data granularity in this scenario differs from past analysis, with one row per product in an invoice reflecting a significant volumetry of information. Furthermore, sales with fewer than two products were excluded for consistency and to avoid deceiving the results.

Following, the calculation of the frequent itemsets using Apriori is needed, to take conclusions from the associations found. The technique of locating sets of items that frequently appear together in a given dataset is referred to as frequent itemsets discovery.

However, some data modification was performed prior to utilizing the One Hot Encoding (OHE) approach. Because the Apriori can execute bitwise operations such as counting the occurrence of itemsets and creating candidate itemsets, this method will make it easier to sift through all the sales data by minimizing memory use and providing efficient data processing. These processes may be faster in terms of computing than processing the original dataset with categorical values. A visual clarification of what this method performs can be observed in Figure *7*.



Ticket Code	Product Code	Sale Date
3521	5532456	11-05-2023
3521	1123564	11-05-2023
1235	1123564	12-05-2023

	Product Codes		
Ticket Code	5532456	1123564	8865435
3521	1	1	0
1235	0	1	0

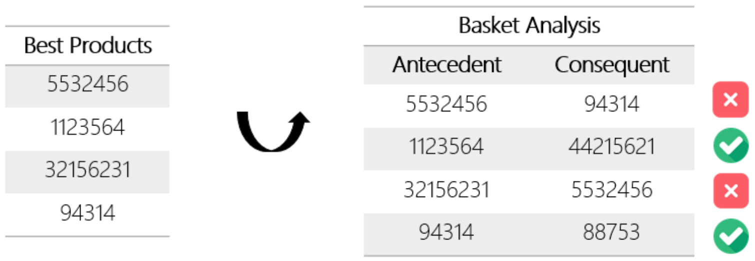
Figure 7 - One Hot Encoding representation

With the rearranged dataset, the process of finding the most frequent itemsets can then be performed. For this specific purpose, the lift measure was the one used since it measures the of the strength of association between two itemsets, demonstrating how likely it is that itemset B will be purchased if itemset A is purchased (for additional clarifications, see section 2.5).

This method, like the Best Products process, requires a particular quantity of products to be added to the final basket. This relates to the number of products added to the Best Products collection using Basket Analysis. Likewise, the user is the one who defines this amount as an input, which will be used in this phase.

Finally, after computing the frequent itemsets and the resulting association rules, the ones with the greatest amount of lift will be chosen, based on the quantity of products to be added by the user.

It is also crucial to note that items discovered in this phase that were previously discovered in the Best Products process are deleted to minimize duplicating campaign suggestions. Also, if a certain item which is found but does not have sufficient stock for the campaign duration, is also removed from the suggestions. This process is illustrated in Figure *8*:



Best Products
5532456
1123564
32156231
94314

Basket Analysis		
Antecedent	Consequent	
5532456	94314	☒
1123564	44215621	✓
32156231	5532456	☒
94314	88753	✓

Figure 8 - Basket Analysis output filtering

Beyond that, and as a last step, the output of this model is combined to the output of the Best Product, yielding the Product Discovery result. This final product basket is then utilized to calculate the campaign, a procedure that will be discussed in the next sections.

The architecture underlying the Basket Analysis flow may then be seen in the image below, which is made up of the orderly flow of stages taken to reach the outcome.

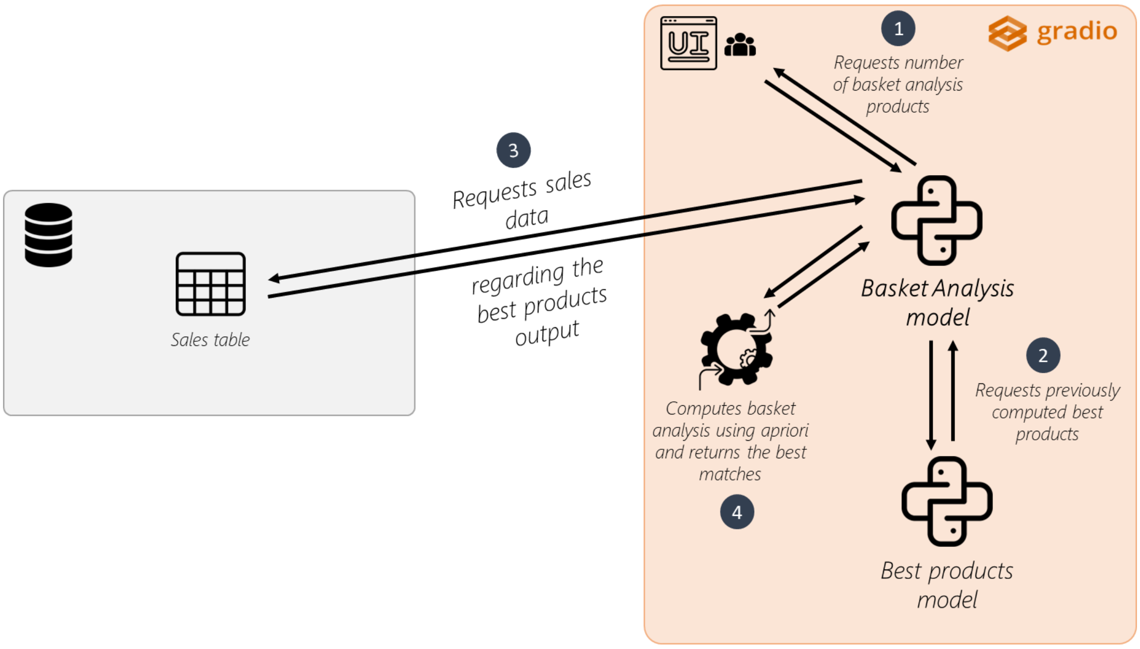

gradio
1
Requests number of basket analysis products
3
Requests sales data regarding the best products output
Sales table
Basket Analysis model
2
Requests previously computed best products
Computes basket analysis using apriori and returns the best matches
4
Best products model

Figure 9 - Basket Analysis architecture

3.5. Campaign Generation flow

This section covers the process of generating the actual campaign, which is the recommender system output. The Price Estimation and Forecasting modules are included in the Campaign Generation flow, which cover the process of assigning prices to products and simulate campaign results.

Working with the previous flow output, which is the basket of products that will comprise the campaign itself, some additional information must be added. The approximate price that each product should have, as well as how that campaign, composed of the previously computed basket of products and prices, will operate, are crucial pieces worth understanding.

This process is critical because it tells the retailer how many units of each product will be sold and at which price, on each day of the campaign and in each store, as well as how much profit the retailer should make (this profit is computed using the equation [3.1], based on the estimated price evaluated by the Price Estimation model, the actual product cost, and the predicted sale quantities produced by the Forecasting model).

Before delving into these flow models, a section devoted to data rearrangements outlines several critical modifications performed to sales data to improve the effectiveness of the Price Estimation and Forecasting models and reduce processing time and computational efforts.

3.5.1. Data Rearrangement

As previously stated in section 3.2, the data utilized for the present solution is derived from a typical Business Intelligence (BI) procedure known as ETL (Extract, Transform, and Load), implying that the data is already structured, processed, and ready for consumption. Otherwise, if this was not the case and the data was raw as it came from the Point of Sales (POS) devices, some additional tasks would need to be completed for the information to be as clean and rearranged as possible, such as removing irrelevant information, dividing the data into different and organized tables with keys and constraints, and so on.

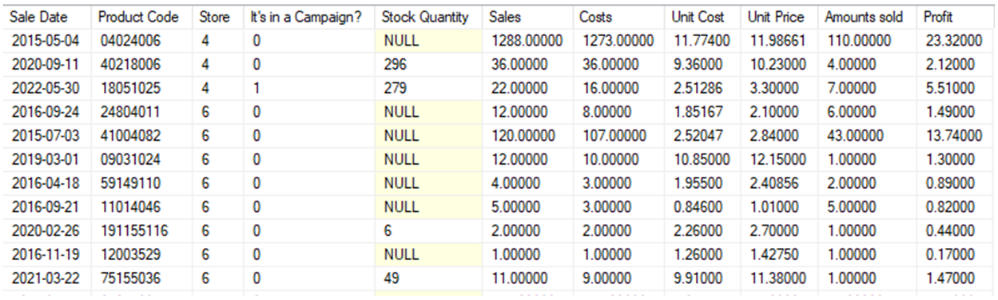
Based on this, transformations are few and less radical, but some need to be performed. Because the system is dealing with data from retail sales, the granularity of the sales table would be one row per product in an invoice on a store in a day, implying a massive volumetry of information. Processing all this data would take a long time and provide a poor user experience, which is why pre-materializing aggregated data or minimizing the need to recompute models at each execution is vital to consider.

As previously stated, several of these time-saving strategies are suggested in the Products Discovery models. In the Best Products method, data is pre-materialized by entering aggregated information on product rankings by category into a table and updating this information daily. In the Basket Analysis method, however, none of these strategies can be employed since the highest granularity of data (one row each product on an invoice on a store in a day) is required to calculate the product associations.

However, pre-materializing data may be rather useful for the following models, since whether assessing past product pricing or training a model to forecast the daily outcomes of each product, the maximum granularity of data is not required, because the concept of invoice is unnecessary.

Based on this, a process to materialize aggregated data into a table is responsible for this part, which updates the data in the materialized table once per day. This process will write the data with a granularity of one row per product in a store on a day, summing information like the sales, costs, stock quantity or amount of units sold for that day in each store, and then compute labels like the profit. Furthermore, this procedure ignores data that can be regarded as outliers or that can affect the data's quality, such as credits, empty invoices, sales with a price of zero, among other things. These outliers may rise due to system faults, and this technique stops them from passing to the models.

A snapshot of this table can be observed in Figure *10*:



Sale Date	Product Code	Store	It's in a Campaign?	Stock Quantity	Sales	Costs	Unit Cost	Unit Price	Amounts sold	Profit
2015-05-04	04024006	4	0	NULL	1288.00000	1273.00000	11.77400	11.98661	110.00000	23.32000
2020-09-11	40218006	4	0	296	36.00000	36.00000	9.36000	10.23000	4.00000	2.12000
2022-05-30	18051025	4	1	279	22.00000	16.00000	2.51286	3.30000	7.00000	5.51000
2016-09-24	24804011	6	0	NULL	12.00000	8.00000	1.85167	2.10000	6.00000	1.49000
2015-07-03	41004082	6	0	NULL	120.00000	107.00000	2.52047	2.84000	43.00000	13.74000
2019-03-01	09031024	6	0	NULL	12.00000	10.00000	10.85000	12.15000	1.00000	1.30000
2016-04-18	59149110	6	0	NULL	4.00000	3.00000	1.95500	2.40856	2.00000	0.89000
2016-09-21	11014046	6	0	NULL	5.00000	3.00000	0.84600	1.01000	5.00000	0.82000
2020-02-26	191155116	6	0	6	2.00000	2.00000	2.26000	2.70000	1.00000	0.44000
2016-11-19	12003529	6	0	NULL	1.00000	1.00000	1.26000	1.42750	1.00000	0.17000
2021-03-22	75155036	6	0	49	11.00000	9.00000	9.91000	11.38000	1.00000	1.47000

...

Figure 10 - Snapshot of materialized aggregated data from sales

As a result, this table will have a daily result for each product, indicating how popular it was at each store, at what price and with what cost, showing the entire profit for the merchant on that product.

Table *7* contains the definition of each field in the materialized table:

Label	**Definition**	**Data Type**
Sale Date	Date in which the invoice was produced.	Date
Product Code	Unique identifier of each product.	String
Store	Unique identifier of the store in which the product was sold.	Integer
It's in a Campaign?	Has the value 1 if the product was on a campaign at that store on that day, and a value of 0 otherwise.	Boolean
Stock Quantity	Represents the amount of stock that the product had on that day at that store.	Integer
Sales	Represents the amount sold, in euros.	Float (currency)
Costs	Represents the costs regarding the sales of that product.	Float (currency)
Unit Cost	Symbolizes the unit cost of that product on that day.	Float (currency)
Unit Price	Symbolizes the unit price of that product on that day.	Float (currency)
Amount Sold	The number of units sold.	Integer
Profit	The total profit achieved, being calculated using the equation [3.1]	Float (currency)
Purpose	Materialize converted sales data in a table to reduce computing processing by lowering the volumetry of data to evaluate.	

Table 7 - Definition of labels present in the aggregated data from sales

Based on this, the materialized table holding the aggregated sales data will be utilized as the basis for the subsequent Price Estimation and Forecasting methods, increasing computation performance, and drastically lowering the volumetry of data to analyze.

3.5.2. Price Estimation model

One key aspect of any campaign in any business segment is the price of each product. Yet, pricing can be a complex matter since lots of aspects can be involved, such as the price applied in competitors, costs, margins, consumer interests or even stock quantities (Grewal et al., 2011). If the goal is price optimization, all these factors must be examined since the ideal price sought tries to maximize or minimize margins, stocks, or public interest.

In this situation, a price estimation will be performed, which will consider previous seasonal behaviors that were applied in each product during a campaign. However, some of the previously specified aspects, such as stocks and margin maximization, are also being explored, but in previous models, as discussed in the Products Discovery section (3.4), by selecting products with enough stock for the current campaign and the most profitable ones considering the season.

To estimate pricing based on product campaign history, past product data, particularly the change in product price when it joins the campaign, must be analyzed. This, however, requires daily analysis of each product's sales data to determine the C Day (day in which the product went to campaign). Even though the previously described materialized sales data table is utilized in this process, one more materialized table with even less data can be built expressly for this operation to minimize processing massive volumes of information.

As described in the previous section (3.5.1), the daily updated materialized sales data has a granularity of "one row per product in a store on a day," which means that if the retailer has three stores, this table will have three rows per product on a day. However, because the price of each product is the same in each existing store, the store label is unnecessary in the dataset. If the store label is subsequently deleted, the number of rows decreases to one-third of the original size of the dataset, indicating a considerable reduction in data. As a result, the pricing estimating model will only use a limited version of the materialized sales data table, without the store component.

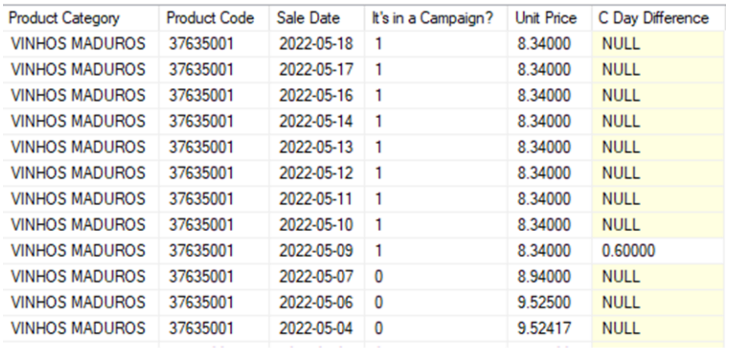
As previously stated, the C Day for each product must be determined to examine historical store pricing practices when the product joins the campaign. This must be done for each product separately since the price changes that a product experiences when it enters a

campaign vary greatly across product categories and between different products within the same category. For example, a more costly wine may receive an average discount of 5 euros when it participates in a campaign, but a cheap wine may only receive a fifty-cent discount.
Another factor to consider is the time of year. Using an example from section 3.4.1, consumption of cod fish in Portugal increases significantly during the winter season while remaining relatively low during the summer, implying that customers will seek this product more frequently during the fourth quarter of the year. As a result, this product may suffer a bigger discount during non-wanted seasons such as summer to boost customer interest in the product, but it may have a tiny discount when it joins the campaign in the winter considering people will purchase this product anyhow.

As a first stage, an analysis per product per day must be performed, checking every time in the past that the product joined a campaign, and if it happened, performing the price difference between the preceding day of the campaign and the campaign price. This can be accomplished using the below formula, for product *p*, campaign *c* and campaign starting day *cd*:

$$C\ Day\ Difference\ (p, c) = Unit\ Price\ (p, cd - 1) - Unit\ Price\ (p, cd) \qquad [3.5]$$

After applying this logic for each product, a result like the one shown in Figure *11* is achieved:



Product Category	Product Code	Sale Date	It's in a Campaign?	Unit Price	C Day Difference
VINHOS MADUROS	37635001	2022-05-18	1	8.34000	NULL
VINHOS MADUROS	37635001	2022-05-17	1	8.34000	NULL
VINHOS MADUROS	37635001	2022-05-16	1	8.34000	NULL
VINHOS MADUROS	37635001	2022-05-14	1	8.34000	NULL
VINHOS MADUROS	37635001	2022-05-13	1	8.34000	NULL
VINHOS MADUROS	37635001	2022-05-12	1	8.34000	NULL
VINHOS MADUROS	37635001	2022-05-11	1	8.34000	NULL
VINHOS MADUROS	37635001	2022-05-10	1	8.34000	NULL
VINHOS MADUROS	37635001	2022-05-09	1	8.34000	0.60000
VINHOS MADUROS	37635001	2022-05-07	0	8.94000	NULL
VINHOS MADUROS	37635001	2022-05-06	0	9.52500	NULL
VINHOS MADUROS	37635001	2022-05-04	0	9.52417	NULL

...

Figure 11 - Price Estimation C Day Difference

The existing labels of the above analysis are described in Table *8*:

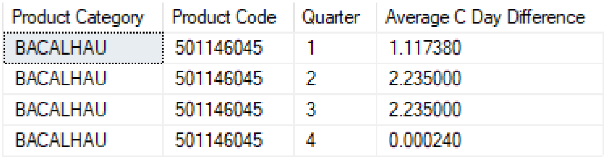
Label	Definition	Data Type
Product Category	An aggregator of products based on their use and type.	String
Product Code	Unique identifier of each product.	String
Sale Date	Date in which the invoice was produced.	Date
It's in a Campaign?	Has the value 1 if the product was on a campaign at that store on that day, and a value of 0 otherwise.	Boolean
Unit Price	Symbolizes the unit price of that product on that day.	Float (currency)
C Day Difference	Represents the price difference when a product enters in a campaign, calculated using formula [3.5]	Float (currency)
Purpose	Compute the C Day Difference for every campaign of each product.	

Table 8 - Content of Price Estimation C Day Difference analysis

Based on this, the data required for the Price Estimation model is nearly complete, but it may be further aggregated. As previously stated, each product's price will be calculated by applying historical behaviors to the current product price, implying that the most recent price will suffer the average difference that was normally applied to that product when it entered prior campaigns. One thing to note is that the term difference, rather than discount, is used. When items are placed in campaigns, their prices are normally reduced; but, in some situations, such as when the stock for that product is typically low or the demand for that product is particularly high, the price may not be reduced. These patterns are typically seasonal, being cached by this process by computing the C Day Difference averages quarterly. This concept is covered by the following equation, for product *p* and quarter *q*:

$$Average\ C\ Day\ Difference\ (p,q) = \frac{\sum C\ Day\ Difference\ (p,q)}{N^{o}\ of\ Campaigns\ (p,q)} \quad [3.6]$$

One example of this process applied to a particular product can be observed in Figure *12*:



Product Category	Product Code	Quarter	Average C Day Difference
BACALHAU	501146045	1	1.117380
BACALHAU	501146045	2	2.235000
BACALHAU	501146045	3	2.235000
BACALHAU	501146045	4	0.000240

Figure 12 - Average C Day Difference applied to a specific product

In this example, the technique is applied to a specific product in the cod fish category, which suffers a decreased price discount during campaigns during seasons with higher demand for that product. So, in most cases, it may generate positive results (which will result in discounts to the real product price) but, in certain situations, it may get negative values (which will produce increases in the price). This can be understood by looking at formula [3.5], because the C Day Difference is calculated by subtracting the product price on the previous day of the campaign from its price on the actual day of the campaign, implying that positive values are categorized as discounts and negative values as price increases.

After this data is computed, it is requested and utilized by the python Price Estimation model, which will only use it for the previously computed basket, composed by the Best Products and Basket Analysis ones. The algorithm will then locate the most recent price for each one of these products and apply the Average C Day Difference computed in the database for the current quarter of analysis, generating the estimated price in this manner.

There are some situations that require special attention, such as when there is no past campaign data for a specific product or when the predicted price is less than the current product pricing.

The model takes the following actions to comply with the first occasion: If the product has no data for previous campaigns in the current quarter, the Average C Day Difference from the previous quarter is used; if the previous quarter information is also missing, an average of the existing quarters with data for that product is used; finally, if the product has never had any campaigns, the Average C Day Difference for every four quarters will be empty, and so the average of this value is computed using data from other products in the same category, in the current quarter.

The second example, defined by predatory pricing, is when a seller sets the price of a product below its cost, which is a circumstance to avoid. The Price Estimation model may produce this if the current quarter's Average C Day Difference is large enough to cause the current product price to fall below the cost. This is also more likely to occur if the retailer's current product price is near to its cost, implying a low margin. To address this, in these cases, the

Price Estimation model will add a 5% margin to the actual product cost and use it as the estimated price.

The architecture for this model, as well as the requests between the Python model and the database, may be seen in the image below.

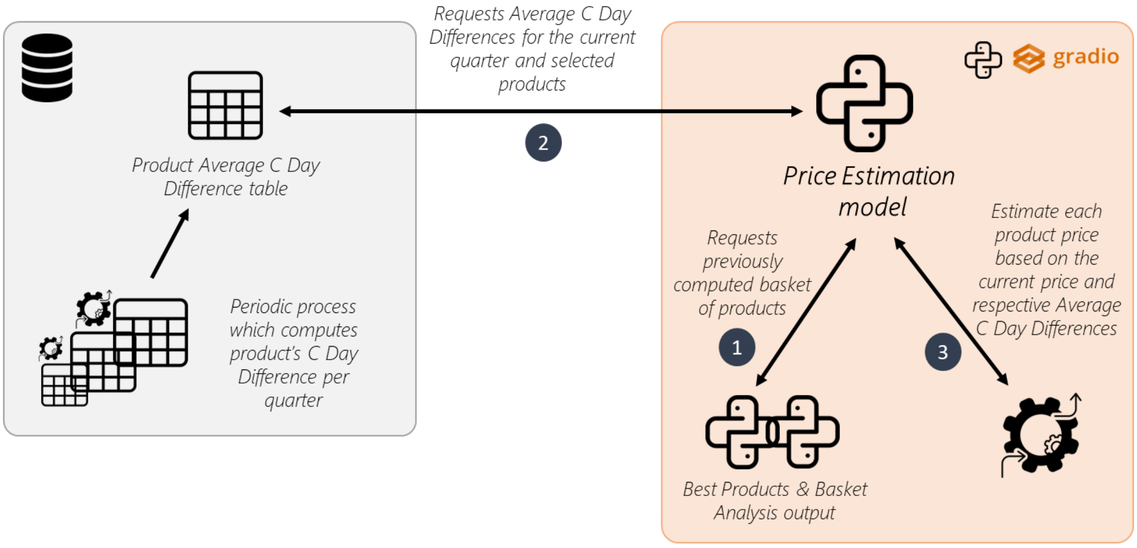

Requests Average C Day Differences for the current quarter and selected products
2
gradio
Product Average C Day Difference table
Periodic process which computes product's C Day Difference per quarter
Price Estimation model
Requests previously computed basket of products
1
Estimate each product price based on the current price and respective Average C Day Differences
3
Best Products & Basket Analysis output

Figure 13 - Price Estimation architecture

3.5.3. Forecasting model

The forecasting model employs a combination of database information and prior model output as a final step to estimate how a campaign made of a basket of products, each with a defined price and cost, would perform. Moreover, once applied, this model is expected to inform how many things will be sold and how much profit that campaign would generate throughout the previously defined number of days that it should endure.

So, in order to accomplish this, this model should learn from previous activity of each discovered product in the newly created campaign, that is, all previous sales information regarding these products, and then predict how many quantities will be sold in each day, taking into account the estimated price in the previous model, as well as its cost (each product cost is daily updated in the database and it comes through the ETL process described in section 3.2). The profit is then determined using the equation [3.1] by knowing the daily forecasted sales quantity for each product in the campaign.

Because the component to be forecasted is numerical (the sales quantity) and is based on a collection of factors (such as the day, store, product price, or product cost), a regression model is the best technique to meet this requirement.

Based on this, as a first step, this model will request outputs from the previous models, such as which products will be contained in the campaign, and what's the corresponding price for each one of them. Second, to train the regression model, the history of sales data for those products must be extracted. Because the needed information is the daily outcome of each product, and a granularity of one row per product in a store on a day is sufficient, the aggregated sales data collected and updated by the method described in section 3.5.1 is then utilized to train the model. The model then extracts the relevant data and performs some simple pre-processing activities such as associating the correct data type with each label. Another procedure related to the feature engineering process that had to be accomplished was dealing with text-based data. Labels like the product code or category are textual, and while they are required for the forecasting model, they cannot be used in regression algorithms. To address this, a process of hashing these columns using the label encoding technique, converting them to integers while maintaining their integrity, was implemented. After training and compute predictions, the hashed values which were previously mapped are then decrypted, returning to their original value.

More extensive data treatment and pre-processing tasks are not required to be performed to the data present on the Data Warehouse (DW) because the data has already been arranged by the ETL process (section 3.2); additionally, the process of creating aggregated sales data (section 3.5.1) already computes similar tasks such as the removal of outliers (invoices with negative or 0 final prices) or the removal of products that do not have sufficient stock considering the campaign duration (done by the Best Products & Basket Analysis models described in sections 3.4.1 and 3.4.2).

With the needed information prepared, the data is then divided into two datasets, one regarding the categories needed for the forecasting operation (*x*) and other containing the label to predict (*y*), as described in Table *9*:

Label	**Definition**	**Data Type**
	x	
Year	Year in which the sale happened.	Integer
Quarter	Quarter in which the sale happened.	Integer
Month	Month in which the sale happened.	Integer

Day	Day in which the sale happened.	Integer
Weekday	Day of the week in which the sale happened.	Integer
Store	Unique identifier of the store in which the product was sold.	Integer
It's in a campaign?	Has the value 1 if the product was on a campaign at that store on that day, and a value of 0 otherwise.	Boolean
Product Category	An aggregator of products based on their use and type (encoded).	Integer
Product Code	Unique identifier of each product (encoded).	Integer
Unit Cost	Symbolizes the unit cost of that product on that day.	Float (currency)
Unit Price	Symbolizes the unit price of that product on that day.	Float (currency)
Purpose	Obtain the history sales data needed to train the regression model	
	y	
Amount Sold	The number of units sold.	Integer
Purpose	Obtain the history sales data needed (amount sold) to train the regression model	

Table 9 - Definition of labels present in the data for training the regression model

This data structure is then used to train the models as further stated in section 4.1.3, first to find the most appropriate regression model and its hyperparameters, and then to do the necessary evaluations.

With this, the model could then be trained with sales data related to previously discovered products (outputted from the best products and basket analysis processes described in the previous sections) and then forecast the amount sold for each product in each day of the campaign and in each store for a certain duration of campaign days defined by the user. The profit and margin obtained with the generated campaign can also be computed using this information, as well as the previously estimated product prices and current product costs. The data flow and architecture of this model are depicted in a more intuitive manner in Figure *14*.

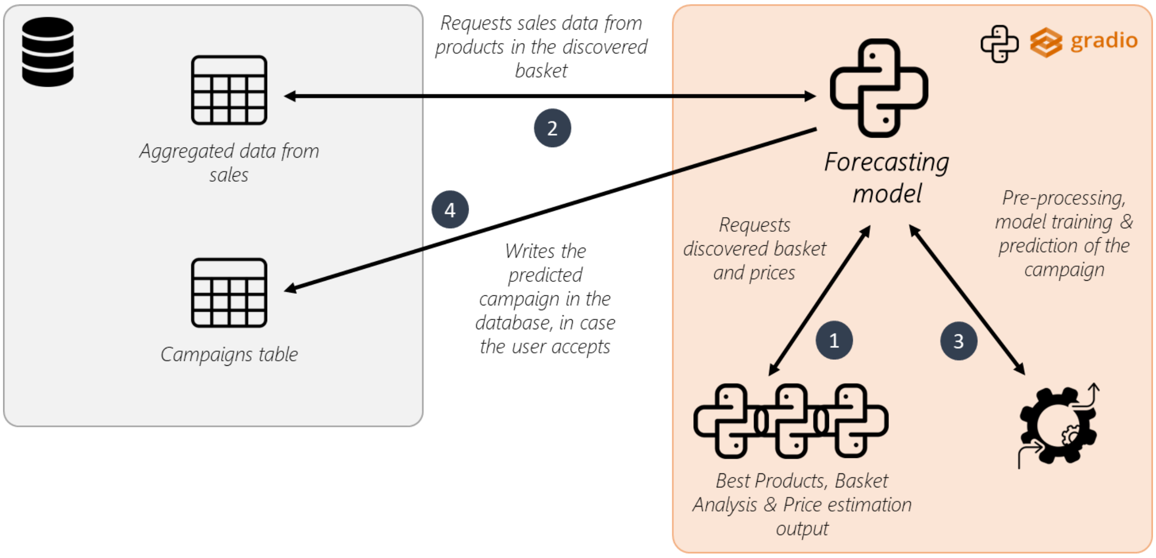

Aggregated data from sales
Campaigns table
Requests sales data from products in the discovered basket
2
4
Writes the predicted campaign in the database, in case the user accepts
gradio
Forecasting model
Requests discovered basket and prices
Pre-processing, model training & prediction of the campaign
1
3
Best Products, Basket Analysis & Price estimation output

Figure 14 - Forecasting model architecture

Now that all models have been described, it is able to evaluate the entire Recommender System as a distinct flow with many components talking with one another. To accomplish this, the system flow is described below, using the acronyms RS User (Recommender System's user), UI (Gradio Python UI Controller that communicates with the models), DB (database in SQL Server), and the model acronyms BP (Best Products), BA (Basket Analysis), PE (Price Estimation), and FM (Forecasting Model).

Component	Action
RS User	Passes as input variables the number of best products -> *varBP*, basket analysis products -> *varBA* and the campaign duration (in days) -> *varCD* Executes the **Products Discovery** process
UI	Ivokes the **BP** model passing variables *varBP* and *varCD*
BP	Executes **DB** procedures to find products with sufficient stock for the campaign duration
DB	Computes product category weights (based on the previous number of products per category) on previous campaings for the same period Ranks products inside each category based on their profitability Outputs the product's treated dataset -> *dbBPProducts*
BP	Based on *dbBPProducts,* applies filters to get the most profitable products per category, based on the category weights, profitability ranks and on *varBP* Returns the modified dataset -> *bpOutput*
UI	Gets *bpOutput* and passes it to the **BA** model, together with *varBA*
BA	Queries **DB** and finds through past sales data the best associations to match with the **BA** basket of products
DB	Returns data -> *dbBAData*
BA	Computes the associations between products based on *dbBAData* and keeps the more correlated ones based on the lift measure and on the *varBP* Outputs the **BA** products -> *baOutput*

UI	Aggregates *bpOutput* and *baOutput* products -> *productsOutput* Shows the *productsOutput* to the **RS User**
RS User	Chooses if the discovered products satisfy the business requirements Yes -> Executes the **Campaign Generation** process No -> Re-runs the **Products Discovery** or exits the system
PE	Executes **DB** procedures for each *productsOutput* product to get the Average C Day Difference
DB	Finds in previous campaigns on the same period the average price difference for each product when it enters in a campaings Outputs Average C Day Difference and the latest available product cost -> *dbPEData*
PE	Modifies the *productsOutput* to add the respective estimated price and last available cost provided from *dbPEData* Outputs the consolidated dataset -> *peOutput*
UI	Invokes the **FM** passing as input the *peOutput* and *varCD*
FM	Requests **DB** previous aggregated sales data regarding the products present in *peOutput*
DB	Returns data -> *dbFMData*
FM	Pre-processes and rearranges *dbFMData* and *peOutput* using the same techniques Trains the regression model Performs predictions Returns dataset with predictions -> *fmOutput*
UI	Computes profit, sale quantities and amount of products aggregations based on *fmOutput* Presents the data to the **RS User**
RS User	Chooses if the generated campaign should be saved in the database Yes -> Writes the campaign and PBI gets refreshed No -> Re-runs the **Campaign Generation** process or exits the system

Table 10 - System's flow

The flow of the Recommender System given in Table *10* explains how each component communicates with each other to obtain the desired result.

The model components (BP, BA, PE, and FM) as well as the DB details were thoroughly covered in this chapter, while the UI component is addressed in Annex section B. Also detailed in the Annex (in section C) is the tabular data model implemented in MS Power BI, which serves as the foundation for the reports analyzed in the following chapter.

4. Evaluation and Results

This chapter describes the evaluation process that was designed to analyze the efficiency of the recommender system's models, as well as an overview of the outcomes and outputs from the system's user perspective.

The evaluation method involves performing tests and assessments on the models, which are divided into two flows: Product Discovery and Campaign Generation. These evaluations are intended to determine whether these models provide answers to the evaluation research questions.

The results of the system are then discussed later in the chapter. To aid in the analysis process, a collection of dashboards was created using the Power BI reporting tool (with the data model specified in section C of the Annex), which is intended to be the primary tool for viewing the recommender system's results. As a result, the dashboards can precisely examine these results, which are composed of a basket of discovered products, their prices, and a daily prediction of quantities sold, revenues, costs, and profits.

4.1. Models Evaluation

Model evaluation is an important step to assess the efficacy of a model. A campaign recommender system is a process that demands precise assessments which, for the current system features, are divided in three categories:

- Are the recommendations accurate and meet the user requirements? In other words, are the products suggested satisfying the company's preferences and needs?
- Are the estimated prices correlated with the past pricing patterns and respect the company's policies?
- Are the forecasts precise in a way that will not mislead the user with inaccurate information?

These topics are critical and demand evaluations and specific analytics that can ease the process of answering them.

4.1.1. Evaluation Protocol

To complete the necessary evaluations, the models of the recommender system were divided into two key groups: Product Discovery and Campaign Generation. To accomplish the most accurate evaluation process, the data used for assessment differs between the two groups.

The whole history of campaigns was used in the Products Discovery evaluation to compute specific metrics, which were then compared to a produced basket of products using the Best Products and Basket Analysis models. For the Campaign Generation evaluation, a random previous campaign from mid-2021 was chosen, and its basket of products was utilized as an input for the Price Evaluation and Forecasting methods, the outputs of which were then compared to the real ones that occurred in the chosen campaign. The models' evaluation measures were also different. The recall at k measure (equation [4.4]) is used in both Products Discovery models to determine whether the proposed basket of products is a popular choice. The MAE (Mean Absolute Error) metric was used in the Price Estimation model to understand how estimated prices differ from actual prices, whereas in the Forecasting model, measures such as MAE, RMSE (Root Mean Squared Error), and R2 (R-Squared) were used for the assessment. These metric's equations are shown below.

$$MAE = \frac{1}{n}\sum |\, real\ value - predicted\ value \,| \quad [4.1]$$

$$RMSE = \sqrt{\frac{1}{n}\sum (real\ value - \ predicted\ value)^2} \quad [4.2]$$

$$R2 = \ 1 - \frac{\sum(real\ value - \ predicted\ value)^2}{\sum(real\ value - \ mean\ of\ the\ dependant\ variable)^2} \quad [4.3]$$

Although with the above methodology a clear validation of the recommender system results can be made, formulating targeted research questions aims to streamline the assessment of model outputs, ensuring a more straightforward and insightful analysis. Considering this goal, this evaluation process intends to answer to the following questions: Are the products discovered by the Best Products and Basket Analysis models (Product Discovery process) popular choices in past campaigns? Is there any product suggested by the Product Discovery process which was never part of any campaign before? Is the estimated price MAE from the Price Estimation model lower than 10 cents? Do the forecasted daily sale quantities in each store follow real data patterns and trends? These questions will be approached in the end of the Models Evaluation section.

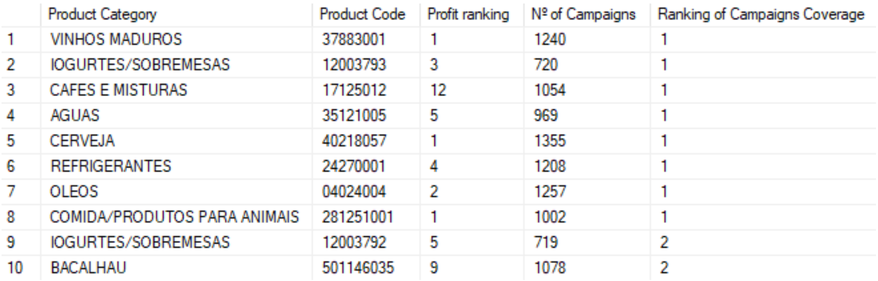
4.1.2. Products Discovery Evaluation

As previously stated in section 3.4, two models fuel the Products Discovery flow: the Best Products model and the Basket Analysis model, and they are linked since the latter works with the outputs of the first.

Starting with the Best Products, this model uses several heuristics and computations to choose the most profitable products with sufficient stock to include in a campaign. It computes this at the category level, learning from previous campaigns the percentage of products per category that are typically included in the campaign's basket and calculating which ones were the most rentable for that season. Aside from that, the quantity of best items discovered by the algorithm is determined by the user and varies with each execution (section 3.4.1).
This model's output is a set of products, each of which corresponds to a category, and it is ensured that the number of products per category to include in the campaign follows past business patterns, and that each category contains the most profitable products, namely, those with the highest success in terms of margins and amounts sold.
Second, the Basket Analysis model explores the sales history for links between different products purchased in the same tickets, identifying patterns of customer behavior, and recommending the most likely item to be sold (consequent) beside a certain item (antecedent). This model takes the basket generated by the Best Products model and looks for the best outcomes for those antecedents. Moreover, the number of products to enrich the basket with basket analysis is also chosen by the user, which may differ within each execution of the model (section 3.4.2).
To perform the assessment, a study of the most included items in the whole history of campaigns was performed, with each product assigned a rank based on its participation in previous campaigns. This information can then be utilized to determine whether the model's output encompasses popular products in campaigns or even products that have never been in campaigns before. This evaluation technique is known as *recall at k*, and it measures the proportion of relevant products successfully recommended to the user among the top k recommended items (Airen & Agrawal, 2022), and it is calculated using the equation below.

$$Recall@k = \frac{Nº\ of\ relevant\ items\ in\ the\ top\ k\ recommendations}{Total\ nº\ of\ relevant\ items} \qquad [4.4]$$

Following that, data needs to be prepared for the evaluation. Therefore, these models were tested with a total of 140 products (which is the average amount of products in campaigns, calculated using past campaigns information) generated by the Best Products and Basket Analysis models and the results were compared to the previous findings on a single analysis, as shown below (for the top ten rows).



	Product Category	Product Code	Profit ranking	Nº of Campaigns	Ranking of Campaigns Coverage
1	VINHOS MADUROS	37883001	1	1240	1
2	IOGURTES/SOBREMESAS	12003793	3	720	1
3	CAFES E MISTURAS	17125012	12	1054	1
4	AGUAS	35121005	5	969	1
5	CERVEJA	40218057	1	1355	1
6	REFRIGERANTES	24270001	4	1208	1
7	OLEOS	04024004	2	1257	1
8	COMIDA/PRODUTOS PARA ANIMAIS	281251001	1	1002	1
9	IOGURTES/SOBREMESAS	12003792	5	719	2
10	BACALHAU	501146035	9	1078	2

Figure 15 - Snapshot of the Products Discovery evaluation process

Figure *15* depicts the analysis's result, which consists of 140 rows, one for each product generated by the models. This data is then supplemented with the findings of an examination of the entire campaign history. Table *11* describes the analysis content.

Label	Definition	Data Type	Provenience
Product Category	An aggregator of products based on their use and type.	String	Best Products
Product Code	Unique identifier of each product.	String	Best Products
Profit Ranking	The product's profitability ranking [3.1] inside each category.	Integer	Best Products
Nº of Campaigns	The number of campaigns in which the product was used.	Integer	Campaign's history analysis
Ranking of Campaigns Coverage	The product's ranking of previous campaigns coverage computed in each category.	Integer	Campaign's history analysis
Purpose	Compare the Products Discovery's outputs to real past campaign's data.		

Table 11 - Definition of labels present in the Products Discovery evaluation process

Based on this, the first ten products shown in the picture represent the most and second most chosen products in previous campaigns for their respective categories (accordingly to the ranking of campaigns coverage).

Firstly, this research revealed that 100% of the 140 products generated by the model were part of campaigns in the past. Secondly, in attempt to acquire more accuracy in the results, the *recall at k* measure was computed for each category. Because categories have a distinct number of products that are included in campaigns, the top 35% most popular items in each category were utilized to establish the cluster of relevant items (denominator of the equation [5.1]). The basket of 140 previously discovered products was then compared to this cluster of relevant items, and the average *recall at k* was 85 percent, indicating that 85 percent of the products discovered by the Best Products and Basket Analysis models represent the top 35 percent most popular products in each category, showing positive outcomes.

Therefore, it was discovered that the products generated by this model are not only rentable and popular, but they are also the most chosen ones for incorporating campaigns in each respective category.

4.1.3. Campaign Generation Evaluation

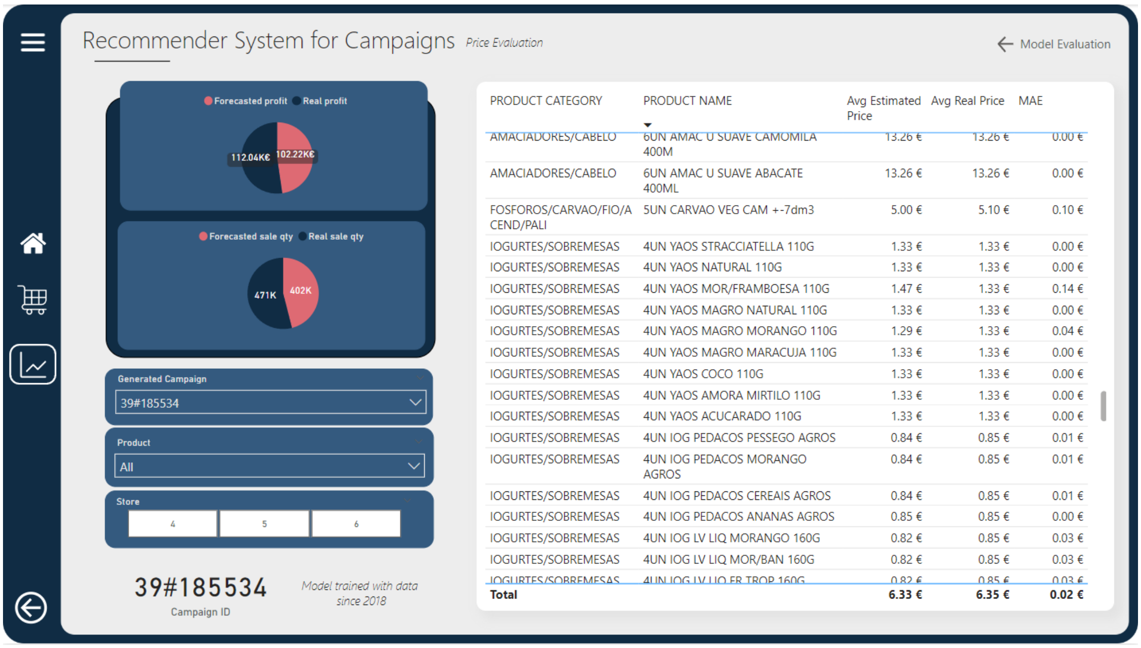
There are two main models for the Campaign Generation component, as explained in section 3.5: Price Estimation and Forecasting.

Because this component of the system is primarily focused on predicting or forecasting specific values, comparing the model's results to actual prior data is a more accurate approach to assess them.

To do so, data from a random campaign that existed in mid-2021, with 235 products, was picked and retrieved. This random campaign was retrieved using a SQL method that retrieved a list of all previous campaigns' unique identifiers (IDs) to a database table, assigned a random ordering, and then selected one campaign ID. Following that, the Products Discovery output was replaced with the products that were truly part of that specific campaign. Finally, the Campaign Generation section was carried out based on those products, for the time and duration of the randomly selected campaign (in this process, the forecasting model was trained with sales data up to the date in which the randomly chosen campaign happened, to avoid biased results).

In terms of price estimation, the model adopts a technique that entails learning the typical patterns of price fluctuations when each product enters a campaign (section 3.5.2). This procedure assigns an anticipated price to a product, which may or may not be suitable based on some circumstances or business strategies.

A specific dashboard was constructed in Power BI (PBI) utilizing a constructed data model (described in section C of the Annex) to examine how each estimated price differed from the real price of each commodity. To refine the study, users can use data filters in this visualization, such as the generated campaign via its unique ID, the product or product category, or even the store. Then, two pie charts show how the created campaign differs from the actual one in terms of profits and sales volume. Finally, a table with a data granularity of one row per product is provided, containing the Price Estimated model's estimated price, the true price applied to that product, and the MAE. Figure *16* contains an example of this analysis, which can be observed below.



Recommender System for Campaigns *Price Evaluation*

← Model Evaluation

Generated Campaign: 39#185534

Product: All

Store: 4 5 6

39#185534

Campaign ID

Model trained with data since 2018

PRODUCT CATEGORY	PRODUCT NAME	Avg Estimated Price	Avg Real Price	MAE
AMACIADORES/CABELO	6UN AMAC U SUAVE CAMOMILA 400M	13.26 €	13.26 €	0.00 €
AMACIADORES/CABELO	6UN AMAC U SUAVE ABACATE 400ML	13.26 €	13.26 €	0.00 €
FOSFOROS/CARVAO/FIO/ACEND/PALI	5UN CARVAO VEG CAM +-7dm3	5.00 €	5.10 €	0.10 €
IOGURTES/SOBREMESAS	4UN YAOS STRACCIATELLA 110G	1.33 €	1.33 €	0.00 €
IOGURTES/SOBREMESAS	4UN YAOS NATURAL 110G	1.33 €	1.33 €	0.00 €
IOGURTES/SOBREMESAS	4UN YAOS MOR/FRAMBOESA 110G	1.47 €	1.33 €	0.14 €
IOGURTES/SOBREMESAS	4UN YAOS MAGRO NATURAL 110G	1.33 €	1.33 €	0.00 €
IOGURTES/SOBREMESAS	4UN YAOS MAGRO MORANGO 110G	1.29 €	1.33 €	0.04 €
IOGURTES/SOBREMESAS	4UN YAOS MAGRO MARACUJA 110G	1.33 €	1.33 €	0.00 €
IOGURTES/SOBREMESAS	4UN YAOS COCO 110G	1.33 €	1.33 €	0.00 €
IOGURTES/SOBREMESAS	4UN YAOS AMORA MIRTILO 110G	1.33 €	1.33 €	0.00 €
IOGURTES/SOBREMESAS	4UN YAOS ACUCARADO 110G	1.33 €	1.33 €	0.00 €
IOGURTES/SOBREMESAS	4UN IOG PEDACOS PESSEGO AGROS	0.84 €	0.85 €	0.01 €
IOGURTES/SOBREMESAS	4UN IOG PEDACOS MORANGO AGROS	0.84 €	0.85 €	0.01 €
IOGURTES/SOBREMESAS	4UN IOG PEDACOS CEREAIS AGROS	0.84 €	0.85 €	0.01 €
IOGURTES/SOBREMESAS	4UN IOG PEDACOS ANANAS AGROS	0.85 €	0.85 €	0.00 €
IOGURTES/SOBREMESAS	4UN IOG LV LIQ MORANGO 160G	0.82 €	0.85 €	0.03 €
IOGURTES/SOBREMESAS	4UN IOG LV LIQ MOR/BAN 160G	0.82 €	0.85 €	0.03 €
IOGURTES/SOBREMESAS	4UN IOG LV LIQ FR TROP 160G	0.82 €	0.85 €	0.03 €
Total		**6.33 €**	**6.35 €**	**0.02 €**

Figure 16 - Price Estimation model's evaluation (dashboard)

As shown in the figure, the MAE was about 0.02 euros, indicating a clear approximation of the real given prices to the products and an understanding of the pricing patterns by the model. Once this data is available and computed, it is added together as the final piece to calculate campaign forecasts however, before that, the appropriate regression model must be chosen.

To start with, in regression algorithms, data must first be divided into a training set and a test set for the model to be evaluated and generalized. The training set aids in model learning and parameter estimation, whereas the test set serves as an independent benchmark to evaluate the model's performance on unseen data, ensuring robustness and preventing overfitting. The typical split between training and test sets is 70 - 80 percent on the training set to 20 - 30 percent on the test set, and in this case the split was 75 percent - 25 percent. Because the training dataset corresponds to sales data with time dimensions, shuffling is avoided while splitting datasets in train and test, avoiding forecasts of past values with predictions of future values.

Afterwards, the next stage is to create the regression model that will forecast the sale quantity per product during the campaign. One critical step in accomplishing this is selecting the appropriate algorithm to perform the operation. There are several regression models that can provide quite accurate forecasts, but they all differ in how they obtain their results, and cross-validation is one method for determining which method is best for the data at hand, which is used to evaluate a model's performance and generalization capacity. By providing a more rigorous estimate of the model's efficacy, this method helps to mitigate concerns such as overfitting, giving a comprehensive evaluation of the model's performance on multiple subsets of the data (Unpingco, 2016).

So, the cross-validation procedure was computed using fourteen of the most adequate regression algorithms for the current problem, and the results of the best four (based on the RMSE and R2 metrics) are represented below, in Table *12*.

Algorithm	RMSE	R2	Specifications
Random Forest	**32.0**	**0.65**	Model split: 75 percent training 25 percent testing Training Dataset: One year of sales data
XGBoost	38.8	0.61	
K-Nearest Neighbors	36.6	0.54	
Decision Trees	43.1	0.37	

Table 12 – Forecasting model cross-validation results

The results of the cross-validation procedures are shown in Table *12*, computed with the dataset represented in Table *9* of section 3.5.3, as it is the proper data structure to train the model and further perform forecasts. Also, one year of sales data was used for this process, corresponding to 1.6 million rows approximately. In addition, as previously stated, the train and test sets were separated without data shuffling to avoid predicting past values with future ones (for the current example with data from one year, having February in the test set while November is in the train set). With Random Forest achieving the best results, with a lower RMSE (Root mean squared error), which is a measure of the average difference between the predicted and actual values, and a higher R2 (R-Squared), which represents the proportion of the variance in the dependent variable that is predictable from the independent variables, indicating how well the model fits the data. This way, when analyzing through these variables, it is intended to choose the model with a lower RMSE and a higher R2 (Géron, 2017).

Even though the errors appear to be somewhat high, they may not be cause for concern since the models were validated using their default parameters. These parameters are known as hyperparameters in the field of regression algorithms, and they differ between algorithms, playing an important role in the model's performance and behavior. Also, there is no such thing as a universally optimal parameter because they can behave quite differently depending on the data being processed. As a result, hyperparameter tuning may be required to discover the optimal hyperparameters for the current model. This process evolves by performing multiple combinations of values to find the optimal configuration to find the optimal values in a specific model, being a computationally intensive process depending on the complexity of the data evolved (Rebala et al., 2019).

This process was executed for the Random Forest algorithm (as the one which presented better results), using a model with a split of 75 – 25 percent, also using the dataset represented in Table 9, similarly to the cross-validation tests yet, in this case, solely 4 months of sales data were used, due to the complexity of calculations performed by the hyperparameter tuning procedure, representing 400 thousand individual rows of data.

Following the discovery of the regression model to employ and the most relevant hyperparameters based on the prior tests, the model may be evaluated. As previously explained (section 3.5.3), the Forecasting model will utilize the history of sales of each product in the campaign basket (previously assembled by the Products Discovery flow) to anticipate how many quantities of each product will be sold over the campaign days.

To test this model, a basket composed by the products from the previously used random campaign was collected (with the same products used for the Price Estimation model evaluation), and all their sales history was then employed to train the model (with about 500.000 rows). Following the pre-processing and feature engineering phases (described in section 3.5.3), the model was trained, and several assessment metrics were derived from it, as shown in Table *13*.

RMSE	MAE	R2	Specifications
48.05	13.61	0.70	Model split: 75 percent training 25 percent testing Training Dataset: Entire sales data for the 235 products (belonging to the randomly chosen campaign)

Table 13 - Forecasting model evaluation measures

Before comparing these values to the cross-validation metrics (in Table 12) for the chosen method (Random Forest), it is important to note that in this case the data volume passed as input to train the model has decreased significantly to approximately 500 thousand rows, and also that the method's hyperparameters were changed to those that better fit the actual data. By examining the above values, it is possible to observe that the RMSE has grown, even though the R2 has also increased. This suggests that, despite having a higher goodness of fit, the error rose. This phenomenon could be caused by several factors such as having a lower volumetry of data, which increases the error, nonetheless, some other factors will be approached further in this chapter.

To better understand the deviations between the predicted and actual values, a Shapley Additive Explanations (SHAP) analysis may be performed. SHAP assigns contributions to each feature in a prediction model to assess their impact on individual predictions. SHAP aids in understanding model decisions and nurturing transparency, which is critical for informed decision-making, by quantifying feature importance and interaction effects.

This research was conducted using the previously mentioned data for the basket of products chosen in earlier evaluations, to understand the amount of impact in the forecasts made by each feature, either to make them higher or lower than the actual values. The SHAP results are shown in the image below, where the color of each dot indicates whether that feature value was high or low for that row of the dataset, and the horizontal location indicates whether the effect of that value led to a greater or lower prediction.

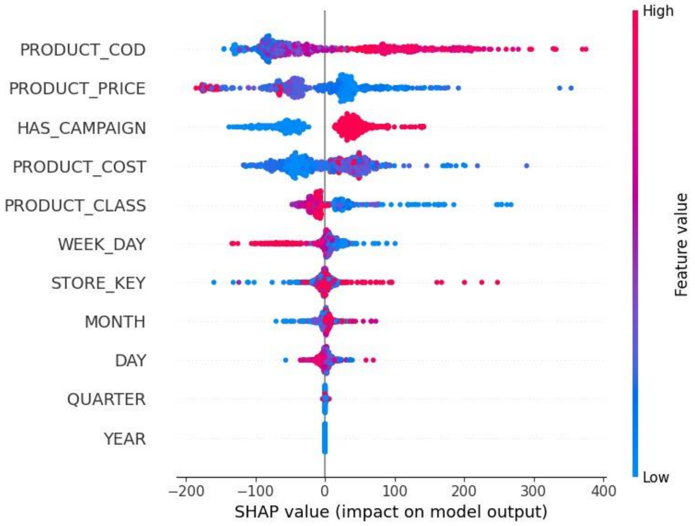
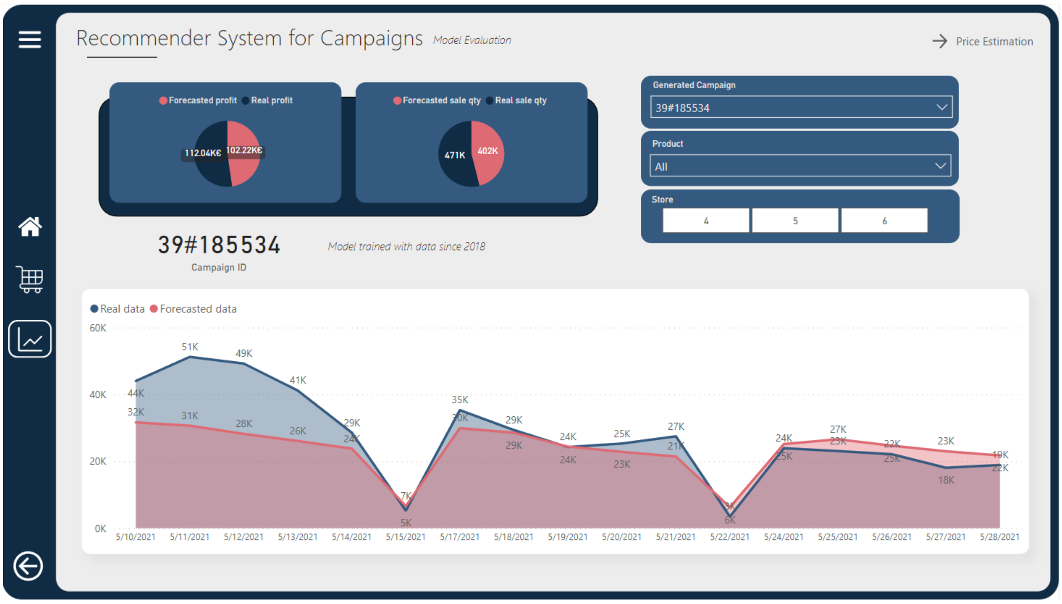


Figure 17 - SHAP analysis

The graphic shows that, as expected, features such as the product, its price and cost, and whether it is in campaign have a significant impact on the forecast. Since the model is trained with the entire sales dataset with a label identifying if the product was or not in campaign at that day, it's comprehensible that the model captures the pattern of increased sales quantities when the product actually is in a campaign, and that's the reason why there is a cluster of red dots (high feature value) on the right side of the vertical line, meaning an high impact on forecasted sale quantities above the real values.

Another important aspect is the product code (textual label, since it may contain letters), which is the product's unique identity and is included so that the forecasting model can understand individual product patterns. In this scenario, this label had a significant impact on forecasts above the real values, showing that certain sales trends from some products may be influencing forecasts for other products, which may result in erroneous results in some cases (later, one approach to overcome this situation is suggested in Chapter 5). Nonetheless, comparing the outcomes of this model to real-world data is another technique to validate them. To do this, the same randomly selected campaign as previously detailed in the Price Estimation evaluation was used, but this time the forecasted number of items sold was compared to the actual amount sold on each day of the campaign, store, and product. This analysis was also created on a dedicated dashboard for the subject, where several filters, like those found on the prior evaluation dashboard, can be used. Although, in this example, the anticipated values are compared to the actual ones on a line chart to better understand not only the model's accuracy, but also how it fits the data trends, which can be critical for the retail sector. If no data filters are applied on the dashboard, the observed quantities sold represent the sum for the total of products and stores, which may not be precise for some assessments; however, by filtering any specific product, category, store, or combination of filters, the data fits into the user's needs. One example of this visualization, regarding the same real campaign, which was approached previously, can be seen in Figure *18*.



Recommender System for Campaigns *Model Evaluation*

Generated Campaign: 39#185534

Product: All

Store: 4 | 5 | 6

39#185534

Campaign ID

Model trained with data since 2018

Figure 18 - Forecasting model's evaluation (dashboard)

By assessing the forecasted and actual values using the line chart, it is possible to determine that there is some fluctuation in the number of products sold in the early days of the campaign. Because the recommender system forecasts the quantity of products purchased at various stores on specific days, it is vulnerable to several external factors that are difficult for the system to comprehend. One of these factors could be the weather, as rainy days make it difficult for customers to go shopping. Other factors that may have an impact on the normal pattern of sales include news in the media or financial problems. These factors are sometimes inaccessible to the system and can explain why some odd behaviors occur.

Nonetheless, the forecasted values followed the true data trend for the rest of the campaign, which is critical in this case, and the variance is not as significant when looking at the overall profit and sale numbers (in the pie charts). The same study but applied to a specific store (rather than evaluating as a whole) can be seen in the Annex (section D), yielding even more precise results not only on a daily view but also on the total profit and sale quantities.

The same analysis but with the first five days removed from the forecast can also be seen in the same Annex section, to determine whether this behavior was due to an issue with the model or an unusual fluctuation of the real values, and from that test it was proven that the forecasted values followed the same trend on the remaining days, leading to a successful outcome.

After evaluating the full recommender system models, the previously indicated research questions (in section 4.1.1) can be evaluated. "*Are the products discovered by the Best Products and Basket Analysis models (Product Discovery process) popular choices in past campaigns?"*, the recall at k measure obtained an 85 percent result in the Products Discovery Evaluation (section 4.1.2), indicating that most of the suggested products corresponded to the most included products in previous campaigns (more popular ones). It is important to note that, as demonstrated in this process section (3.4), the discovered products by the Best Products and Basket Analysis models are chosen based on their profitability (equation [3.1]), which includes prices, costs, and sale quantities, rather than their previous popularity among campaigns (although the number of products to include per category is).

"*Is there any product suggested by the Product Discovery process which was never part of any campaign before?*", also as demonstrated by the Products Discovery evaluation, 100 percent of the suggested basket of 140 products has previously been used in campaigns.

"*Is the estimated price MAE from the Price Estimation model lower than 10 cents?",* the Price Estimation evaluation (section 4.1.3) found that the average MAE was 0.02 euros (or 2 cents), indicating a remarkably low error and a MAE lower than 10 cents.

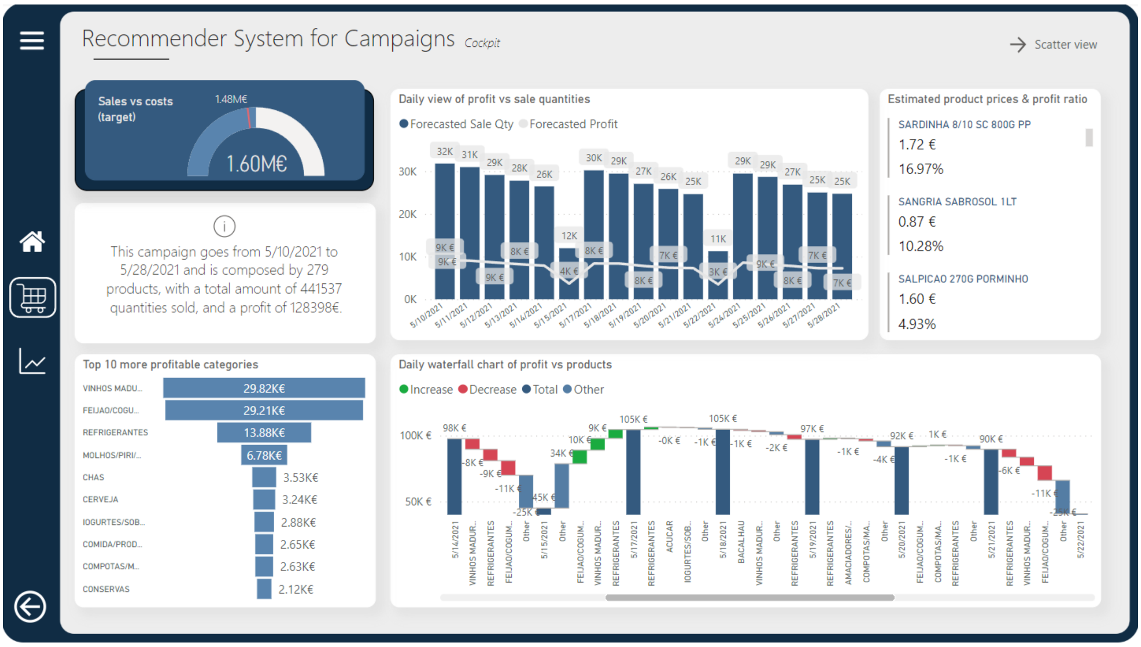
"*Do the forecasted daily sale quantities in each store follow real data patterns and trends?*", when comparing the predicted values with the real ones for a specific past campaign in the forecasting model evaluation, it was possible to observe that the forecasted data follows the real data trend, although in the few initial days of that specific campaign showed some deviation, it did not appear to be a recurrent behavior but rather a single occurrence which was proven in section D of the Annex when the deviation days were not considered on the forecast.

4.2. Reporting

Dashboards, as a visual representation of key performance data, enable users to easily monitor system outputs. Advanced analytics go beyond surface-level interactions to show subtle patterns and preferences in user behavior. This level of information allows for constant algorithmic improvement, matching the recommender system with changing user dynamics. The integration of dashboards and analytics is a compelling requirement; this combination supports adaptive strategies, employing data-driven insights to successfully create and refine user experiences.

As previously stated, MS Power BI (PBI) was the tool of choice for producing reports, with a data model and metrics established in accordance with all modeling best practices (section C of the Annex). Then, using this model, some analyses were built to simplify the process of assessing the Price Estimation and Forecasting models (section 4.1.3). Finally, in the same report, the main analytical tool, termed cockpit, was created, having all the information required for the user to keep track of the generated campaigns and all their information. The lateral buttons control the flow between the cockpit and the evaluation analyses (as can be seen on the figures of section 4.1.3).

This dashboard is split into three sections: the cockpit itself, scatter, and the tabular view. The first is a simple yet effective way to view all essential information and Key Performance Indicators (KPIs). In the second view, a scatter chart displays an overall comparison of the profit rate, revenue, and sale quantities. Ultimately, in the last the generated campaign data is displayed as a matrix for a more direct assessment. Figure *19* illustrates a high-level perspective of the cockpit for a single filtered campaign.



Recommender System for Campaigns *Cockpit*

This campaign goes from 5/10/2021 to 5/28/2021 and is composed by 279 products, with a total amount of 441537 quantities sold, and a profit of 128398€.

Estimated product prices & profit ratio

SARDINHA 8/10 SC 800G PP	1.72 € 16.97%
SANGRIA SABROSOL 1LT	0.87 € 10.28%
SALPICAO 270G PORMINHO	1.60 € 4.93%

Top 10 more profitable categories

VINHOS MADU...	29.82K€
FEIJAO/COGU...	29.21K€
REFRIGERANTES	13.88K€
MOLHOS/PIRI/...	6.78K€
CHAS	3.53K€
CERVEJA	3.24K€
IOGURTES/SOB...	2.88K€
COMIDA/PROD...	2.65K€
COMPOTAS/M...	2.63K€
CONSERVAS	2.12K€

Scatter view
Sales vs costs (target)
1.48M€
1.60M€
Daily view of profit vs sale quantities
● Forecasted Sale Qty ● Forecasted Profit
30K
20K
10K
0K
32K 31K 29K 28K 26K 12K 30K 29K 27K 26K 25K 11K 29K 29K 27K 25K 25K
9K € 9K € 9K € 8K € 4K € 8K € 8K € 7K € 3K € 9K € 8K € 7K € 7K €
5/10/2021 5/11/2021 5/12/2021 5/13/2021 5/14/2021 5/15/2021 5/17/2021 5/18/2021 5/19/2021 5/20/2021 5/21/2021 5/22/2021 5/24/2021 5/25/2021 5/26/2021 5/27/2021 5/28/2021
Daily waterfall chart of profit vs products
● Increase ● Decrease ● Total ● Other
100K €
50K €
98K € -8K € -9K € -11K € -25K € 45K € 34K € 10K € 9K € 105K € -0K € -1K € 105K € -1K € -2K € 97K € -1K € -4K € 92K € 1K € -1K € 90K € -6K € -11K € -25K €
5/14/2021 VINHOS MADUR... REFRIGERANTES FEIJAO/COGUM... Other 5/15/2021 Other FEIJAO/COGUM... VINHOS MADUR... REFRIGERANTES 5/17/2021 REFRIGERANTES ACUCAR IOGURTES/SOB... Other 5/18/2021 BACALHAU VINHOS MADUR... Other REFRIGERANTES 5/19/2021 REFRIGERANTES AMACIADORES/... COMPOTAS/MA... Other 5/20/2021 FEIJAO/COGUM... COMPOTAS/MA... REFRIGERANTES Other 5/21/2021 REFRIGERANTES VINHOS MADUR... FEIJAO/COGUM... Other 5/22/2021

Figure 19 - Cockpit dashboard overview

To begin, not only the cockpit, but all dashboards in this report, have a menu. This menu offers various filters, including the campaign unique identifier (seen in the picture above as being filtered to a specific created campaign), date filters to refine the campaign period, product category, name and code slicers, and store. All these filters can be used, and they are kept in sync throughout all the report's displays (an image of the menu can be seen in section D of the Annex).

Moreover, this dashboard focuses on five key metrics: sales, costs, pricing, profits, and quantities. These measurements are provided in a variety of displays across the cockpit. The top bar chart compares the quantity sold to the profit made on each day of the campaign, while the bottom bar chart (displayed as a waterfall chart) shows the daily profit fluctuations and the three primary product categories responsible for that shift, whether positive or negative. In addition, the left tornado chart depicts the top 10 most profitable product categories, and the right card offers a list of all the products developed during that campaign, along with their estimated price and overall profit ratio. Finally, the top gauge shows the total predicted sales and how they correspond to the campaign costs, while the below card presents the campaign description in plain text in terms of the previously described metrics.

In the second view, an intriguing overall analysis of the campaign outcomes can be examined, providing the viewer with unique insights into the performance of the generated campaign's basket. A scatter chart is provided in this dashboard, comparing the profit ratios (y axis) with the revenues (x axis) of each product (which can be filtered by category through the menu). This analysis is demonstrated in Figure *20*.

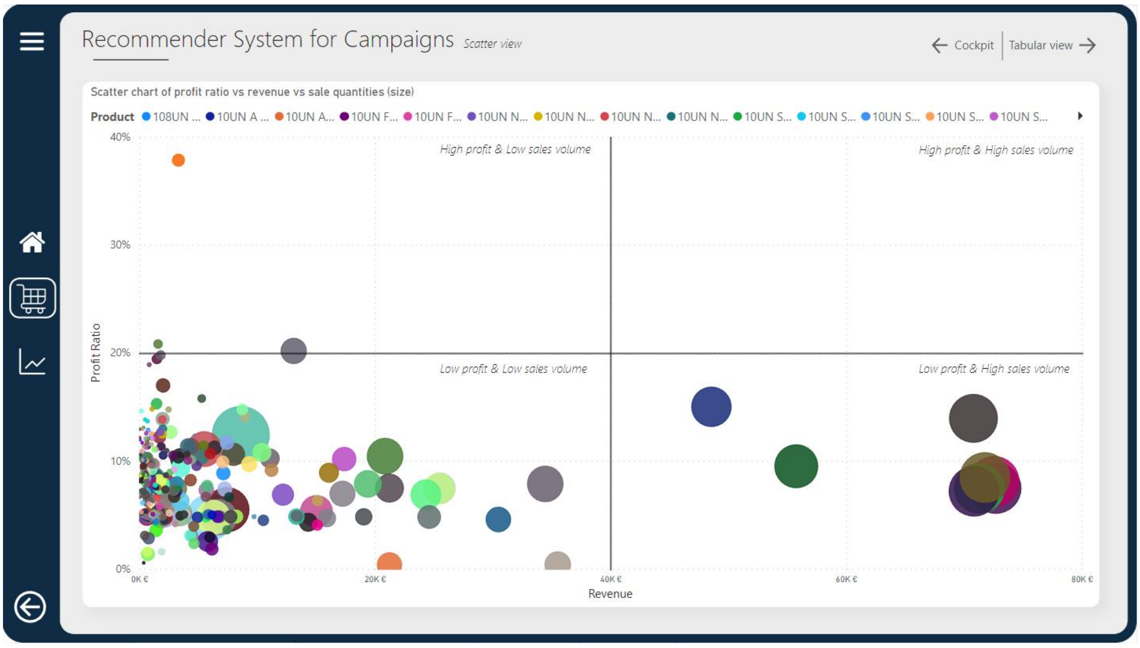
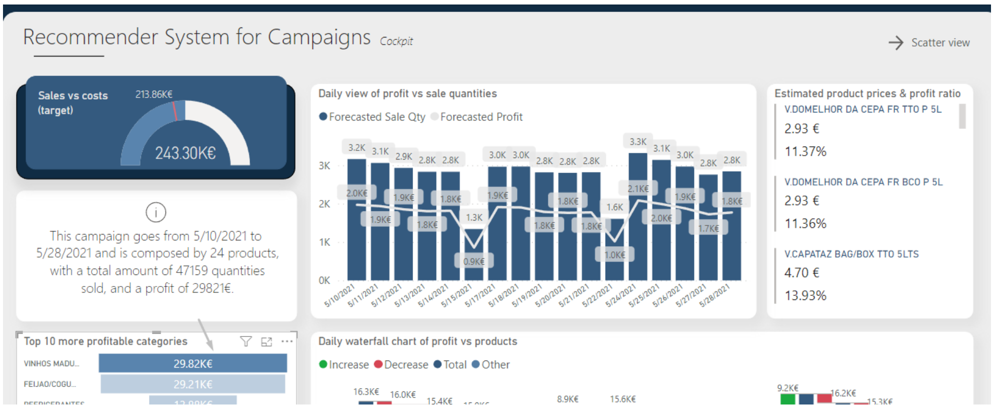

Recommender System for Campaigns
Scatter view
Cockpit
Tabular view
Scatter chart of profit ratio vs revenue vs sale quantities (size)
Product ● 108UN ... ● 10UN A ... ● 10UN A... ● 10UN F... ● 10UN F... ● 10UN N... ● 10UN N... ● 10UN N... ● 10UN N... ● 10UN S... ● 10UN S... ● 10UN S... ● 10UN S... ● 10UN S...
High profit & Low sales volume
High profit & High sales volume
Low profit & Low sales volume
Low profit & High sales volume
Profit Ratio
40%
30%
20%
10%
0%
0K €
20K €
40K €
60K €
80K €
Revenue

Figure 20 - Scatter chart overview

As can be seen, each marker refers to a distinct product, with the size varying in proportion to the number of units sold. This view is also separated into four quadrants to aid in the detection of interesting or irrelevant products, enabling essential decision making. Most products, as is common in the retail industry, have a low profit and a low sales volume, but the amounts sold influence the relevance of marketing such items. Furthermore, products with high profits but low sales volume must be promoted to increase sales quantities and, as a result, revenues. Finally, products with low profits but large sales volume are at the basis of the campaign, and actions may be taken to minimize costs or increase sales volume to increase profitability.

Finally, the third view is a tabular view with a matrix that connects the daily view of the product hierarchy (category, names, and codes) with the previous analyses' primary metrics for a more direct examination of the data (a snapshot of this view can be seen in section D of the Annex).

In addition, some other relevant elements were introduced to facilitate quicker assessments and detail. One example is cross filtering, which is a natural PBI function that has been strengthened by the built data model architecture, allowing interactive filtering across visuals with clicks in specific data points, promoting faster analyses without the need to go to the menu.



Recommender System for Campaigns *Cockpit*

This campaign goes from 5/10/2021 to 5/28/2021 and is composed by 24 products, with a total amount of 47159 quantities sold, and a profit of 29821€.

Estimated product prices & profit ratio

V.DOMELHOR DA CEPA FR TTO P 5L
2.93 €
11.37%

V.DOMELHOR DA CEPA FR BCO P 5L
2.93 €
11.36%

V.CAPATAZ BAG/BOX TTO 5LTS
4.70 €
13.93%

Scatter view
Sales vs costs (target)
213.86K€
243.30K€
Daily view of profit vs sale quantities
● Forecasted Sale Qty ● Forecasted Profit
3K
2K
1K
0K
3.2K 3.1K 2.9K 2.8K 2.8K 1.3K 3.0K 3.0K 2.8K 2.8K 2.8K 1.6K 3.3K 3.1K 3.0K 2.8K 2.8K
2.0K€ 1.9K€ 1.9K€ 1.8K€ 1.8K€ 0.9K€ 1.9K€ 1.8K€ 1.8K€ 1.0K€ 2.1K€ 2.0K€ 1.9K€ 1.7K€ 1.8K€
5/10/2021 5/11/2021 5/12/2021 5/13/2021 5/14/2021 5/15/2021 5/17/2021 5/18/2021 5/19/2021 5/20/2021 5/21/2021 5/22/2021 5/24/2021 5/25/2021 5/26/2021 5/27/2021 5/28/2021
Top 10 more profitable categories
VINHOS MADU...
29.82K€
FEIJAO/COGU...
29.21K€
Daily waterfall chart of profit vs products
● Increase ● Decrease ● Total ● Other
16.3K€ 16.0K€ 15.4K€ 8.9K€ 15.6K€ 9.2K€ 16.2K€ 15.3K€

Figure 21 - Cross filtering

The image above depicts an example of cross filtering, in which by filtering a certain product category (vinhos maduros, shown by an arrow in the lower left corner), all the remaining dashboards are also filtered, displaying data for that specific category.

Furthermore, the drill through functionality was implemented, which allows the user to right click a certain data point (like cross filtering) and then navigate to a tabular view including all data related to that data point, whether it be product categories, names, stores, or campaign days (an image of this analysis can be seen in section D of the Annex).

Finally, features such as exporting the data behind each visual to Excel or exploring the data model through Power Pivot are accessible, allowing the user to share reports and use the recommender system data for external analysis.

5. Conclusions & Future Work

This research investigated the design and implementation of a recommender system built exclusively for the retail sector, with a focus on recommending campaigns based on previous sales patterns and data, hence expediting the campaign creation process. Various aspects of data analytics in retail, campaign management, artificial intelligence methodologies, data mining, and forecasting strategies were examined during this journey.

Handling and analyzing large volumes of data has become a problem for retailers in today's data-driven landscape, and the integration of automated processes validates this effort by incorporating thorough campaign performance metrics simulation. Such simulations give retailers the invaluable ability to predict and assess individual product popularity, associated costs, revenues and, ultimately, campaign profitability, allowing them to fine-tune their strategies in real-time, adapt to market dynamics, and maximize the efficiency and effectiveness of their promotional efforts.

This way, a comprehensive framework for product discovery, pricing estimation, and campaign simulations was successfully built by leveraging the potential of these insights. The evaluation results demonstrated that the models are effective in improving the process of selecting a basket of products to incorporate into a campaign by optimizing product recommendations, in assessing the process of selecting product prices by accurately estimating them based on past patterns, and in easing the process of simulating various detailed information about how that campaign would perform daily per store using forecasting techniques. This study, however, is more than just a conclusion; it is a basic steppingstone toward the improvement of retail automation, providing a solid platform for future research and innovation in this dynamic field.

As this research concludes, it is critical to recognize that the road of improving recommender systems in the retail industry is continuing. Because of the ever-changing nature of the retail industry and the exponential growth of data, ongoing development is required. Further developments in recommender systems, as well as all aspects of retail automation, are anticipated in the future. The refinement and optimization of data mining and AI-powered systems will stay at the forefront of research and innovation. These systems are the foundation of the ability to deal with the intricacies of today's retail world, and their ongoing research and improvement will result in even more precision, efficiency, and effectiveness in directing retail initiatives. As a result, there are several topics that serve as upgrades to the produced campaign recommender system and to make it more automated, flexible, and accurate.

To begin, the Best Products (3.4.1) and Price Estimation (3.5.2) heuristic operations to locate the ideal products and estimate prices may be enhanced. What these models have in common is that they undertake extensive data mining operations on seasonal patterns as a foundation, improving model accuracy by examining them for temporal relationships. As these relationships become shorter, the models' ability to produce better results improves. Currently, the seasonal components of the models focus on quarters; however, utilizing monthly seasonal components might enhance overall accuracy despite increasing the computer power required to execute the calculations. Furthermore, the Products Discovery (3.4) flow might benefit from UI improvement by including a wizard that allows the recommender system user to alter the recommended basket, allowing products to be added or removed before the Campaign Generation section executes, providing the user greater flexibility.

Moreover, the Price Estimation model's (3.5.2) estimation of prices could be enhanced. As of now, this model analyzes past pricing strategies and patterns when products enter campaigns and replicates them seasonally; however, including a price greediness metric (as an input) would increase the recommender system's flexibility by allowing the user to define a low or high greediness when estimating prices, where lower means a more conservative price, and higher means a more ambitious price (based on the original estimated prices).

The regression model (3.5.3) could potentially benefit from some changes, which would undoubtedly improve its accuracy in estimating sales quantities. Because this model predicts with a high level of detail (sale quantity for a product with a price and cost, on a given day in a store), and because it deals with real-world sales data that is susceptible to external factors, any other label that aids in understanding would significantly improve its outputs. Some of them are store closing days, which are easily detectable by the model when they are consistent (always on the same weekday, for example), but when they are punctual (like vacations or local holydays) and differ between stores, they may be difficult for the forecasting model to detect, as feeding this information to the model would resolve the issue. Similarly, access to weather data is critical; wet days are usually not inviting for people to go shopping, and this is especially true in the retail sector, where customers must sometimes carry significant volumes of groceries. So, telling the forecasting model whether the day would have heavy rain or not would undoubtedly improve accuracy, detecting unexpected reductions in sales quantities (on rainy days) or increases in sales quantities (probably in the day prior to the rainy day). Furthermore, observing the Campaign Generation Evaluation section (4.1.3), one thing to notice is that the product code label (unique product identifier) had a significant impact on forecasts above the real values (can be seen by analyzing the SHAP plot in Figure *17*), indicating that certain sales trends from some products may be influencing forecasts for other products, which may result in inaccurate results. One possible explanation for this scenario is that the regression model is trained with all the sales data (with the discovered products as input), and thus training it individually (one model per product) would cluster each product pattern on a closed model, better captioning each product's data. This method would greatly improve the accuracy of the results, but it would require far more computer power.

References

Ahmad, H. W., Zilles, S., Hamilton, H. J., & Dosselmann, R. (2016). Prediction of retail prices of products using local competitors. International Journal of Business Intelligence and Data Mining, 11(1). https://doi.org/10.1504/IJBIDM.2016.076418

Airen, S., & Agrawal, J. (2022). Movie Recommender System Using K-Nearest Neighbors Variants. National Academy Science Letters, 45(1). https://doi.org/10.1007/s40009-021-01051-0

Amornchewin, R. (2018). The Development of SQL Language Skills in Data Definition and Data Manipulation Languages Using Exercises with Quizizz for Students' Learning Engagement. IJIE (Indonesian Journal of Informatics Education), 2(2). https://doi.org/10.20961/ijie.v2i2.24430

Atkinson, L., Müller-Bady, R., & Kappes, M. (2020). Hybrid bayesian evolutionary optimization for hyperparameter tuning. GECCO 2020 Companion - Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. https://doi.org/10.1145/3377929.3389952

Brockwell, P. J., & Davis, R. A. (2002). Introduction to Time Series and Forecasting - Second Edition. In Springer-Verlag.

Cao, L. (2021). Artificial intelligence in retail: applications and value creation logics. International Journal of Retail and Distribution Management, 49(7). https://doi.org/10.1108/IJRDM-09-2020-0350

Chen, Y. L., Tang, K., Shen, R. J., & Hu, Y. H. (2005). Market basket analysis in a multiple store environment. Decision Support Systems, 40(2). https://doi.org/10.1016/j.dss.2004.04.009

Chouksey, S. (2018). STOCK PRICE PREDICTION USING TIME SERIES MODELS.

Chui, M., Francisco, S., Paris, E. H., Washington, S. R., & London, T. A. (2017). Artificial Intelligence the Next Digital Frontier ? McKinsey Global Institute.

Davenport, T. H. (2018). The AI Advantage How to Put the Artificial Intelligence Revolution to Work (Management on the Cutting Edge). Information Research, 24(1).

Davenport, T., & Harris, J. (2007). Competing on Analytics: The New Science of Winning.

Ferrari, A., & Russo, M. (2018). Introducing Microsoft Power BI. In Introducing Microsoft Flow.

Ferreira, K. J., Lee, B. H. A., & Simchi-Levi, D. (2016). Analytics for an online retailer: Demand forecasting and price optimization. Manufacturing and Service Operations Management, 18(1), 69–88. https://doi.org/10.1287/msom.2015.0561

Géron, A. (2017). Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems. In O'Reilly Media.

Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. http://oreilly.com/safari

Grau, G. R. (2017). Market Basket Analysis in Retail. Dept.of Computer Science (UPC), February.

Grewal, D., Ailawadi, K. L., Gauri, D., Hall, K., Kopalle, P., & Robertson, J. R. (2011). Innovations in retail pricing and promotions. Journal of Retailing, 87(SUPPL. 1). https://doi.org/10.1016/j.jretai.2011.04.008

Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques. In Data Mining: Concepts and Techniques. https://doi.org/10.1016/C2009-0-61819-5

Hunt, W., & Rolf, S. (2022). Artificial Intelligence and Automation in Retail - Benefits, challenges and implications (a union perspective).

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. Electronic Markets, 31(3). https://doi.org/10.1007/s12525-021-00475-2

Kallier, S. M. (2017). The influence of Real-time Marketing campaigns of retailers on consumer purchase behavior. International Review of Management and Marketing, 7(3).

Kallier Tar, S. M., & A Wiid, J. (2021). Consumer perceptions of real-time marketing used in campaigns for retail businesses. International Journal of Research in Business and Social Science (2147- 4478), 10(2). https://doi.org/10.20525/ijrbs.v10i2.1075

Kelley, J. (2020). Impact of Artificial Intelligence, Machine Learning, and Automation in Operations Management: an Analysis of Healthcare, Manufacturing, and Retail Sectors. Honors Theses.

Khaydukova, M., Cetó, X., Kirsanov, D., del Valle, M., & Legin, A. (2015). A Tool for General Quality Assessment of Black Tea—Retail Price Prediction by an Electronic Tongue. Food Analytical Methods, 8(5). https://doi.org/10.1007/s12161-014-9979-3

Kimball, R., & Ross, M. (2013). The Data Warehouse Toolkit, The Definitive Guide to Dimensional Modeling. In Wiley.

Krishnamurthi, S., & Indiramma, M. (2021). Sign Language Translator Using Deep Learning Techniques. 2021 4th International Conference on Electrical, Computer and Communication Technologies, ICECCT 2021. https://doi.org/10.1109/ICECCT52121.2021.9616795

Lalou, P., Ponis, S. T., & Efthymiou, O. K. (2020). Demand Forecasting of Retail Sales Using Data Analytics and Statistical Programming. Management and Marketing, 15(2). https://doi.org/10.2478/mmcks-2020-0012

Lekhwar, S., Yadav, S., & Singh, A. (2019). Big data analytics in retail. Smart Innovation, Systems and Technologies, 107. https://doi.org/10.1007/978-981-13-1747-7_45

Lucas, A., Pegios, K., Kotsakis, E., & Clarke, D. (2020). Price forecasting for the balancing energy market using machine-learning regression. Energies, 13(20). https://doi.org/10.3390/en13205420

Molinaro, A. (2013). SQL Cookbook. In Journal of Chemical Information and Modeling (Vol. 53, Issue 9).

Müller, A. C., & Guido, S. (2017a). Introduction to Machine Learning with Python.

Müller, A. C., & Guido, S. (2017b). Introduction to Machine Learning with Python - A Guide for Data Scientists.

Pearson, M., Knight, B., Knight, D., & Quintana, M. (2020). Introduction to Power BI. In Pro Microsoft Power Platform. https://doi.org/10.1007/978-1-4842-6008-1_16

Pradana, M. R., Syafrullah, M., Irawan, H., Chandra, J. C., & Solichin, A. (2022). Market Basket Analysis Using FP-Growth Algorithm on Retail Sales Data. International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2022-October, 86–89. https://doi.org/10.23919/EECSI56542.2022.9946478

Prasad, J. P., & Venkatesham, T. (2021). Big Data Analytics- In Retail Sector. International Journal of Computer Science and Mobile Computing, 10(7). https://doi.org/10.47760/ijcsmc.2021.v10i07.005

Rebala, G., Ravi, A., & Churiwala, S. (2019). An Introduction to Machine Learning. In An Introduction to Machine Learning. Springer.

Rhanoui, M., Yousfi, S., Mikram, M., & Merizak, H. (2019). Forecasting financial budget time series: Arima random walk vs lstm neural network. IAES International Journal of Artificial Intelligence, 8(4). https://doi.org/10.11591/ijai.v8.i4.pp317-327

Ricci, F., Shapira, B., & Rokach, L. (2015). Recommender systems handbook, Second edition. In Recommender Systems Handbook, Second Edition. https://doi.org/10.1007/978-1-4899-7637-6

Rob J Hyndman, & George, A. (2014). Forecasting: Principles and Practice. Principles of Optimal Design, September.

Russell, S. (2019). Book Reviews Human Compatible: Artificial Intelligence and the Problem of Control. In Cato Journal (Vol. 40, Issue 2).

Sagin, A. N., & Ayvaz, B. (2018). Determination of Association Rules with Market Basket Analysis: Application in the Retail Sector. Southeast Europe Journal of Soft Computing, 7(1). https://doi.org/10.21533/scjournal.v7i1.149

Unpingco, J. (2016). Python for probability, statistics, and machine learning. In Python for Probability, Statistics, and Machine Learning. https://doi.org/10.1007/978-3-319-30717-6

Vaisman, A., & Zimányi, E. (2014). Data warehouse systems: Design and implementation. In Data Warehouse Systems: Design and Implementation. https://doi.org/10.1007/978-3-642-54655-6

Venuturumilli, S., Peyyala, P. R., & Alamuri, S. (2016). Evaluating the Impact of Business Intelligence Tools on Organizational Performance in Food and Groceries Retail. Journal of Economics & Business Research, 22(2).

Yee, C. G., Aziz, M. F. A., & Hasan, S. S. (2010). Applying instant business intelligence in marketing campaign automation. 2nd International Conference on Computer Research and Development, ICCRD 2010. https://doi.org/10.1109/ICCRD.2010.180

Ziegel, E. R. (2003). The Elements of Statistical Learning. Technometrics, 45(3). https://doi.org/10.1198/tech.2003.s770

Annex

A. Tools

a. SQL Server database

As described in section 3.1, the data for the study is stored in a SQL Server database. It is prudent to use SQL databases' tools since they give performance and scalability in data management and analytics (Molinaro, 2013).

Aside from that, some data preparation steps must be completed to organize the data that will be used as input to the models. Furthermore, because of the volume of information in the database (approximately 60 million records), it is critical to provide well-structured analytical views to avoid overloading the model with irrelevant data.

As previously noted, some DDL operations will be introduced to do this. This procedure is divided into three major steps:

- Views: developing analytical views that join only the essential data from many tables in the database. Views in SQL Server are objects like tables, with the exception that they do not contain data (they only store metadata) because they are built of a SQL query that produces a certain output, which is executed every time the view is accessed (Molinaro, 2013). The major objective of these views is to avoid unnecessary data from being entered into the model, as well as to keep all necessary data structured in a few items.
- Tables for model outputs: to store all the model's information regarding the generated campaigns, some tables need to be created. These tables will materialize the model's output, consisting in the forecasted prices for each product in the campaign, and the information about the products that should be sold together that is given by the basket analysis model.
- Table for process logs: it's critical to understand what happens when the model runs. To do this, a table holding information on the model's outputs and user interaction will be built, with the goal of keeping when the model was performed, any errors that may occur, and which campaigns were generated (by storing the campaign ID).

b. Python

Python is a popular machine learning programming language due to its ease of use and versatility. It provides a diverse set of libraries that give sophisticated data analysis and model creation tools (Müller & Guido, 2017a).

Because most of the operations in the current job will be analytical and ML related, several specific libraries may be useful owing to the tools they provide and the trust that the global community has in them. One example is Scikit-learn, which is a sophisticated Python machine learning package. It provides a comprehensive set of tools and methodologies for developing intelligent systems, including supervised and unsupervised learning algorithms, preprocessing and feature extraction tools, and assessment measures. This library is constructed on top of NumPy and SciPy, as indicated in the book "Hands-On Machine Learning with Scikit-Learn and TensorFlow," and provides an easy and uniform interface for implementing machine learning algorithms. It also works nicely with other scientific Python tools like matplotlib for data display and pandas for data management (Géron, 2017).

As previously stated, it is intended to create the market basket analysis model and the price forecast model in Python, both of which require communication with the SQL database to collect and write data.

c. User Interface & Analytics

One important step is to provide a way for the user to interact with the model. User engagement with ML models is critical since it enables the input of preferred parameters as well as the observation of results in an orderly and intuitive manner. This allows the user to customize the model to their individual requirements and acquire a better grasp of the model's output. Furthermore, this interaction enables the user to assess the model's performance, detect any flaws or inconsistencies, and make appropriate improvements.

This increases transparency and trust in the model while also giving the user more control over the model's output.

Gradio is a reasonable option for interacting with the model, which is essentially the User Interface (UI) for inputting preferred variables and executing the process. Gradio is a Python library for creating interactive machine learning models that is both user-friendly and efficient. It's simple to integrate with existing code and has a streamlined interface for user inputs (Krishnamurthi & Indiramma, 2021).

On the other hand, the analytical component of the models' outputs is also crucial. One tool that offers plenty of options for data modeling and presentation is Microsoft Power BI (PBI). PBI has been increasing in popularity over the last years due to its ease-of-use tools for conducting dashboards, and for its flawless integration with other Microsoft products (Pearson et al., 2020).

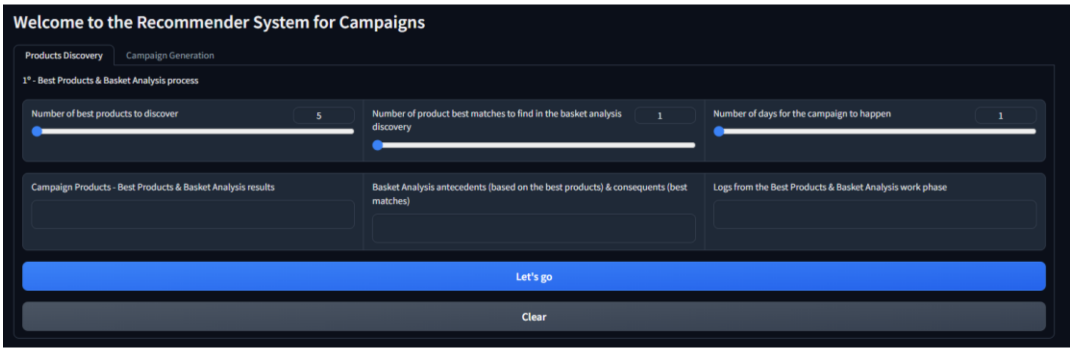
This way, it is expected to employ Gradio's simple yet effective UI to simplify parameter input in the models, as well as PBI's analytical capabilities for generating dashboards using the model's generated campaigns.

B. User Interface

The way a recommendation system interacts with the user is an important aspect to consider. Since the User Interface (UI) serves as the "face" of any system, it should be intuitive, effective, and appealing.

Gradio was used to create the UI for the recommender system in the current solution. This tool, as previously described in subsection c (section A of the Annex), is a Python library that allows you to create a front-end web page that is seamlessly connected to the back-office models which are running in the background.

Using this library, a web page containing all the necessary user inputs for the models as well as a summary of the model's outputs was created, with a focus on usability and functionality. The web page is composed of two main tabs, as the flows represented in section 3.3: Product Discovery and Campaign Generation, as can be seen in Figure 1.


Welcome to the Recommender System for Campaigns

1° - Best Products & Basket Analysis process

Number of best products to discover: 5

Number of product best matches to find in the basket analysis discovery: 1

Number of days for the campaign to happen: 1

Campaign Products - Best Products & Basket Analysis results

Basket Analysis antecedents (based on the best products) & consequents (best matches)

Logs from the Best Products & Basket Analysis work phase

Figure 1 - UI initial page

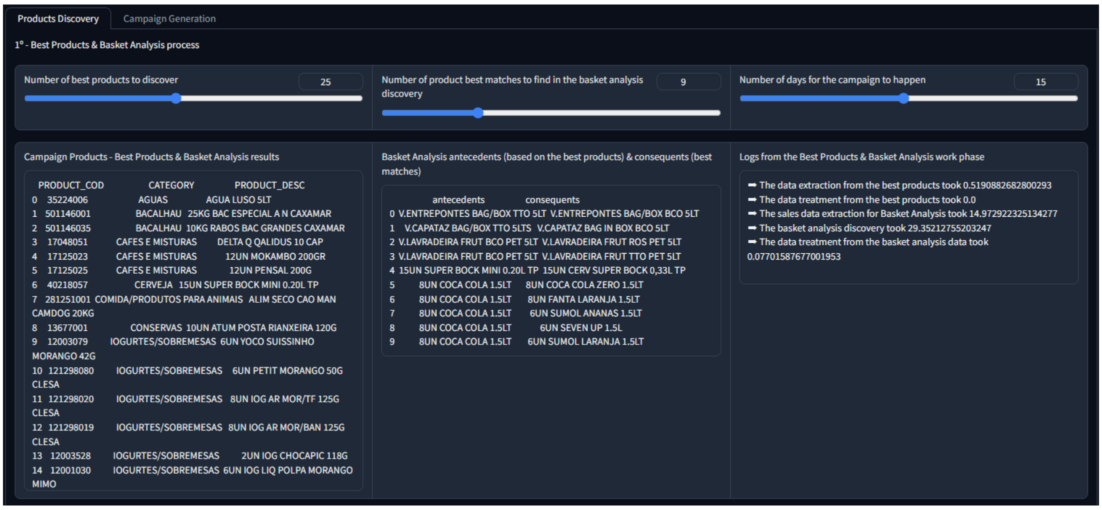
In the first tab, all the necessary user inputs are displayed:

- Number of best products to discover: represented as a slicer with a range of 1 to 50. This input will be sent to the Best Products model, which will determine how many best products should be included in the campaign (considering the ratios between most profitable products per product category, as explained in section 3.4.1).

- Number of product best matches to find in the basket analysis discovery: also shown as a slicer ranging from 1 to 30. Similarly, the Basket Analysis model receives this input, which represents the number of best matches to consider based on the lift ratio of matches (as described in section 3.4.2).
- Number of days for the campaign to happen: also shown as a slicer ranging from 1 to 30. This input will indicate to various models the total duration of the generated campaign. Although this information is more important in the Campaign Generation section, it is also critical in the Products Discovery section, where products with insufficient stock for the duration of the campaign will be discarded and not used in the entire process.

When these components are filled, the Products Discovery flow can be launched by pressing the execute button. Following execution, three prompts are displayed containing the model's output information, such as the Best Products (section 3.4.1) that were discovered along with their category, the Basket Analysis (section 3.4.2) best matching products along with each antecedent, and finally some execution logs focusing on each sub-process duration.

An image containing the already filled user inputs, as well as the outputs of the Products Discovery can be observed in Figure 2.



1º - Best Products & Basket Analysis process

Number of best products to discover: 25

Number of product best matches to find in the basket analysis discovery: 9

Number of days for the campaign to happen: 15

Campaign Products - Best Products & Basket Analysis results

```
  PRODUCT_COD          CATEGORY          PRODUCT_DESC
0   35224006            AGUAS          AGUA LUSO 5LT
1  501146001          BACALHAU  25KG BAC ESPECIAL A N CAXAMAR
2  501146035          BACALHAU  10KG RABOS BAC GRANDES CAXAMAR
3   17048051       CAFES E MISTURAS      DELTA Q QALIDUS 10 CAP
4   17125023       CAFES E MISTURAS         12UN MOKAMBO 200GR
5   17125025       CAFES E MISTURAS          12UN PENSAL 200G
6   40218057           CERVEJA  15UN SUPER BOCK MINI 0.20L TP
7  281251001 COMIDA/PRODUTOS PARA ANIMAIS  ALIM SECO CAO MAN
CAMDOG 20KG
8   13677001          CONSERVAS  10UN ATUM POSTA RIANXEIRA 120G
9   12003079      IOGURTES/SOBREMESAS  6UN YOCO SUISSINHO
MORANGO 42G
10  121298080      IOGURTES/SOBREMESAS   6UN PETIT MORANGO 50G
CLESA
11  121298020      IOGURTES/SOBREMESAS   8UN IOG AR MOR/TF 125G
CLESA
12  121298019      IOGURTES/SOBREMESAS  8UN IOG AR MOR/BAN 125G
CLESA
13   12003528      IOGURTES/SOBREMESAS       2UN IOG CHOCAPIC 118G
14   12001030      IOGURTES/SOBREMESAS  6UN IOG LIQ POLPA MORANGO
MIMO
```

Basket Analysis antecedents (based on the best products) & consequents (best matches)

```
             antecedents             consequents
0 V.ENTREPONTES BAG/BOX TTO 5LT  V.ENTREPONTES BAG/BOX BCO 5LT
1    V.CAPATAZ BAG/BOX TTO 5LTS   V.CAPATAZ BAG IN BOX BCO 5LT
2 V.LAVRADEIRA FRUT BCO PET 5LT  V.LAVRADEIRA FRUT ROS PET 5LT
3 V.LAVRADEIRA FRUT BCO PET 5LT  V.LAVRADEIRA FRUT TTO PET 5LT
4 15UN SUPER BOCK MINI 0.20L TP  15UN CERV SUPER BOCK 0,33L TP
5        8UN COCA COLA 1.5LT       8UN COCA COLA ZERO 1.5LT
6        8UN COCA COLA 1.5LT       8UN FANTA LARANJA 1.5LT
7        8UN COCA COLA 1.5LT        6UN SUMOL ANANAS 1.5LT
8        8UN COCA COLA 1.5LT           6UN SEVEN UP 1.5L
9        8UN COCA COLA 1.5LT       6UN SUMOL LARANJA 1.5LT
```

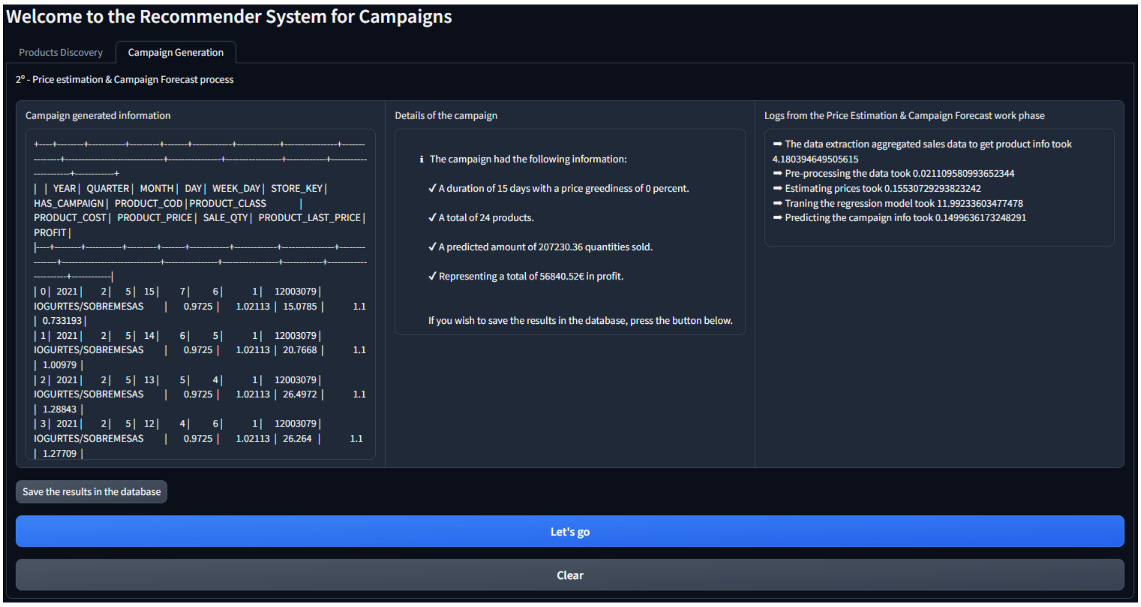
Logs from the Best Products & Basket Analysis work phase

```
➡ The data extraction from the best products took 0.5190882682800293
➡ The data treatment from the best products took 0.0
➡ The sales data extraction for Basket Analysis took 14.972922325134277
➡ The basket analysis discovery took 29.35212755203247
➡ The data treatment from the basket analysis data took
0.07701587677001953
```

Figure 2 - UI for the Products Discovery section

After carefully reviewing the output of this phase, the user can alter any of the existing sliders and re-run the models to reach the desired result.

Once the user is satisfied with the basket for the current campaign, the next step is to generate it. This next phase refers to the process of estimating the product prices based on past behaviours (section 3.5.2) and to generate each day of the campaign through the existing company's stores, by forecasting the amount sold per product (section 3.5.3). With this information, measures such as cost, quantity sold, sales, and profits may be estimated, allowing the user to determine whether the campaign is appropriate for the current time, products, and stocks, or whether changing the basket, duration, or timing would be better strategies. This can be done in the second tab, "Campaign Generation," without providing any further information. Figure 3 shows an example of the results of this section.



Welcome to the Recommender System for Campaigns
Products Discovery
Campaign Generation
2º - Price estimation & Campaign Forecast process
Campaign generated information
Details of the campaign
Logs from the Price Estimation & Campaign Forecast work phase
Save the results in the database
Let's go
Clear

```
+---+-------+---------+--------+------+----------+-----------+-------------+------
-------+-----------------------+-------------+--------------+----------+----------
----------+----------+
|   | YEAR | QUARTER | MONTH | DAY | WEEK_DAY | STORE_KEY |
HAS_CAMPAIGN | PRODUCT_COD | PRODUCT_CLASS          |
PRODUCT_COST | PRODUCT_PRICE | SALE_QTY | PRODUCT_LAST_PRICE |
PROFIT |
|---+-------+---------+--------+------+----------+-----------+-------------+------
-------+-----------------------+-------------+--------------+----------+----------
----------+----------|
| 0 | 2021 |    2 |    5 |  15 |     7 |      6 |        1 |  12003079 |
IOGURTES/SOBREMESAS    |    0.9725 |    1.02113 | 15.0785 |       1.1
|  0.733193 |
| 1 | 2021 |    2 |    5 |  14 |     6 |      5 |        1 |  12003079 |
IOGURTES/SOBREMESAS    |    0.9725 |    1.02113 | 20.7668 |       1.1
|  1.00979 |
| 2 | 2021 |    2 |    5 |  13 |     5 |      4 |        1 |  12003079 |
IOGURTES/SOBREMESAS    |    0.9725 |    1.02113 | 26.4972 |       1.1
|  1.28843 |
| 3 | 2021 |    2 |    5 |  12 |     4 |      6 |        1 |  12003079 |
IOGURTES/SOBREMESAS    |    0.9725 |    1.02113 | 26.264  |       1.1
|  1.27709 |
```

ℹ The campaign had the following information:

✓ A duration of 15 days with a price greediness of 0 percent.

✓ A total of 24 products.

✓ A predicted amount of 207230.36 quantities sold.

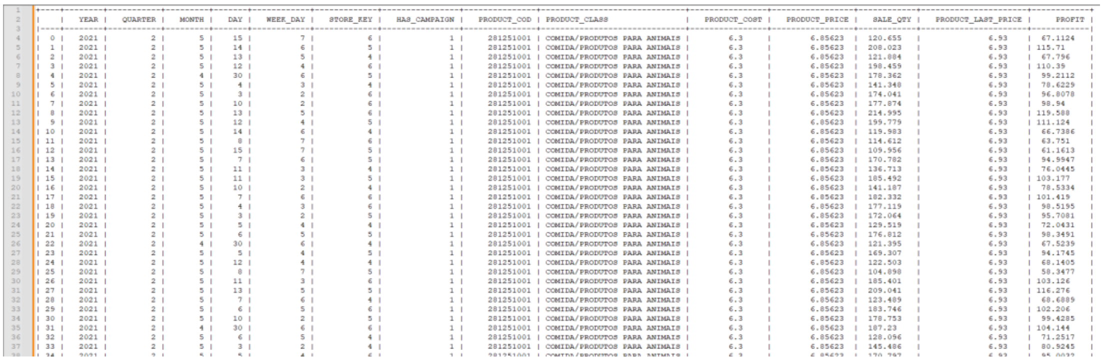
✓ Representing a total of 56840.52€ in profit.

If you wish to save the results in the database, press the button below.

```
➡ The data extraction aggregated sales data to get product info took
4.180394649505615
➡ Pre-processing the data took 0.021109580993652344
➡ Estimating prices took 0.15530729293823242
➡ Traning the regression model took 11.99233603477478
➡ Predicting the campaign info took 0.1499636173248291
```

Figure 3 - UI for the Campaign Generation section

After the campaign is generated, the recommender system's final output is fulfilled. To determine whether the campaign meets the user's needs, three output windows are displayed, the first of which contains the entire content of the campaign with a data granularity of one row per campaign day, store, and product, displaying the estimated price, forecasted amount sold, cost, and profit (despite the content seem unreadable, the intention is for the user to copy and paste it into a notepad or word document, as can be seen in the Figure 4).



```
+----+------+---------+-------+-----+----------+-----------+--------------+-------------+------------------------------+--------------+---------------+----------+--------------------+----------+
|    | YEAR | QUARTER | MONTH | DAY | WEEK_DAY | STORE_KEY | HAS_CAMPAIGN | PRODUCT_COD | PRODUCT_CLASS                | PRODUCT_COST | PRODUCT_PRICE | SALE_QTY | PRODUCT_LAST_PRICE |   PROFIT |
|----+------+---------+-------+-----+----------+-----------+--------------+-------------+------------------------------+--------------+---------------+----------+--------------------+----------|
|  0 | 2021 |       2 |     5 |  15 |        7 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 120.655  |               6.93 |  67.1124 |
|  1 | 2021 |       2 |     5 |  14 |        6 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 208.023  |               6.93 | 115.71   |
|  2 | 2021 |       2 |     5 |  13 |        5 |         4 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 121.884  |               6.93 |  67.796  |
|  3 | 2021 |       2 |     5 |  12 |        4 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 198.459  |               6.93 | 110.39   |
|  4 | 2021 |       2 |     4 |  30 |        6 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 178.362  |               6.93 |  99.2112 |
|  5 | 2021 |       2 |     5 |   4 |        3 |         4 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 141.348  |               6.93 |  78.6229 |
|  6 | 2021 |       2 |     5 |   3 |        2 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 174.041  |               6.93 |  96.8078 |
|  7 | 2021 |       2 |     5 |  10 |        2 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 177.874  |               6.93 |  98.94   |
|  8 | 2021 |       2 |     5 |  13 |        5 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 214.995  |               6.93 | 119.588  |
|  9 | 2021 |       2 |     5 |  12 |        4 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 199.779  |               6.93 | 111.124  |
| 10 | 2021 |       2 |     5 |  14 |        6 |         4 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 119.983  |               6.93 |  66.7386 |
| 11 | 2021 |       2 |     5 |   8 |        7 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 114.612  |               6.93 |  63.751  |
| 12 | 2021 |       2 |     5 |  15 |        7 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 109.956  |               6.93 |  61.1613 |
| 13 | 2021 |       2 |     5 |   7 |        6 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 170.782  |               6.93 |  94.9947 |
| 14 | 2021 |       2 |     5 |  11 |        3 |         4 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 136.713  |               6.93 |  76.0445 |
| 15 | 2021 |       2 |     5 |  11 |        3 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 185.492  |               6.93 | 103.177  |
| 16 | 2021 |       2 |     5 |  10 |        2 |         4 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 141.187  |               6.93 |  78.5334 |
| 17 | 2021 |       2 |     5 |   7 |        6 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 182.332  |               6.93 | 101.419  |
| 18 | 2021 |       2 |     5 |   4 |        3 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 177.119  |               6.93 |  98.5195 |
| 19 | 2021 |       2 |     5 |   3 |        2 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 172.064  |               6.93 |  95.7081 |
| 20 | 2021 |       2 |     5 |   5 |        4 |         4 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 129.519  |               6.93 |  72.0431 |
| 21 | 2021 |       2 |     5 |   6 |        5 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 176.812  |               6.93 |  98.3491 |
| 22 | 2021 |       2 |     4 |  30 |        6 |         4 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 121.395  |               6.93 |  67.5239 |
| 23 | 2021 |       2 |     5 |   5 |        4 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 169.307  |               6.93 |  94.1745 |
| 24 | 2021 |       2 |     5 |  12 |        4 |         4 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 122.503  |               6.93 |  68.1405 |
| 25 | 2021 |       2 |     5 |   8 |        7 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 104.898  |               6.93 |  58.3477 |
| 26 | 2021 |       2 |     5 |  11 |        3 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 185.401  |               6.93 | 103.126  |
| 27 | 2021 |       2 |     5 |  13 |        5 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 209.041  |               6.93 | 116.276  |
| 28 | 2021 |       2 |     5 |   7 |        6 |         4 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 123.489  |               6.93 |  68.6889 |
| 29 | 2021 |       2 |     5 |   6 |        5 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 183.746  |               6.93 | 102.206  |
| 30 | 2021 |       2 |     5 |  10 |        2 |         5 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 178.753  |               6.93 |  99.4285 |
| 31 | 2021 |       2 |     4 |  30 |        6 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 187.23   |               6.93 | 104.144  |
| 32 | 2021 |       2 |     5 |   6 |        5 |         4 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 128.096  |               6.93 |  71.2517 |
| 33 | 2021 |       2 |     5 |   3 |        2 |         4 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 145.486  |               6.93 |  80.9245 |
| 34 | 2021 |       2 |     5 |   5 |        4 |         6 |            1 |   281251001 | COMIDA/PRODUTOS PARA ANIMAIS |     6.3      |       6.85623 | 170.797  |               6.93 |  95.0032 |
```

Figure 4 - Price Estimation Output (when copied to an external document)

Following, the second window displays the campaign details, representing the most important information in this section. This data refers to the campaign as a whole and not to individual products, and it indicates the user, for the predefined duration, discovered products and estimated prices, how much profit will it achieve and how many quantities will be sold. Lastly, the third window displays the execution logs regarding the model's executions. Finally, if the results meet the company's standards, the user can save them in the database by pressing the corresponding button (displayed in the figure). Once pressed, the data is saved in the recommender system database and can be evaluated, as shown in the following sections.

C. MS Power BI

As the need to thoroughly study the recommender's outputs grows, a tool to facilitate this process is required. In this regard, Microsoft Power BI (PBI) is a safe choice due to its well-implemented data integration and modeling capabilities, as well as a large range of configurable representations that may be designed.

This tool was used to build the system's data model, which would then be used in the Model Evaluation (discussed in the Models Evaluation section 4.1) and campaign results visualization (addressed in Reporting section 4.2). In PBI, the next stage is to develop the data model after defining the source and proceeding with the data extraction. To ensure performance and scalability, model building should conform to several modeling criteria, particularly when dealing with enormous amounts of data.

A PBI model, which is a tabular model, separates data into tables that are linked together through relationships, and this type of model guarantees rapid query results by utilizing capabilities such as column and in-memory storage (Ferrari & Russo, 2018).

However, the model's efficiency is determined by how tables are produced, and relationships are built. As previously discussed in section 3.2, tables in a common Data Warehouse (DW) are defined as dimensions and facts, with dimensions often storing categorical information and facts numerical information. Furthermore, the relationships should always move from dimensions to facts, never from facts to facts. Dimensions can be tied to each other in some circumstances (Snowflake Schema), however the model with the maximum effectiveness does not contain dimensions connected to each other (Star Schema). Table relationships should also be one-to-many, which means that one value from one table corresponds to numerous values from the related one, with dimensions on the "one" side and facts on the "many" side of the relationships. This ensures that the best standards in tabular modeling are followed, resulting in an effective and scalable model (Kimball & Ross, 2013).

This way, every query produced through the model (made by applying any filter on the reporting component, or even by showing any data on a simple chart) flows from dimensions to facts, guaranteeing more rapid responses (Vaisman & Zimányi, 2014).

The figure below depicts the data model produced using PBI, which contains the generated campaigns created through the Recommender System's UI (section B of the Annex), as well as a table with real data from prior campaigns, allowing the recommender's outcomes to be compared and evaluated.

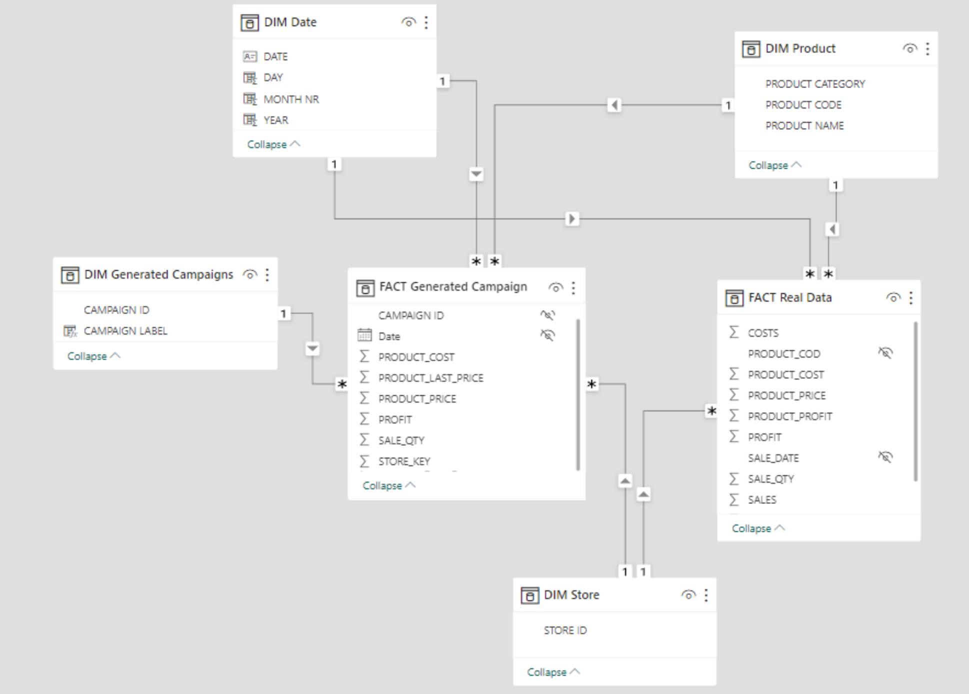


Figure 5 - Recommender System’s data model in PBI

The previously described practices can be evaluated using this paradigm. For starters, it is a Star Schema since no dimension tables (tables with the prefix DIM) are associated with one another. Second, no FACT table (tables with the prefix FACT) is related to one another. Furthermore, the model is made up of four dimension tables that contain categorical information about all of the dates in the dataset, products, generated campaigns, and existing stores, as well as two fact tables that contain numerical information about the generated campaigns (basket of products per campaign created along with forecasted daily quantities sold, profits, and costs) and real past campaigns information (to be used for model evaluations).

D. Evaluation and Results

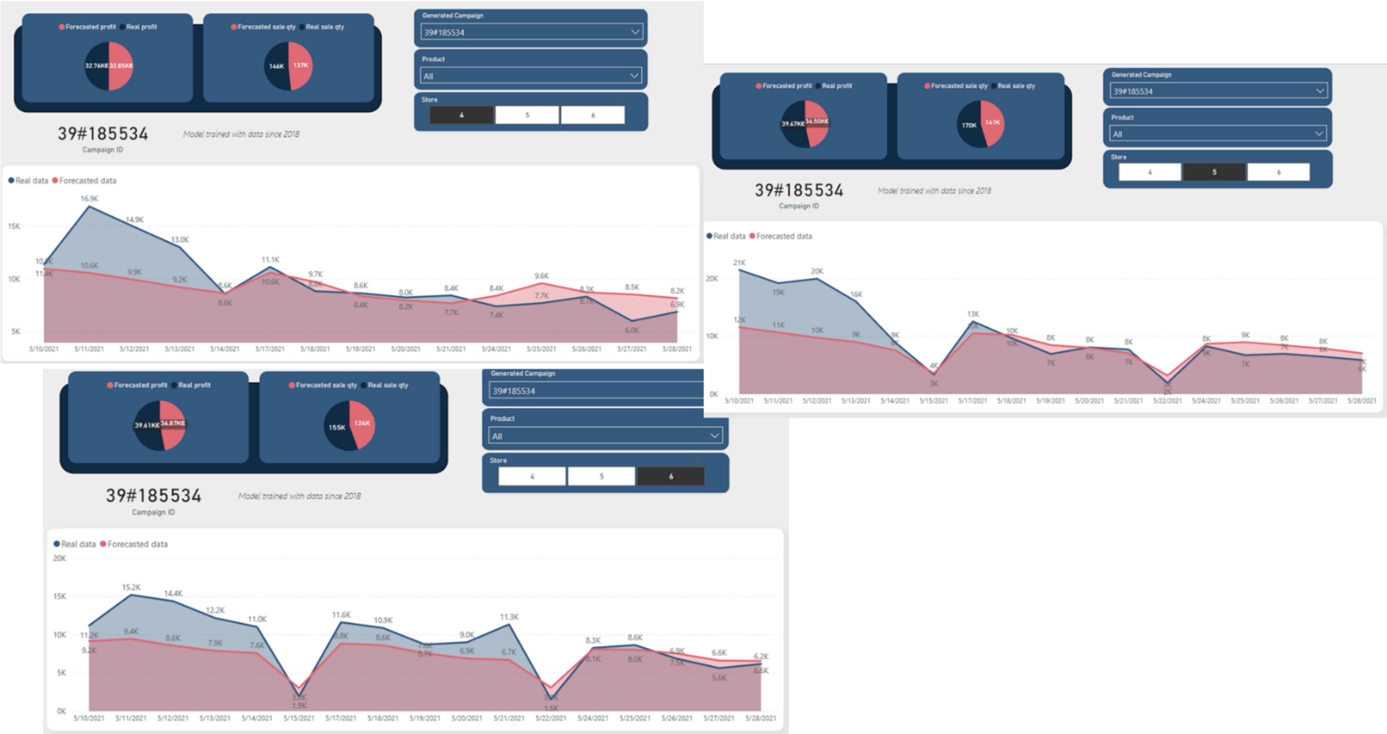
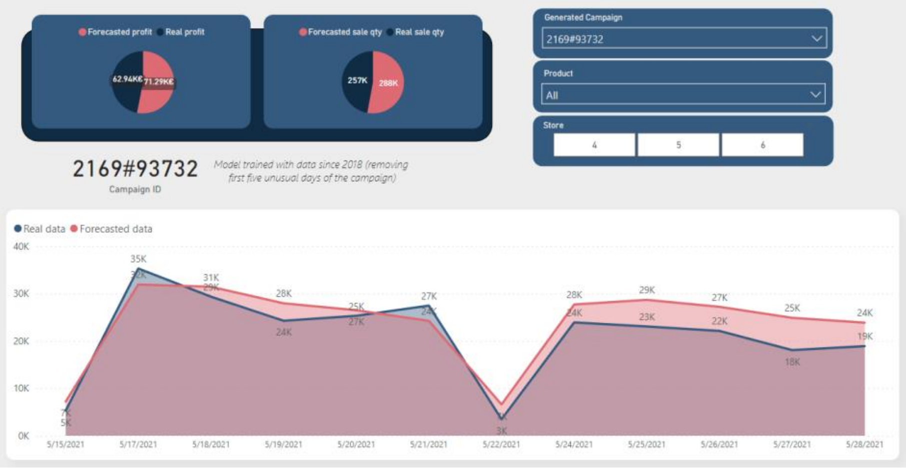


Figure 6 - Analysis specific stores (section 4.1.3)



Generated Campaign: 2169#93732

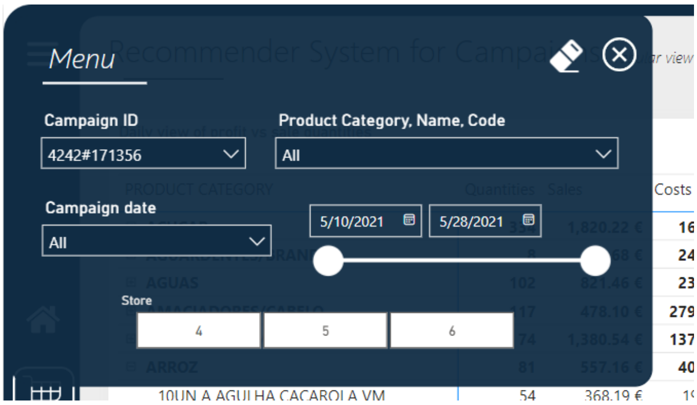
Product: All

Store: 4 5 6

2169#93732
Campaign ID

Model trained with data since 2018 (removing first five unusual days of the campaign)

Figure 7 - Tests without the first unusual 5 days (section 4.1.3)



Campaign ID: 4242#171356

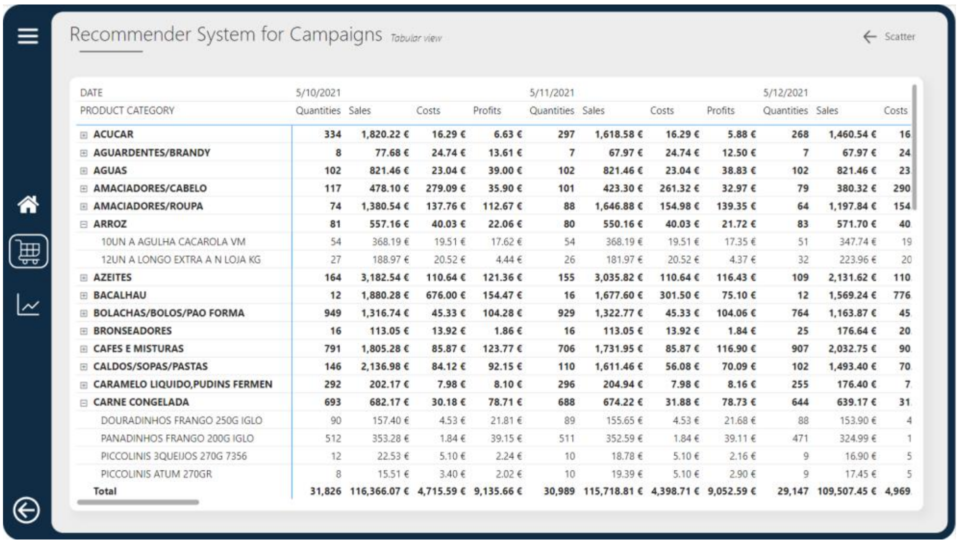
Product Category, Name, Code: All

Campaign date: All

5/10/2021 – 5/28/2021

Store: 4 5 6

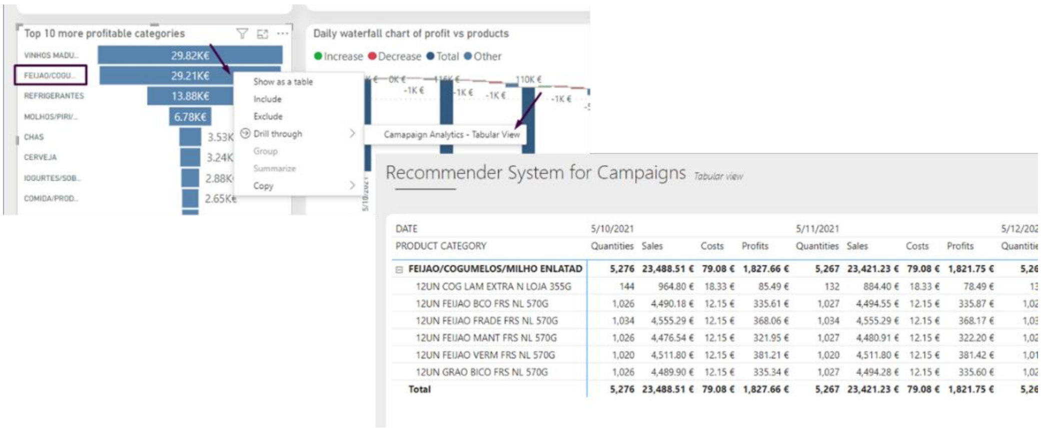
Figure 8 – Dashboard's menu (section 4.2)



Recommender System for Campaigns *Tabular view*

DATE	5/10/2021				5/11/2021				5/12/2021		
PRODUCT CATEGORY	Quantities	Sales	Costs	Profits	Quantities	Sales	Costs	Profits	Quantities	Sales	Costs
⊞ ACUCAR	334	1,820.22 €	16.29 €	6.63 €	297	1,618.58 €	16.29 €	5.88 €	268	1,460.54 €	16
⊞ AGUARDENTES/BRANDY	8	77.68 €	24.74 €	13.61 €	7	67.97 €	24.74 €	12.50 €	7	67.97 €	24
⊞ AGUAS	102	821.46 €	23.04 €	39.00 €	102	821.46 €	23.04 €	38.83 €	102	821.46 €	23
⊞ AMACIADORES/CABELO	117	478.10 €	279.09 €	35.90 €	101	423.30 €	261.32 €	32.97 €	79	380.32 €	290
⊞ AMACIADORES/ROUPA	74	1,380.54 €	137.76 €	112.67 €	88	1,646.88 €	154.98 €	139.35 €	64	1,197.84 €	154
⊟ ARROZ	81	557.16 €	40.03 €	22.06 €	80	550.16 €	40.03 €	21.72 €	83	571.70 €	40
10UN A AGULHA CACAROLA VM	54	368.19 €	19.51 €	17.62 €	54	368.19 €	19.51 €	17.35 €	51	347.74 €	19
12UN A LONGO EXTRA A N LOJA KG	27	188.97 €	20.52 €	4.44 €	26	181.97 €	20.52 €	4.37 €	32	223.96 €	20
⊞ AZEITES	164	3,182.54 €	110.64 €	121.36 €	155	3,035.82 €	110.64 €	116.43 €	109	2,131.62 €	110
⊞ BACALHAU	12	1,880.28 €	676.00 €	154.47 €	16	1,677.60 €	301.50 €	75.10 €	12	1,569.24 €	776
⊞ BOLACHAS/BOLOS/PAO FORMA	949	1,316.74 €	45.33 €	104.28 €	929	1,322.77 €	45.33 €	104.06 €	764	1,163.87 €	45
⊞ BRONSEADORES	16	113.05 €	13.92 €	1.86 €	16	113.05 €	13.92 €	1.84 €	25	176.64 €	20
⊞ CAFES E MISTURAS	791	1,805.28 €	85.87 €	123.77 €	706	1,731.95 €	85.87 €	116.90 €	907	2,032.75 €	90
⊞ CALDOS/SOPAS/PASTAS	146	2,136.98 €	84.12 €	92.15 €	110	1,611.46 €	56.08 €	70.09 €	102	1,493.40 €	70
⊞ CARAMELO LIQUIDO,PUDINS FERMEN	292	202.17 €	7.98 €	8.10 €	296	204.94 €	7.98 €	8.16 €	255	176.40 €	7
⊟ CARNE CONGELADA	693	682.17 €	30.18 €	78.71 €	688	674.22 €	31.88 €	78.73 €	644	639.17 €	31
DOURADINHOS FRANGO 250G IGLO	90	157.40 €	4.53 €	21.81 €	89	155.65 €	4.53 €	21.68 €	88	153.90 €	4
PANADINHOS FRANGO 200G IGLO	512	353.28 €	1.84 €	39.15 €	511	352.59 €	1.84 €	39.11 €	471	324.99 €	1
PICCOLINIS 3QUEIJOS 270G 7356	12	22.53 €	5.10 €	2.24 €	10	18.78 €	5.10 €	2.16 €	9	16.90 €	5
PICCOLINIS ATUM 270GR	8	15.51 €	3.40 €	2.02 €	10	19.39 €	5.10 €	2.90 €	9	17.45 €	5
Total	31,826	116,366.07 €	4,715.59 €	9,135.66 €	30,989	115,718.81 €	4,398.71 €	9,052.59 €	29,147	109,507.45 €	4,969

Figure 9 – Tabular view (section 4.2)



Recommender System for Campaigns *Tabular view*

DATE	5/10/2021				5/11/2021				5/12/202
PRODUCT CATEGORY	Quantities	Sales	Costs	Profits	Quantities	Sales	Costs	Profits	Quantitie
⊟ **FEIJAO/COGUMELOS/MILHO ENLATAD**	**5,276**	**23,488.51 €**	**79.08 €**	**1,827.66 €**	**5,267**	**23,421.23 €**	**79.08 €**	**1,821.75 €**	**5,26**
12UN COG LAM EXTRA N LOJA 355G	144	964.80 €	18.33 €	85.49 €	132	884.40 €	18.33 €	78.49 €	13
12UN FEIJAO BCO FRS NL 570G	1,026	4,490.18 €	12.15 €	335.61 €	1,027	4,494.55 €	12.15 €	335.87 €	1,02
12UN FEIJAO FRADE FRS NL 570G	1,034	4,555.29 €	12.15 €	368.06 €	1,034	4,555.29 €	12.15 €	368.17 €	1,03
12UN FEIJAO MANT FRS NL 570G	1,026	4,476.54 €	12.15 €	321.95 €	1,027	4,480.91 €	12.15 €	322.20 €	1,02
12UN FEIJAO VERM FRS NL 570G	1,020	4,511.80 €	12.15 €	381.21 €	1,020	4,511.80 €	12.15 €	381.42 €	1,01
12UN GRAO BICO FRS NL 570G	1,026	4,489.90 €	12.15 €	335.34 €	1,027	4,494.28 €	12.15 €	335.60 €	1,02
Total	**5,276**	**23,488.51 €**	**79.08 €**	**1,827.66 €**	**5,267**	**23,421.23 €**	**79.08 €**	**1,821.75 €**	**5,26**

Figure 10 – Drill through (section 4.2)