STORE LAYOUT OPTIMIZATION FOR A LUXURY FASHION RETAILER

MIRI MSc Thesis

Student: OLGA FETISOVA
Supervisor: ALFREDO VELLIDO
Advisor: SERGI MARIN

Master in Innovation and Research in Informatics - Data Science
Universitat Politècnica de Catalunya

Barcelona
18 June, 2018

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

A thesis presented by OLGA FETISOVA
in partial fulfillment of the requirements for the MSc degree on
Innovation and Research in Informatics

Acknowledgements

I would like to express my gratitude towards Clariba company where I had the opportunity to develop my Master's Thesis. Special thanks to my manager Lluis Aspachs and my advisor Sergi Marin for their help in sharing the knowledge and providing support.

For the completion of this work it was also very important the consistent feedback given by my supervisor and former-professor, Alfredo Vellido. I am grateful for his encouraging words and professional attitude towards my project.

I would like to thank my family for the endless support provided all the way during my studies. Special thanks to Ward Taya for his love, patience and enormous assistance that helped me to achieve better results on my work.

Abstract

The store layout and the customer interaction with the items in the store are the crucial factors to business effectiveness in a luxury jewelry industry. Understanding shopping behavior and knowing which items are the most important for the customer are essential for the success of the retailer. This paper focuses on an experimental investigation of the human behavior within the store, in which it examines how this behavior helps to adjust a layout that optimizes sales. We are exploring the possibility of applying machine learning techniques to predict customers' behavior and influence their choice. Firstly, we designed and implemented ETL process that includes image processing, web scraping, and other data preparation techniques. Then, we extracted the necessary features, followed by pattern mining, clustering, and similarity matrix calculation using NLP. Lastly, we proposed an algorithm based on Decoy theory to optimize the store design layout of Bulgari, Dubai. To the extent of our knowledge, the novelty of the research concludes in providing a solution for building customer behavior patterns based on extracted data from raw filmed heatmap photos taken from Prism Skylabs. Besides that, we combine Decoy theory with user behavior patterns and design principles, while optimizing sales and increasing the overall average value of the receipt. The results of our work aim at increasing the store performance and improving its profitability.

Contents

1 Introduction **1**

1.1 Retailer layout 2

1.2 Luxury fashion industry 4

1.3 Problem Statement 5

2 Background **7**

2.1 Store Design 7

2.2 Tracking Users Behavior 10

2.3 Data Sources 14

2.4 Challenges 15

2.5 Goals and Objectives 16

3 Related Work **18**

3.1 Store Layout Design Techniques 18

3.2 Data mining techniques for retailing 20

3.2.1 Clustering 21
3.2.2 Similarity for Item-based Recommenders 23
3.3 Data Mining tools and approaches 24

4 Proposed Solution **27**

4.1 Initial Assumptions 29
4.2 Data Set Compilation 30
4.2.1 Storing data in the database 30
4.2.2 Completion of the datasets with further available information 30
4.3 Layout Model 34
4.3.1 Clustering 35
4.3.2 Items Similarity 37
4.3.3 Pattern Mining 40
4.3.4 The Decoy Effect 41
4.3.5 Proposed Model for Layout 42

5 Implementation and Experiments **44**

5.1 Overview of the System 44
5.2 Data Integration 45
5.3 Data Preprocessing 47
5.3.1 Data Quality 47
5.3.2 Data Preparation 53

5.4 Feature Extraction 55
5.4.1 Missing prices extraction - Web Scraping 55
5.5 Image Processing 59
5.5.1 Activity Map Data Extraction 59
5.5.2 Feature Creation 62
5.6 Items Similarity 70
5.7 Shopping Patterns 71
5.8 Layout Development 72

6 Proposed Evaluation **76**
6.1 A/B Testing 76

7 Conclusion and Future Work **79**
7.1 Concluding remarks of the presented approach 79
7.2 Limitations and obstacles 80
7.3 Future Work 81

A Dashboard Development **82**

Bibliography **84**

List of Figures

2.1 Bulgari Store Layout 8

2.2 Bulgari Counter Layout 9

2.3 Camera view: Sections 11

2.4 Heatmap: reveals what people interact with most 12

2.5 Heatmap: reveals where people spend time 12

2.6 Activity map 13

4.1 Workflow Elaboration 29

4.2 Blue 32

4.3 Dark Blue 32

4.4 HTML Document Structure 34

4.5 Hierarchical Clustering 36

4.6 Items Description 39

4.7 Layout Model 43

5.1 System overview 44

5.2 Postman Request Example 46
5.3 Tickets Distribution 51
5.4 Customer ID distribution 51
5.5 Items' Frequency 52
5.6 Prism Dataset Sample 53
5.7 Heatmap 53
5.8 Activity map 53
5.9 Missing Values Treatment 54
5.10 Missing values treatment 55
5.11 Bulgari Web Page 56
5.12 Web Scraping Example 57
5.13 Web Scraping UK Price Example 58
5.14 Description Example 59
5.15 Counters Split in Prism 61
5.16 Color Transformation 61
5.17 Value Comparison, Counter 4 62
5.18 Value Comparison, Counter 5 62
5.19 Divisive Clustering 65
5.20 Agglomerative Clustering 66
5.21 Assessing Divisive Clustering 67
5.22 Assessing Agglomerative Clustering 68

5.23 Divisive Elbow 68

5.24 Agglomerative Elbow 68

5.25 Divisive Silhouette 69

5.26 Agglomerative Silhouette 69

5.27 Hierarchical Clustering - Dendrogram 70

5.28 Items Description 71

5.29 Purchase - Dwell Location 72

5.30 Necklace 1 74

5.31 Necklace 2 74

5.32 Necklace 3 74

5.33 Bracelet 1 75

5.34 Bracelet 2 75

5.35 Bracelet 3 75

6.1 A/B Test 1 77

A.1 Dashboard 82

List of Tables

4.1 Desired Table A 27

4.2 Desired Table B 28

4.3 Decoy Example 1 41

4.4 Decoy Example 2 41

5.1 Obtained Table 1 Example 73

5.2 Obtained Table 2 Example 73

Chapter 1

Introduction

The concept of a shop has changed during the past years, becoming not only the place where customers go to buy a specific product, but also a place where customers spend part of their time. For luxury brands, the physical experience has always been one of the core elements in the industry [ins13]. The store guides people not only by the price, but the ambiance of shopping, professional consultation and the ability to touch-and-feel the product are also important factors in the retailing experience. Therefore, it is very important to study the consumers' behavior so as to investigate the elements of the decision-making process of purchase that determine a particular consumer choice and how marketing strategies can influence the customer to buy a specific item.

One of the keys to a retailer's success is understanding shopper's behavior. In particular, it is necessary that managers know which retail attributes are important to which shoppers, and their main goal is to improve the consumer shopping experience [FMRZ17]. This way, retailers can arrange the items the way it is attractive for the customers and, therefore, beneficial for the store. Itamar Simonson emphasizes in his recent research [Sim99] the importance of the store image as a criterion to affect consumer behavior and store layout design as a critical determinant towards the creation of that store image.

A similar attitude towards this research can be found in a recent work by Merrilees and Miller [MM01]. The authors report that store layout design is one of the most important determinants of store loyalty. Simonson [DS99] also adds that the store layout design plays a key role not only in satisfying buyers' requirements, but also in influencing their wants and preferences. Similarly,

Grewal and Baker [BGP94] state that store layout affects consumers' price acceptability, which is also related to purchase intentions. Baker, Grewal, and Levy [BLG92] also found out that there is a relationship between positive experiences in a retail context and willingness to buy.

Simonson [Sim99] also points out that, apart from customer satisfaction, it is possible to influence buyers wants and preferences by applying product assortment "tricks". We summarize this idea as follows:

- Management can use the assortment subset considered by buyers to improve the possibility of a purchase and to affect the selection of a specific option in the store.
- The way the set of the available options is presented also affects customer preferences and final purchase decisions.

Fast-fashion is a relatively new phenomenon that has developed within the luxury sector. It means that the luxury products lifecycles have become shorter. As a result, "It" fashion items change every week [Oko16]. Therefore, many fast fashion retailers or grocery stores have turned to the use of product assortment tricks. The challenge of luxury retailers is subtly different from fast-fashion and mass-market consumer business, whose main goal is to quickly capture current fashion trends. On the contrary, luxury brands aim to drive future sales and lead the market. Thus, when it comes to the luxury fashion, the implementation of such techniques gets harder because the number of customers decreases, all of them need individual attention and cannot be divided under the basic criteria. Coupling the existing customer and consumer insights with additional behavioral information on how the luxury consumers are evolving and act in the store can help luxury brands to build deeper understanding and enable more tailored marketing - and ultimately improve longer-term value creation [del].

1.1 Retailer layout

Store layout is one of the core elements of the success of the retailer. It brings a significant contribution to the store image and manipulates customer traffic flow. In traditional conventional retailing, such as supermarkets, the store layout choice follows the standard most commercial pattern, which, in most cases, is the "Grid-Flow" layout. Such layout can be described as long vertical and horizontal rows of display units of goods with a single entrance and a separate

single exit. Such layout is widely favored by the grocery stores and supermarkets as the visiting customers usually have purchases planned beforehand and such layout helps them to identify the desired pre-selected products easily [VODS04].

Another general and commonly used pattern is the "Guided-Flow" layout, where the path is long and predetermined from the store entrance to its exit. It is also known as the "Racetrack/boutique" layout. Such type usually offers an unusual and entertaining shopping experience [Lew94].

In mass fashion retail, the most common and most feasible layout is the Free-Flow Layout. This pattern allows the free movement of customers around the store featuring few entrances and exits. Such option allows to enhance the image of the store and improve the overall consumer's mood and feeling offering an asymmetric arrangement of displays and aisles, employing a variety of different shapes, sizes, and styles for display [VODS04]. Additionally, consumers most likely will stay longer in such store if the atmosphere is complimentary. However, the disadvantage of such flow is that shoppers might fail to spot several products as a result of the absence of the main entrance and a central exit [Oko16].

In addition to these essential considerations, there are other important factors that should be considered especially when it comes to store atmosphere in the luxury sector. One of such factor is to ensure that the layout corresponds to the prestigious image of the store and complements its luxury atmosphere. Another factor is to ensure that space is optimized and provides the most comfortable space and distance for the customer to have a pleasant shopping experience. Moreover, each brand has its own special arrangement that reflects the brand personality. In addition to the store unique layout, the color scheme adopted by the store is essential to maintain the image and positioning of the brand. This ensures synchronization of the brand's identity with the store representation, and cannot be over-emphasized.

For example, the *Chanel*[1] monochrome black and white coloring in the store design and in the other aspects of the brand communication evokes classic chic. The gold and brown colors of *Louis Vuitton*[2] symbolize the prestige and high-class of the store and are meant to be in harmony with the overall visual identity of the brand.

All these aspects mentioned above are highly important for the brand and its perception among customers. However, once the layout is chosen and the colors are selected, the store

[1] https://www.chanel.com/en_US/
[2] https://www.louisvuitton.com/

should be consistent with this approach and try to align the rest of the factors with it. At this step, it is important to know what the consumer is looking for and try to influence his or her decisions while offering additional items that the customer might not normally consider. That is where the store needs to know about customers' behavior and preferences.

1.2 Luxury fashion industry

There are many different vocabulary definitions of luxury, but most of them share the same six core criteria:

1. a very qualitative hedonistic experience,
2. its price far exceeds the actual value of the item,
3. tied to its heritage and culture,
4. available in limited and controlled distribution,
5. offered with personalized accompanying services, and
6. representing a social marker, where the purchaser feels special with a sense of being privileged [KB12].

Thus, luxury brands can be characterized by their nature as unique, social, high quality, and emotional. Ultimately, this concept of luxury influences consumers' perceptions and is identified by their motives depending on the culture [VJ17]. The symbolic value of such brand, acquired through quality, status, and creativity, is directly proportional to its success. And its brand image is the most important component of brand's equity, defined as the "differential effect that brand knowledge has on consumer response to the marketing of that brand" [KAG08].

Recent studies show that the value of an art is essentially linked to the context under which it is consumed. As an example, it has been realized that there is a drop in the perceived value of the concert performed as a street art versus a symphony hall. This concept is referred to as an "art without a frame". The luxury consumer has the same requirements for the retailer. This frame helps to create a value for the store. The physical experience has always been the core of such frame. The ability to touch-and-feel the product, the relationship the stuff builds with the

consumer, the service received in the store, are all parts of this frame [del]. With the time, the frame becomes more digitized. That is where all the luxury brands need to find a unique balance against protecting the core and the heritage of the brand.

1.3 Problem Statement

Bulgari is one of the oldest Italian jewelry and luxury goods brands. Its rich product lines include jewelry, watches, fragrances, accessories, and hotels. Bulgari belongs to the luxury retailer group called LVMH (Louis Vuitton Moet Hennessy)[3]. The brand has a well-recognized name that is famous all over the world. The company honors its past even within modern design and thus, the update of the layout of each store was manually personalized and designed by domain experts: designers in Rome, Italy. Nowadays, this approach seems to have become too time-consuming and less effective, due to the fact that domain expertise is not always available in real time for the great number of stores (over 300, located in the most prestigious parts of the world). This means that the customers' preferences may vary upon the country and other criteria should be taken into consideration, such as specific store location, its entrance, doors location, customer's behavior, etc., when developing a layout plan.

This research focuses on developing an optimal data-based, multi-criteria layout for one of the Bulgari stores, located in Dubai, as a proof of concept of the feasibility of the approach. We are exploring the possibility of applying machine learning (ML) techniques to foresee customers' behavior and influence their choice. The results of our work aim to increase store performance and improve its profitability.

The main goals of this thesis can be summarized as follows:

1. Providing an overview of the ML process involved in the development of the models addressing store layout problems and domain-specific data challenges, such as understanding data requirements, data acquisition, ETL (extract, transform, load) process and feature extraction;

2. Extracting valuable data from the camera images, using image processing techniques, in order to analyze customers' behavior and describe purchasing patterns;

[3] https://www.bulgari.com/en-es/celebrating_explored

3. Detecting mistakes in the current design layout and proposing a new one based on literature and best business practices.

Our main contribution with this thesis is two-fold and can be described as follows:

1. We propose an end-to-end solution for extracting and analyzing recorded camera data of customer behavior. To the best of our knowledge, this is the first research that focuses on the analysis of data extracted from raw filmed heat-map photos taken from Prism Skylabs, in order to build customer behavior patterns.
2. We apply the Decoy Pricing Effect theory to propose a new optimal layout. To the best of our knowledge, this is the first research to combine Decoy theory with pattern mining techniques in the fashion industry.

The thesis is organized as follows: In Chapter 2, we describe the origin of the data, the objectives of the project and its primary goals. Chapter 3 contains the work related to the discussed topics. The next chapter focuses on the proposed solution and its formalization including layout model proposition. Chapter 5 describes the implementation process and evaluation methods. The following Chapter offers a proposed evaluation method to our approach. Finally, Chapter 7 provides conclusions and some possible avenues for future work.

Chapter 2

Background

In this chapter, we provide an introduction to the store under analysis and then we describe the dataset and its sources. Our data is divided into two parts. The first part of the data is the sales data that is provided by the store itself. The second part is the data provided by a cloud platform, namely Prism Skylabs.

2.1 Store Design

In this part of the thesis, we describe the store used as a proof of concept for the analyses. Despite the fact that the company headquarters are located in Rome, Italy, we focus on a specific store located in Dubai Mall, Dubai, Arab Emirates. This is a medium-size Bulgari store located on the first floor, next to its main competitors, such as Cartier[1], Rolex[2], and Tiffany[3], which brings only a specific audience to that part of the mall. The main products are displayed in the room upon the entrance to the store. This room includes 20 counters with various items on it, such as watches, jewelry and glasses. There are around 1,000 items being displayed in the store on a daily basis. The item layout can be defined as stable, which means that the items do not change their shelf positions. However, there are two small groups of items that substitute each other every 4 to 7

[1] http://www.cartier.com/
[2] https://www.rolex.com/
[3] https://www.tiffany.com/

weeks. One group replaces the other and takes its location.

Apart from the main room, there is an accessory room, located at the back of the store. The information about this room is not provided by the management. Moreover, there is a leisure area with a sofa for customers to relax. The information regarding this zone is not available either. Therefore, these two rooms are disregarded in our work.

Figure 2.1 provides an overview of the layout. We can see the location of the rooms and the main counters that are monitored by the camera, used in our study.

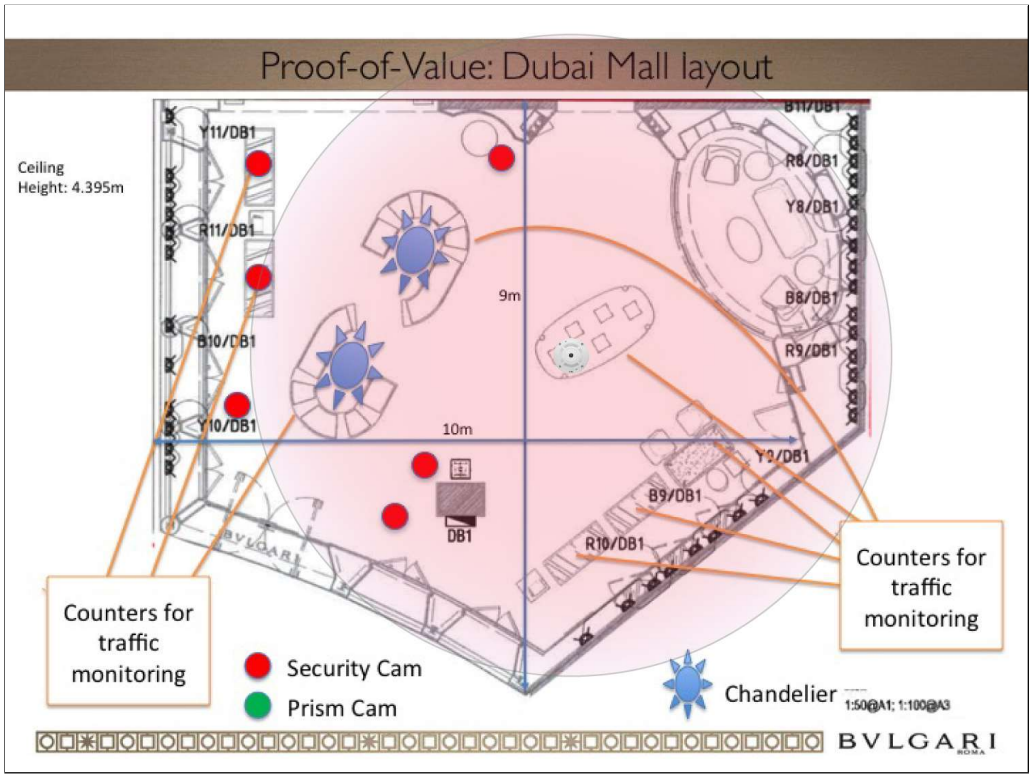

Figure 2.1: Bulgari Store Layout

Figure 2.2 is the Prism camera view with the counters numbers. Further on, in this work, we will be referring to the specific numbering that is displayed in the picture below. The counters are numbered 1 to 20 for the ease of reference. Counters 1, 2 and 3 are called *Front Counters* of the store. Counters 4,5,6,7 and 8, grouped together, and 9,10,11,12,13, grouped together, are named *Sushi Counter 1* and *Sushi Counter 2* respectively. The rest of the tables do not have any

special names for reference.

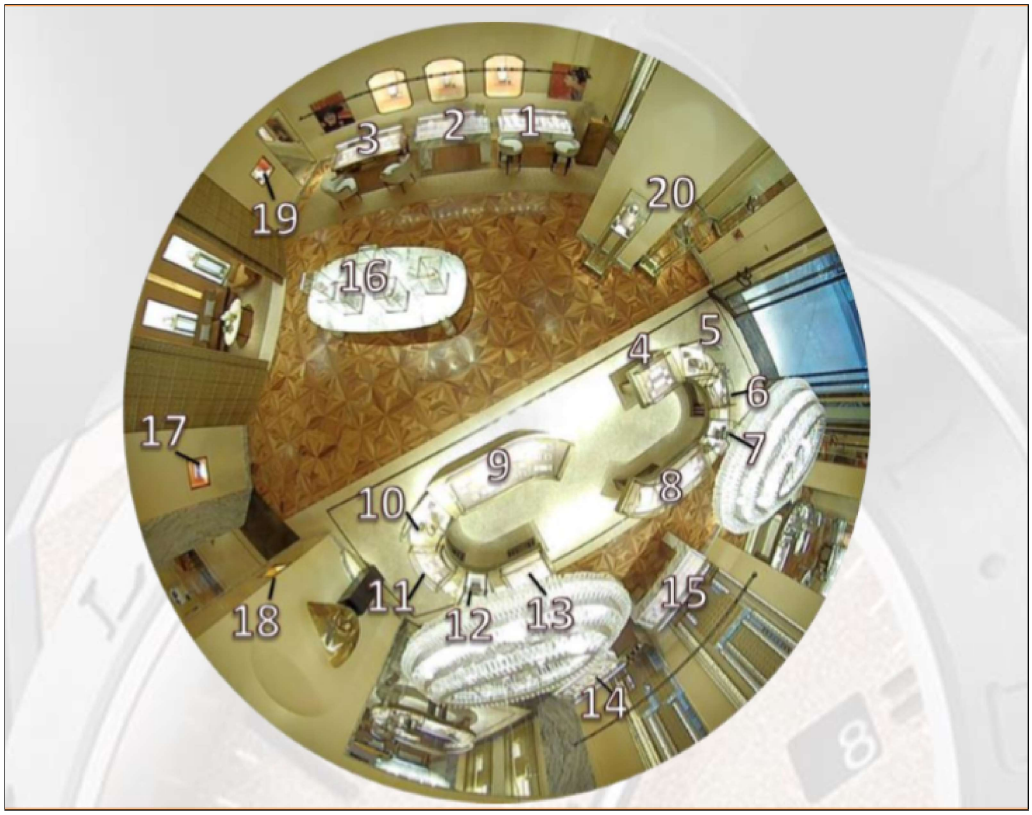

Figure 2.2: Bulgari Counter Layout

The store has a history of over 130 years and some of the factors of the layout design remain constant and are not allowed to change, including the Free-Flow Layout, the colors of the brand and the distribution of the sections. When it comes to the arrangement of the items, we would like to propose the new optimized layout that will be automated and furthermore, benefit the store.

The information the store provides is nothing more than a dataset, which is based on two different sources. The sales data is provided by the luxury fashion retailer - Bulgari, Dubai. The second part comes from its partner - Prism Skylabs (PRISM), which provides the cloud-based software to collect the customers' behavior metrics from the store.

This work is held at Clariba Consultancy - SAP partners in Europe and the Middle East, as a

part of the thesis project for the Master in Innovation and Research in Informatics Program at the Polytechnic University of Catalunya. Bulgari, a luxury fashion brand, is a customer of Clariba. This research is fully performed in parallel to an analytics project between Clariba and Bulgari.

2.2 Tracking Users Behavior

Prism Skylabs (PRISM) is the technology company that connects cameras with businesses providing fast analytics on the store performance. It is a comprehensive cloud intelligence platform that processes camera data and produces valuable information from it[4].

The main idea is that the store installs a circular *fisheye* camera lens [Bro82] that intends to create a wide hemispherical image of the main room in the store. With its special settings, it records the information about the people entering the store and monitors their behavior next to the specified counters. PRISM provides its cloud platform for the customers to be able to manipulate the views and adjust the necessary settings to rapidly produce the analytics of the required area.

In Figure 2.1, we see that the camera in our case is installed in the center of the main room. Figure 2.2 shows its view and the counters it includes in its range. Once we enter the cloud platform, we immediately see the main room of the store where we can select any area and create a new zone for which we might want to collect information. Figure 2.3 shows the areas in the store that we are interested in. These are the preselected areas that the management can create for their needs in order to collect the information about them.

As an example, the blue zone in the picture is the front counters area that we have selected. It combines tables $\{1,2,3\}$ together and produces information for one zone. From the moment this zone is created, the system automatically starts collecting information - the number of people entering the zone, the average time they spend there, and how many people are in the current zone on average within a time range.

There are two different types of metrics we can collect: “insights” and “tripwires count”. The first one - insights, provides the following information:

1. *count* - displays the current number of people in a given area within a given period of time;

[4] `https://prism.com`

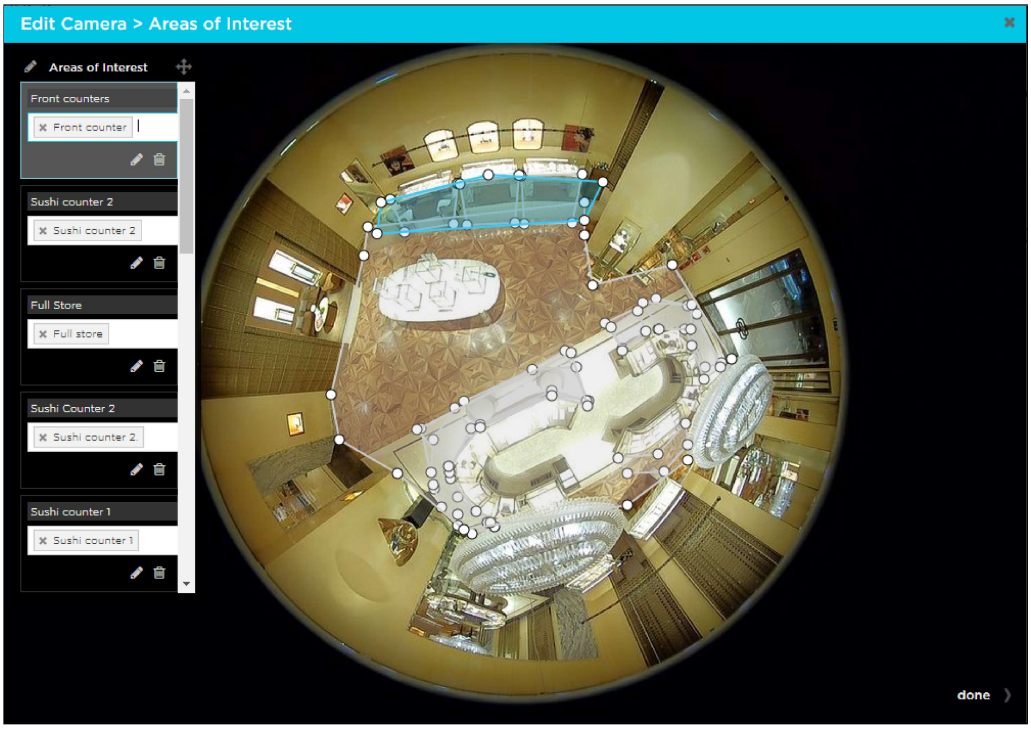

Figure 2.3: Camera view: Sections

2. *dwell time* - average number of seconds people stay in a given area;

3. *occupancy* - average number of people at a given counter. The second one counts the number of people crossing the line (one or both directions).

Moreover, the camera data provides a heatmap of the store for each week to show the performance of each highlighted area, which area of the store brings more customers (colored in red) and which lacks attention (colored in blue). The heatmap has 4 levels of color: blue (no people/lack of people), green (medium occupancy of the area), red (crowded), red-and-white (very crowded). An example of such type of image is presented in Figure 2.4. It reveals that, during the week 17, 2017, customers had a high interest in the items located in the front counters {1,2,3}, on the contrary, the rest of the store seemed less attractive to them.

Figure 2.5 shows an example of the heatmap where we can see how much time people spend next to the specific counters. Thus, during the same week 17, 2017, Front counters {1,2,3}

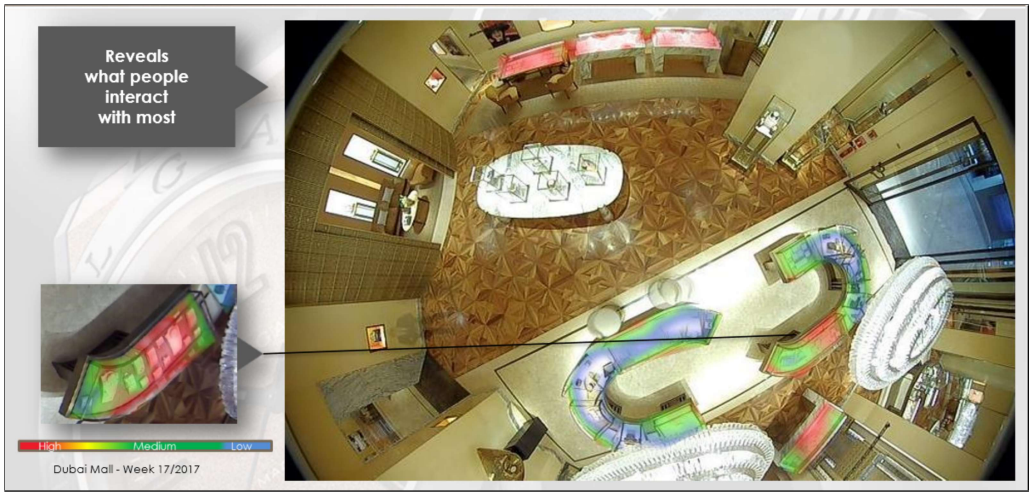

Figure 2.4: Heatmap: reveals what people interact with most

brought customers attention but also, some of the areas in the center of the store were occupied with people (red, red-white zones).

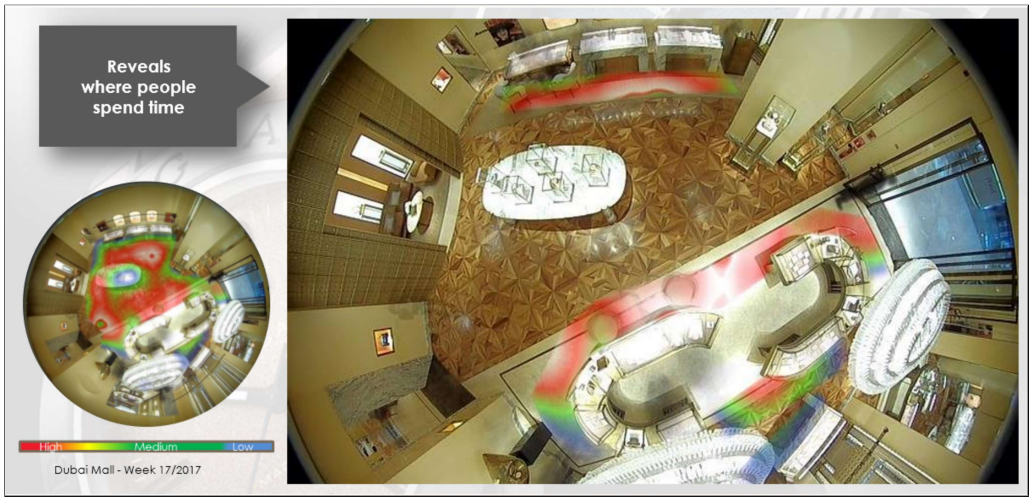

Figure 2.5: Heatmap: reveals where people spend time

Apart from that, the system provides the activity map, which adjusts its blue color intensity to the number of people and the occupancy of the store. Such map is available for each 15 minutes

of the records, which means the picture changes every quarter of the hour.

Example:

Figure 2.6: Activity map

Figure 2.6 shows the activity in the store for a certain hour. As we can see, the color is more intense in some of the areas and less visible in the others, which means that the most colored areas bring more people and thus are more popular. We can assume that the items in these areas attract more customers and are more appealing to the audience.

All this information gives a basic visual overview of the store performance. However, it is impossible to guide customers and relocate items based on just visual interpretation and general assumptions. For this reason, we need more sophisticated analysis and algorithms to address our problem.

2.3 Data Sources

Here, we describe in some detail the data sources provided to us fro analysis by the management. As mentioned before, our dataset was extracted from the luxury jewelry store, Bulgari and it portrays the activity and customer behavior during the period of the year 2017. For confidentiality matters, the analyzed dataset is not available for public usage as a part of the company's requirements and privacy agreement. This section focuses on explaining the origin and the structure of the datasets in detail.

For the current project to test our findings we used real data from the customers POS/ERP, such as SAP ERP[5] (centralized enterprise resource planning software developed by SAP company), which provides relevant transactional data including sales, stock, salesperson, store, price and items location. Sales data are available from January 2017 to January 2018. The data holds all the transactions with clients IDs, purchased items, and receipts values. Moreover, we know which items are available in the store, their location, and stock availability during one year period (January 2017 - January 2018). The customer provides shift distribution of the salespeople and who is in charge of the specific sale and each transaction.

The second source of data is PRISM Skylab, a cloud analytics platform. As mentioned earlier, Bulgari cooperates with the PRISM company, providing all the data for its further analysis. Specifically, the following data features are available from PRISM for extraction:

1. *count*
2. *dwell time*
3. *occupancy* .

This information (*count*, *dwell*, *occupancy*) is available for the period of 1 year, namely 2017, for two main zones: tables 1, 2, 3 grouped together, called *Front Counters* and *Sushi zone*, which are the counters 9 and 10 merged together, as they were preselected on January 15, 2017, and the camera has been collecting this information. At project onset, we realized that this information was not enough. We selected all of the counters separately and started collecting the full store data on February 26, 2018. Such fully split information is available until March 3, 2018 (6 days)

[5] `https://www.sap.com/products/enterprise-management-erp.html`

as that is the end of the contract period for the camera installed in the store. Unfortunately, this contract was not prolonged and we had to make do with the data collected until March 3.

Apart from that, we are using activity maps (blue heatmaps as shown in Figure 2.6) to analyze people's distribution for each hour and to fulfill the missing information. Such map displays the picture of the whole store, moreover, it colors the areas of the room in blue according to the number of people in the area and time they spend there. As mentioned before, our camera produces such map for every quarter of an hour. Along with that, we have heatmaps of the activity on the shelves and near the counters. Such maps are available only on a weekly basis.

For all the counters in the store, we want to collect insights and obtain all the possible metrics. For the entrance to the store and paths to an accessory room and leisure zone, we want to know the number of people crossing the entrance line. This will help us to estimate the amount of people in the main room with respect to the number of people entered overall.

2.4 Challenges

At the beginning of 2017, the idea of the project was not clear from the company's point of view. Therefore, after camera installation and selecting only two zones (Front Counters 1-3 and Tables 9-10) as a demo, no further actions have been taken. Therefore, for the whole year of 2017, the only available camera metrics we have for the analysis are for the following areas: 1-3 (grouped together) and 9-10 (grouped together). From this limitation, the main challenge is to fulfill the whole store information in order to be able to discover shopping patterns and optimize the layout.

Upon confirmation of the project in February 2018, we highlighted all the zones in the store that we are interested in and started collecting the necessary metrics. One week later, we were informed that the license expired and the contract with PRISM Skylabs came to an end. While the new agreement was still under discussion, we had to move on with our project due to the time limitations and decide on the further actions. As the fully split counters were only available for a short period of time - one week, we decided to rely on the data from the year 2017 and use the complete data for testing and evaluation purposes.

One of the options to overcome the obstacle of separating counters is to analyze the available heatmap graphs and perform reverse engineering using image processing techniques. Another option is to extract the split information from the blue activity maps according to the color den-

sity. As the number of people in the current area increases, the color becomes more intense. These two approaches are discussed in Chapter 5.

It is worthwhile to mention that unlike mass brands, luxury brands do not strive to please everyone. On the contrary, they aim to attract those customers whose beliefs are similar to theirs. For example, *Hermés* customers have to form a long-term, intimate relationship with the store before they are given the opportunity to buy one of the brand's "it" bags [GEP14]. Unfortunately, such information for Bulgari is not available and we carry our project regardless such limitations.

Within the year, the sales pattern changes and the reasons for that vary. In most of the cases, the main reason is seasonal (holidays). It is common, for example, that before Christmas people tend to buy more, and their behavior becomes less *rational* [Ban79]. The Middle East attracts people from all over the world during all the seasons. Holidays there vary to please the majority of the cultures. However, due to the data limitation and its availability only for one year, we cannot compare and adjust our analysis to the seasonal sales. We will rely on previous research and assumptions.

Another limitation worth mentioning is that the sales data is only available for the year of 2017 due to confidentiality matters of our customer. Therefore, we are not able to detect holiday or seasonal purchase behavior. We assume there is a change in the purchase behavior during certain periods of time (such as a tourist boom during Christmas and shopping festival in January) based on the previous research. Regardless this, we are not able to evaluate such findings, we leave this matter for discussion in Chapter 7.

2.5 Goals and Objectives

In our work, we propose an innovative approach to a build sales prediction model based on users' behavior and discover the factors that influence this behavior. We believe that access to physical browsing information of shoppers in retail stores can provide crucial insights about shoppers' needs and interests. We use this information to reveal the effectiveness of the store layout.

We are interested to examine that ability to guide users to buy certain items by adjusting the store's layout. This notion is followed by the interests of increasing sales in the store, optimizing stock, improving sales training effectiveness and increasing customers satisfaction in a competitive high-end retail market. We want to measure the consumer's reaction to a variety of variables

including price, counter location, or even specific details of jewelry styling i.e: line, collection, etc. Finally, we propose the optimal layout that takes into consideration all the mentioned factors.

Chapter 3

Related Work

In this Chapter, we present the current state of related research topics. First, we discuss studies related to the store layout techniques, followed by applied Data Mining methods in retail and existing tools for the identification of shopping patterns and clusters.

3.1 Store Layout Design Techniques

Understanding shopper behavior is one of the keys to retailer success. It is necessary for the management to know the exact needs of the customer and thus be able to identify primary and secondary products. Much research has been conducted to answer the question of how people shop according to the analysis of their behavior. The outcomes are not yet welcomed by the majority of retailers. As a result, many stores rely on local and anecdotal shopper insight and domain expertise to justify making changes in the stores. In this Section, we review such research in accordance with our work.

Simonson [Sim99] has analyzed purchasing behavior in stores and stated that in most of the cases, consumers construct their preferences when faced with a specific purchase decision, rather than retrieve performed evaluations of product features and alternatives. This means that, regardless of how much information people have about all of the products in the store, their purchase behavior is still based on the specific set of items and decision tasks that depend on the

particular characteristics of the considered options and the manner in which they are evaluated [TS93].

Such findings bring very important implications for marketing campaigns in the store, suggesting that products assortment cannot be assessed separately and solely on the basis of the characteristics of the items. Instead, the whole subset of items offered to the consumer has to be carefully analyzed and evaluated including the other marketing mix variables such as, for example, price. These implications suggest the notion that consumers have difficulty assessing the value of an option when it is considered in isolation. One might assume that customers evaluate all the options that they have been exposed to in the past in comparison with each other. However, [BC83] suggests that prior knowledge is less salient and has to be retrieved from the memory, so that consumers are likely to evaluate the attractiveness of a product relative to the other options that are considered simultaneously [BC83].

Simonson [Sim99] supports such findings with a great example about three bread makers. Suppose that a page in a catalog presents three bread makers; consumers are more likely to evaluate each of these options relative to the other two, as opposed to comparing it to bread makers encountered in the past and information stored in memory. Consequently, if one option appears clearly superior to the others under consideration, consumers may conclude that it is an attractive alternative, which increases the likelihood of making a purchase.

In [Sim99], the following two implications from the analysis above are suggested. First, that the total assortment that the store offers has less impact on the purchase decision of the customer than the specific subset the customer considers. Therefore, if retailers could control which specific option the customer is interested in, then they could design the given subsets in a way that would increase the likelihood of making a purchase of high-margin products.

Second, that a particular set of items can affect purchase probability in a predictive way. Consistently with the above implications, adding a relatively inferior option to the existing one but not to the third one can shift choices from a lower-priced option to a higher-priced option. The paper provides several illustrative experiments using microwaves and suggesting the following concepts: selling two type of microwaves (simple - cheap *vs.* more sophisticated - expensive), allows the customer to choose on the basis of her or his needs; however, adding a third option with a significantly higher price to the previously most expensive option increases the selling of the latter, that is now seen as reasonably priced.

Simonson [Sim99] reaches the following conclusions:

- The probability that the purchase will take place increases if one of the options from an assortment is significantly superior to the other one.

- The choice probability can be increased for any target option by adding an option that is significantly inferior similar to the target option but not to the other ones under the same product assortment.

The approach of grouping items in a similar set of three is related to our work and will be discussed further in the thesis as it is one of the most important concepts in the retailing industry. Moreover, we will discuss the related concept, called Decoy Effect [AW95] as a way to force a specific choice for a consumer by adding an unreasonably high-priced item with poor characteristics to the offered set. Section 6.4 of *The Oxford Handbook of Pricing Management* [JH95] discusses restaurant pricing and how the *Decoy Effect* helps to emphasize one or another meal on a menu. As an example, the author shows how overpriced wine makes the other wine options look more appealing and affordable. By providing 'decoy' wine, restaurants can increase sales of the other wins on the menu. Moreover, one survey, conducted in the U.K., showed that 25% of dinners decide to order the second cheapest wine on the menu. The common explanation of such phenomenon is based on the idea that people purchase wine in bottles for a group and the person choosing the wine often aims to be frugal but not be seen as "cheap". Thus, the group chooses the second cheapest but not the cheapest option. While this explanation is commonly cited, there is, to our knowledge, no empirical confirmation of such behavior.

3.2 Data mining techniques for retailing

The concept of similarity is fundamentally important in almost every scientific field. It also brings its significant value to retailing. Usually, retailing approaches are based on business theories, which involve finding a particular group of customers, necessary product characteristics, and so on. Clustering, distance-based outlier detection, classification, regression and search are major data mining problems that involve the computation of similarities between instances and hence the choice of a particular similarity measure can turn out to be a major cause of success or failure of the algorithm. Here we would like to focus on some papers and approaches that are related to our current work.

3.2.1 Clustering

Clustering is an approach widely used in the retailing industry to identify the segment the store belongs to, find the target market, offer a certain product to the specific audience, etc. In addition to improve productivity, clustering plays a vital role in a company's ability to innovate. As stated in [Por98], some of the same characteristics that enhance the productivity of the store have an even more dramatic effect on innovation and productivity growth. Charrad, Ghazzal and Boiteau define clustering as the partitioning of a set of objects into groups the way that objects within the same group are more similar to each other. Many clustering algorithms depend on some assumptions in order to organize data in the few subgroups [CGBN12]. A common issue in cluster analysis is that there is no single correct answer to the question of how many clusters provide the best outcome, since cluster analysis and the practical assessment of outcomes often involve human subjective judgment.

Many studies have tried to group clusters by their types. The most common algorithmic approaches are hierarchical methods, partitioning techniques and density estimation, as well as clumping techniques and some other methods not falling into any of the categories. This structure was suggested by Everitt as early as 1974 [Eve74].

More recently, Sheikholeslami, Chatterjee, and Zhang categorized clustering methods into the following types: partitional clustering, hierarchical clustering, density-based clustering and grid-based clustering [SCZ00]. To this day, a few more algorithms and approaches have being added to these, but their description is beyond the scope of the thesis. Most of these algorithms take some input parameters, such as a predefined number of clusters, its density, or at least the number of points to group a cluster. Different clustering methods allow us to create different groups of data, even selecting different parameters for the same algorithm leads to different clusters and may significantly affect the results. Therefore, it is very important to follow effective evaluation standards in order to be confident in the clustering analysis outcome [CGBN12].

Another effective way to obtain as precise results as possible is to select the most important data features that help to find clusters efficiently, thus helping to understand the data and reducing their size (domensionality reduction). The drawback of such procedure is that the task of finding important features for unsupervised data is a topic lacking systematic research. Traditional feature selection algorithms work only for supervised data where the data label is available. If the feature is important, it may help in creating the correct cluster; on the contrary, an unimportant feature might significantly affect and blur the cluster. Such features should be removed from the

dataset prior to or as part of the clustering procedure due to their confounding behaviour and irrelevance [DL00]. When it comes to nominal data, it is hard to choose the correct approach to perform clustering.

There are five main techniques to cluster nominal data:

a) Create a dummy code and compute inter-subject distances and use hierarchical clustering procedure on the derived data.

b) Create dummy data and use a standard clustering algorithm such as k-means.

c) Use correspondence analysis to derive spatial coordinates for each subject and then perform a standard clustering algorithm such as k-means.

d) Use latent class procedures for contingency table analysis.

e) Use Hartigan's Ditto Algorithm for categorical data.

Researchers explain that, in most cases, these algorithms have some drawbacks and are not efficient with categorical data. Thus, in the first approach, we need to decide on the distance measure among all the candidates. K-means is one of the most well-known clustering algorithms to partition the data. It is very efficient when it comes to processing large datasets, however, it is suitable only for numerical data and very sensitive to outliers. In [AJ14], k-means is extended by using a simple matching dissimilarity function suitable for categorical data. The mean value is substituted with mode values and a frequency based method for updating the clustering process that reduces the cost of the function. Such algorithm is suitable for categorical data.

[NS51] presents a simple procedure for clustering categorical data. This procedure is an analog to k-means. Similarly, authors of [CGC01] suggest to focus on the K-modes approach, which does not make any assumptions regarding the data - nonparametric, circumvents the need to define an *ad hoc* distance measure on the categorical data to be clustered and is as fast as the standard k-means [CGC01]. The k-modes algorithm is an extension of k-means for categorical data, by replacing k-means with k-modes.

On the other hand, Edelbrock emphasizes his research [Ede79] on hierarchical clustering as the method that does not produce a discrete number of clusters but rather a hierarchical arrangement between objects. The advantage of such algorithm is that it becomes easier to see the clusters visually and choose the most suitable number of clusters if the label is not available. In

order to perform such clustering, it is important to choose the correct dissimilarity measure as an input, where dissimilarity measure between A and B, as the total mismatch of the corresponding attribute categories of the two objects. The greater the number of mismatches is, the more distinct the objects are, and reverse. In other words, it is recommended to find a suitable algorithm to calculate the distance between the objects for a better clustering.

There are few methods to calculate a distance value between nominal data. The most recommended method is using Gower's similarity coefficient [Gow71], which compares all the attributes and indicates the average absolute discrepancy between all pairs of observations [Ede79]. For nominal data, the idea is the following: variables of k categories are first converted into k binary columns and then the Dice coefficient is applied. Gower method is known to be sensitive to outliers present in continuous variables and to non-normality; it is also computationally expensive to keep in memory an *NxN* distance matrix for large data samples. This limitations make transformations a preprocessing step that might be necessary.

3.2.2 Similarity for Item-based Recommenders

The rapid growth of e-commerce has led to the development of recommender systems - a personalized information filtering technology used to identify a set of N items that are predicted to be of interest to a certain user [Kar01]. User-based collaborative filtering technologies are considered to be the most successful ones. Unfortunately, such technology requires customers' information on certain products and moreover, the computational cost grows linearly as the number of customers increase. To address such issue and scalability concerns, item-based recommender techniques have been developed. Similarity search is one of the core operations for such recommender systems. Mamoulis and Cheung in their paper [MCL03] show how a hierarchical index can be used to process efficiently similarity search and other related query types on sets and categorical data.

Dissimilarity Functions for Categorical Variables The case of categorical variables for static and sequential data is outlined. For the static case, the following distances are well-known:

- Chi-Squared
- Hamming Distance

- Levenshtein Distance
- Damerau-Levenshtein Distance

Karypis [Kar01] discusses a few similarity techniques and provides guidelines for their evaluation. His idea is to determine the similarities between various items and then to identify which set of the items to recommended. Thus, one of the methods Karypis focuses on is to model the items as vectors in the user space and to find its similarity based on cosine function. He proves that such method achieves substantially good results. In particular, the cosine-based algorithm is, on the average, 15.7% better than the user-based recommendation algorithm.

As we know, The sample available to us for analysis in this thesis is relatively small and we can afford to keep the distance matrix in memory for our analysis. Moreover, we do preprocessing (Chapter 5) to ensure that our data is clean and ready for further computations. In the following chapters, we explain in detail our choice of algorithms and the approach we follow.

3.3 Data Mining tools and approaches

The goal of retail market [Pic04] managers is not only to provide high-qmost interesting products and services but at the same time to be able to react appropriately to changes in customer needs. To help the decision making process, data mining can be applied. It helps to identify useful customer behavior patterns from large amounts of customer and transaction data [GP02].

Over the past decades, there have been significant developments in data mining techniques. Some of these developments are implemented in customized service to develop customer relationship. Customized service is crucial in retail markets. Marketing managers can develop long-term and pleasant relationships with customers if they can detect and predict changes in customer behavior. In the dynamic retail market, understanding changes in customer behavior can help managers to establish effective promotion campaigns. This study integrates customer behavioral variables, demographic variables, and transaction database to establish a method of mining changes in customer behavior. For mining change patterns, two extended measures of similarity and unexpectedness are designed to analyze the degree of resemblance between patterns at different time periods. The proposed approach for mining changes in customer behavior can assist managers in developing better marketing strategies.

There are numerous methods [VODS04] [BLG92] [Gri05] to analyze the store layout and propose diverse solutions. One of the most recent implementations involving video recording and human movement analysis was conducted in the following paper [LZP14]. The main idea of this algorithm is that the system performs the background subtraction method after a video stream from the RGB-D camera has been acquired. Such method is widely used when it comes to detecting moving objects and regions within a sequence of images. In addition, there are numerous different approaches to address the face background detection problem. Piccardi [Pic04] provides a complete review on some of the most interesting techniques, such as Gaussian average, where the background is detected independently at each pixel location; a mixture of Gaussians, assuming that over some time different background objects appear at the same pixel location; Eigen backgrounds, based on eigenvalue decomposition, etc. With the growth of complexity, the accuracy of these algorithms increases and it becomes easier to distinguish the moving object on the background as it is a very common issue until nowadays.

The approach selected by Grottini Lab keeps a model of the background and performs a pixel-by-pixel subtraction with the current frame, getting the blob related to objects. Each pixel keeps the depth information, and the background image is dynamically updated. Also, the research focuses on the description of the situations that happen in such context: when the customer picks the product from the shelf - positive relation, when the customer picks the product and repositions it - negative relation, and nothing is taking - neutral relation [LZP14]. The core idea of this work was to propose an intelligent system that analyses and classifies the behavior of customers within a retail store. This way the researchers implement the heatmap based on the movements, which makes it visual to the managers to know the weak areas of the store. The proposed system ensures a rather high reliability, especially in the ideal condition in which the shelves are those considered to arm height.

As we will mention in the following Chapter 4, it is essential for the data to keep the most important features. However, there are cases when the provided features are not enough to obtain the most precise data analysis. One of the widely used techniques to obtain missing features is to use web data scrapers to extract information from the web. González-Peña and Lourenço [GPLLF$^+$13] present some of the most relevant approaches to web scraping technologies. Their research provides a comparison of the well-known libraries and frameworks addressing the drawbacks of each one and highlighting its advantages. Regarding the web scraping libraries, the authors suggest to focus on Apache HttpClient package for Java and its substitution - BeautifulSoup for Python. This library can be combined with language native support for HTTP connections. Current research emphasizes on the simplicity of the framework. It suggests

that programmers are able to create Web data scrapers in one or few lines of code by simply piping operating system command-line programs inside shell scripts [GPLLF+13].

Chapter 4

Proposed Solution

With the goal of proposing an optimal store layout that maximizes profitability, the data has to be subjected to a number of treatments in order to have it ready for further processing and analysis. To perform this complex task, the ideal dataset would require having accurate information regarding the time customers spend at each counter and a precise location for each item in the store. These features are not fully available in one data source. It is rather a combination of raw data extracted from multiple sources, then transformed as a part of full ETL [KC11] pipeline. These desired (loaded, extracted and transformed) data would assure that the counters that bring less attention should be rearranged in terms of the layout according to the customers' choices of the items. A simplified example of the desired dataset (after ETL process) is presented in Tables 4.1 and 4.2.

Table 4.1: Desired Table A

Metrics / Date	Count #1	Count #2	...	Count #n	Dwell #n	Occupancy #n
Date 1						
Date 2						
...						
Date n						

Table 4.2: Desired Table B

About / Item	Feature #1	Feature #2	...	Description	Ranking	Location
1						
2						
...						
n						

- Table 1 includes the information regarding the count, dwell and occupancy of all the zones in the store for each working hour.
- Table 2 includes all the items with their features, unique description, ranking, location and price.

We query the raw data, detailed in Table 4.1 and Table 4.2, from two different resources: 1) PRISM Skylab, and; 2) Organizational SAP ERP systems. The data is stored in the database that we use during this master thesis research. Then, we perform a series of transformations to reach the targeted format, as of obtaining metrics of *dwell*, *occupancy*, and *count*. Lastly, we normalize the data, extract features, and merge the data (the steps are detailed in Sections 5.1 and 5.2).

To summarize, the pipeline presented in this chapter is as described in the following steps:

1. Preparing two datasets: 1. Items dataset that includes the set of initially provided features (described in Section 5.3), as well as additionally extracted ones (Section 5.4), and; 2. a dataset of counters metrics split for each location.

   The preparation stages are as follow:

   (a) Store the data in the database

   (b) Complete the Items dataset with the available information. This stage includes:

      i. Extracting data from the web page

ii. Assigning rankings to the items

(c) Complete the metrics dataset by extracting data from the activity heatmaps

2. Developing an optimal layout model, which includes:

(a) Clustering as a feature creation

(b) Items Similarity

(c) Pattern Mining

(d) Decoy Pricing

(e) Model characterization

A workflow elaboration is presented in Figure 4.1.

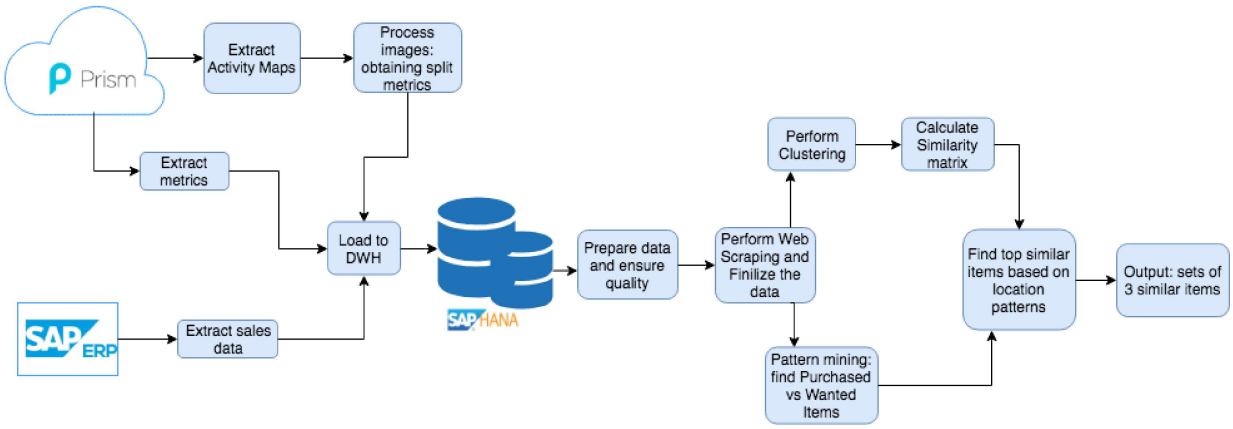

Figure 4.1: Workflow Elaboration

4.1 Initial Assumptions

Before discussing the framework design, we present assumptions that define the approach taken for our proposed solution. These are recommendations on the analysis procedure from the data analytics experts, at Clariba, who act as domain experts and advisors in this thesis:

1. Sales data from 2018 is not currently available but, for testing purposes, we assume that the location of the items remains the same within the year. Since all the items kept their location for the period of one complet year, we propose our solution assuming that in 2018 the layout has not changed.

2. When it comes to data storing, the preference is given to SAP HANA due to organizational reasons, such as resources, knowledge and support.

3. As visual results are very important for the customer, it is necessary to be able to provide them. For the last step after the evaluation, we are expected to deliver an *insights dashboard* to the customer (Bulgari). Visualization should be handled in SAP tools or other supported ones. Dashboard is presented in the Appendix.

4.2 Data Set Compilation

4.2.1 Storing data in the database

After querying the data from the original data sources, we needed to store the data in one database that acts as the main source of truth. The reason behind this is to make a flow of data from PRISM Skylabs directly to the database, which will be updated on a daily basis. The choice of the database (SAP HANA database[1]) is explained by the project requirements and management suggestion to use a DBMS by SAP. The decision is based on the following reasons: it simplifies the integration between the rest of the components the company has used before; it produces immediate real-time results due to the fact that the data resides in the RAM; also, it processes any type of data and can handle huge amounts of them. All these are taken into consideration as required for future purposes as a part of the permanent solution for Bulgari project.

4.2.2 Completion of the datasets with further available information

Due to the project time limitations and data sources availability, some of the features were not provided by the management. To overcome this, several alternative approaches were discussed. We demonstrate these approaches in the following sub-sections.

[1] `https://www.sap.com/products/hana/features.html`

4.2.2.1 Extract data from activity heatmaps

After the data is integrated and stored in the database, we need to analyze it and obtain the missing information. The camera connected to PRISM Cloud platform provides analytical information for the period of one year, which was the contract duration period for Bulgari (from January 2017 to March 2018). This version was installed to provide sample data examining the camera performance and exhibiting the usage of the heatmaps to identify the most attracted areas in the store. Within this period of time the camera has collected detailed information for two main areas, as mentioned in Section 2.3, "Front Counters" (store's tables 1 to 3 together) and "Sushi counter" (tables 9 and 10 together). The rest of the information was not collected in 2017. Thus, one of the challenges of the current project was to overcome the data limitation problem by reproducing the data for the whole store in order to be able to conduct a complete research analysis.

In order to restore some of the information, we used a digital image processing technique. This approach helped us to extract useful information from it. It is a type of signal processing where the input is an image and the output may be image or characteristics or any features associated with that image.

Image processing can be divided into the following three steps:

- Image importing (image acquisition tools can be used)
- Analyzing and performing manipulations on the image
- Collecting output as an image or report that is based on an image analysis.

An image is a two-dimensional function *f(x,y)*, where *x* and *y* are the spatial coordinates and the amplitude of *f* at any pair of coordinates is the intensity of the image. If these values are finite and discrete, then the image is digital, in which the elements are called pixels. Each pixel's intensity value varies based on the used color model, where each color within the pixel has its own value (i.e. between 0 and 255). One of the color models we considered is called RGB model, where red, green and blue are added together in various combinations to reproduce a broad array of colors [Bou04]. The same way, we can distinguish a value of any color in the image. Thus, pure blue has the values of (0,0,255) where the third number stands for blueness. Given the fact 255 is the maximum value allowed, the lower the value, the darker the color, where tuple (0,0,0)

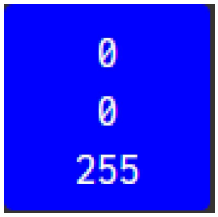

Figure 4.2: Blue

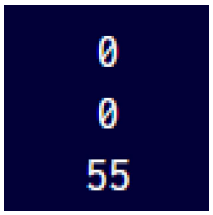

Figure 4.3: Dark Blue

represents black. An example is presented in Figures 4.2[2] and 4.3[3]. The combination of all three value such as Red, Green, and Blue creates different shades of blue as well.

Such technique of identifying a color by each of the pixels of the image allows us to analyze an image in terms of colors and find the value of each pixel. The PRISM system provides two types of images - the heatmap and the activity map (Chapter 2.4). The heatmap holds the information for every week of the year. The values of the metrics, along with the number of people passing by and the time they spend at the counters, keep changing every minute or even every second in the store. That is why we consider one-week information as a very rough approximation for our project. This is not sufficient to restore our data for the rest of the counters. In addition, another factor that makes extracted information not valuable, is that such map has only 4 color levels (blue, green, red, and white-red mixture) without its density. In other words, we will be able to represent the counters occupancy in only 4 levels respectively, which is a poor estimation for 20 counters in total.

Instead, the activity map, which is presented as a blue map of blue color density, is available for every 15 minutes during the whole year. In such (blue) map, density represents the number of people and the time spent at the counters. The darker the area the more people it has at a certain period of time. That is why we decided that the *blue activity map* can be valuable when it comes to the counter metrics.

Therefore, in order to overcome the raw data problem we propose the following idea: by extracting the activity maps from our data source (PRISM Skylabs), we apply image processing techniques, namely reverse engineering, to obtain the raw data (or; needed features) from the picture. We analyze the amount of blue color in each area of the store by extracting the actual

[2] http://rgb.to/rgb/0,0,255
[3] http://rgb.to/rgb/0,0,55

values of the color. After normalizing the calculations, we obtain the percentage values of blue in each area. That said, we obtain the percentage values of how much the area is crowded with respect to the whole store at a certain hour of time. This process and the obtained results are described in detail in Section 5.5.

4.2.2.2 Web data extraction

Bulgari is a luxury fashion company that is widely known company around the world. It has its webpage translated to a number of different languages, which makes it easily accessible for the representatives of 17 countries. The data that describes the items and their characteristics is publicly available and easy to access through the web. In this subsection, we aim at describing how to get such data and extract the information we are interested in.

There are several options to obtain the data from the web when it is not explicitly provided. The data can be obtained from the web-based API, extracted from PDF, or by screen scraping the web site. The main goal of all of the above solutions is to get access to machine-readable data.

Our project is held at a real jewelry store and some of the information remains unavailable for public purposes due to confidentiality reasons. Therefore, we try to extract as much information as possible from the webpage, because it can provide a better description of our items and improve the overall results. The advantage of web scraping from the rest of the options is that even if it does not have an API for raw data, it is still possible to request the data. In the case of Bulgari, there was no other source for additional items description besides the website. That is why we decided to use web scraping for this project.

Web scraping entails extracting and combining contents of interest from the web in a systematic way. Web scrapers establish communication with the target Web site through the HTTP protocol, parse it and extract the content of interest. The general idea behind web scraping is to retrieve data that exists on a website and convert it into a format that is suitable for analysis. Most of the information is usually included in the HTML code. The HTML tags contained in the angled brackets provide structural information, which is useful for selecting the content relevant to our needs. Most modern browsers, such as Chrome, for example, have a parser that reads the HTML document, parses it into a Document Object Model (DOM) structure, and then renders the DOM structure. An example of such structure can be found in Figure 4.4[4].

[4] (http://web.stanford.edu/~zlotnick/TextAsData/Web_Scraping_with_Beautiful_

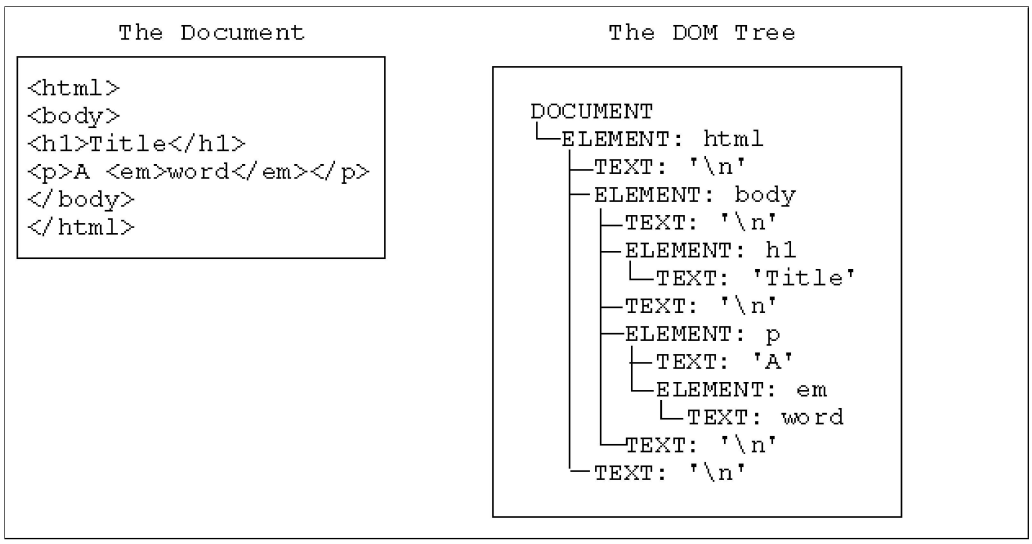

Figure 4.4: HTML Document Structure

In order to transform the extracted data into a structured representation that is suitable for further analysis and storage, we need to get the name of each item in the list in the structure that is associated with the value we are looking for [GPLLF+13]. This way, we will be able, for instance, to extract the missing prices for some of the items. Besides that, we are interested in the complete items description as an additional feature. Such description is unique for each of the items and represents the main characteristics of the product.

4.3 Layout Model

With the goal of proposing a layout model for the store, we aim to analyze the available data using the techniques described below in this section. First of all, we want to define a similarity between items in order to compare them and, thus, be able to recommend similar products on content-basis. Then, we want to uncover customers behavior patterns in the store, such as, for instance, which are the areas of buyers preference and which counters are selling the most. This way we will decide which products should be relocated and from which counters. Finally, we provide our solution based on the decoy pricing theory to rearrange the items in the store.

`Soup.html`

4.3.1 Clustering

In order to define items similarity, we strive to have as precise description of the items as possible. This will help us to obtain better similarity values. To do so, we want to create a new feature by grouping the items into clusters. After that, each cluster will be assigned to the items description, extracted from the web as explained in the previous subsection 4.2.2.2 .

There is a number of clustering algorithms available to us, though most of them are designed to handle numerical data [CD05], where the proximity measure is usually defined as a geometrical distance. For categorical data, which has no order relationship, one available general method is to transform data into a binary format. However, such binary mapping can lose the meaning of the data, which results in the formation of the incorrect clusters [GRS00]. Many algorithms for categorical data [IM09] extend existing methods with a proximity measure for categorical data.

ROCK is an example of a clustering algorithm that is suitable for categorical data. It is a bottom-up clustering algorithm that performs based on a similarity function for the number of common neighbors, which is defined by the Jaccard coefficient. In this case, objects are the more similar to each other as the more common neighbors they have. The disadvantage of this algorithm is that its complexity is quadratic as it clusters a randomly sampled dataset and then partitions the entire dataset based on these clusters.

Another option we can use is *k-modes*, which is based on the well-known k-means algorithm with the adoption of a new similarity function to handle categorical data. A cluster center in this case is represented by some virtual object, which is formed by the most frequent attribute values in the cluster. It extends the *k-means* paradigm to cluster categorical data by applying matching dissimilarity measure for categorical objects; using modes instead of means for clusters; and a frequency based method to update modes in the same way as *k-means* to minimize the clustering function cost. As its fundations are the same as those of *k-means*, the efficiency of the latter algorithm is preserved [HDX06]. The disadvantage of such clustering approach is that the number of desired clusters must be specified prior to all the calculation. However, unlike most existing methods for clustering categorical data, the *K-modes* procedure explicitly optimizes a loss function based on the L0 norm (defined as a limit of an Lp norm as p approaches zero) [CGC01].

Hierarchical clustering is a third type of clustering approach we can perform. It is more exploratory and can be approached in two ways: either through agglomerative (bottom-up) or divisive (top-down) clustering. As we can not evaluate our clusters based on labels (they are not

provided), this exploratory algorithm should provide us some insights on the number of clusters and suggest the potentially best options.

Agglomerative clustering starts with n clusters, where n is the number of observations, assuming that each of them is its own separate cluster. Then the algorithm will try to find most similar data points and group them. As a result, these groups start forming clusters.

In contrast, divisive clustering performs the other way around, by assuming that the n data points are one big cluster at the onset and splitting the most dissimilar ones into separate groups.

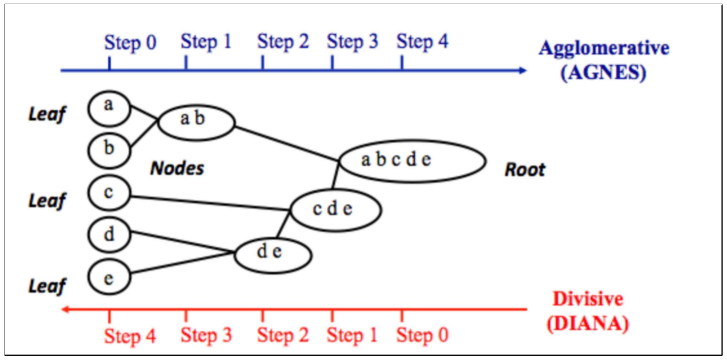

Figure 4.5: Hierarchical Clustering

Figure 4.5 portraits the difference between agglomerative and divisive clustering approaches. As research shows [ZMRA13], agglomerative clustering is usually better at discovering small clusters and is the option used by default by most standard software, whereas divisive clustering is usually better at discovering larger clusters. For our data, it is worth trying both approaches and comparing them.

One of the last steps in the clustering procedure is to assess these clusters. Working with categorical variables, we might end up with nonsensical clusters because the combination of their values is limited - they are discrete, and so is the number of their combinations. As it often happens with assessment, there is more than one way possible, therefore, we can complement the idea with our own judgment relying on characteristics of the clusters and so on.

Conceptually, when clusters are created, our interest lies in discovering distinct groups of data points, such that the distance between them within clusters is minimal while the distance between groups is as large as possible. Distance between points is a measure of their dissimilarity derived

from a dissimilarity matrix. Hence, the assessment of clustering is built around an evaluation of compactness and separation.

Once we decide on the clustering approach, we need to evaluate our clusters and choose the correct number of groups. There are two main evaluation techniques that we will follow to assess our clusters:

- Elbow method: suggested when the compactness of clusters, or similarities within groups are most important for the analysis.
- Silhouette method: as a measure of data consistency, the silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters.

In practice, as shown in [MDP+07], these two methods are very likely to provide different results that might be confusing: different numbers of clusters will correspond to the most compact / most distinctively separated clusters. As a result, our personal judgment and understanding about the data and their context will be a significant part of making the final decision.

The obtained clusters are added to the items' description as an additional feature to improve the similarity results. This is described in sections 5 and 6 of the following Chapter 5.

4.3.2 Items Similarity

Similarity in a data mining context can be defined as a distance in a multi-dimensional space with dimensions representing features of the objects [Gos12]. A small distance would indicate a high degree of similarity and a large distance would correspondingly indicate a low degree of similarity. The ways to calculate the distance vary depending on the data type and available information.

In our research, we want to be able to recommend items to assign next to each other, as well as pick the counter to place them on. In order to do so, we would like to approach the idea of an item-based recommender system. The idea lies in creating a similarity function for all the items in the store and relocating them based on their price and similarity distance.

In order to group the most suitable products together and be able to change the customers perception, we want to know how similar our items are. The idea is to define a similarity based

on the following characteristics: the description of the items that is extracted from the web page using web scraping technique, while taking into consideration the other features provided by the management with the given data. In the previous Subsection 4.3.1, we discussed how to group products according to the features provided and assign them to clusters. In order to take into account this information for the similarity function, we simply add cluster numbers to the description of each item. In the previously discussed paper by Karypis [Kar01], the cosine similarity algorithm outperforms other well-known similarity algorithms. We decided to follow this research and base our work on cosine similarity as it fits our data description.

4.3.2.1 Cosine-Based Similarity

One of the approaches to compare the items is *tf-idf* vectors construction followed by cosine similarity calculation [SWY75]. The *tf-idf* (term frequency–inverse document frequency) is a measure commonly used in the field of information retrieval to compare how significant a word is to a given document. Here, there is a value for each pair term-document. So if we want to analyze how related a query is to a document we will have a vector representing a query. The *tf* refers to the calculation of the frequency of a given term (or word) in the document (the higher this frequency, the higher the relationship between the term and the document).

As described by Karypis [Kar01], one of the ways to calculate the similarity between items is to treat each item as a vector in the space of items and use the cosine function between these measures of similarity. The cosine between two vectors is given by

$$sim(v, u) = cos(\vec{v}, \vec{u}) = \frac{\vec{v} \cdot \vec{u}}{|| \vec{v}||_2 || \vec{u}||_2}$$

where '.' denotes the vector dot-product operation.

From the equation above we can conclude that the similarity between two items is high if item 1 has the same features as item 2. If the vectors are close to parallel, we assume that both values are 'similar' to each other, and consequently, the set of words is similar in two vectors. Whereas, if the vectors are orthogonal, then we assume the sentences are independent or "not similar".

In [Kar01], the similarity is higher for the frequently purchased items, whereas, in our case, the most similar items should be the items from the same, or similar, set of words in the de-

scription, with respect to our corpus. Thus, for instance, a pair of rings is likely to have higher similarity than a pair of ring and watch.

Figure 4.6: Items Description

RING

Grown from the Roman roots of the brand into an elegant fusion of culture and modernity, the BVLGARI BVLGARI ring is an effervescent, contemporary statement of classiness. The trademark double logo was initially inspired by the curved inscriptions on ancient coins, whilst today it has evolved into playful interpretations, framing an exquisite diamond. BVLGARI BVLGARI 18 kt rose gold flip ring with a diamond (0.25 ct). Also available in 18 kt white gold.

RING

Grown from the Roman roots of the brand into an elegant fusion of culture and modernity, the BVLGARI BVLGARI ring is an effervescent, contemporary statement of classiness. The trademark double logo was initially inspired by the curved inscriptions on ancient coins, whilst today it has evolved into playful interpretations, framing hard gemstones and pavé diamonds.
BVLGARI BVLGARI 18 kt white gold flip ring with black onyx and pavé diamonds.

NECKLACE

Staying true to its revolutionary spirit, the legendary B.zero1 icon has been reinterpreted by a legend of design, Zaha Hadid. Born from a Roman inspiration, the Colosseum, and reimagined by the greatest woman in architecture of all times, the B.zero1 Design Legend jewelry is a perfect collision of languages, where geometry merges with fluidity and modernity exalts tradition.

Let us consider the example in Figure 4.6. According to the vector similarity, Item 1 and Item 2 are supposed to have a very high similarity index - close to 1, due to the very similar description, where Item 2 and Item 3 have very few words in common and thus, the vector similarity is significantly lower than in the previous case.

4.3.3 Pattern Mining

As for the next step in this master's thesis, we want to discover customer behavior patterns. Specifically, we want to find different groups of counters that attract people and see if buyers thoughtfully observe the items they are willing to buy, or if certain items provoke buying from a different counter. That said, that customers can buy from the counter they spend most of their time at, or on the contrary, they may prefer to spend their time at the counters with more "exciting" items while observing, but buy something else due to reasons such as price, appearance, etc. In order to discover such patterns, we use a pattern mining approach. Specifically, we focus on the *frequency algorithm*, which finds the most frequent patterns among our set of highest dwell and counters where the purchases have been made during a specific hour. In order to discover such patterns, we use the previously obtained information about the dwell of each counter, as this is the information that emphasizes where people spend the most time in the store. In this case, we are interested in the counter number that has the highest dwell for each hour.

Beyond that, we analyze the transactional data and split it by hours corresponding to our dwell information. Thus, we have the table of the areas where people spend the most time while in the store and where the purchase is coming from within the same hour. Taking into consideration the information provided by the company management, people need at least 10 minutes to conduct a purchase from the moment of entering the store due to the fact that, in most cases, the customer spends some time in front of the counters to compare items and only after that confirms the purchase. Thus, if the purchase has been made within the first 10 minutes of each hour, we assume that this transaction belongs to the dwell of the previous hour.

4.3.4 The Decoy Effect

Here, we describe Decoy pricing theory [5], which we have used to rearrange the items in order to suggest an optimal layout. As previously mentioned in Chapter 3, the main idea of this theory is to place similar items in groups of three, with various characteristics and prices, according to which people tend to change their preferences between two main choices when presented with an asymmetrically dominating third item. Let us refer to Table 4.3 and Table 4.4 for an explanation:

Table 4.3: Decoy Example 1

Watch 1	Watch 2	Watch 3
Plastic	Metal, waterproof	Gold, waterproof, 3 years warranty
100€	220€	250€

According to the description above, most of the customers will choose *Watch 3*, because its characteristics are much better than the middle option and the price is only 30€ more expensive. However, if we omit the second option, the customers decision will be based, mainly, on their needs: whether they need a luxury watch or a basic one for functional purposes, as in this case the difference between the two options will be 150€.

Let us slightly change the setting, as exemplified in Table 4.4, by adjusting the price of the third option to 700€. Now the difference between *Watch 2* and *Watch 3* is large and, thus, most of the customers will go for the second option (*Watch 2*). The middle, *Watch 2*, offers better quality than the first one, and the difference in price is low comparing to the third option.

Table 4.4: Decoy Example 2

Watch 1	Watch 2	Watch 3
Plastic	Metal, waterproof	Gold, waterproof, 3 years warranty
100€	200€	700€

Summing up the idea of the Decoy theory: we can guide customers towards one or another

[5] https://www.lokad.com/decoy-pricing-definition

product by adjusting the price of the item nearby in the same set. In our case, we are not allowed to change the pricing strategy of the store; moreover, it is not the purpose of this project. However, we will still be able to suggest which item should be placed together to increase sales and guide users toward more expensive products.

4.3.5 Proposed Model for Layout

We want to analyze the discovered patterns and figure out which of them are the most interesting for us. For the initial proposal, we take all the different pairs where people spend the most time (with the highest *Dwell*) and where they buy the product. After discovering such patterns, we analyze each pair separately proposing the three options for relocation that are of highest interest for us. Consequently, we take each pair of high dwell location/ purchase location and find top three purchased items from the following location where the purchase was made. For each of these items, we find the most similar item from our similarity matrix (the items with the highest similarity score with respect to the given item) applying the following condition: the new similar item should come from a location of the counter with the highest dwell of the same pair, and its price should be higher than the previously purchased option as long as the ranking is lower than before (in order to boost the sales of the less popular items). Therefore, obtaining a group of three items with one of them significantly higher than the first one (the so-called Decoy item), we need to acknowledge that, from such combination, this item will be the least purchased one. We would expect that, instead of the cheapest option that was popular before, customers will shift towards the second most expensive item with a medium value. This idea is graphically represented in Figure 4.7.

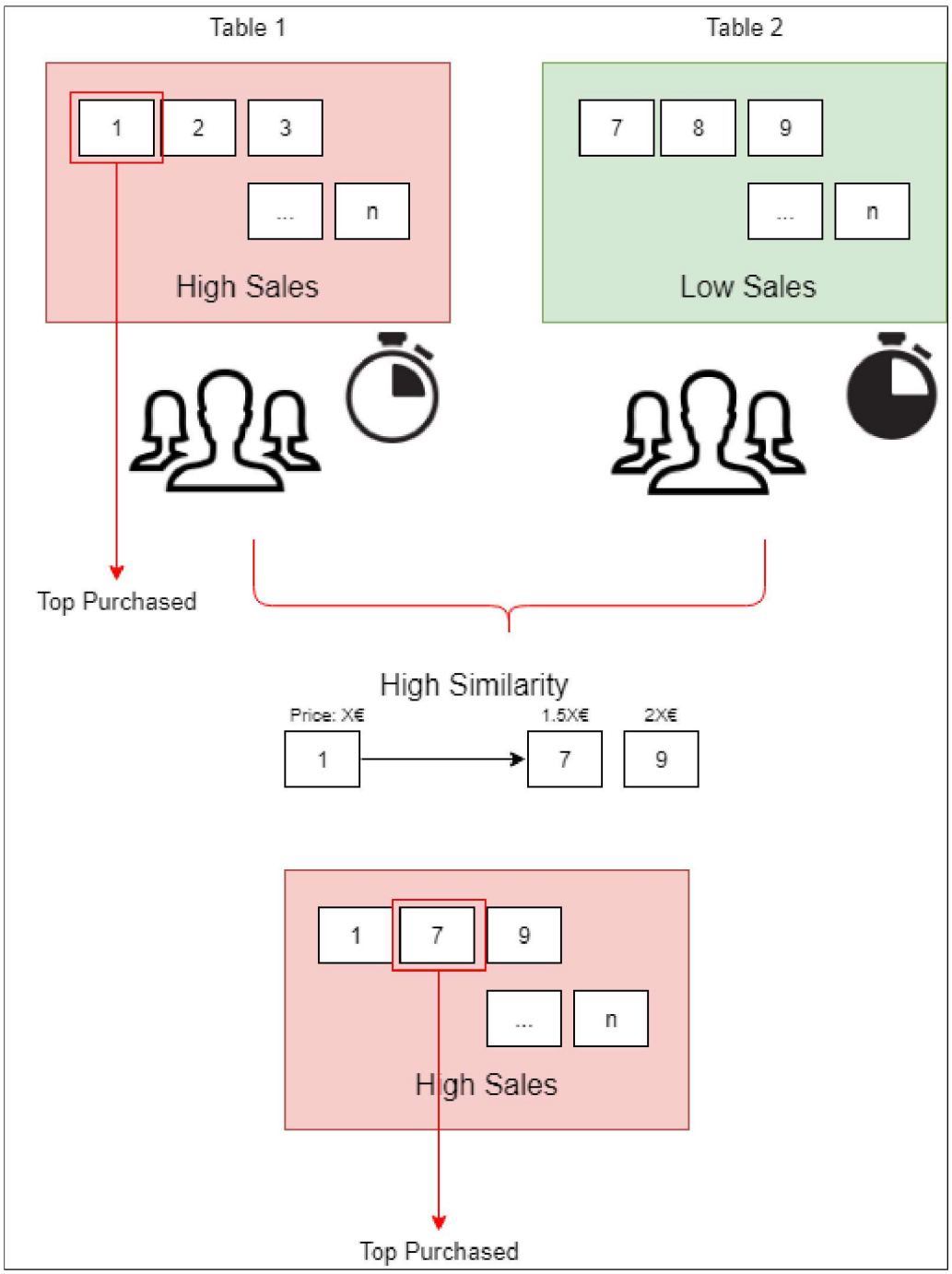

Figure 4.7: Layout Model

Chapter 5

Implementation and Experiments

In the previous chapter we focused on providing a general description of the proposed approach. That consists of the notion to analyze the data of a store belonging to the luxury retailer (Bulgari), while being oriented toward the main goal of store layout optimization. In this chapter, we provide detail of the implementation of techniques, data preparation and design and execution of the experiments.

5.1 Overview of the System

As we represent in Figure 5.1, our system consists of 4 main components: image processing, pattern mining, similarities finding and an exploration/browsing tool that uses the previously obtained information to propose new layout options. These components are described according to the following steps:

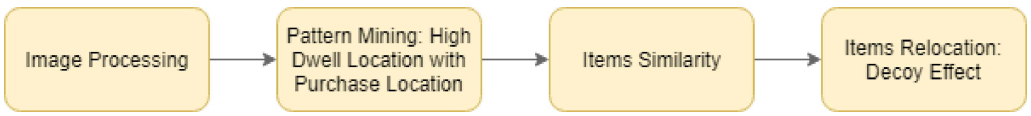

Figure 5.1: System overview

Specifically, the main idea of our work lies in the following concept: First of all, we need to obtain the data in the following format, presented below:

- New features are extracted to fulfill items description;
- The full description of the items is available to determine item-to-item similarities;
- *Occupancy*, *Dwell* and *Count* information is extracted for each area of the store for a period of one month;
- Item prices are available to propose a new relocation within counters and product pairs, based on similarity and price features.

In order to satisfy these criteria, we need to extract the data and store it in the database for our convenience. After that, we provide a data preparation assessment and decide how to obtain the proposed structure.

5.2 Data Integration

One of the key elements of our work was the extraction of the necessary data from the provided sources. The main source was the PRISM Skylabs and the data was located in its (PRISM) cloud platform. It could be accessed through a connection to its custom-made API. As a part of this project requirements, we stored the data in the SAP Hana database for the ease of the integration in the future with the rest of the customer systems, such as SAP ERP. This guarantees a solution that is sustainable for the company.

In order to get the data from the PRISM cloud platform, we needed to ensure a connection with the source. For this purpose, we used Postdot Technologies' Postman application [1]. It is a free application and it handles almost all types of standard requests and responses. We can use it to connect and test the endpoints of the API that we are calling. This application allows us to see the structure of the data and discover its features by simply entering an authorization token provided by the company and sending a request for the information we want. Figure 5.2 provides an example of sending a request to obtain information regarding different zones that are specified by the PRISM application.

[1] `https://www.getpostman.com/`

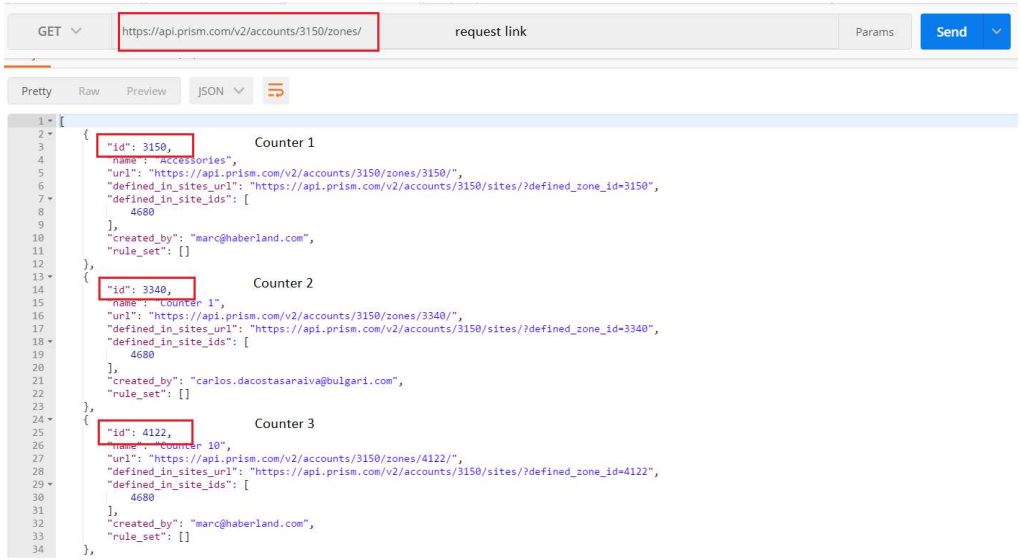

Figure 5.2: Postman Request Example

We then needed to create actual tables in the database to store the data. For that, we used a Hana Editor from the workbench, where we established a connection with Postman in JavaScript:

```
Var aConnection = $.db.getConnection("API::annonuser")
```

We needed to provide a statement to load the data into the table, to set the schema and delete the existing data in the table so as to avoid having duplicate entries. As a result, we obtained a clean table. We request the data just as in Postman so that we could parse it into the database form. It is very important to specify the request link and the format of the data (JSON structure in our case).

Taking into account that JSON can describe either one or more objects, and we can get more than one object per API call, it is essential to make sure we are not losing any data by creating a loop. For some columns, we tried to clean any null data that may come from the API so that everything was a clearly defined type, such as *site_id* is defined as an integer, *name* is a string.

We provided such queries for each of the counters we want to store the data. As a result, we obtained the table storing information about *dwell*, *occupancy* and *count* for each hour of camera records and for each of the created tables (see Chapter 4.2.2).

One goal of our thesis work, is to propose a permanent solution for the information to be analyzed in real time. For that, we want the database to send a request to the PRISM API and download newly available information on a regular basis. Therefore, we scheduled a job to be executed to store loadings on a daily basis at 01:01 AM. It is common to do such data loadings at night as we are aiming to load the previous day information.

5.3 Data Preprocessing

In this part of the thesis, we provide a general data analysis on each of the tables. For these purposes, we use R language with R Studio [2]. We preprocess the data to understand the quality of the data, drop redundant columns, extract the necessary features and, surely, create our own.

As described in the previous subsection, the data is stored in a Hana Database. We ensure a connection of R with Hana in order to be able to retrieve the necessary tables form the database. Library RJDBC allows to easily connect Hana with R by specifying a username and a password and do the necessary data manipulations in R Studio.

5.3.1 Data Quality

In order to start processing our data, let us present the datasets available to us. We start with the SAP ERP data provided by the management of the store. The first table holds all the items from the store with the description. Specifically, we have:

- **Storage location** - character type, the value is persistent for all the records in the table as it represents the location of our store, which is the only store we have (Dubai Mall);
- **Store name** - character type, the value is persistent for all the records in the table as it represents the name of our store, which is the only store we have (Bulgari);
- **Material** - numeric value, it is unique for each record of our table, thus, we treat it as our primary key in the database (item ID number);

[2] https://www.rstudio.com/

- **Description** - character value, consists of letters and numbers, it has 1095 unique values out of 1112 records;
- **Aesthetic line** - categorical data represented by characters, represents different groups the jewelry belongs to;
- **SBU** - character value, has three different values: JEW:845 records - Jewelery, JHJ: 28 records - High Jewelery, WTC:239 records - Watch;
- **Classification** - character value, has six different types;
- **Category** - categorical data represented by characters, has these following options: Catalog Fine, Catalog Precious, Ladies Watches Repet, Men Watches Repetiti, Catalog HMH Jew, Ladies High End Watc, Other;
- **Business** - categorical data represented by characters, has the following options: Fine Jewelry, Precious Jewelry, Ladies Watches, Men Watches, HMH Jewelry, Ladies High End Watc, Other;
- **Family** - categorical data represented by characters, has the following options: Rings-F-C, Rings-P-C, Ladies Watches Repet, Bracelets-F-C, Men Watches Repetiti, Necklaces-F-C, Other;
- **Catalog** - string values, represent the unique catalog number for each record of the data;
- **Open Orders January 10th 2018** - numerical value, represents the number of pre-orders for January 2018;
- **In Transit January 10th 2018** - numerical value, represents the number of transit orders for January 2018;
- **Actual Stock Qty January 10th 2018** - numerical value, represents the stock quantity for January 10, 2018;
- **Target Stock January 10th 2018** - numerical value, represents the expected stock quantity for January 10, 2018;
- **Unit Sold Past 12 month** - numerical value, represents the amount sold for the past year for each item in the store;
- **Counter # Week #** - for each week there is a counter number for each item, it is a numerical value that represents a specific location number.

The described items dataset has 1,112 records with 69 attributes. The first two columns belong to the store description. In our case, focus is on one specific store and, as a result, all of the values in these two columns are the same. Such information is redundant for our purposes. The following nine columns provide the description of each of the items (its ID, collection, line, category, catalog number, etc.), where *Material* is our unique identifier for the items. The data in these columns is categorical. The next five columns show the information about current open orders, stock information, and items sold within the year; values are numerical and have a lot of blank cells due to the fact that not all of the items are either pre-ordered for the next year or even were purchased last year. The last set of columns provides the information about the location of each item per every week of the year. The reason it is displayed weekly is that some of the items replace previous ones and, thus, there are weeks when some items are not available, but substituted by another ones. However, there is a set of items that never leave their counters.

By analyzing the summary of the dataset, it appears to have many missing values in the *order description* and *location description* columns (as expected and mentioned above). And this actually makes sense because if the item is in the open order it has the number of the ordered items. However, if the item is not pre-ordered there is nothing displayed, which results in a null value in our case. When it comes to locations, the items that are not displayed in a certain week do not have a location number for that week. This brings another set of missing values for us, which are in fact just zeros or non-existing values.

Now, let us introduce the second dataset. The second table includes the transactional data for the period of sales from January 2017 to January 2018 that is received from the customer's ERP.

- **Storage location** - character type, the value is persistent for all the records in the table as it represents the location of our store, which is the only store we have (Dubai Mall);
- **Store name** - character type, the value is persistent for all the records in the table as it represents the name of our store, which is the only store we have (Bulgari);
- **Material** - numeric value, it is unique for each record of our table, thus, we treat it as our primary key in the database (item ID number);
- **SBU** - character value, has three different values: JEW:845 records - Jewelery, JHJ: 28 records - High Jewelery, WTC:239 records - Watch;
- **Business** - categorical data represented by characters, has the following options: Fine Jewelry, Precious Jewelry, Ladies Watches, Men Watches, HMH Jewelry, Ladies High

End Watc, Other;

- **Catalog** - string values, represent the unique catalog number for each record of the data;
- **Unit Sold Past 12 month** - numerical value, represents the amount sold for the past year for each item in the store;
- **Sales Person** - string, represented by a character value, holds the name of the person responsible for the current transaction;
- **Date** - date of the purchase, presented in the following format - dd/mm/yyyy;
- **Invoice value** - numerical value that represent the price of the item euros;
- **Client ID** - numerical value, which is unique for each customer;
- **Ticket** - numerical value, unique transaction ID number for each purchase.

The dataset we are going to discuss is the *Transactions* with the information regarding purchases. It consists of 13 variables and 7,561 records of receipts for the period of one year. As in the previous dataset, the first two columns represent the information about the store, in our case it is the same for all records. The next set of columns is the basic item description with the item ID - *Material*, the main reference from the Items dataset. The column *Unit Sold Past 12 month* shows the quantity of the current item that is purchased per transaction (in most cases it is 1), which yields a low variability of distinct values for such attribute, less effective and therefore, not relevant for our research target. The next set of columns is the information regarding the transaction itself: *Sales person, Date, Time, Invoice value* (price of the items in euros), *Client ID, Ticket* (receipt ID).

The dataset does not have any missing values. However, we can see that one of the columns has a negative value - *Price*. This is unusual for this category, therefore we consider it as an artifact. This might be a return of the purchase or mistake in the records. However, such price distribution might significantly affect our findings later on. Our data does not have any duplicated values. Therefore, no actions should be taking regarding this problem.

By analyzing our transactions, it becomes clear that basket analysis[3] does not fit our approach. Figure 5.3 demonstrates tickets distribution where the horizontal axis represents all the receipts within a year and the vertical axis shows the number of purchased items. We see that, in

[3] http://www.albionresearch.com/data_mining/market_basket.php

most cases, people purchase one or two items at a time. There is a reasonable explanation of that: this is a luxury retailer where the items are relatively expensive and it is not a mass shopping type retailer (for which a market basket analysis might be a reasonable choice).

The customers in such stores are loyal due to the reason they share similar beliefs and value the products (Chapter 1). They visit the store several times a year and buy few items. If we take a look at the *clients' Id* distribution (Figure 5.4), we see that it appears more frequently than the transactional references. Indeed, some of the buyers buy just once, but there are others who pay a visit more often. We see that some of the customers have purchased over 50 items within the past year, but, again, these cases are quite unique and we cannot rely on that and base our layout approach on such purchases.

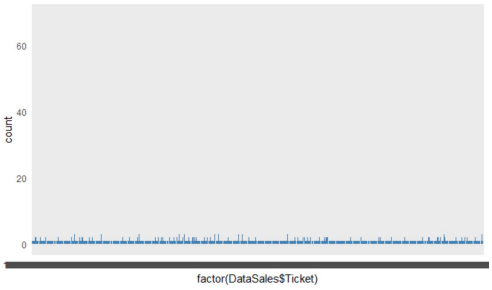

Figure 5.3: Tickets Distribution

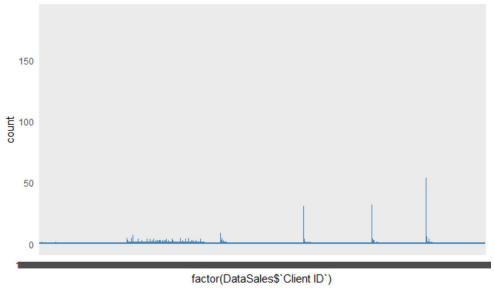

Figure 5.4: Customer ID distribution

Another important aspect we analyze is the frequency of certain purchases. Figure 5.5 displays the frequency plot of the top purchased items within the year 2017.

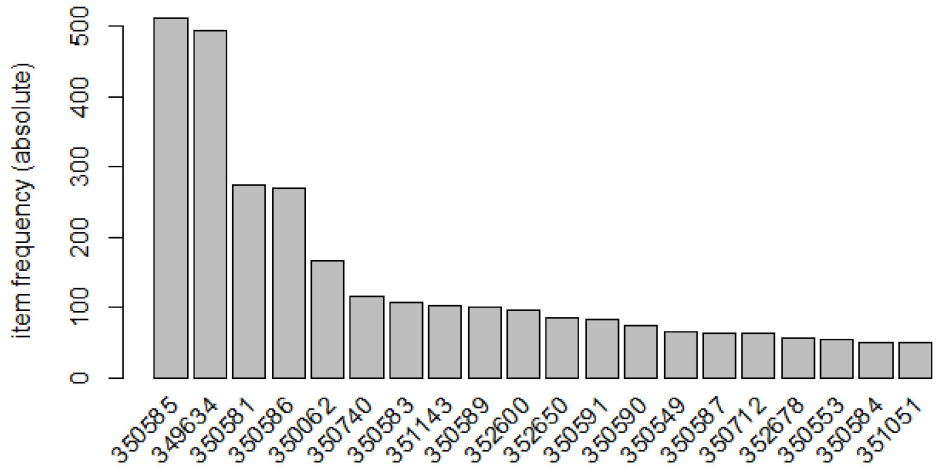

Figure 5.5: Items' Frequency

The plot in Figure 5.5 shows the popularity of the items. It can be observed that some of the items have been purchased around 500 times, where the rest appear just in a few transactions. Also, we know that 343 items have not been purchased at all during 2017 as they do not appear in the transactions, but only on the shelves in the store. This ranking suggests a popularity feature as a way to determine customers preferences.

The next data source we are discussing is the PRISM cloud platform. The camera metrics dataset has 9 variables with 101,070 records, which we successfully loaded to the HANA Database. We automated the loading of new values from the API directly to HANA database every day. For test purposes, we downloaded the data from January 2017 and for a period of one year. The sample dataset is presented in Figure 5.6. Each record represents a specific zone at a certain hour. The unique zone identifier is *zone_id* with its further description. It follows by *site_id*, which is a unique store id - in our case, it is the same due to the fact that we are working with the same store. *Period* is always specified as 'hour', thus it is a redundant column. *Business_hour* has always the same value of "1", which does not bring any variance to our data. The most significant columns are *count*, *avg_dwell*, *avg_occupancy*, *start* and *stop* (the period for the time the information is provided).

```
> head(data)
  zone_id site_id period business_hour count avg_dwell avg_occupancy               start                stop       Date     Time
1    3341    4680   hour             1     0         0             0 2017-09-26T22:00:00 2017-09-26T23:00:00 2017-09-26 22:00:00
2    3341    4680   hour             1     0         0             0 2017-10-02T19:00:00 2017-10-02T20:00:00 2017-10-02 19:00:00
3    3342    4680   hour             1     0         0             0 2017-02-01T16:00:00 2017-02-01T17:00:00 2017-02-01 16:00:00
4    3150    4680   hour             1     7         0             0 2017-10-25T22:00:00 2017-10-25T23:00:00 2017-10-25 22:00:00
5    3342    4680   hour             1     0         0             0 2017-08-09T22:00:00 2017-08-09T23:00:00 2017-08-09 22:00:00
6    3148    4680   hour             1     0         0             0 2018-02-03T19:00:00 2018-02-03T20:00:00 2018-02-03 19:00:00
```

Figure 5.6: Prism Dataset Sample

The data that we also extracted from the PRISM Cloud Platform is the heatmaps and activity maps. Heatmaps are available for each week for the year of 2017. An illustrative example is presented in Figure 5.7. Three colors: red, blue and green change according to people's interaction with the shelves on a weekly basis. In the example below, we see that front counters (tables 1 to 3) were very popular among the customers, whereas the rest of the counters did not bring much attention.

Activity maps represent the crowdedness on the area. People's activity is represented by the blue color which varies depending on the number of people. The more intense it gets, the more people and activity within the area it represents. The example is presented below in Figure 5.8. There we can see that within that particular hour all the front counters were busy with people, where counters at the back (4,5,6,7,8,9,10,11,12,13,14,15) barely had visitors.

Figure 5.7: Heatmap

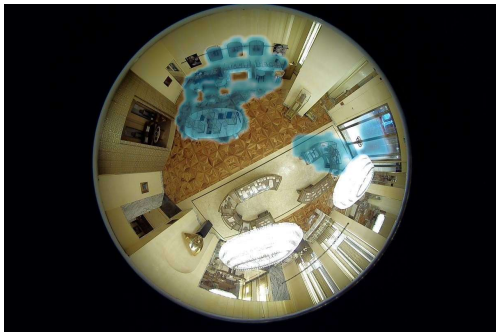

Figure 5.8: Activity map

5.3.2 Data Preparation

In order to prepare the data for the pattern mining and items similarity analysis, we initially decide on the important features to be extracted, merged and modified. We list the required characteristics as follows:

- Identify the location of the items;
- Create items ranking feature;
- It is essential to obtain items' descriptions and missing prices.

First of all, we change all the items' descriptions into categorical values to ease the identification of groups. Moreover, as mentioned earlier in section 5.3.1, the *Items Description* dataset has a number of missing values in the location identifier. To overcome this problem, we merge the columns that describe the locations into one and replace the missing values this way by substituting them from different weeks. An illustration of this process is presented below (Figure 5.9).

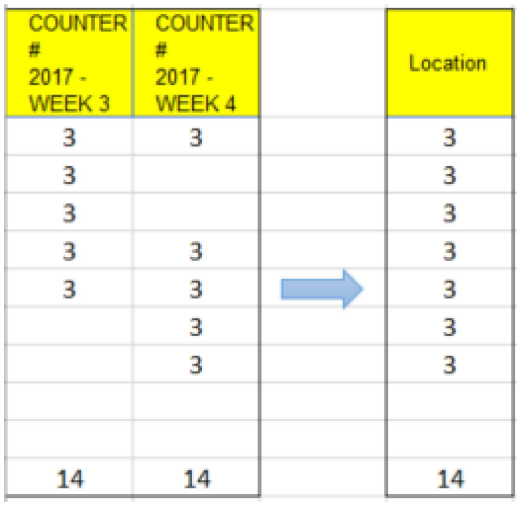

Figure 5.9: Missing Values Treatment

From this image, we can see that some of the items have not been displayed within the year at all and thus, could not be purchased. For our analysis we remove them from the dataset as they are irrelevant for our analysis.

In both datasets (*Items Description* and *Sales Transactions*), there are columns with the store information (*Storage location* and *Store name*) that we find irrelevant for analysis and they are thus removed from the dataset. Also, we decided to remove columns with the stock information

(*Open Orders, In Transit, Actual Stock Qty, Target Stock, Unit Sold Past 12 months*), due to their of variability.

In order to obtain more information about our items, we take the price of the items from the invoices and add it to our description. Some of the items have not been purchased within the year and thus, we do not have any invoice information for these items. Such cases are describe in the next section: Feature Extraction, where we extract price information for them.

The PRISM cloud dataset that we obtained in the previous subsection 5.2, includes 74,858 entries of *dwell*, *occupancy*, and *count* for each zone in the store for each hour of the 2017 year. A significant part of the dataset has zero values due to two different reasons. First of all, we obtained the information for all the hours the camera was working. The business hours of the store are from 10 am to 11 pm from Sunday to Thursday and from 10 am to 00 am on Friday and Saturday as the week starts on Sunday in the Middle East. Therefore, we need to remove such zero values for each non-working hour.

From the PRISM data, we also decided to remove *site_id*, which holds the store id number. In our case, it is the same and thus, redundant for our analysis. Also, the *start* and *stop* columns are displayed as timestamps, representing the beginning and the end times for a certain operation such as *count*, *dwell* or *occupancy*. From such records, we want to obtain date and time separately.

For example (Fig. 5.10),

Figure 5.10: Missing values treatment

5.4 Feature Extraction

5.4.1 Missing prices extraction - Web Scraping

As mentioned in chapter 2, the customer does not provide us with an accurate pricing information about the items. Therefore, we are using the invoice values, defined in Section 5.3, to determine

the price of the items. Yet, the challenge lies, mainly, in the list of items that have not been purchased within the provided data year, 2017, in which data was recorded. That means, we lack any information in respect to the value of the items, despite its importance for the further analysis and items relocation consideration. Our goal is to increase sales by targeting the purchase of specific items. And, that makes the insights about the actual values of the products essential for the target. To overcome this problem, we rely on Web Scraping techniques, described in the paper [GPLLF+13], as one of the methodologies for data extraction.

Bulgari's website, described in Section 4.2.2., acts as an international online store. That said, all the prices are available online. We want to retrieve price information for the items that has no price in our datasets. For this purpose, we use a Python Web Scraping library named Beautiful Soup[4], suggested by Daniel Glez-Peña and Anália Lourenço [GPLLF+13], as a Python library.

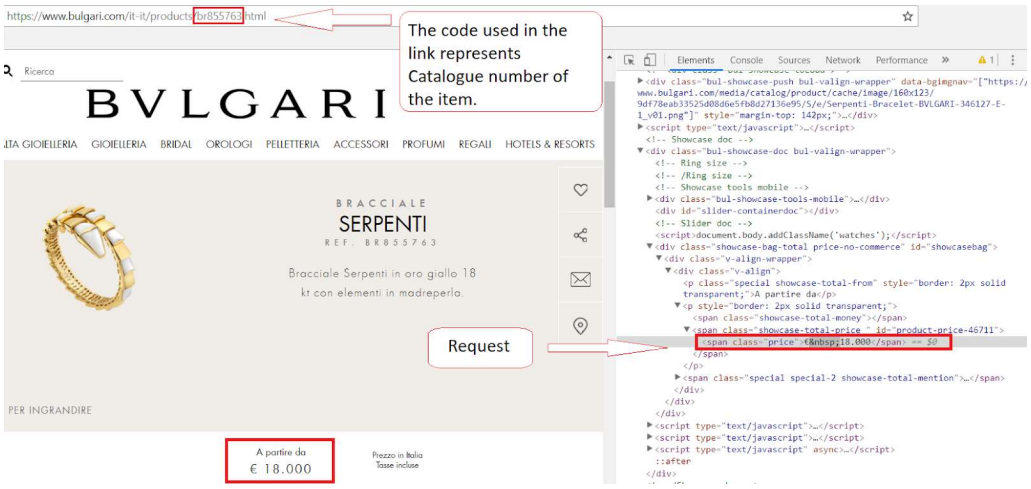

Figure 5.11: Bulgari Web Page

[4] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

```python
from urllib2 import Request, urlopen
from bs4 import BeautifulSoup
import re
import unicodedata

def getBeautyPrice(quote_page):
    req = Request(quote_page, headers={'User-Agent': 'Mozilla/5.0'})

    webpage = urlopen(req).read()
    soup = BeautifulSoup(webpage, 'html.parser')

    price_box = soup.find('span', attrs={'class':'price'})
    price = price_box.text
    return unicodedata.normalize('NFKD', price).encode('ascii','ignore')


print getBeautyPrice("https://www.bulgari.com/it-it/products/br855763.html")

```

```
18.000
```

Figure 5.12: Web Scraping Example

One of the challenges here is that the URL link of the item is not always consistent and thus, there are two different ways to build the correct request. The first one is that the link can be built by putting together the prefix `https://www.bulgari.com/it-it/products/` that follows by the material number (ID of the item) and a fixed extension. For example, "https://www.bulgari.com/en-it/products/350062-*e.html*", where number stands for the item id number and italic extension is the fixed extension suitable for these cases.

The second option is that the fixed prefix follows by the Catalog number. For example, "https://www.bulgari.com/en-it/products/br855118.*html*", where br855118 is the catalog number of the item and in italic is the fixed suffix for these cases. Therefore, we need to consider these two cases in order to be able to extract as many prices as possible. Figures 5.11 and 5.12 demonstrate the process of obtaining the price directly from the webpage of the customer.

Another issue that we had to face is that the online store does not have all the products available that are in the physical store, however, the various changes based on the selected country. We decide to use Italy and Great Britain as our main online stores due to the fullest variety of our products and no need in changing the currency based on the exchange rate as the prices in UK pounds correspond to exactly the same prices in euro.

Firstly, we extracted the information from the Italian version of the website (European format). We found that some of the items' prices were still not retrieved. We decided to move to the UK webpage following the assumption that the prices are similar compared to those in other versions of the website. However, we encountered a different issue, the same request was not available anymore due to the different HTML format of a price element (Figure 5.13). We considered different cases of markup elements, text patterns, and CSS alignment. Hence, we removed the British Pound (GBP) character from the scrapper response and considered different text encodings before saving it in the right clean format.

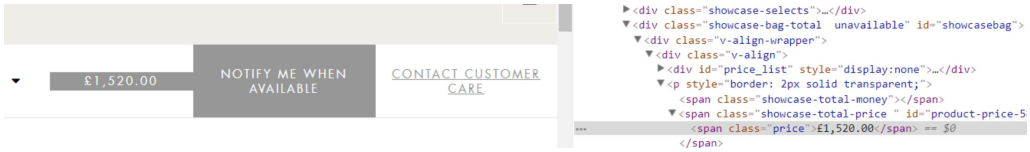

Figure 5.13: Web Scraping UK Price Example

The pseudocode of the described procedure is presented below:

```
For each item (row) in the dataset:
        HTTP_URL = url_prefix + item['material'] +url_postfix
        if item['price'] == 0:
                item['price'] = getHtmlElement(HTTP_URL)
```

The function *getHtmlElement* considers the website in different locations, languages, and currencies (compared to Euro). We optimized this function until reaching a success rate of 82%, that left us only with 61 items zero-priced. For data quality considerations, these items were omitted from the dataset. To be precise, such method provides us 282 values out of 343 missing records. This way we could collect the prices of 921 out of 982 items.

Apart from price extraction, we perform similar (web scraping) procedure to obtain the description of the items. Example from the official Bulgari web page is presented in Figure 5.14. This provides us an additional feature that will be used later on. It is very important to extract the information in the same language to be able to compare our records.

Figure 5.14: Description Example

5.5 Image Processing

As part of our thesis proposal, we are going to use the data obtained by the activity heatmaps. In order to extract meaningful information from the given maps, we adopted the reverse engineering techniques described below.

5.5.1 Activity Map Data Extraction

In this subsection, we focus on data preparation and ETL process that involve image processing. The novelty of the research is emphasized on the way we deal with the deficit of the raw data, as described in Chapter 2. In fact, there was no raw data provided for each counter separately to distinguish the number of people at each zone. Under such condition, the best solution would be to analyze the blue heatmaps where the density of the color (levels of blueness) represents the number of the people in each area. As we do not know the exact number of people, occupancy and dwell for each counter separately, we decided to represent each zone of the heatmap in the percentage format (rational numbers between 0 and 1). And, then approximate the metrics obtaining more data for the further analysis.

To ungroup the counters and obtain the missing data we reverse engineered the areas by using image processing frameworks. The steps are as follows:

1. We define the exact pixels area in each counter, as shown on Figure 5.15.

2. We subtract an original camera shot (with exact same resolution and dimensions) of the store from the snapshot with heatmaps as proposed by [LZP14]. This gives a clear new image with shades of blue only as shown on Figure 5.16 (left image is the original heatmap, right image is the subtracted one). The subtraction is done for each pixel in both photos respectively.

3. We calculate the 'blueness' of each zone. Considering the fact pixels are described as sets of Red/Green/Blue in the shape of (R,G,B) tuples, we are interested in the average B from all the pixels in each zone.

4. We normalize the numbers among each area, and among the overall store (depends on the different needs). For instance, we can tell that at a specific moment the density of blue of the grouped table of the counters $\{1,2,3\}$ was distributed as of: counter_#1=30%, counter_#2=45%, counter_#3=25%.

This way we obtained the values for the front counters (tables 1 to 3). By knowing the exact values of these counters, we calculated the values of each counter separately with respect to the whole store.

Figure 5.15: Counters Split in Prism

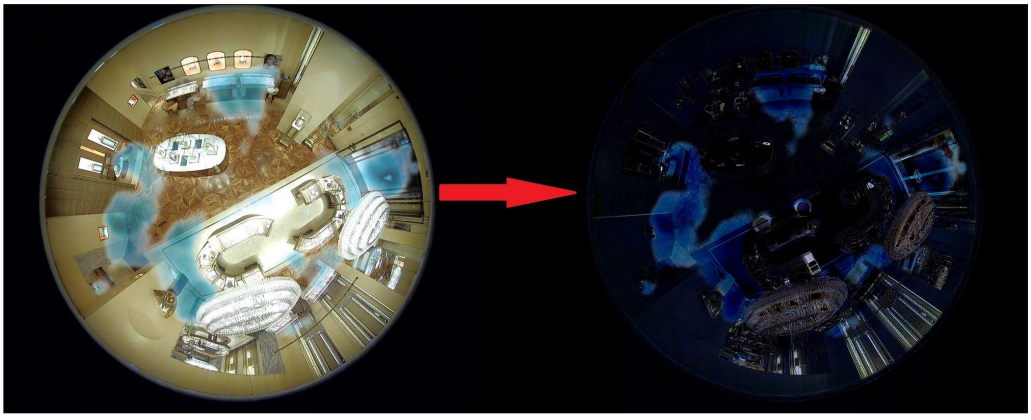

Figure 5.16: Color Transformation

As an example, we provide the following values that we managed to extract from the picture displayed in Figure 5.16:

```
{'upper': (0.11, 0.42, 0.48)}
```

This says that by analyzing three upper tables (counter 1, 2 and 3 as defined in Figure 2.2.), we obtain the following results: 11% of the overall activity within these counters belongs to the first table, 42% represents table 2 and counter 3 has 48% of the activity. Also, as shown, the last table (Counter 3) has the highest blueness comparing to the other two neighboring tables.

In order to evaluate our results and understand how close we can estimate the numbers to the real values, we use the full data collected from the period February 26 to March 2, 2018. We extract the data from the heatmaps, then compare it to the real values provided by the camera. The results are presented in Figures 5.17 and 5.18:

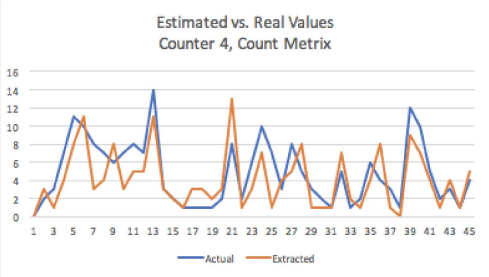

Figure 5.17: Value Comparison, Counter 4

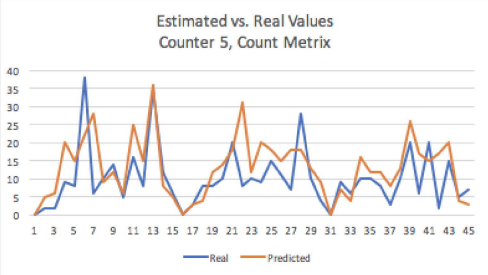

Figure 5.18: Value Comparison, Counter 5

The chart above shows the difference between estimated and real values for the count of people for 3 days period. As an example, we present counter 4 and 5 with the 'count' metrics. The orange values represent the estimated values that we extracted via heatmap reverse engineering procedure. The blue values are the ones we could collect from the counters 4 and 5 from the camera in that specific period (February 26 to March 2, 2018). As we can see the difference in values lies within a small range.

5.5.2 Feature Creation

5.5.2.1 Clustering

Generally, some useful variables can be hidden in a large quantity of raw data, and thus can be obtained through data integration and transformation by applying some of the techniques. The

main idea of clustering our items is to put the similar once in the same group to be able to place them together based on certain characteristics later on.

Our idea of clustering is to create a new feature for the *Description* of the Items, which we use later on to build a similarity matrix. For this purpose, we are going to use items features and the purchase information we have. In order to create an additional feature for better clustering, we apply apriori rules to detect shopping patterns. The idea of such algorithm is to find the items that are usually bought together. Taking into consideration the fact that we are talking about the luxury store and most of the receipts have only one item purchased at a time, specifically there are 6269 unique transactions out of 7561 total records, which means that less than 17% of people buy several items at a time. To overcome this problem, we decide to perform apriori analysis based on customers ID and group all their purchases within a year in one transaction per each customer. This approach gives us 5109 unique transaction, which means that 32% of people buy several items - twice more than the standard market basket analysis allows us to collect. This approach will help us to find out which items the customers might be interested in after purchasing something before, however, 68% of the purchases with only one item (when the customer visited the store) is still not sufficient in order to be able to perform the layout rearrangement. Therefore, we decide to use transaction frequency function to determine the most purchases items and assign popularity feature. Such feature will help to perform better clustering.

The feature we want to find is the items ranking or its popularity within customers. In order to do so, we are using frequency algorithm as described in Chapter 4.3. The values are distributed between 0 and 1. We assign the highest items frequency to the items that are purchased most of the times. On contrary, the items that are never purchased within the whole year get the value 0 as we assume they are not popular and do not attract buyers.

Now we can proceed with the clustering approach. The clustering process itself contains 3 distinctive steps:

- Calculating dissimilarity matrix - is arguably the most important decision in clustering, and all the further steps are going to be based on the dissimilarity matrix you've made;
- Choosing the clustering method;
- Assessing clusters.

Dissimilarity Matrix The crucial step for clustering is to build a dissimilarity matrix. Dissimilarity matrix is a mathematical expression of how different, or distant, the points in a data set are from each other. It helps, later on, to group the closest ones together or separate the furthest ones - which is a core idea of clustering.

This is the step where data types differences are important as dissimilarity matrix is based on distances between individual data points. It is easier to imagine distances between the numerical data point, however, when it comes to the categorical data, there is no such an obvious way. In order to compute dissimilarities in this case, we decide to use a Gower distance[5], which calculates the pairwise distances between observations in the data set. In order to create a dissimilarity function, we are using daisy() with metric = c('gower') from the cluster package in R.

Hierarchical Clustering The main input for our clustering algorithms is the dissimilarity matrix obtained in the previous subsection. There are various functions available in R to compute hierarchical clustering. The commonly used ones are:

- **hclust** (in *stats* package) and agnes (in *cluster* package) for agglomerative hierarchical clustering (HC)

- **diana** (in *cluster* package) for divisive HC

Divisive Clustering For Divisive clustering we are using diana function from the cluster package in R. We draw a dendrogram for a better visualization. The results are presented in Figure 5.19.

[5] https://www.rdocumentation.org/packages/cluster/versions/2.0.6/topics/daisy

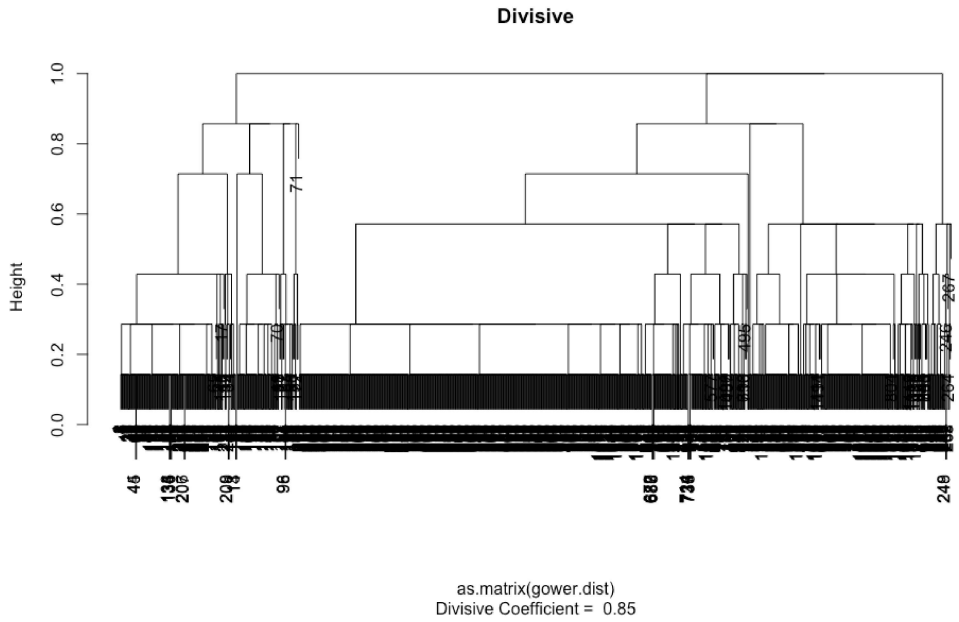

Figure 5.19: Divisive Clustering

Agglomerative Clustering The results are presented in Figure 5.20.

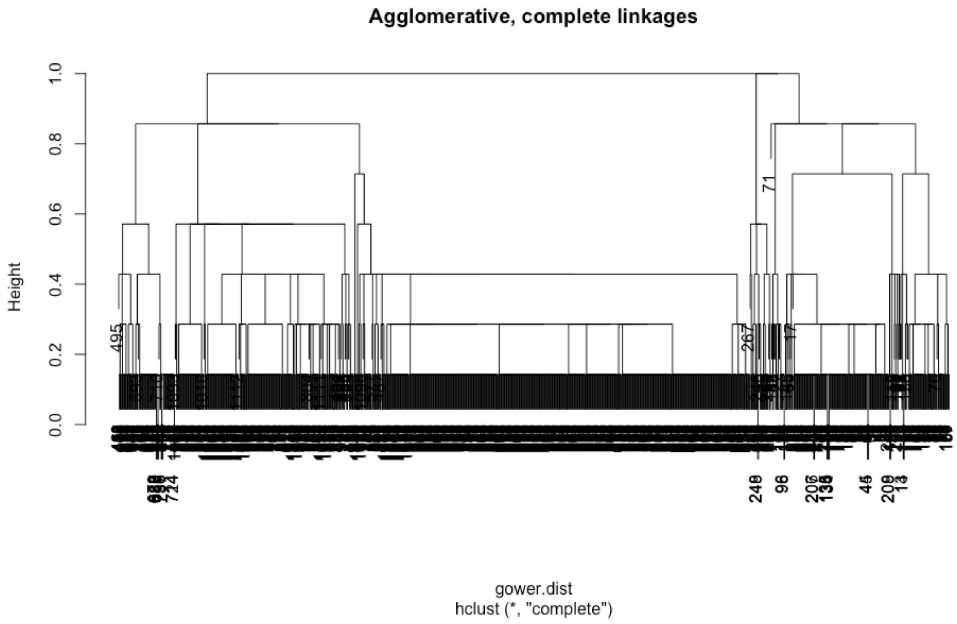

Figure 5.20: Agglomerative Clustering

In the dendrograms displayed above (Figure 5.19 and Figure 5.20), each leaf corresponds to one observation. As we move up the tree, observations that are similar to each other are combined into branches, which are themselves fused at a higher height. Due to the high number of values, it is nearly impossible to distinguish them on the plot as our horizontal axis.

The height of the cut to the dendrogram controls the number of clusters obtained. It plays the same role as the k in k-means or k-modes clustering. In order to identify subgroups we can cut the dendrogram but for that we need to evaluate our clusters and decide on the number of clusters we want to obtain.

Assessing clusters As previously discussed in Chapter 4, we will be focusing on two main approaches to evaluate our clusters. We start with some of the measurements to analyze (Figure 5.21 and Figure 5.22):

```
> stats.df.divisive
                   Test 1  Test 2  Test 3  Test 4  Test 5  Test 6
cluster.number       2.00    3.00    4.00    5.00    6.00    7.00
n                 1112.00 1112.00 1112.00 1112.00 1112.00 1112.00
within.cluster.ss  185.81  169.92  150.78  143.12  140.55  139.98
average.within       0.52    0.50    0.49    0.49    0.49    0.49
average.between      0.90    0.90    0.89    0.89    0.89    0.89
wb.ratio             0.58    0.55    0.55    0.55    0.55    0.55
dunn2                1.58    1.57    1.40    1.41    1.39    1.34
avg.silwidth         0.41    0.43    0.45    0.46    0.46    0.46
Cluster- 1  size   239.00  239.00  154.00  154.00  154.00  154.00
Cluster- 2  size   873.00   28.00   85.00   59.00   59.00   59.00
Cluster- 3  size     0.00  845.00   28.00   26.00   18.00   18.00
Cluster- 4  size     0.00    0.00  845.00   28.00    8.00    7.00
Cluster- 5  size     0.00    0.00    0.00  845.00   28.00    1.00
Cluster- 6  size     0.00    0.00    0.00    0.00  845.00   28.00
Cluster- 7  size     0.00    0.00    0.00    0.00    0.00  845.00
```

Figure 5.21: Assessing Divisive Clustering

From the table (Figure 5.21) above we can see that average.within, which is an average distance among observations within clusters, is shrinking, so does within cluster SS. Average silhouette width has the reverse relationship. The size of clusters is disproportional in its turn.

```
> stats.df.aggl
                      Test 1  Test 2  Test 3  Test 4  Test 5  Test 6
cluster.number          2.00    3.00    4.00    5.00    6.00    7.00
n                    1112.00 1112.00 1112.00 1112.00 1112.00 1112.00
within.cluster.ss     184.72  169.92  163.67   99.09   98.59   90.31
average.within          0.51    0.50    0.49    0.35    0.35    0.34
average.between         0.90    0.90    0.83    0.79    0.79    0.79
wb.ratio                0.56    0.55    0.60    0.44    0.44    0.43
dunn2                   1.41    1.57    0.94    0.75    0.76    0.82
avg.silwidth            0.41    0.43    0.15    0.34    0.33    0.34
Cluster- 1  size      267.00  239.00  239.00  239.00  238.00  220.00
Cluster- 2  size      845.00   28.00   28.00   28.00    1.00   18.00
Cluster- 3  size        0.00  845.00  769.00  530.00   28.00    1.00
Cluster- 4  size        0.00    0.00   76.00  239.00  530.00   28.00
Cluster- 5  size        0.00    0.00    0.00   76.00  239.00  530.00
Cluster- 6  size        0.00    0.00    0.00    0.00   76.00  239.00
Cluster- 7  size        0.00    0.00    0.00    0.00    0.00   76.00
```

Figure 5.22: Assessing Agglomerative Clustering

Agglomerative complete linkage hierarchical clustering is more balanced when comparing on the number of observations per group (Figure 5.22).

Now let us focus on the number of clusters and decide how many to keep. We start with the Elbow method. It shows how the within sum of squares - as a measure of closeness of observations: the lower it is the closer the observations within the clusters are - changes for the different number of clusters. Ideally, we should see a distinctive 'bend' in the elbow where splitting clusters further gives only minor decrease in the SS.

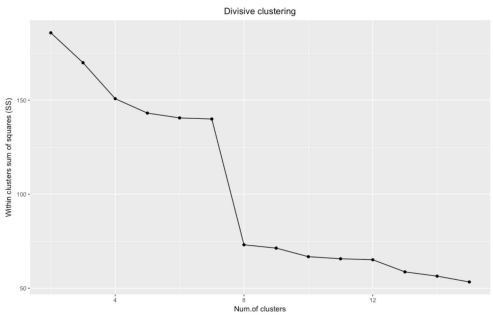

Figure 5.23: Divisive Elbow

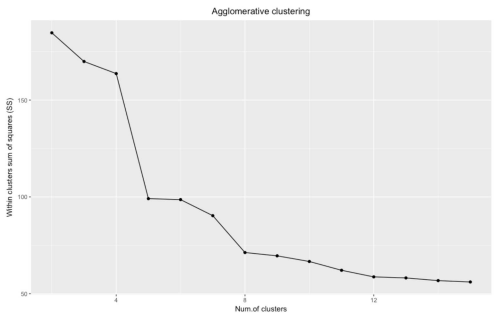

Figure 5.24: Agglomerative Elbow

Both algorithms provide different choices for the number of clusters. It is not straightforward from the plots above (Figure 5.23 and Figure 5.24) to decide on the number of clusters as the elbow is not direct. Figure 5.23 shows two elbows, which is not correct according to the method. In this case, splitting the picture in two, we can say that the best number of clusters will be either 4 or 8. Figure 5.24 does not provide the precise number either, we assume that the elbow is at number 5 or 8.

Since the above evaluation is not convincing, we move to the next method - silhouette.

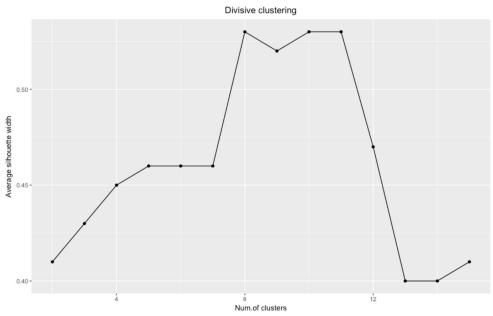

Figure 5.25: Divisive Silhouette

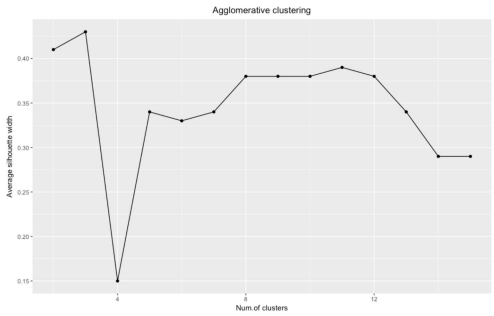

Figure 5.26: Agglomerative Silhouette

In the silhouette method, the rule indicates that we should choose the number that maximizes the silhouette coefficient because we want clusters to be distinctive (far) enough to consider them separate. Figure 5.25 suggests that the best number of clusters will be at 8, however, on the other hand, Figure 5.26 maximizes its silhouette coefficient at 3.

The nature of categorical data type poses some limitations on the data visualization. We want to

- see how observations are clustered,
- know is how observations are distributed across categories - a colored dendrogram created.

Evaluation methods suggest that the best number of clusters lies at the number 8. However, if we recall the number of observations per group in the clusters (Figure 5.21 and Figure 5.22), one of them turns to have only 1 value. We do not want that as it is not the goal. The idea of our clustering is to group the most similar items together based on their characteristics. Such cases

happen when there is an outlier in the dataset. We have already evaluated our data and disregarded the outlier option. Therefore, we decide to move on with 5 clusters using agglomerative method as it provides the most balanced groups of items. The colored dendrogram is presented below in Figure 5.27.

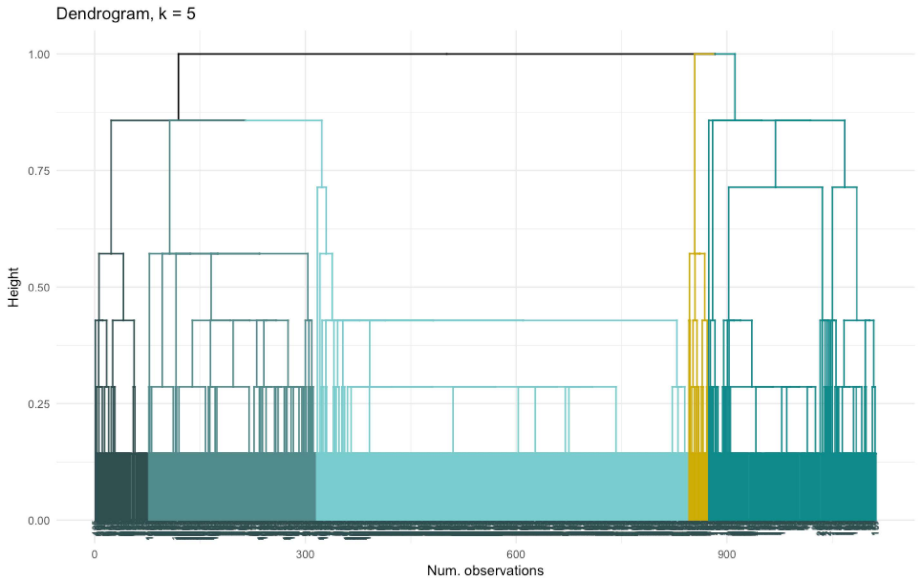

Figure 5.27: Hierarchical Clustering - Dendrogram

After we decided on the number of clusters, we add them as a new feature to our dataset and moreover, we add them to the description of the items to perform a similarity in the next section 5.6.

5.6 Items Similarity

To begin with, we start our approach with organizing the similarity matrix between the item located in the store. We have a full table of items with the description in the following format (Figure 5.28):

102179	BZ23BSCL	Earrings Merging two of the most iconic symbols of Bulgari design, the Serpenti earrings coil the sinuosity of the snake with the contemporary soul of tubogas. Evoking both the sensual curves of a woman and the fluid shape of the serpent, the earrings are crafted with the shapely lines of the tubogas technique, with a flexible and tubular litheness. Radiating glamour and a truly individual style, this Serpenti jewel is magnetic as its effortless. Serpenti Tubogas short earrings in 18 kt rose gold with pave diamonds.
102087	BZ23BSGCC/12.M	Watch Quartz movement. 23 mm steel and 18-ct rose gold case, 18-ct rose gold bezel, black ceramic ring. Black lacquered dial set with diamond indexes. Medium size black ceramic bangle.

Figure 5.28: Items Description

The description has been extracted from the web using web scraping techniques. The textual content is unique for each individual item. There are items with the similar or almost the same description, which gives them a higher similarity value in a matrix. In order to take into consideration the characteristics mentioned in Section 5.5, such as *Family Line, Collection, Category*, etc., we assign cluster identifiers to the items and merge it with its description. Concerning the textual processing (or: natural language processing), we tokenize the text to get separate tokens for better performance of the similarity algorithm. Experimentally, we found that stemming or lemmatizing lead our corpus to poor results, hence, it was avoided.

As discussed in the previous Chapter 4, we are using cosine similarity matrix. The implementation is done in Python using sklearn library. In order to implement this algorithm we align item's full description in one column, creating a vector of attributes, where each of them represents the frequency of a particular word,phrase contained in the table (document). Our functions compares the angles between the vectors using dot product and provides, as a result, the similarity matrix.

The diagonal of the matrix is always 1 as it refers to the similarity between the same item. Also, we observe that the first few items have a higher similarity between each other as the first and last items. We can explain it as the items from the same collection or category are usually grouped and displayed together in our original dataset.

5.7 Shopping Patterns

The next step in our pipeline is to find shopping patterns. For this, we are using *Dwell* information of each counter for each working hour of the day and find the location of the maximum dwell per hour. This is the counter where customers spend the most time at a certain hour in the store. This means that if the highest dwell at 11 am is at the counter 2, within the last hour the majority of

customers spend most of the time at that counter. However, it is not always true that these people buy from the same table. This is why we are aiming to find such combinations of counters where people spend time at one place but buy from a different one. To find that, we use transactional data to obtain the information about the counters where the purchase have been made. We group all our transactions, as described in the previous chapter 4, by hour and find the counter form which the item has been purchased.

After that, we join the result by the hour and find the top most frequent combination. As a result, we obtain Figure 5.29, where *PurchaseLocation* represents the counter number the item has been purchased from, *MaxDwell* represents the counter with the highest dwell for a certain hour and count stands for the number of cases such combination appears.

PurchaseLocation	MaxDwell	Count
4	D1	104
4	D15	97
4	D19	71
19	D1	44
1	D15	43
4	D3	39
19	D15	31
1	D1	30
4	D2	23
13	D15	23
19	D19	21
19	D2	21
19	D3	20

Figure 5.29: Purchase - Dwell Location

5.8 Layout Development

The next step in our pipeline is actually to develop the algorithm which tells us how to group some of the items in order to boost sales. There are few techniques that lead to systematic deviations from a standard of rationality or good judgment when shopping. That is what we focus on.

At this stage of the project we have data in the following format:

- Table 1

Table 5.1: Obtained Table 1 Example

PurchaseLocation	MaxDwell	Count

- Table 2

Table 5.2: Obtained Table 2 Example

Material	Descrip.	Aesthetic Line	SBU	Classific.	Category	Business	Family	Location	Invoice value

- Table 3

  Similarity matrix of the items.

We apply the our idea of using the items that people buy the most and finding the appropriate group of three to place them together.

The pseudo-code is described below:

```
For each Location in Table 1:
    Find 3 most ranked items in Table 2:
      For each item find the 2 items with highest similarity form Table 3, where:
        each newItem.Location = HighDwellLocation from Table 1
        and
        newItem.Price > Item.Price
        and
        newItem.Rank < Item.Rank
```

This way we propose three new improvements for each of the tables in the store. The items that used to be in a high demand before, now are offered as a cheaper version of the similar item and, according to the proposed idea, customers should shift their preferences towards the item with a higher price as subconsciously it looks like a better deal in the given combination.

As a result, we obtain sets of three with one of the items significantly cheaper than the other two. The algorithm searches for the closest item to the given one from a specified location.

Sometimes it happens that two locations have a different type of products, such as watch and rings. It seems that the items have nothing in common, but research proves that in this case, people might consider purchasing the combination of these items as an addition to the originally desired one.

As a result of our findings we obtain the following combination of items:

Figure 5.30: Necklace 1

Figure 5.31: Necklace 2

Figure 5.32: Necklace 3

The first item (Figure 5.30) is the one that has the highest rank overall, comes from location number 4, its price is 2300€. The other two items belong to the counter number 1 with the lower ranking and higher price. Item two has a price of 3860€ and item 3 has a price of 4600€. According to the Decoy effect, the Decoy item in our case is the third one as its price is significantly higher and comparing to the second necklace and it does not have any diamonds. This reason should trigger customers to switch from item 1 towards item 2.

Another example in Figures 5.33, 5.34, and 5.35 follows the same principle. The second (5.34) and third (5.35) items are significantly more expensive than the previous one. There two items offer better characteristics and thus, should trigger customer towards the middle option.

Figure 5.33: Bracelet 1

Figure 5.34: Bracelet 2

Figure 5.35: Bracelet 3

The results presented in Figure 5.33, Figure 5.34 and Figure 5.35, according to our approach should trigger customer to purchase bracelet 2. The price of the first item is 560€, and it has the highest ranking among customers. It can be explained by the pricing as it is relatively low comparing with the other bracelets in the store. Second item costs 2600€. It is an elegant golden bracelet with amethyst stones. Third item is a thick heavy bracelet, has few stones, and its price is 7630. It is more than 10 times of what the customer is planning on spending. As a result of such rearrangement, it will be easier to convince customer to purchase the second item as in such content the first one seems simple and cheap.

The full set of results can be found in Appendix A.

Chapter 6

Proposed Evaluation

Our project has been conducted having strict time and resource limitations. The duration of the project was set to be four months, specifically from February until June 2018. During the project, we were limited with our resources and could not evaluate our solution. In this chapter, we propose a plan for the evaluation of the developed approach and future assessment guidelines that we recommend the management to follow, or rely on.

According to the empirical studies on stores layout design [LZ07], customers tend to purchase the items in the middle of their path as these items are the most noticeable. We propose to start the changes in the main counters located in the center of the room. Also, the desired item (the one we are aiming to switch the customer to) should be in the middle of our three options.

6.1 A/B Testing

One of the simple ways to know about customer preferences is to conduct statistically powerful A/B tests. These tests are increasingly popular in business as a policy to evaluate potential innovations. The key insight of this testing is that the optimal experimentation strategy depends on whether it gains accrue from typical innovations or from rare and unpredictable large successes that can be detected using tests with small samples. The [AAMO$^+$18] with its theoretical results and empirical analysis suggests that even simple changes to business practices dramatically

increase innovation productivity.

A/B testing is a very popular approach in the web development industry. Companies test the best web page layout in order to increase the number of people engaged in some activity. The idea is that one half (or: some) of the visitors will be shown version A – otherwise known as the control page – and another half (or: the remaining) will be shown version B, the variation page. At the end of the testing period, both pages are compared to find out which one outperforms the other [AAMO+18].

When it comes to the physical store layout testing, we would like to propose the following technique: half of the store will be rearranged according to our approach and another half maintains the initial layout chosen by the management of the store. The suggested evaluation is offered for the duration of 10 weeks. The implementation is based on the ranking of the items (the highest ranked item is the most purchased and vice versa). We propose to start with the less popular items changing the layout gradually. To recall the original items layout described in Chapter 2, there are two types of items: ones that are always on the counters and others that appear every four to seven weeks.

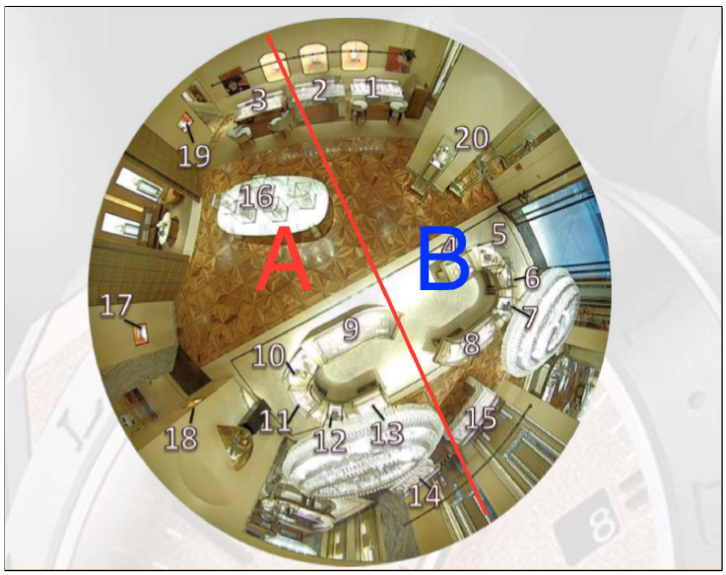

Figure 6.1: A/B Test 1

Testing will be split into two parts: firstly, changing the layout in section A (Fig 6.1) for the

period of 5 weeks (as it is an average change of the layout), while section B (Fig 6.1) maintains the layout defined by the management. After this period, we propose a change where section A returns to its initial layout and section B changes the layout according to the one suggested by us. After ten weeks period of both A and B rotations, we will obtain customer behavior information for both parts of the store with two different layout patterns. Then, we will be able to compare two layout schemas and evaluate if the 'middle' item approach works and how it benefits the store. The goal is to see if the purchases of the proposed items have increased, taking into consideration the overall sales within this period of time (10 weeks).

It is important to mention that the suggested testing technique is recommended to be conducted during a stable sales period. By stable period we define the period of no important holidays, such as Christmas [1], Eid Al-Fitr [2], Eid Al-Adha or Mother's Day, etc. Also, we need to assure that no special promotions are launched within the testing time that can affect our sales. Another benefit is emphasized in a [SSH99] with respect to the fact that the middle choice preference can add an additional value to the strategy. By placing the preferred option in the center, we add another influencing factor that will help customers choose that option.

Such evaluation techniques provide clarification to the innovative approach and explain the overall logic behind it. In fact, the results of the evaluation can be taken as indicator exhibiting the added value of our solution. Then, the organization can decide whether it worth adoption or not. Further work will be discussed according to the obtained results of the evaluation.

[1] https://www.britannica.com/topic/Christmas

[2] https://www.aljazeera.com/indepth/features/2016/07/eid-al-Fitr-160701164352978.html

Chapter 7

Conclusion and Future Work

7.1 Concluding remarks of the presented approach

In this master thesis, we deliver a proof of concept for layout rearrangement of a luxury jewelry store, Bulgari, using Machine Learning techniques. The used data in our research consists of 1) The store sales data of 2017, 2) Recorded customer behavior analytics (Prism), and; 3) Scrapped online shop data. Firstly, we designed and implemented ETL process to load, store, transform, and merge the data. This process involved image processing, web scraping, and other data preparation and feature extraction techniques. Using the outcome of this step, we further performed pattern mining, clustering, and similarity matrix calculation using NLP. Lastly, we proposed an algorithm based on Decoy theory to optimize the store design layout of Bulgari, Dubai.

We present our contribution as of twofold. First, we propose an end-to-end solution for an automated extraction of recorded camera data of customer behavior. To the best of our knowledge, this is the first research that provides a solution for building customer behavior patterns based on extracted data from raw filmed heatmap photos taken from Prism Skylabs. Second, we provide a new layout for a luxury fashion retailer based on the Decoy Pricing theory, optimizing customer's sales and increasing the average receipt value. To the best of our knowledge, this is the first research to combine Decoy Theory with user behavioral patterns and design principles.

Due to the novelty of the research and overall complexity of the layout-related studies, the

quality metrics for validating the approach were missing, which called for an exploratory analysis. Chapter 6 suggests an evaluation technique in order to assess our findings and develop a future strategy. We expect this kind of work to help luxury fashion retailers to monitor purchasing behavior of the customers. In addition to that, our approach suggests the necessary layout rearrangements in order to maximize sales and increase the average receipt value.

7.2 Limitations and obstacles

As for any work that deals with complex information, during the elaboration of our study we came across the certain limitations described in this section. Firstly, we were guaranteed by the customer to obtain the following data:

- Sales data for both years 2017 and 2018;
- Prism analytics per counter in the store form 2017 and 2018.

Within the first phase of the project, we learned that Bulgari's Prism subscription has come to an end. That said, Prism data could no longer be available. We could use everything that has been recorded up to that moment (e.g. heatmaps, activity maps, metrics recorded for several counters), but could not obtain any new values. That left us with only one year of data (2017) delivered by Bulgari's management.

Taking into consideration the lack of provided data, we had to adjust our approach accordingly. We decided to focus on one-year data, extracting the necessary metrics from the activity maps obtained from Prism cloud platform during 2017. To do so, we applied an image processing technique to reconstruct values for *occupancy*, *dwell* and *count* for each counter separately by analyzing the degree of the blue color (heat density) of each image.

Furthermore, the similarity calculation phase presented some difficulties with respect to the algorithm(s) of choice. The main challenge hereby was comparing items using only the available categorical description, especially due to the fact that many items have the same characteristics. This will cause our similarity matrix to have many values of '1' (or: 100%), which indicates full similarity. To overcome this problem, we extracted individual items' description form the webpage, using web scraping. Then, we applied NLP techniques and created a vector for each

item in the store. This helped us to obtain a vector similarity matrix while enriching the corpus with text obtained directly from the official online store of Bulgari.

One more significant challenge is reflected in the fact we could not evaluate our solution using the new data (2018), as promised by the management at the beginning of the project, in February 2017. This imposed a limitation to provide a quality metrics for validating our approach. To overcome this obstacle, we provide a detailed evaluation methodology which consists of AB testing validation technique and a future plan of the implementation strategy.

7.3 Future Work

Chapter 6 suggests an assessment strategy to be applied in order to obtain a quality metrics for the evaluation of our approach. This is the first step in the future work plan. After testing our model, we will obtain results on how well it performs. The necessary adjustments will be applied according to the obtained quality metrics.

As a next step, in the further development of this work, we suggest following [TTW07] that proposes the idea of clustering stores based on similar sales and finalizing the layout for each group specifically. The idea is to choose a test store in each cluster that minimizes the cost of forecast errors where total seasonal sales are the predictor variable for its cluster. In this case, it is important to pay attention not only to the sales mix and how it differs across the stores but, also, to the timing of the sales. According to the authors of the book, Southern stores sell springsummer merchandise earlier than northern stores. We propose to explore this approach across several stores within the Bulgari chain.

Appendix A

Dashboard Development

For our customer the visual results are very important. In this chapter we present a dashboard created in SAP Lumira[1]. The dashboard highlights the main activity in the store.

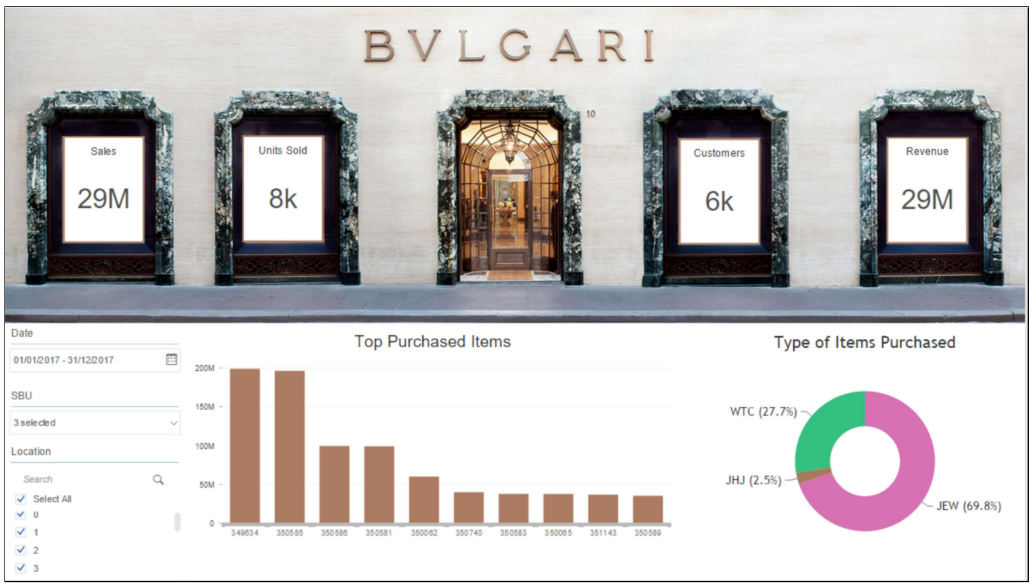

Figure A.1: Dashboard

[1] https://saplumira.com/

Figure A.1 elaborates the visualization of the store performance dashboard. The dashboard is divided into three subgroups: KPI views, most purchased items, and; sales per product type. It offers a summary view of the sales performance within a specific period of time.

The first group of KPI corresponds to the sales in euros, units sold, a number of customers, and total revenue for a specific amount of time. The second group displays the most purchased items. It is set to visualize the top ten most popular items in the store. The last chart delivers the information regarding the type of the items purchased. All of the charts can be sorted by a certain time period, a group of the product, and its location. That said, the management can see the top sold products from a certain counter, the performance of the counters, and the revenues they bring.

After we obtain the results of the proposed evaluation (Chapter 6), we will create a second part of the dashboard to visualize the layout corresponding to our findings.

Bibliography

[AAMO+18] Eduardo M Azevedo, Deng Alex, Jose Montiel Olea, Justin M Rao, and E Glen Weyl. A/b testing. 2018.

[AJ14] Elavarasi Anitha and Akilandeswari J. Categorical data clustering using frequency and tf-idf based cosine similarity. In *disciplinary Research in Egineering and Technology*, pages 39–43. Icidret, 2014.

[AW95] Dan Ariely and Thomas S Wallsten. Seeking subjective dominance in multidimensional space: An explanation of the asymmetric dominance effect. *Organizational Behavior and Human Decision Processes*, 63(3):223–232, 1995.

[Ban79] Sharon K Banks. Gift-giving: A review and an interactive paradigm. *ACR North American Advances*, 1979.

[BC83] Gabriel Biehal and Dipankar Chakravarti. Information accessibility as a moderator of consumer choice. *Journal of Consumer Research*, 10(1):1–14, 1983.

[BGP94] Julie Baker, Dhruv Grewal, and Ananthanarayanan Parasuraman. The influence of store environment on quality inferences and store image. *Journal of the academy of marketing science*, 22(4):328–339, 1994.

[BLG92] Julie Baker, Michael Levy, and Dhruv Grewal. An experimental approach to making retail store environmental decisions. *Journal of retailing*, 68(4):445, 1992.

[Bou04] Nicholas Boughen. *Lightwave 3D 8 Lighting*. Wordware Publishing, Inc., 2004.

[Bro82] David Brooks. *Lenses and lens accessories: a photographer's guide*. Curtin & London, 1982.

[CD05] Chia-Hui Chang and Zhi-Kai Ding. Categorical data visualization and clustering using subjective factors. *Data & Knowledge Engineering*, 53(3):243–262, 2005.

[CGBN12] Malika Charrad, Nadia Ghazzali, Véronique Boiteau, and Azam Niknafs. Nbclust package: finding the relevant number of clusters in a dataset. *UseR! 2012*, 2012.

[CGC01] Anil Chaturvedi, Paul E Green, and J Douglas Caroll. K-modes clustering. *Journal of classification*, 18(1):35–55, 2001.

[del] The luxury opportunity. *Deloitte Fashion and Luxury*.

[DL00] Manoranjan Dash and Huan Liu. Feature selection for clustering. In *Pacific-Asia Conference on knowledge discovery and data mining*, pages 110–121. Springer, 2000.

[DS99] Ravi Dhar and Itamar Simonson. Making complementary choices in consumption episodes: Highlighting versus balancing. *Journal of Marketing Research*, pages 29–44, 1999.

[Ede79] Craig Edelbrock. Mixture model tests of hierarchical clustering algorithms: The problem of classifying everybody. *Multivariate Behavioral Research*, 14(3):367–384, 1979.

[Eve74] Brian Everitt. Cluster analysis 122, 1974.

[FMRZ17] Emanuele Frontoni, Fabrizio Marinelli, R Rosetti, and Primo Zingaretti. Shelf space reallocation for out of stock reduction. *Computers & Industrial Engineering*, 106:32–40, 2017.

[GEP14] Vadim Grigorian and Francine Espinoza Petersen. Designing luxury experience. 2014.

[Gos12] A Ardeshir Goshtasby. Similarity and dissimilarity measures. In *Image registration*, pages 7–66. Springer, 2012.

[Gow71] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

[GP02] Paolo Giudici and Gianluca Passerone. Data mining of association structures to model consumer behaviour. *Computational Statistics & Data Analysis*, 38(4):533–541, 2002.

[GPLLF+13] Daniel Glez-Peña, Anália Lourenço, Hugo López-Fernández, Miguel Reboiro-Jato, and Florentino Fdez-Riverola. Web scraping technologies in an api world. *Briefings in bioinformatics*, 15(5):788–797, 2013.

[Gri05] David A Griffith. An examination of the influences of store layout in online retailing. *Journal of Business Research*, 58(10):1391–1396, 2005.

[GRS00] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366, 2000.

[HDX06] Zengyou He, Shengchun Deng, and Xiaofei Xu. Approximation algorithms for k-modes clustering. In *International Conference on Intelligent Computing*, pages 296–302. Springer, 2006.

[IM09] Dino Ienco and Rosa Meo. Distance based clustering for categorical data. In *SEBD*, pages 281–288. Citeseer, 2009.

[ins13] Dec 2013.

[JH95] Bharath M Josiam and JS Perry Hobson. Consumer choice in context: the decoy effect in travel and tourism. *Journal of Travel Research*, 34(1):45–50, 1995.

[KAG08] Kevin Lane Keller, Tony Apéria, and Mats Georgson. *Strategic brand management: A European perspective*. Pearson Education, 2008.

[Kar01] George Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 247–254. ACM, 2001.

[KB12] Jean-Noël Kapferer and Vincent Bastien. *The luxury strategy: Break the rules of marketing to build luxury brands*. Kogan page publishers, 2012.

[KC11] Ralph Kimball and Joe Caserta. *The Data WarehouseÂ ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. John Wiley & Sons, 2011.

[Lew94] Dale M. Lewison. *Instructors manual and transparency masters to accompany retailing*. Macmillan College Pub. Co., 1994.

[LZ07] J Lynchjr and G Zauberman. Construing consumer decision making. *Journal of Consumer Psychology*, 17(2):107–112, 2007.

[LZP14] Daniele Liciotti, Primo Zingaretti, and Valerio Placidi. An automatic analysis of shoppers behaviour using a distributed rgb-d cameras system. In *Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on*, pages 1–6. IEEE, 2014.

[MCL03] Nikos Mamoulis, David W Cheung, and Wang Lian. Similarity search in sets and categorical data using the signature tree. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 75–86. IEEE, 2003.

[MDP+07] Bruno Meunier, Emilie Dumas, Isabelle Piec, Daniel Bechet, Michel Hebraud, and Jean-Francois Hocquette. Assessment of hierarchical clustering methodologies for proteomic data mining. *Journal of proteome research*, 6(1):358–366, 2007.

[MM01] Bill Merrilees and Dale Miller. Superstore interactivity: a new self-service paradigm of retail service? *International Journal of Retail & Distribution Management*, 29(8):379–389, 2001.

[NS51] Jerzy Neyman and ELIZABETH L SCOTT. *Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, 1951.

[Oko16] Uche Okonkwo. *Luxury fashion branding: trends, tactics, techniques*. Springer, 2016.

[Pic04] Massimo Piccardi. Background subtraction techniques: a review. In *Systems, man and cybernetics, 2004 IEEE international conference on*, volume 4, pages 3099–3104. IEEE, 2004.

[Por98] Michael E Porter. *Clusters and the new economics of competition*, volume 76. Harvard Business Review Boston, 1998.

[SCZ00] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. Wavecluster: a wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal—The International Journal on Very Large Data Bases*, 8(3-4):289–304, 2000.

[Sim99] Itamar Simonson. The effect of product assortment on buyer preferences4. *Journal of Retailing*, 75(3):347–370, 1999.

[SSH99] Jerel E Slaughter, Evan F Sinar, and Scott Highhouse. Decoy effects and attribute-level inferences. *Journal of applied psychology*, 84(5):823, 1999.

[SWY75] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[TS93] Amos Tversky and Itamar Simonson. Context-dependent preferences. *Management science*, 39(10):1179–1189, 1993.

[TTW07] Christopher S Tang, Chung-Piaw Teo, and Kwok-Kee Wei. *Supply chain analysis: a handbook on the interaction of information, system and optimization*, volume 119. Springer Science & Business Media, 2007.

[VJ17] Franck Vigneron and Lester W Johnson. Measuring perceptions of brand luxury. In *Advances in Luxury Brand Management*, pages 199–234. Springer, 2017.

[VODS04] Adam P Vrechopoulos, Robert M O'keefe, Georgios I Doukidis, and George J Siomkos. Virtual store layout: an experimental comparison in the context of grocery retail. *Journal of Retailing*, 80(1):13–22, 2004.

[ZMRA13] Marie Lisandra Zepeda-Mendoza and Osbaldo Resendis-Antonio. Hierarchical agglomerative clustering. In *Encyclopedia of Systems Biology*, pages 886–887. Springer, 2013.