From Aisles to Insights: Decoding Association Rules and Seasonal Cues in Grocery Shopping

Ariana-Andra Șerpar
Faculty of Mathematics and Computer Science
West University of Timișoara
Timișoara, Romania
ariana.serpar00@e-uvt.ro

Erik Zsolt Varga
Faculty of Mathematics and Computer Science
West University of Timișoara
Timișoara, Romania
erik.varga98@e-uvt.ro

***Abstract*—The amount of data created in commercial contexts calls for the implementation of Data Mining (DM) techniques. This allows companies to gain important information about the behavior of the customers as well as the popularity and performance of the products.**

This paper aims to show how DM can be applied to rudimentary transactional data and how much knowledge can be extracted from seemingly basic information. In order to do that, we implement association rule mining on a very simple dataset and analyze seasonal trends by calculating various measurements. Additionally, we perform clustering to gain more insight about the behavior of the clients. The results reveal promising relationships between products, the impact of seasonal changes and potential client categories.

***Index Terms*—Data Mining, association rules mining, seasonal trends, market basket analysis, clustering**

I. Introduction

Now more than ever, the quantity of data is steadily increasing and shows no signs of stopping soon [11]. Huge amounts of data are collected in a large variety of domains, such as healthcare, business or production.

However, all collected data is utterly useless if it is not used in an efficient manner. In order to mean something, the data must be structured in a way that allows us to gain knowledge from it, in whatever way possible. This is the moment when data becomes information, and the most reliable way to not only create information, but also learn from it, is through Data Mining (DM).

DM is a complex process that allows us to identify underlying patterns, trends or relationships between pieces of data [2]. These relations are most often invisible to the human eye and must be performed by computers. DM can be divided into five essential steps that guarantee effective insight extraction:

- *Collection*, which involves extracting raw data from a variety of sources (web logs, sensors, transactions, etc.)
- *Cleaning*, a step that consists of removing inconsistencies and errors, to get more accurate results
- *Processing*, which relies on various techniques to eliminate missing data and present what is left in a standardized manner
- *Analysis*, a task directly tied to identifying regularities (and thus irregularities), associations and trends within the data
- *Knowledge absorption*, arguably the most important part, as the absence of it deems all other steps insignificant. The ability to interpret, explain and understand the obtained results in a human way is the very aim of DM.

DM helps with predictions, decision-making, detecting anomalies as well as outliers and also provides recommendations. One of the fields that seems to make the most out of the opportunities provided by DM is marketing [5]. The large amount of data in marketing can be easily explained in relation to human-related activities. This aspect is to be expected, as humans generate data in a continuous manner, whether they are aware of it or not.

From a business perspective, companies can gain valuable information through the use of DM techniques. The promise of better understanding the customers, along with their behavior and way of thinking is an enticing premise for any business. By knowing their clients, companies will be able to attract more supporters, which of course results in more profit and opportunities for growth.

The aim of this paper is to put into perspective how important DM truly is in a business context, by presenting methods and results that show how much information can be acquired from basic transactional data.

The structure of this paper involves a Theoretical Framework presented in Sec. II, which includes details about association rules mining and clustering. Section III describes the dataset we have chosen, with its two separate files. A short introduction to the implementation and the packages used is presented in Sec. IV. Subsequently, we dive into association rules mining in Sec. V, then we take a look at seasonal trends in Sec. VI. Section VII provides information about the implementation of a clustering algorithm. Lastly, conclusions are drawn in Sec. VIII.

II. Theoretical framework

A. Association Rules

Association rules mining is a type of DM task that is specifically designed for the analysis of transactions [6]. The main idea of this approach is to pinpoint the co-occurrence of items or sets of items in lists of transactions. This concept is the foundation of the "Frequently Bought Together" sections that exist on e-commerce websites.

As the name suggests, the final results of this technique involve a set of rules that contain the items of the transactions. The quality of the established rules is quantified using three specific measures: support, confidence and lift [4].

Support. This metric shows the reliability of an association rule, by taking into consideration how many times the items are indeed bought together out of all transactions. To put it plainly, the support is the frequency of an itemset.

Let X and Y be two items that exist in a list of transactions. In that case, the support of X and Y will be calculated using the following formula:

$$\textit{Support}\{X,Y\} = \frac{\textit{Nr. of transactions with both X and Y}}{\textit{Total nr. of transactions}}$$

Confidence. Calculated through the support metric, confidence is meant to present how probable the co-occurrence of the items in a transaction truly is. It is worth mentioning that the confidence threshold must be supplied by the user, as he decides the minimum value that is still considered acceptable.

Taking into consideration the notations previously used in the case of the support metric, confidence is computed using the following equation:

$$\textit{Confidence}\{X,Y\} = \frac{\textit{Support}\{\text{X, Y}\}}{\textit{Support}\{\text{X}\}}$$

Lift. Also called "interest", this particular metric is supposed to quantify the novelty of a rule. If a rule is interesting and can be used to gain insights, its lift would not be close to 1. Therefore, in order for the lift to be considered satisfactory, its value must be either lower or higher than 1.

Like confidence, lift is also calculated by using support and by applying the formula:

$$\textit{Lift}\{X,Y\} = \frac{\textit{Support}\{\text{X, Y}\}}{\textit{Support}\{\text{X}\}*\textit{Support}\{\text{Y}\}}$$

One of the most common algorithms employed for generating rules for frequent itemsets is the Apriori algorithm [1]. The first step of the algorithm is to compute the support of each item individually. After analyzing the results, a support threshold must be established, as the items that have values lower than the threshold value cannot be considered frequent.

Subsequent to the removal of the infrequent items, the support of each itemset is computed. An itemset is constructed by making all possible combinations between the frequent items. It is best practice to calculate the support of itemsets with two items first and then move on to three items. Therefore, the last itemset will be the largest, containing all frequent items.

Upon performing all the necessary computations, what is left to do is to generate the association rules and compute the confidence for each rule. An association rule between X and Y can be read as "IF X THEN Y". After confidence is calculated, the last step is to compute lift.

B. Clustering

Clustering is a DM technique dedicated to categorizing data. The base idea is that items that belong in the same cluster are more similar to each other than they are similar to items from other clusters [7]. Clustering can be used for image segmentation and data summarization (where documents with the same content are grouped together), but it is most frequently used for customer segmentation.

Depending on the particularities of the dataset and what we want to achieve, clustering can be done using various types of algorithms. Clustering algorithms can be hierarchical, density-based, partitional and probabilistic. As there is a large amount of variation between each type of algorithm (with each one having diverse techniques), we chose to focus particularly on the hierarchical algorithms, as they are the ones we employed in the implementation portion of our paper.

Hierarchical clustering algorithms combat one of the biggest shortcomings of partitional algorithms, the fact that the number of clusters must be supplied prior to seeing any results. This category of algorithms aims to arrange the partitions in a hierarchical manner (as the name would suggest) [9].

There are two main approaches to hierarchical clustering, which we will describe in the following paragraphs.

- *Agglomerative clustering*, which starts off with only one item in each cluster and then merges the clusters that are similar until all items belong to one large cluster.
- *Divisive clustering*, which begins the implementation with one cluster that contains all values and then recursively splits it until each cluster contains only one item.

It is worth mentioning that, in our implementation, we considered more appropriate to use the agglomerative clustering method. When it comes to deciding which clusters should be merged together, there are various (dis)similarity measures that can be applied. The characteristics of each measure are described below.

- *Single linkage*, which shows the smallest distance between points that belong to different clusters.
- *Complete linkage*, the opposite of single linkage, presents the largest distance between two points that do not belong to the same cluster.
- *Average linkage*, which quantifies the average distance between all the points that belong to different clusters.
- *Ward criterion* [8], the most complex dissimilarity measure, merges the two clusters that produce the smallest variance within the cluster. In our implementation, we considered that the Ward linkage would provide the best results due to its intricacy.

In order to perform clustering in our implementation, we used Jaccard similarity to ultimately compare the clients. If I_A and I_B are two sets that contain item A and B, the Jaccard similarity is calculated by performing the following computation [3]:

$$J(A,\ B) = \frac{I_A \cap I_B}{I_A \cup I_B}$$

III. Dataset description

The selected dataset for this particular paper is called “Groceries dataset for Market Basket Analysis (MBA)” and was posted on Kaggle by Rashik Rahman [10]. The dataset has a Kaggle usability score of 10,00 and a Creative Commons license, which makes it available to the public at large.

The dataset is divided into separate files, to aid the implementation of various tasks. The most rudimentary one is meant to be used in combination to the Apriori algorithm and is called “basket.csv”.

This specific *.csv* file has 10 columns and 14,963 rows. Each row is meant to be interpreted as a transaction of variable length. Naturally, the length of each transaction in this file cannot exceed 10 products, as it would not be registered in its entirety.

The products that exist in each transaction are presented in the form of nominal attributes and range from household items to edible articles. If a transaction is shorter than the 10 column length, the rest of the spaces are completed with the NaN descriptor.

The second and last file of our chosen dataset is dubbed “Groceries data.csv” and it is arguably more complex than the “basket.csv” file. Due to its attributes, this dataset can be used for a variety of tasks, unlike the previous dataset (which was designed only for association rules usage). The size of this dataset is considerable, with 7 columns and 38,765 entries. The name of the features and the data type are described in Table I.

TABLE I
”Groceries data.csv” feature description

Column Name	**Data Type**
Member_number	numerical
Date	date (YYYY-MM-DD)
itemDescription	categorical
year	numerical
month	numerical
day	numerical
day_of _week	numerical

IV. Implementation prerequisites

For transparency and reproducibility reasons, we deem necessary to disclose the Python (version 3.10.12) packages used throughout our implementation and their respective versions. All information is supplied in Table II.

TABLE II
Package versions

Package	**Version**
google.colab	0.0.1a2
numpy	1.22.4
pandas	1.5.3
sklearn	1.2.2
scipy	1.10.1
matplotlib	3.7.1
opendatasets	0.1.22
apyori	1.1.2

V. Association Rules Implementation

A. Preprocessing

After importing the dataset and visualizing the items in the “basket.csv” file as a dataframe, we may begin the preprocessing steps. These are mandatory, as we have to transform the data regarding transactions into a form that can be used in combination with the Apriori algorithm.

The items from each row are transposed into arrays of various lengths (that match the length of the transactions). We remove the NaN values by comparing each item to itself. If the bool value of the operation is *True*, we append the item to the transaction array. Otherwise, we do not. This comparison is possible because a NaN value will never be equal to itself.

The resulting transaction arrays are concatenated into a larger array that centralizes all the useful data from the dataframe. It is important to keep the transactions separated within the large array, otherwise we will not achieve what we aim for. By achieving this, the preprocessing stage comes to a close and we can begin the exploratory analysis of the transactions.

B. Apriori Algorithm Implementation

The implementation of the algorithm starts off with defining a function that would present the association rules in a more appealing and easily understandable manner. We decided to print the base item or itemset first, then the additional items. For interpretation purposes, the confidence and lift of each rule are also displayed.

The array containing all transactions is fed into the Apriori algorithm and the required parameters are set. Because we want to see all the rules at first, the initial parameters are set as low as possible. Therefore, support is set to 0.001, while confidence and lift are both equal to 0.

After displaying and analyzing all the resulting rules, a new threshold is set, in order to allow us to only keep the association rules that are indeed useful. The support remains the same, with a value of 0.001, but the confidence is set to 0.14 and the lift to 1.1. The results are shown, explained and discussed in the following subsection.

C. Results

Ultimately, the initial application of the Apriori algorithm (with lower thresholds) is able to find 750 association rules within the 14,963 transactions. This may seem like a significant amount, however, most of the rules just associate the base item or itemset with itself. This produces a large number of rules where the lift is 1, which does not allow us to gain any insights in regards to the transactions. Such results are not useful neither from an exploratory nor from a marketing perspective.

When analyzing the confidence of the generated rules, we were able to determine that most of them do not exceed a value of 0.1. On the contrary, most of the values are closer to a tenth of this number. We attribute these low confidence values to the large number of transactions, because the increase in transaction number lowers the probability of two or more items being bought together very frequently.

After the confidence and lift thresholds are lifted, the association rules that pair an item with itself are successfully removed. The combination of 0.14 confidence and 1.1 lift results in 9 association rules that are shown in Table III.

The best confidence is achieved by the association rule "IF {yogurt, sausage} THEN {whole milk}", which has a value of 0.2558. The best lift is 1.9117, and it belongs to the association rule "IF {whole milk, sausage} THEN {yogurt}".

The high number of rules that contain the items *rolls/buns* and *whole milk* is attributed to the high frequency of these items in this specific list of transactions.

TABLE III
TOP 9 BEST ASSOCIATION RULES

Base Item/Itemset	Additional Item	Confidence	Lift
packaged fruit/vegetables	rolls/buns	0.1417	1.2884
processed cheese	rolls/buns	0.1447	1.3157
seasonal products	rolls/buns	0.1415	1.2863
semi-finished bread	whole milk	0.1760	1.1148
rolls/buns, soda	other vegetables	0.1404	1.1506
rolls/buns, sausage	whole milk	0.2125	1.3455
sausage, soda	whole milk	0.1797	1.1383
whole milk, sausage	yogurt	0.1641	**1.9117**
yogurt, sausage	whole milk	**0.2558**	1.6198

VI. Seasonal Influence on the Shopping Pattern

A. *Preprocessing*

Given the dataset with columns such as those from Table I, the preprocessing for this task revolves around establishing the season each transaction was made in.

Therefore, we assigned seasonal labels to each month of the year, in a intuitive manner. Thus, the transactions associated with the month values 1, 2 or 12 will have an additional label that reads "Winter", the ones with month values 3, 4 and 5 are labelled "Spring" and so on. These labels are included in a new column of the dataframe, called "season".

B. *Shopping Pattern Analysis*

To help us analyze seasonal patterns, we defined several measures that we considered susceptible to change. The first measure is the total number of items sold during each season. As we can see in Figure 1, the highest number of items was sold during summer, followed by spring, while winter and fall follow closely behind with almost identical values.

The second measure is highly related to the first one, as we counted the number of items sold in each season again, but this time we included the days of the week in the measurements. We might have expected to see more transactions made on weekends, as Saturday and Sunday are usually the days when families tend to shop for the whole week ahead. However, as presented in Figure 2 this was not the case. With respect to the seasonal changes, there were no differences between the number of items the clients bought on each day of the week.

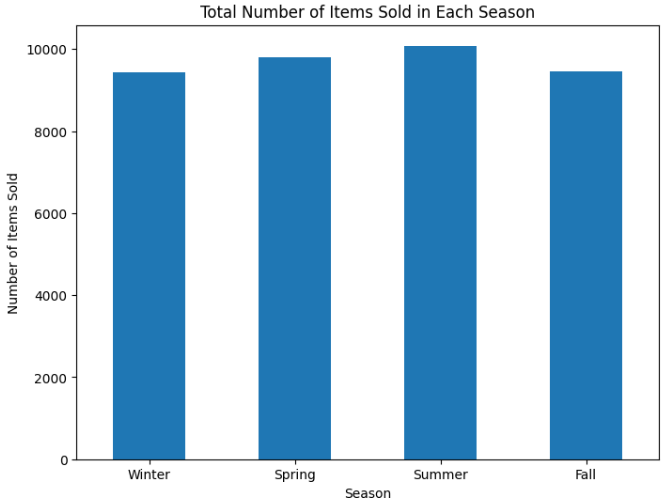

Fig. 1. Total number of items sold in each season.

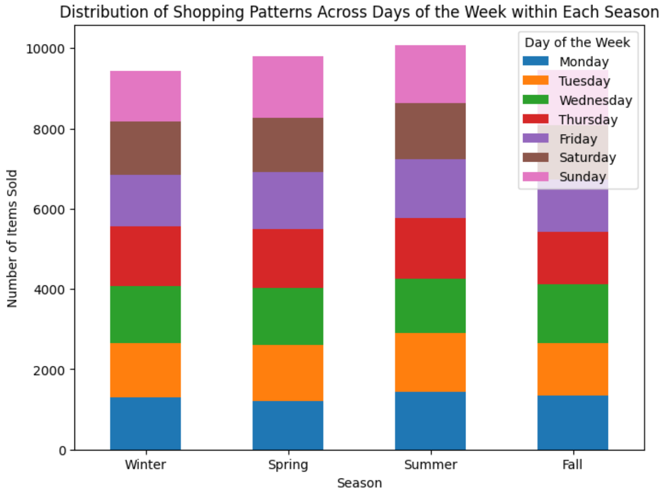

Fig. 2. Distribution of shopping patterns across days of the week within each season.

Going further, we analyzed the total number of transactions the clients have made in each season. The number of transactions in each season (Figure 3) is highly correlated with the number of items the clients purchased in each season. As expected, the more transactions the clients make, the more items will be sold in total.

Regarding the average number of items purchased per transaction in each season, although their distribution across seasons (Figure 4) still looks very similar to the number of items and number of transactions in each season (with winter and fall being the lowest), we can see that the differences between seasons are almost insubstantial compared to the previous measurements.

The last measurement for seasonal influence that we conducted was to compare the top 10 most sold items in each season (Table IV). Until the 6th rank, all seasons share the same top 5 most sold items. Starting from the 6th rank, we

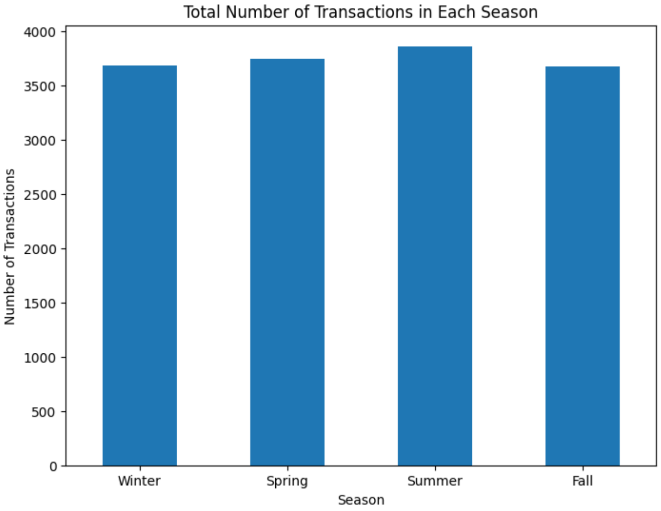

Fig. 3. Total number of transactions in each season.

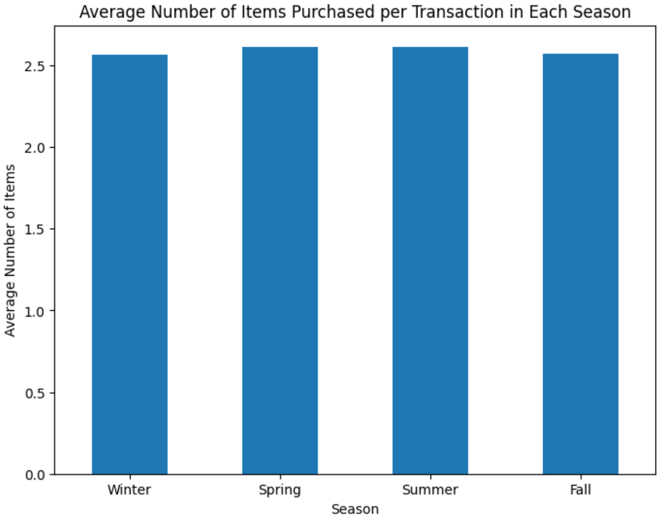

Fig. 4. Average number of items purchased per transaction in each season.

can observe that in the spring season, the 6th most sold item is “tropical fruit” instead of “root vegetables” like in the other seasons. As for the 7th rank, we find that the item “tropical fruit” is the 7th most sold item (which was the 6th in the case of spring), with the exception of the spring season where “bottled water” takes the 7th spot. Ranks 8, 9 and 10, although they exhibit some similarities, we cannot attribute a specific item to each place, rather there are several items.

C. *Results*

As an overall review of this section, we can clearly state that summer (closely followed by spring, winter and fall) was the best season if we consider the number of items and transactions as the most important factors for evaluating the market. Despite the fact that the season has a small influence on shopping patterns, as observed in all measurements, the main shopping patterns remain the same across all seasons. For example, the top 5 most sold items are always the same.

VII. Client clustering based on items

A. *Preprocessing*

For preprocessing, we created four dictionaries, one for each season, to store the items purchased by clients without duplicates. We iterated through the grocery data and assigned each item to the corresponding season’s dictionary. Using a defined function, we calculated the Jaccard similarity between lists of items for each client. We obtained four similarity matrices, one for each season, representing the similarities between clients based on their purchased items. It is important to note that while we separated the data by season in this section, our main focus was on comparing similarities and clustering, rather than inspecting the influence of seasons on shopping patterns.

B. *Client Similarity Analysis*

Before delving into the analysis of similarities between clients, an interesting finding that emerged was the distribution of clients across seasons, as shown in Table V. These findings are intriguing because, in contrast to the results presented in Section VI, the highest number of clients was observed in spring rather than in summer, despite the latter season witnessing the highest volume of item sales and transaction count.

Despite dividing the measurements into four sections for each season separately, we can draw the same conclusion regarding the client similarity measurements. As shown in Table VI, regardless of the season, the most similar clients consistently possess a significant number of items, with these items typically being the most popular ones. Consequently, these clients exhibit high similarity to other clients. On the other hand, the least similar clients have very few items in all cases, reducing the likelihood of similarity to other clients, as none of their items appear in the top 10 most sold items. On average, the most similar clients display a similarity of around 10% to all other clients, whereas the least similar clients exhibit a similarity of approximately 0.1%.

C. *Agglomerative Clustering*

For clustering, we applied an agglomerative clustering algorithm separately for each season, using three different distance thresholds (9, 8, and 8.5). This resulted in a total of 12 clustering models. Despite using the same distance threshold for each season, it led to a different number of clusters for each (Table VII). In this paper, although the clusters differ across seasons, we can still draw conclusions by analyzing just one season.

Regarding the winter season, a distance threshold of 9 resulted in 2 clusters: (1) a cluster with a diverse range of clients, including all clients that could not be assigned to the second cluster, and (2) a cluster consisting of clients who purchased “other vegetables”. When the distance threshold was set to 8.5, the number of clusters increased to 3: (1)

TABLE IV
TOP 10 MOST FREQUENTLY SOLD ITEMS IN EACH SEASON

	Winter		**Spring**		**Summer**		**Fall**	
Rank	**Item**	**Frequency**	**Item**	**Frequency**	**Item**	**Frequency**	**Item**	**Frequency**
1	whole milk	559	whole milk	652	whole milk	662	whole milk	629
2	other vegetables	471	other vegetables	445	other vegetables	506	other vegetables	476
3	rolls/buns	433	rolls/buns	409	rolls/buns	429	rolls/buns	445
4	soda	371	soda	375	soda	389	soda	379
5	yogurt	305	yogurt	367	yogurt	328	yogurt	334
6	root vegetables	282	**tropical fruit**	282	root vegetables	273	root vegetables	284
7	tropical fruit	248	**bottled water**	258	tropical fruit	247	tropical fruit	255
8	**bottled water**	232	**sausage**	253	**sausage**	243	**bottled water**	216
9	**sausage**	231	**root vegetables**	232	**bottled water**	227	**citrus fruit**	198
10	**citrus fruit**	183	**citrus fruit**	211	**pastry**	221	**sausage**	197

TABLE V
NUMBER OF CLIENTS IN EACH SEASON

Season	**Number of Clients**
Winter	2406
Spring	2454
Summer	2433
Fall	2430

TABLE VI
MOST AND LEAST SIMILAR CLIENTS IN RELATION TO OTHER CLIENTS

Season	**Client ID**	**Similarity**	**Items**
Winter	4274	0.092	bottled beer, beef, soda, whole milk, rolls/buns, other vegetables
Winter	1529	0.001	soups, preservation products
Spring	3818	0.096	soda, whole milk, curd, tropical fruit, yogurt, other vegetables
Spring	3308	0.001	tea, light bulbs
Summer	1879	0.105	rolls/buns, whole milk, yogurt, other vegetables
Summer	2605	0.001	ketchup, potato products
Fall	1188	0.103	bottled beer, soda, whole milk, rolls/buns, other vegetables
Fall	4838	0.002	honey, flower (seeds)

a mixed cluster containing clients that did not fit into any other clusters, (2) a cluster with clients who purchased “other vegetables”, and (3) a cluster with clients who purchased “whole milk”. Lastly, with a distance threshold of 8, there were 5 clusters: (1) a mixed cluster, (2) a cluster with clients who purchased “other vegetables”, (3) a cluster with clients who purchased “whole milk”, (4) a cluster with clients who purchased “soda”, and (5) a cluster with clients who purchased “rolls/buns”.

The dendrograms will appear consistent within a season regardless of changing the distance threshold (number of clusters), but they will vary across different seasons (Figures 5, 6, 7, and 8).

As a general rule, irrespective of the season, applying clustering to the client similarity matrix will result in $N-1$ clusters where clients have purchased a specific item, and the remaining cluster will be a mixed cluster containing all other

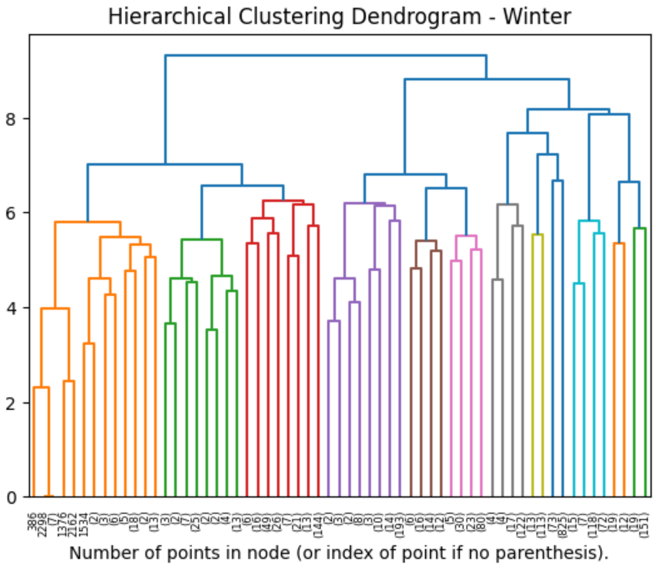

Fig. 5. Hierarchical clustering dendrogram for winter.

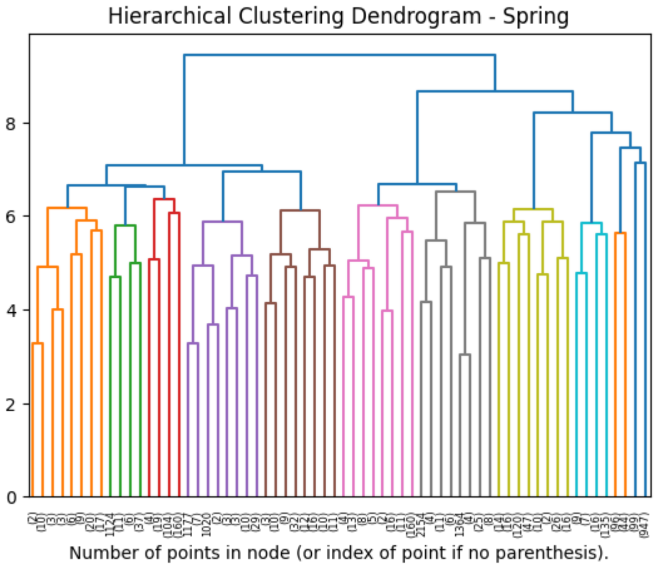

Fig. 6. Hierarchical clustering dendrogram for spring.

TABLE VII
NUMBER OF CLUSTERS FOR EACH DISTANCE THRESHOLD

Season	Distance Threshold	Number of Clusters
Winter	9.0	2
Winter	8.5	3
Winter	8.0	5
Spring	9.0	2
Spring	8.5	3
Spring	8.0	4
Summer	9.0	2
Summer	8.5	4
Summer	8.0	5
Fall	9.0	3
Fall	8.5	4
Fall	8.0	5

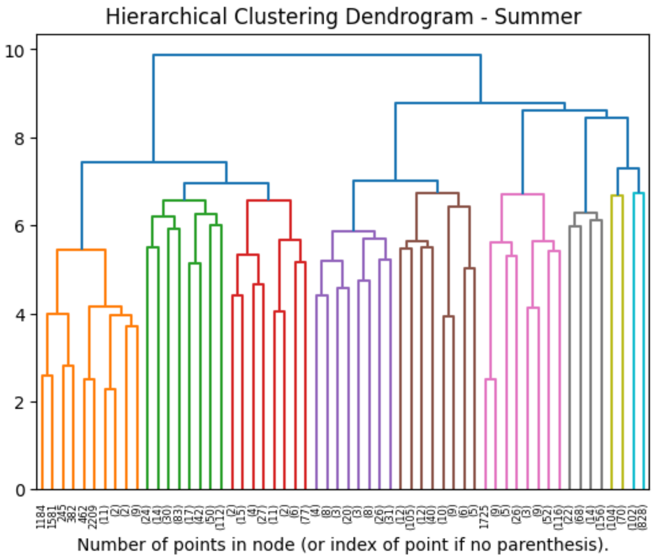

Fig. 7. Hierarchical clustering dendrogram for summer.

clients (in the case of N clusters). Additionally, we would like to emphasize that even if a client has purchased a specific item for which a cluster was created, there is still a chance for them to end up in the last mixed cluster, although the likelihood is low.

D. Results

Considering the entirety of the similarity measurements and clusterings conducted, we can derive two important conclusions. Firstly, clients who purchase a greater quantity of popular items will always be more similar to a larger number of clients compared to those who purchase fewer and less popular items. Secondly, with regard to clustering, the clusters will mostly consist of clients who have bought a specific item. However, it is possible for some clients, even if they have purchased an item on which a cluster was based, to end up in the "last" mixed cluster which consists of clients who could not be associated with any other clusters.

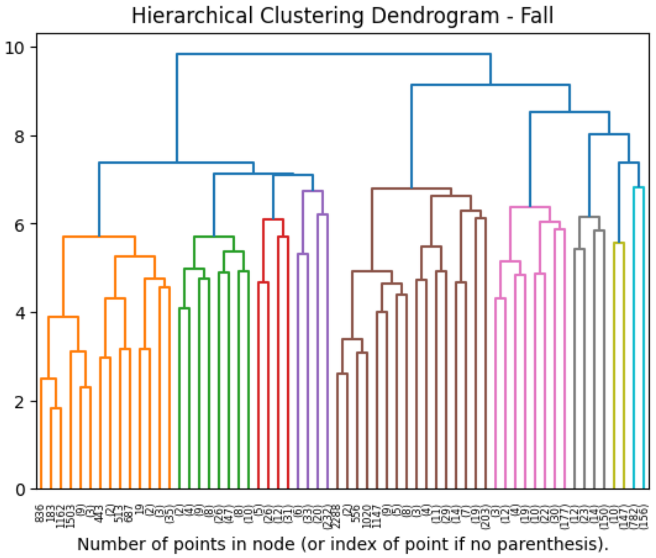

Fig. 8. Hierarchical clustering dendrogram for fall.

VIII. CONCLUSION AND FUTURE WORK

In this paper, several DM techniques were employed in order to aid marketing-related decision-making. Association rules mining was used to provide insight into what sets of products are bought together most often. From the transactional data, 750 rules were obtained by applying the Apriori algorithm. After lifting the parameter thresholds, we were able to pinpoint the best 9 rules, with a lift over 1.1.

An exploratory analysis on how seasonal changes affect buying patterns was also conducted. Summer was the strongest season in terms of items sold and total number of transactions. However, no major seasonal changes could be unearthed, not even in terms of the most frequently sold items for each season. The conclusion we can draw from this analysis is that the available products are not necessarily seasonally impacted.

Agglomerative clustering was also applied in this paper, in order to assess and better understand the behavior of the customers. A seasonal component was added here as well. During the analysis, we were able to pinpoint the client IDs of the customers who exhibited the most similar behavior to others or the most contrasting buying pattern. We have concluded that client similarity depends directly on the number of items purchased and also the popularity of the items.

Ultimately, we can state that a vast amount of information can be gleaned from simple transactional data. We consider DM to be a powerful asset when it comes to business endeavors and it is of great help when it comes to marketing strategies.

Further work may include exploring other DM methods in relation to business contexts, such as time series analysis, classification and regression models. We consider that applying DM solutions in other fields (other than business) is also a premise worth investigating.

REFERENCES

[1] Rakesh Agarwal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. of the 20th VLDB Conference*, volume 487, page 499, 1994.

[2] Shivam Agarwal. Data mining: Data mining concepts and techniques. In *2013 international conference on machine intelligence and research advancement*, pages 203–207. IEEE, 2013.

[3] Sujoy Bag, Sri Krishna Kumar, and Manoj Kumar Tiwari. An efficient recommendation generation using relevant jaccard similarity. *Information Sciences*, 483:53–64, 2019.

[4] Fuguang Bao, Linghao Mao, Yiling Zhu, Cancan Xiao, and Chonghuan Xu. An improved evaluation methodology for mining association rules. *Axioms*, 11(1):17, 2021.

[5] Michael JA Berry and Gordon S Linoff. *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons, 2004.

[6] Trupti A Kumbhare and Santosh V Chobe. An overview of association rule mining algorithms. *International Journal of Computer Science and Information Technologies*, 5(1):927–930, 2014.

[7] T Soni Madhulatha. An overview on clustering methods. *arXiv preprint arXiv:1205.1117*, 2012.

[8] Fionn Murtagh and Pierre Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 31:274–295, 2014.

[9] Jelili Oyelade, Itunuoluwa Isewon, Olufunke Oladipupo, Onyeka Emebo, Zacchaeus Omogbadegun, Olufemi Aromolaran, Efosa Uwoghiren, Damilare Olaniyan, and Obembe Olawole. Data clustering: Algorithms and its applications. In *2019 19th International Conference on Computational Science and Its Applications (ICCSA)*, pages 71–81. IEEE, 2019.

[10] Rashik Rahman. Groceries dataset for market basket analysis (mba).

[11] Petroc Taylor. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025.