NOVA IMS
Information Management School

CUSTOMER SEGMENTATION

A Key to Unlocking Business Growth and Success

REPORT

Machine Learning II | BSc in Data Science, 2024-2025
Professors Fernando Bação and Ivo Bernardo

Gonçalo Beirão Catarino Saldanha Palhoto, 20231639

Tiago Martins da Cruz, 20231682

João Pedro Gouveia Nunes de Sousa, 20231711

NOVA Information Management School

June 9th, 2025

SUMMARY

EXECUTIVE SUMMARY

This project applies unsupervised machine learning techniques to segment a retail customer base using demographic and transactional data. By analysing the datasets, we identify meaningful customer groups based on shared characteristics and purchasing behaviours. The resulting segments inform the development of targeted marketing strategies, enabling personalized promotions and improved customer engagement.

Methodology and Approach

To elaborate this project, data preprocessing was done in the first place, using KNN imputation to address missing values and DBSCAN to detect and remove multidimensional outliers. Principal Component Analysis was applied for feature contribution assessment before the clustering section, where six algorithms were systematically evaluated, namely: Hierarchical clustering, K-Means, Self-Organizing Maps (SOM), DBSCAN, Mean-Shift, and two tandem approaches combining multiple algorithms. Finally, association rules were developed using the Apriori algorithm to identify product relationships within each segment.

Key Findings and Results

The analysis successfully identified seven distinct customer segments using a tandem approach that combined Hierarchical clustering with K-Means refinement. Association rules analysis revealed the most popular products for each segment, enabling the development of targeted promotional campaigns. The segmentation provides actionable insights for personalized marketing strategies, with each segment showing distinct spending patterns across product categories and varying levels of store engagement.

Technical Implementation

Jupyter notebooks were utilized as drafts for the different stages of the project, while the final, production-ready files are maintained as simple Python files. This approach ensures both analytical flexibility during development and operational reliability for business deployment, supporting the transition from exploratory analysis to practical implementation.

INTRODUCTION

Understanding and responding to customer needs is a cornerstone of business growth in today's competitive retail landscape. Customer segmentation, which consists of dividing a broad customer base into smaller, more manageable groups based on shared characteristics, enables businesses to tailor marketing strategies, maximize engagement, and foster loyalty.

This project leverages unsupervised machine learning techniques to reveal groups within a retail dataset comprising customer demographics and transaction history. By analysing the provided datasets, we aim to identify meaningful segments that reflect both purchasing behaviours and demographic profiles, moving beyond traditional, predefined marketing categories.

Through detailed exploratory analysis and clustering, we generate actionable insights into customer preferences. These insights inform the design of targeted promotional campaigns, supporting data-driven decision-making and enhancing the effectiveness of marketing initiatives. Ultimately, this approach empowers businesses to better understand their customers and unlock new opportunities.

EXPLORATORY DATA ANALYSIS AND DATA PREPROCESSING

The data sources used for this project were the datasets *customer_info* and *customer_basket* provided by the Teaching Assistant.

Exploratory Data Analysis (EDA)

The first dataset represents the primary source of analytical data and includes demographic features such as basic personal information, household composition, and location. Behavioural features capture shopping patterns including distinct stores visited, typical shopping hours, customer service interactions, and loyalty program participation. The spending features represent the core analytical variables, encompassing lifetime spending across ten product categories, total distinct products purchased, percentage of promotional purchases, and customer seniority.

Regarding the *customer_basket* dataset, it contains a substantial sample of customer purchasing behaviour. Each transaction record includes invoice identification, the list of purchased goods and customer identification, providing granular insights into purchasing patterns and product associations. It was seen that when the exact same items are purchased in two different instances, e.g., two purchases where `list_of_goods = ['candy bars', 'soup', 'cake']`, the `invoice_id` will be the same, meaning that this variable cannot be used as the index columns.

Missing Values

Examining data completeness of the datasets revealed varying degrees of missing information in the *customer_info* dataset (Figure 1), most notably, roughly 32% of customers did not have a loyalty card number. This missingness likely reflects voluntary program participation rather that data collection issues and thus represents a potential opportunity for improving customer engagement. There were no missing values observed for the *customer_basket* dataset (Annex 1).

	Missing Count	Missing %
customer_name	0	0.00
customer_gender	0	0.00
customer_birthdate	341	1.00
kids_home	749	2.20
teens_home	783	2.30
number_complaints	1022	3.00
distinct_stores_visited	681	2.00
lifetime_spend_groceries	0	0.00
lifetime_spend_electronics	0	0.00
typical_hour	1362	4.00
lifetime_spend_vegetables	1022	3.00
lifetime_spend_nonalcohol_drinks	0	0.00
lifetime_spend_alcohol_drinks	681	2.00
lifetime_spend_meat	0	0.00
lifetime_spend_fish	1703	5.00
lifetime_spend_hygiene	0	0.00
lifetime_spend_videogames	0	0.00
lifetime_spend_petfood	0	0.00
lifetime_total_distinct_products	0	0.00
percentage_of_products_bought_promotion	0	0.00
year_first_transaction	0	0.00
loyalty_card_number	10908	32.03
latitude	0	0.00
longitude	0	0.00

Figure 1: Missing data in the customer_info *dataset.*

Univariate Analysis

To begin with, customer birthdates were converted to datetime format, allowing the creation of a new age column by calculating the difference between the current date and the birthdate obtained from the dataset.

Next, some histograms were plotted on the *customer_info* dataset, allowing some conclusions to be taken (other plots may be found in Annex 2):

- Most households have either no kids or just one kid at home, with fewer households having more than two. Regarding teens, the situation is similar (Figures 2 and 3);
- The most frequent shopping hours are around 9 a.m. and 1 p.m. (Figure 4);
- Few people have filed more than one complaint (Figure 5).

Moving on to the *customer_basket* dataset, the `list_of_goods` was first converted from strings to lists, and then a function which returns a sorted dictionary of the most purchased items was created (using merge sort). The plot in Figure 6 illustrates the top 15 most frequently purchased products, which show oil and cooking oil as market leaders.

Moving on to outlier detection, some boxplots were plotted (Annex 3), which, in conjunction

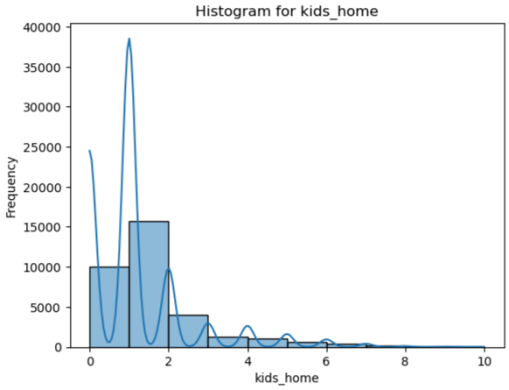


Figure 2: Histogram for kids_home.

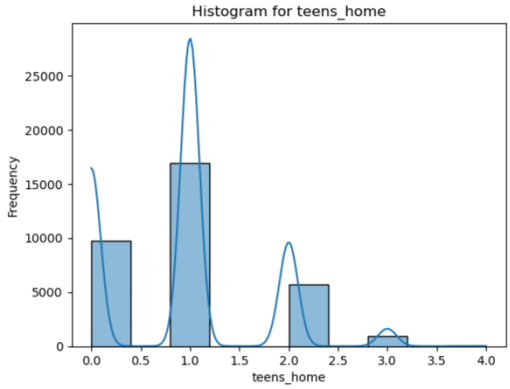


Figure 3: Histogram for teens_home.

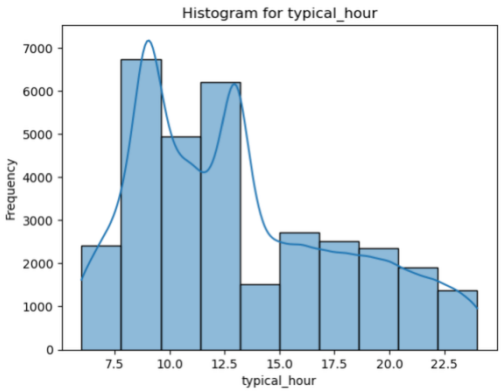


Figure 4: Histogram for typical_hour.

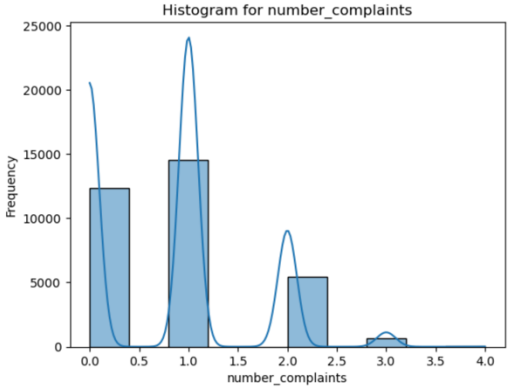


Figure 5: Histogram for number_complaints.

with the previous histograms, confirmed that most variables in the *customer_info* dataset were right-skewed, which is typical of financial data, as it always starts at 0 and is bound to have extreme values to the right. In addition, it was also noted that the percentage of products bought on sale is a percentage, but some negative values persist which must be handled in the preprocessing stage.

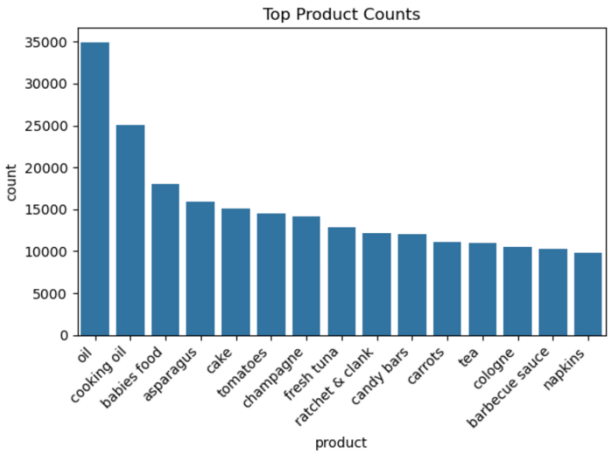


Figure 6: Top 15 most frequently purchased products.

Multivariate Analysis

Plotting the correlation heatmap of the numerical variables in the *customer_info* dataset (Figure 7) reveals some relevant correlations (above 0.5):

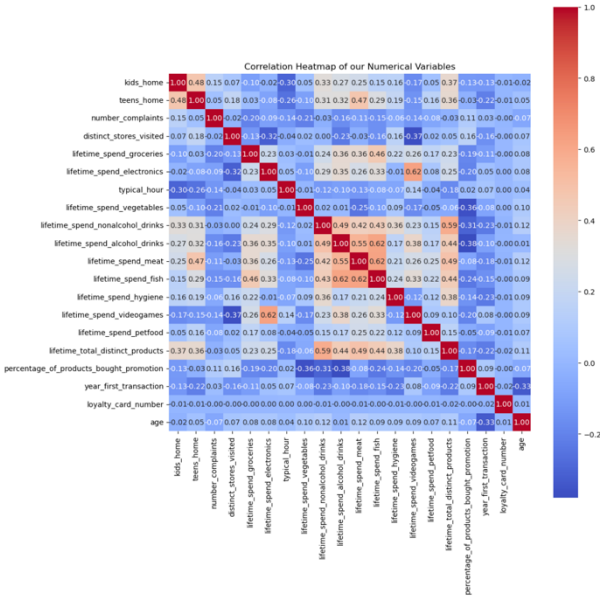


Figure 7: Numerical variables' correlation heatmap.

- *lifetime_total_distinct_products* and *lifetime_spend_nonalcohol_drinks*
- *lifetime_spend_videogames* and *lifetime_spend_electronics*
- *lifetime_spend_meat* and *lifetime_spend_alcohol_drinks*
- *lifetime_spend_fish* and *lifetime_spend_alcohol_drinks*
- *lifetime_spend_fish* and *lifetime_spend_meat*

Based on the previous results, some scatter plots were done (Figure 8), but no strong linear correlations between the variables were observed.

Lastly, a spider chart was plotted (Figure 9), which allows for the analysis of how much money is spent on each category. It should be noted that groceries and electronics were left out since its values were considerably larger than the remaining variables, affecting their visualization (histograms were plotted for these two variables - Annexes 4 and 5).

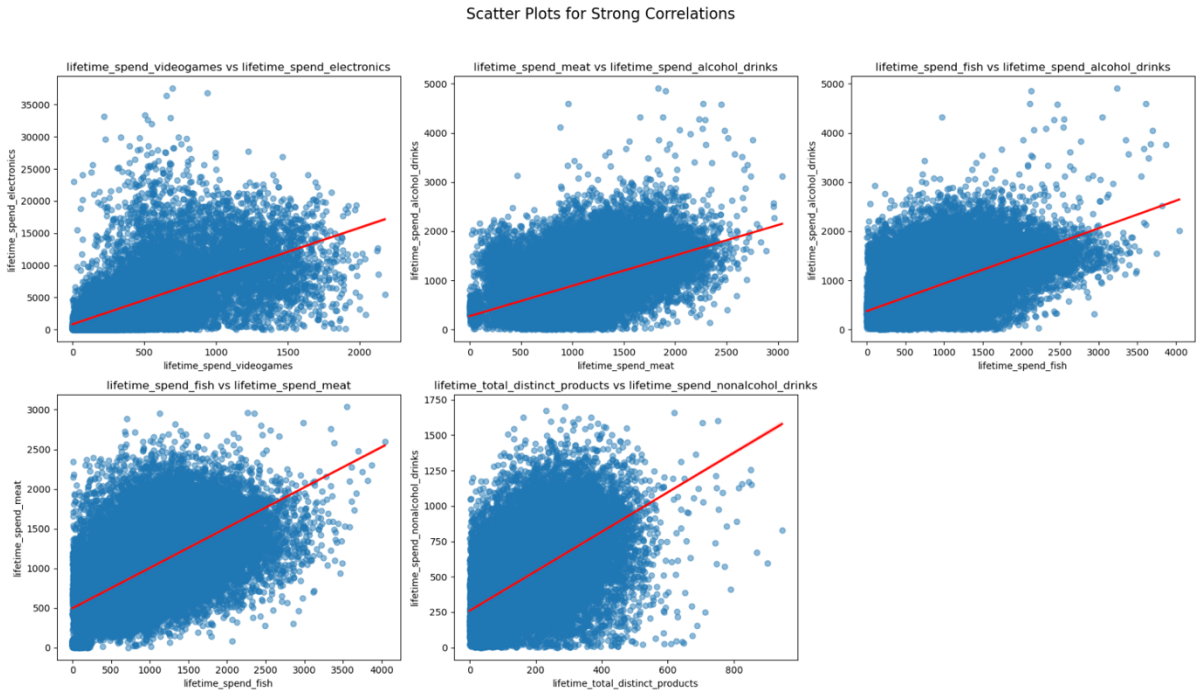


Figure 8: Scatter plots for observed strong correlations.

From the spider chart, it was possible to take some conclusions:

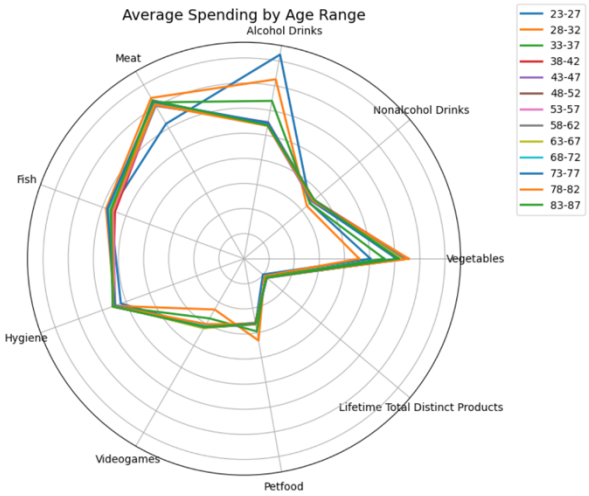


Figure 9: Spider chart representing the average spending according to customers' age ranges.

- The younger the customers are, the more alcohol will be purchased;
- Regarding meat, only the lowest age range (23-27) differs from the remaining ones, demonstrating they purchase less than older customers;
- The distribution is similar across age ranges for fish, non-alcoholic drinks, hygiene products and distinct items bought;
- Older age groups seem to differ the most when analysing videogames and pet food, buying less of the former and more of the latter;
- Vegetables are notably more consumed by older people.

It should be noted that the analysed variables constitute a lifetime spend, hence larger values for older age groups could simply mean they have had more time to buy these

products when compared to younger customers. Following this logic, it is possible to conclude that younger people are purchasing far more alcohol, since they far outweigh older age groups.

Lastly, a coordinate map was created by matching latitude and longitude to identify clusters of people (Annex 6). However, no clear patterns or insights emerged from this analysis, apart from a cluster with a higher lifetime total in East Lisbon.

Data Preprocessing

From the EDA stage, it was understood that there were duplicated rows on *customer_basket* (when the basket content was the same), roughly 32% of customers did not have a loyalty card number and there were negative values in the percentage of products purchased on sale.
It was also seen that since there were no very highly correlated variables, nothing would be dropped in the preprocessing stage. In addition, considering the small datasets and their predominantly numerical content, the data types were kept unchanged.

Feature Selection (Part 1)

Considering the earlier creation of an age column, the `customer_birthdate` column was dropped.
After that, the variable `customer_gender` was encoded, replacing "female" with 0 and all other values with 1.
Lastly, `loyalty_card_number` was transformed into a binary variable, yielding *1* if the customer has a loyalty card and *0* otherwise.

Data Cleaning

Regarding duplicates, on *customer_basket*, even though there are no duplicate rows, some refer to the same customer making different purchases (same `customer_id`) while others reflect different customers buying the same set of items (same `invoice_id`).
For *customer_info*, since `customer_id` was set as the index, we may conclude that the dataset contains no duplicates.

Moving on to the analysis and treatment of missing values, as seen earlier, there are no missing values for the *customer_basket* dataset.

In order to impute missing data in the *customer_info* dataset, K-Nearest Neighbours (KNN) was used. To evaluate the optimal number of neighbours, only rows where age was not missing were selected, and the data was split into training and test sets. In the latter set, age was artificially removed to simulate missingness, and therefore the true values were kept for comparison.

KNN imputation was then applied using different values for *k* (Annex 7), and the accuracy of the imputed values was measured by calculating the RMSE against the original age values in the test set. $k = 10$ was found to be the best option, since larger values for *k* began to show signs of underfitting (there was a progressive increase in predicted values around the median of age – 50-60 years).

Finally, the negative percentages of products bought on sale were assumed to be zero and hence were replaced.

Feature Selection (Part 2)

In this second stage of feature selection, some columns were removed, namely:

- `year_first_transaction`, since the value of this variable can also be found in age (the older a person is, the smaller will the year of the first transaction be);
- `loyalty_card_number`, which only contained a set of digits with no meaning for those who had a loyalty card. It could, though, be used in the future when identifying in the clusters who has a card or not;
- `latitude` and `longitude`, as they were only used for the map in the EDA, which showed no visible patterns.

Unidimensional Outlier Detection

As seen in the EDA, the lifetime variables all displayed a significant number of outliers, along with considerable skewness. A log transformation was applied to the financial features due to their highly right-skewed distributions. This transformation helps to reduce skewness, stabilize variance, and limit the influence of extreme values.

Scaling

Robust Scaler was used in this step, since it was observed during the EDA stage that the data contains a significant number of outliers.

Outlier Detection

At this stage, DBSCAN will be used to determine the presence of multidimensional outliers, as well as analyse the presence of unidimensional outliers.

As a starting point, it will be considered that the minimum number of points is equal to 1 plus the total number of features. After finding the suggested eps and running the DBSCAN, it became evident that this approach wasn't very fruitful, so the range of attempts and combinations of eps and minimum number of points should be expanded. It was found that as either eps rises or the minimum number of points decreases, the number of outliers decreases, and vice-versa. The final selection was `eps = 2.5` and `min_samples = 20`, yielding a single cluster with 733 outliers, which were promptly removed.

Feature Selection (Part 3)

The final step of the data preprocessing was to apply Principal Component Analysis (PCA), a technique with high interpretability.

To assess which features are most important, PCA loadings, which measure how strongly each original feature contributes to each principal component, were used. Since some components explain more variance than others, the loadings were squared (to focus on their strength regardless of sign) and then multiplied by the proportion of variance each component explains. After summing these values across all components, it was possible to obtain an overall score on each feature's contribution to the total variability in the dataset (Fig. 10).

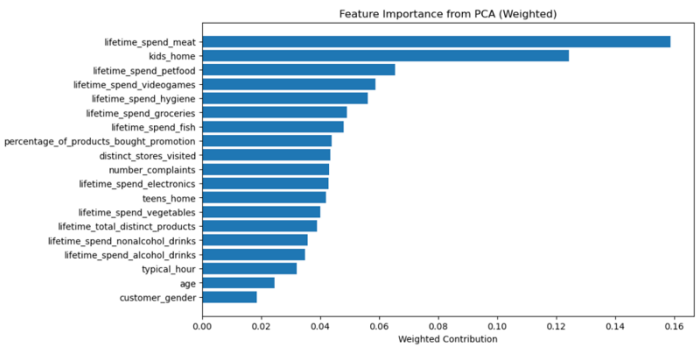


Figure 10: Weighted feature importance obtained from the PCA.

The cumulative explained variance plot (Annex 8) shows that the first principal component explains around 26% of the total variance, with diminishing contributions from subsequent components. In total, 10 components are required to explain roughly 83% of the variance, and all 19 components are necessary to reach 100% of explained variance. This indicates that the variance is relatively spread across many components, suggesting no overwhelming dominance of a small subset of features.

The loadings matrix further confirms this. While certain variables such as `kids_home`, `lifetime_spend_videogames`, `lifetime_spend_groceries` and `lifetime_spend_electronics` show higher weighted contributions, the remaining features also provide relevant information. Importantly, no feature shows an extremely low or negligible contribution that would justify its removal based solely on PCA results.

As a result, all variables will be kept for now, and only the ones deemed unnecessary for running the clustering algorithms will be removed for that.

CUSTOMER SEGMENTATION

Clustering

Through the PCA done earlier, it was decided to only keep the lifetime variables for the clustering. Nonetheless, the remaining columns should be saved to be used at a later stage, when analysing their densities on the clusters.

Hierarchical Clustering

This algorithm groups data points into clusters based on their similarity and does not require pre-specifying the number of clusters. However, it may struggle to correctly partition datasets with complex or overlapping structures, which is the case with the dataset being studied. From Figure 11, the most appropriate division seemed to be at around distance 140, where 4 clusters would be formed. The UMAP projection (Figure 12) showed a decent fit for the data, with the only issue being the merger of the two bottom left clusters. Re-running the model with more clusters (Annex 9) allowed for higher granularity and finer segmentation, as seen with 6 clusters, but some overlapping is still visible.

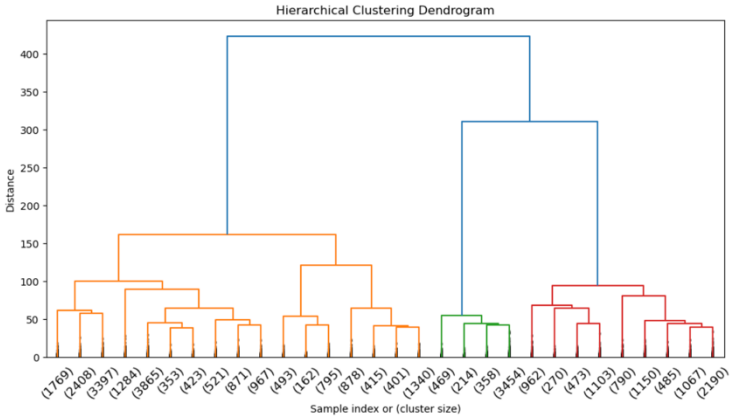


Figure 11: Hierarchical Clustering Dendrogram.

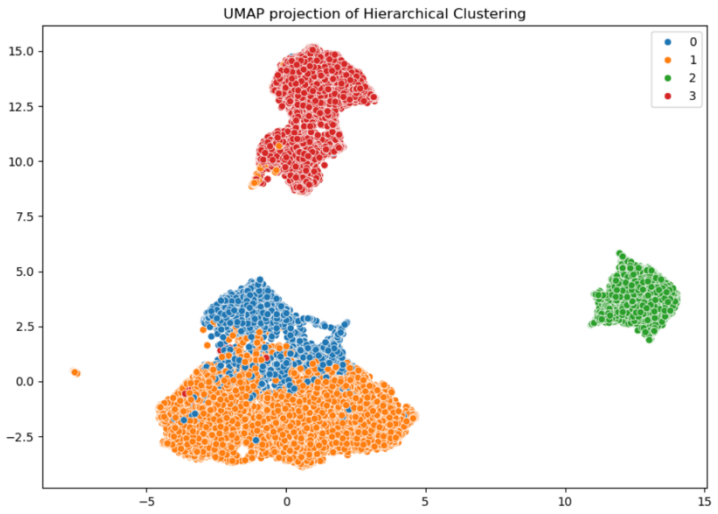


Figure 12: UMAP Projection with 4 clusters.

K-Means

This algorithm divides data into k clusters by minimizing distance between points and their cluster center and is relatively simple to implement. Nonetheless, it assumes clusters are spherical and equally sized, which does not happen with this dataset.

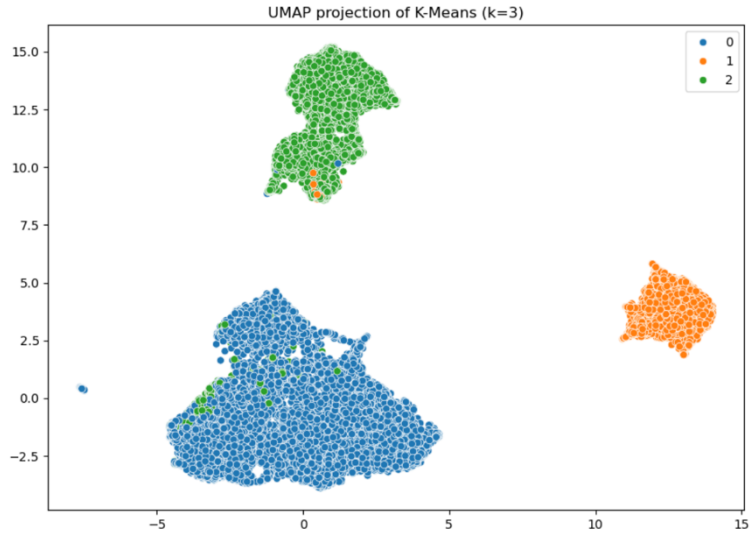


Figure 13: UMAP Projection with 3 clusters.

Since the number of clusters must be defined in advance, the point where the Silhouette Score is the highest and the Davies-Bouldin Score is lowest was found to be at $k = 3$ (Annex 10), and the result shows clear separation of the three clusters (Figure 13).

While the result is acceptable, it is clearly an oversimplification of the actual clusters, so testing with larger values for k provided relatively well-separated and stable clusters, although some overlapping occurred in the UMAP projection (Annex 11).

K-Medoids

This algorithm was not used in this project as it is computationally more expensive than K-Means, especially on large datasets. Even so, it is more robust to noise and outliers than the previous algorithm, since it uses actual data points as cluster centers.

Self-Organising Map (SOM)

This algorithm turns complex data into a two-dimensional grid where similar points are placed close together, making it easier to reveal clusters and patterns that other algorithms may find harder to detect. Nevertheless, it requires careful hyperparameter tuning and can also be computationally expensive for large datasets.

To evaluate and compare the different parameter combinations of the SOM, the Silhouette score was used as a rough indicator of cluster quality. Although it did not prove to be very useful, it provides a simple way to assess how well the data points assigned to

each neuron are grouped together, helping to discard parameter settings that produce poorly separated or unstable mappings.

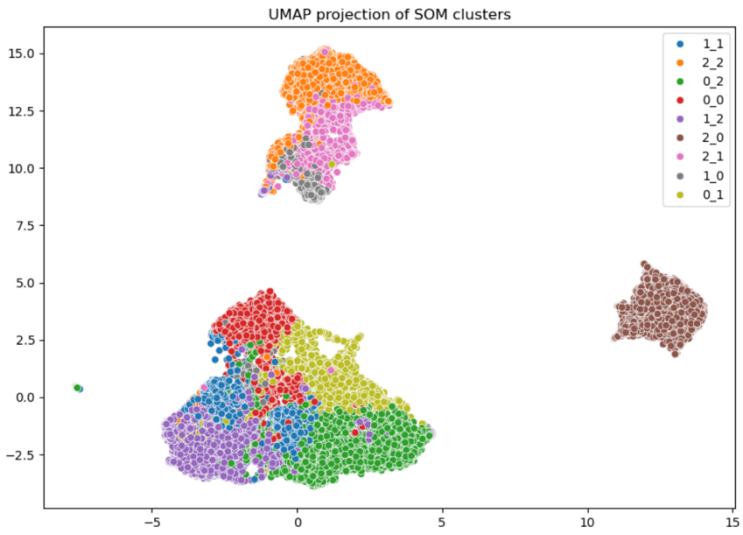


Figure 14: UMAP Projection of SOM Clusters.

After tuning the SOM parameters to values well-suited for the dataset, the resulting clusters still show considerable overlap and dispersion in the UMAP projection (Figure 14). This suggests that, given the underlying structure of the data, the SOM algorithm struggled to find clearly separated and stable segments.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

This algorithm groups data based on density, identifying core points with enough nearby neighbours. It does not require defining the clusters beforehand and can detect noise and outliers, but it is sensitive to the choice of `eps` and `min_samples`.

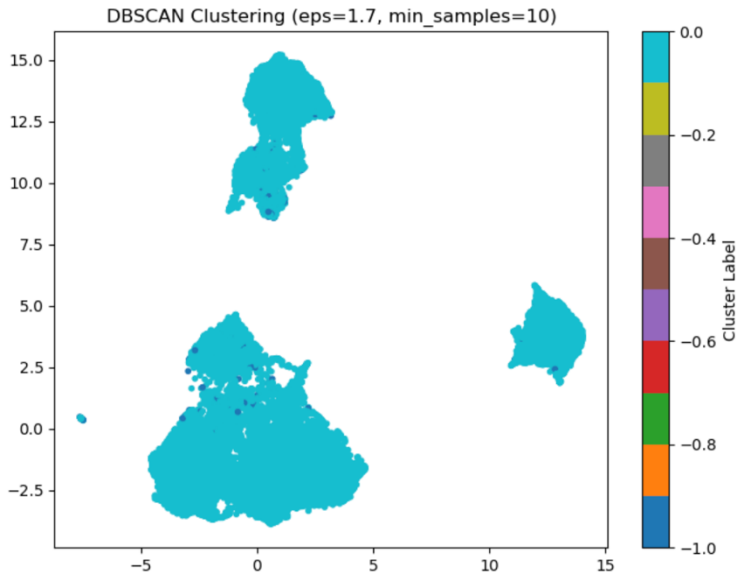


Figure 15: UMAP Projection of DBSCAN with eps=1.7 and min_samples=10.

Multiple combinations were tested, and, in all cases, the algorithm consistently found one large cluster with no outliers, suggesting the transformations applied in the preprocessing stage were successful (Figure 15 & Annex 12). The clustering results showed little sensitivity to the parameter values within the tested ranges.

Despite this, DBSCAN was not able to produce a stable or meaningful clustering solution. Small changes in the parameters always led to most data being into a single cluster. This suggests that the data structure may not be well-suited for density-based clustering with DBSCAN, as no clear density separation could be consistently captured.

Mean-Shift

This algorithm iteratively shifts data points towards the densest areas of the dataset. Like the last few algorithms, it does not require the definition of clusters beforehand, and can find clusters of arbitrary shape and size, but may struggle with them.

Figure 16: UMAP Projection with bandwidth=1.0.

The strategy for finding the ideal bandwidth (how far the algorithm looks for dense regions) was trial and error. Despite testing a wide range of values (Figure 16 & Annex 13), Mean-Shift was not able to properly capture the underlying cluster structure of the dataset, with an over-fragmentation of the data when bandwidth was low, and excessive merging otherwise, meaning the algorithm struggles to adapt to the complex density distribution of the data.

Tandem Approach (Hierarchical Clustering + K-Means)

In this situation, Hierarchical clustering is first used to reveal the data's structure and suggest the number of clusters. The centers of these groups are then used to initialize K-Means, which refines the cluster assignments by minimizing variance within each group.

It does this by finding the cluster centers (centroids) that minimize the sum of squared distances between each point and its assigned centroid. The goal is to make points in each cluster as similar as possible (low variance), and as different as possible from points in other clusters.

4 clusters (Figure 17) were used as a starting point (the same as in the previous Hierarchical clustering), and this number was progressively increased (Annex 14). Applying the K-Means over the Hierarchical improves the solution slightly, making some clusters more stable as they merge less. Thus, this Tandem Approach can be a good solution.

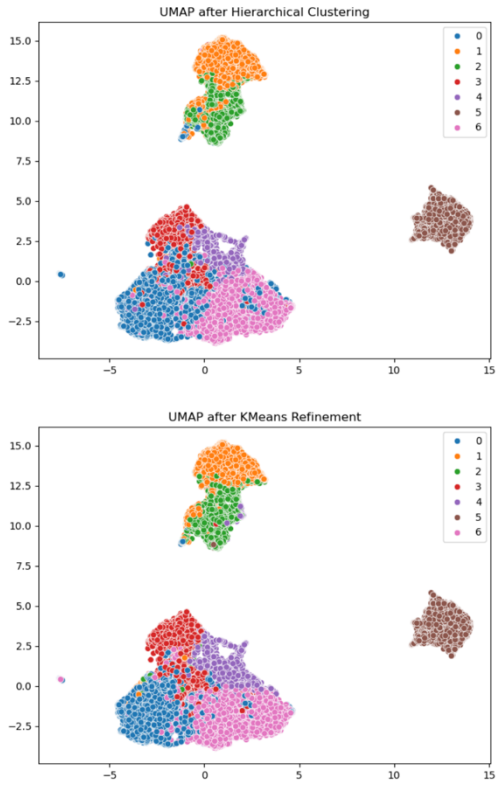


Figure 17: UMAPs for 4 clusters.

Tandem Approach (K-Means (with large *k*) + Hierarchical Clustering)

The previous situation is now inverted. K-Means is first applied with many clusters to over-segment the dataset into small, homogeneous groups. This simplifies the structure of the data and reduces its complexity. Then, Hierarchical clustering is applied to merge these small groups into a final set of meaningful clusters. If, for instance, $k = 100$, then the Hierarchical algorithm will cluster with only those 100 points (Annex 15).

Observation of the dendrogram reveals that $k = 5$ should be the starting point for the K-Means algorithm (Annex 16). Even after trying larger values for k (Annex 17), the best solution remains with 5 clusters (Figure 18), hence $k = 100$ will be increased to $k = 200$, and the rest of the process redone.

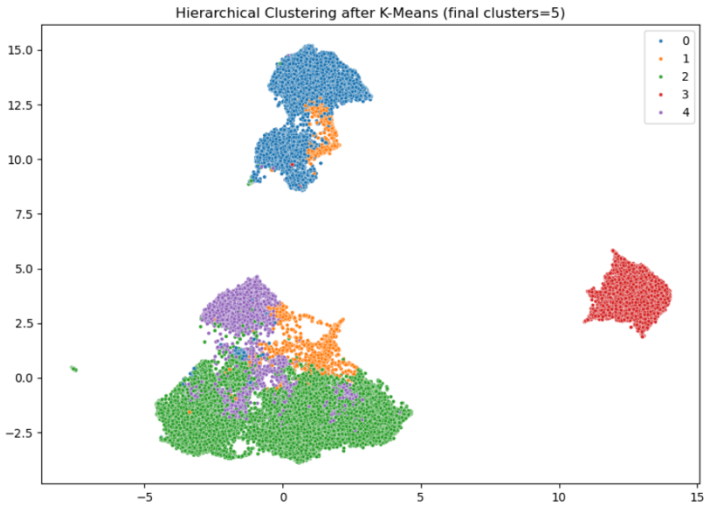


Figure 18: UMAP for 5 clusters.

Now, the dendrogram reveals that $k = 6$ should be the starting point (Figure 19 & Annex 18). Again, after trying larger values for k (Annex 19), the Tandem Approach clearly struggles to find the underlying patterns on the data, forming unclear and unstable clusters that mix and overlap with each other.

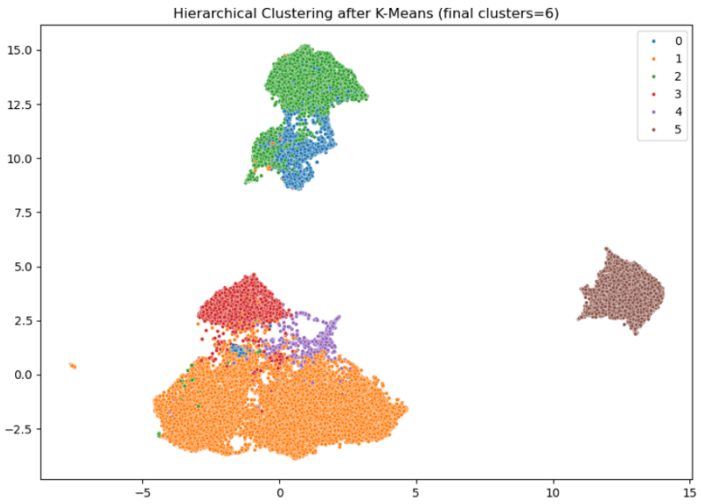


Figure 19: UMAP for 6 clusters.

Comparison, Profiling and Selection

After selecting the clustering solution that appeared the most appropriate, the next step is to analyse and profile the resulting clusters. While UMAP visualizations give an initial sense of how the data points are distributed, they do not always reflect the full extent of the differences between clusters. To better understand the nature of each segment, the average values of key features within each cluster will be computed. This allows for a detailed comparison of customer characteristics across segments and helps to assess whether the clusters capture meaningful behavioural patterns that can later be used to support targeted business strategies.

From all developed solutions, the ones which best fit the data were:

- Hierarchical with 7 clusters;
- K-Means with 4 clusters;
- Hierarchical followed by K-Means with 4 clusters;
- Hierarchical followed by K-Means with 7 clusters.

The solutions with only 4 clusters were excluded right away, as their granularity is rather poor, even when taking into consideration that they apply a good division to the data.
From the remaining solutions, the **Hierarchical followed by K-Means with 7 clusters** segregates the final clusters the best and avoids overlapping in-between segments (Figure 20).

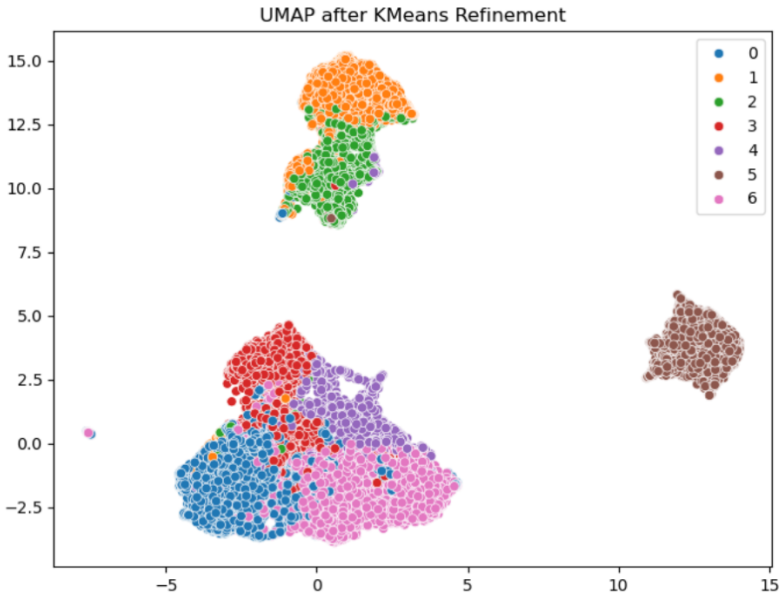


Figure 20: Final solution - Hierarchical clustering, followed by K-Means refinement with 7 clusters.

From the heatmap in Figure 21, obtained after the final clustering, it is possible to find some patterns in the clusters and categorize them:

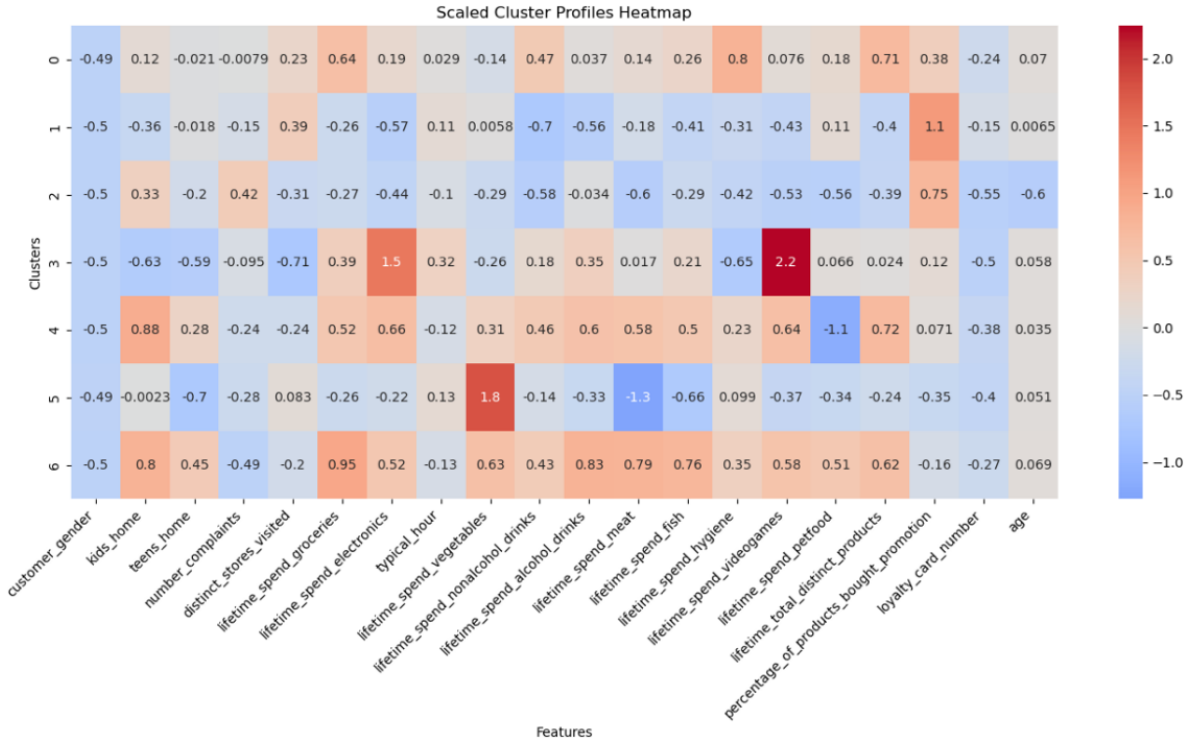


Figure 21: Scaled Cluster Profiles' Heatmap.

- **Cluster 0: "Hyper-Hygienic"** - This cluster showed, by far, the highest amount of money spent on hygiene products. It also showed a medium-high value for groceries shopping, meaning their engagement with the store is considerably high.
- **Cluster 1: "Discount Driven"** - These are the people who visit multiple shops and make purchases mostly when they find a promotion or discount. They have the highest value for both `percentage_of_products_bought_promotion` and `distinct_stores_visited`. However, their overall engagement seems to be lower than most, implying they only go to the stores to look for discounts.
- **Cluster 2: "Coupon Karen"** - Like cluster 1, these customers buy a lot of products on sale yet show a high number of complaints. They also have low engagement with the stores.
- **Cluster 3: "Gadget Geeks"** - This segment is directed at gamers, as we can tell from the extreme values in `lifetime_spend_eletronics` and `lifetime_spend_videogames`. Most of their engagement with the store is for these two types of products. Curiously enough, they are also the group who spends the least on hygiene.
- **Cluster 4: "Anti-PAN Families"** - These are large families, characterized by a high overall engagement with the store for most products except pet food, where `lifetime_spend_petfood` assumes an extremely low value.
- **Cluster 5: "Vegetarians"** – A simple, factual name for this group. This clearly refers to the people who have a vegetable-based diet, as we can observe by the high `lifetime_spend_vegetables` and low `lifetime_spend_meat`. Their engagement is only high for vegetables, as they barely purchase any other products.
- **Cluster 6: "Regular Families"** - This final cluster encompasses high engagement, all-round large families. Their spendings are rather high and homogeneous for all lifetime variables, meaning their engagement is also high.

ASSOCIATION RULES AND TARGETED PROMOTIONS

Association Rules

In this section, the Apriori algorithm will be used, as it provides a better overview by evaluating multiple metrics. Before each association rules' implementation, *TransactionEncoder* will be applied.

The most popular products for the cluster are identified by highest antecedent support and consequent support values (Annex 20). These metrics show the proportion of transactions containing the given products or product combinations.

- For **Cluster 0: "Hyper-Hygienic",** the most popular products are cologne and cider.
- For **Cluster 1: "Discount Driven"**, the most popular products are oil, muffins, cake and tea.
- For **Cluster 2: "Coupon Karen",** the most popular products are beer, salt, white wine and cider.
- For **Cluster 3: "Gadget Geeks",** the most popular products are energy bars, protein bars, energy drinks, pancakes and gadgets for TikTok streaming.
- For **Cluster 4: "Anti-PAN Families",** the most popular products are champagne, fresh tuna, dessert wine and cider.
- For **Cluster 5: "Vegetarians",** the most popular products are asparagus, carrots, tomatoes, cauliflower, shallot, zucchini and avocado.
- Finally, for **Cluster 6: "Regular Families"**, the most popular products are spaghetti, fresh tuna, cottage cheese and frozen smoothie.

Targeted Promotions

Finally, the previous insights will guide the design of targeted promotional campaigns, directed towards each individual cluster and with the goal of enhancing the effectiveness of marketing initiatives.

Firstly, a general campaign could be discounts applied specifically to people with loyalty cards, therefore boosting loyalty. Then:

- For **Cluster 0: "Hyper-Hygienic",** focus on hygiene and health-related cross-sells. Leverage their high spend in hygiene to introduce related consumables.
  - Buy one get one: on hygiene branded products, which are usually more expensive, meaning more profit. E.g.: Buy one Axe Spray deodorant, get two
  - Bundle Promotions: Mixing everyday necessary items such as bread or oil with the hygiene products they already consume (like shampoo). E.g.: Bundle of deodorant and bread or oil.
  - Bundle Promotions option 2: bundle items of the same kind, for instance, for showering products. E.g.: Bundle of shower gel, shampoo and deodorant.

- For **Cluster 1: "Discount Driven",** these customers respond only to price. They should be incentivised with visible, limited-time promotions across various categories to pull them into consistent spending patterns.
  - Slash price: present diverse discounts weekly, to preserve engagement. E.g.: 20% slash price on oil (making them more likely to buy cake and tea)
  - Three for two promotions. E.g.: 3 branded beers for 2 (on slightly more expensive brands is ideal)
  - Coupon packs with expiration dates to create urgency. E.g.: After purchasing a pack of beers, get a limited-time discount on gums. (Making them more likely of buying tea and cooking oil)

- For **Cluster 2: "Coupon Karen",** the approach should be similar to that of cluster 1, only adapting to their complaining habit.
  - Customer care coupons: offer discounts for accumulated positive reviews. E.g.: If a customer with a loyalty card does a positive review, apply 10% discount on beer (making them more likely to get a loyalty card and buying salt)
  - Complaints-free bonus: give discounts for complaint-free streaks. E.g.: If a customer goes 2 months without making complaints, make a discount on beer (making them less complaining and more likely to buy white wine)

- For **Cluster 3: "Gadget Geeks",** the most popular products are energy drinks, protein bars, AirPods, pancakes and gadgets for TikTok streaming. These are high-

value electronics buyers, so relevant tech products and bundle deals that relate to gamer style should be pushed.

- Bundle deals: join products they often buy and make bundles E.g.: Bundle of an energy drink and a protein-bar (Making them more likely to buy pancakes)
- Bundle deals v2: join products they often buy with everyday products. E.g.: Bundle a gadget for tik tok streaming with items like bread or milk

- For **Cluster 4: "Anti-PAN Families",** these are large households that skip pet food. Push essentials and reward basket size.
  - Spend-threshold rewards: Set an arbitrary value as a threshold for discounts. E.g.: Spend €50, get free dessert wine.

- For **Cluster 5: "Vegetarians",** supporting their lifestyle with clean, plant-based product combos is crucial.
  - Veggie baskets: E.g.: Veggie baskets with asparagus and shallot at a flat 20% off (making them more likely to buy tomatoes and carrots).
  - "Meat-free Monday" bundles: auto discount on vegetarian purchases on Mondays.
  - Recipe-based combos: focus on common vegetarian recipes and sell bundles based on their ingredients. E.g.: Basket with zucchini, cauliflower and cooking oil sets.

- For **Cluster 6: "Regular Families",** this segment represents high-value, balanced buyers. It's recommended to cross-sell across categories to increase cart diversity.
  - Mix-and-match: Buy any 3 products, 1 from fish, 1 from meat and 1 from vegetables, get a discount. E.g.: Buy spaghetti, fresh tuna and cottage cheese and get 10% off the total
  - Mega family packs: curated bundles with groceries, hygiene, fish/meat, and household products. Preferably combinations related to a certain meal. E.g.: Turkey, mushrooms and heavy cream as a basket (for a strogonoff)

CONCLUSIONS AND RECOMMENDATIONS

The project used unsupervised machine learning to identify seven distinct customer segments in retail data, moving beyond basic demographic categories. Combining Hierarchical clustering with K-Means produced stable, interpretable segments with clear business value, outperforming single-method approaches.

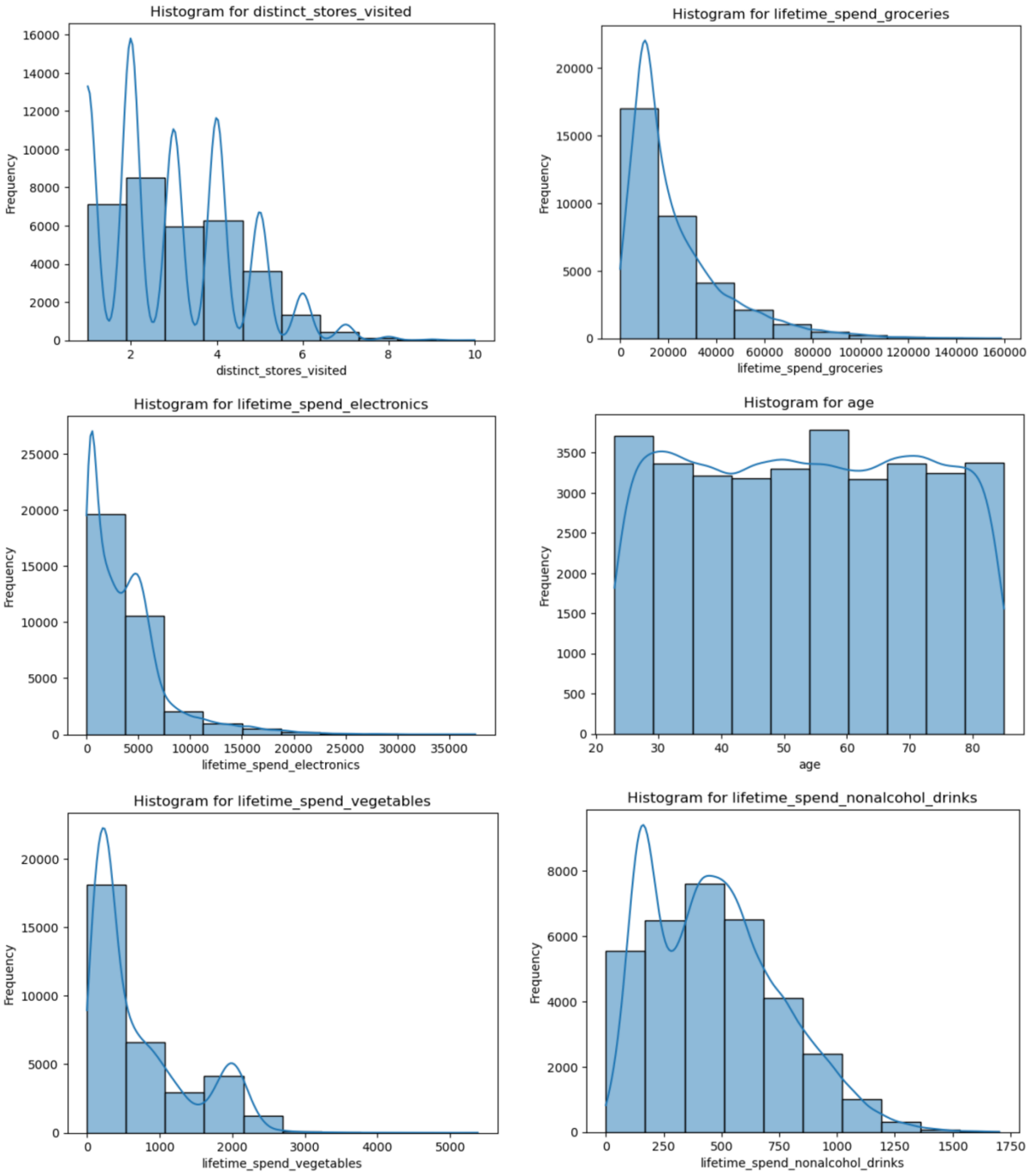
Thorough data preprocessing, including KNN imputation and DBSCAN outlier removal, was essential for reliable clustering. Evaluation of six algorithms showed single approaches failed to capture complex customer behaviours, highlighting the need for tailored algorithm selection. The segments enable targeted retail strategies, and the analysis of association rules identified cross-selling opportunities and revealed high complaint rates among price-sensitive segments, informing customer service strategies.

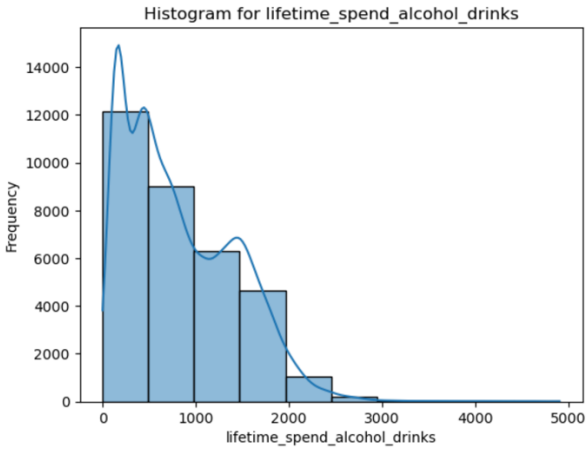
Effective segmentation requires rigorous methodology, algorithm comparison, and business context understanding. The tandem clustering approach proved the value of methodological innovation and provides a repeatable framework for sustainable, data-driven retail advantage.

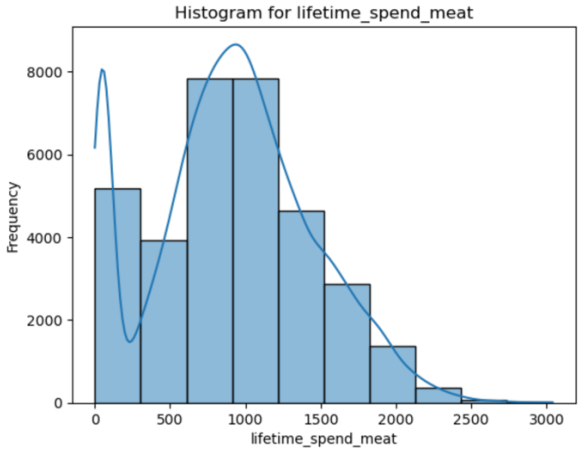
ANNEXES

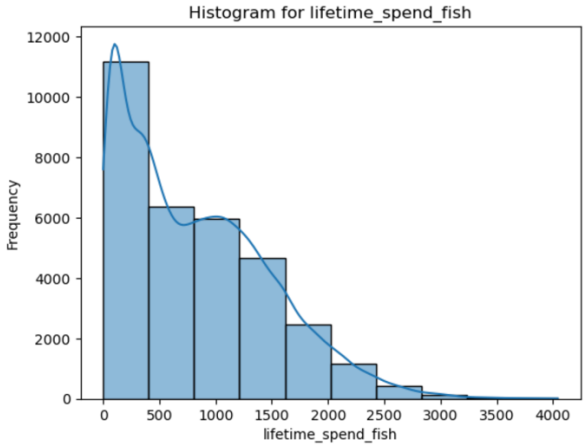
	Missing Count	Missing %
invoice_id	0	0.0
list_of_goods	0	0.0
customer_id	0	0.0

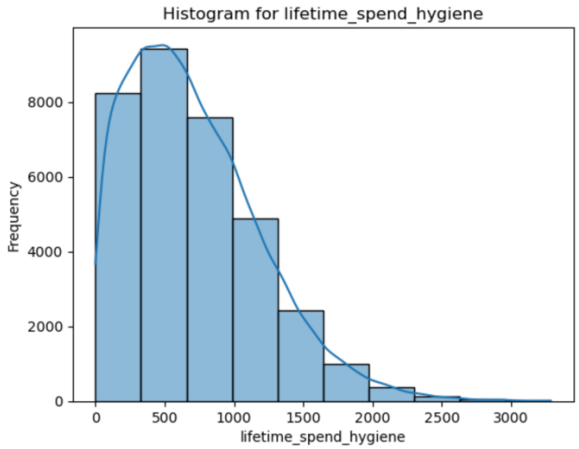
Annex 1: Missing data in the customer_basket *dataset.*

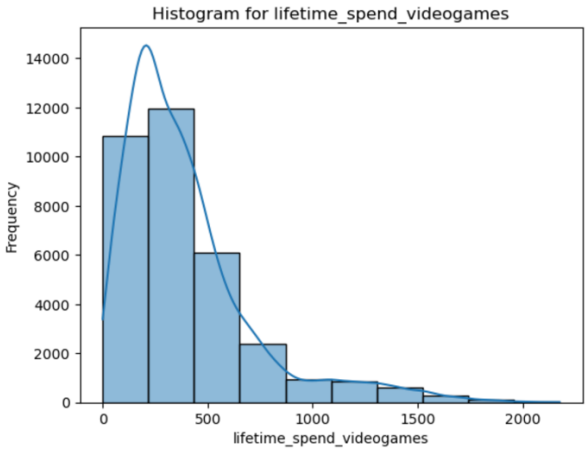


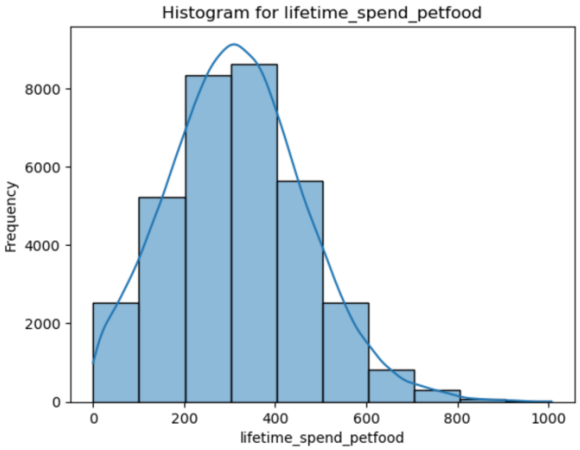

Histogram for lifetime_spend_alcohol_drinks
Frequency
lifetime_spend_alcohol_drinks

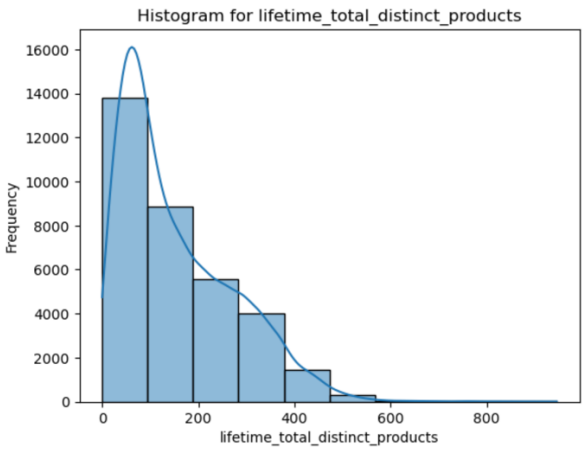

Histogram for lifetime_spend_meat
Frequency
lifetime_spend_meat

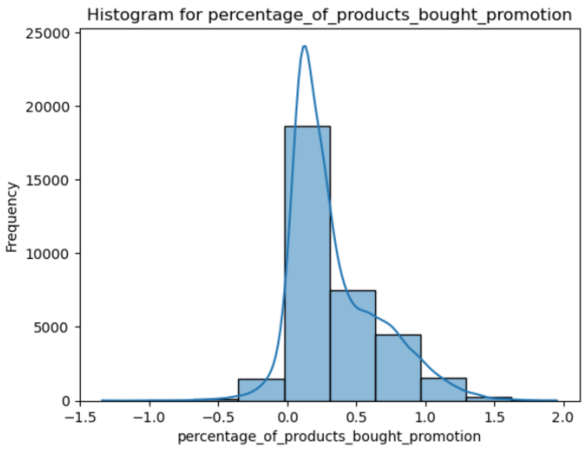

Histogram for lifetime_spend_fish
Frequency
lifetime_spend_fish

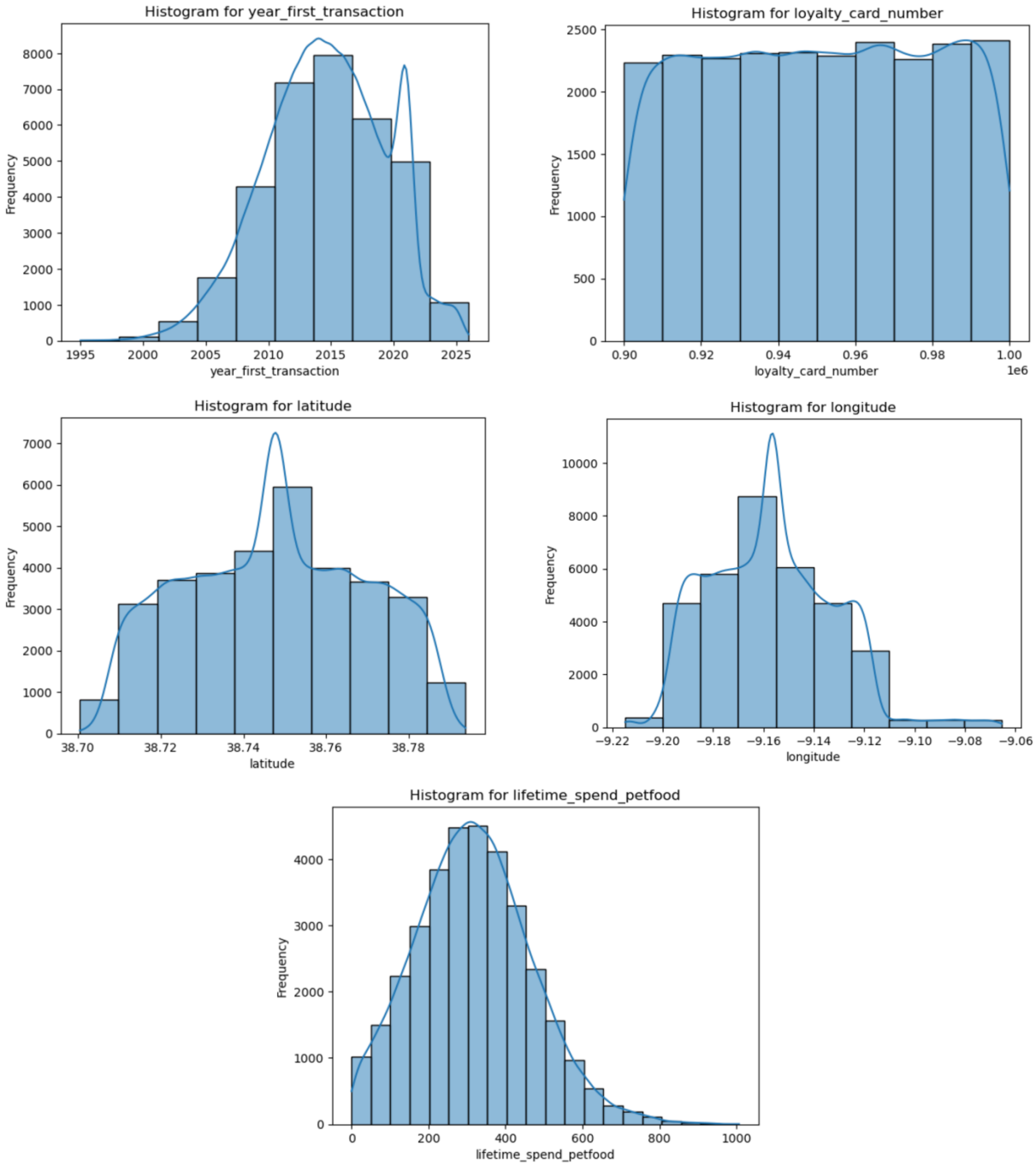

Histogram for lifetime_spend_hygiene
Frequency
lifetime_spend_hygiene


Histogram for lifetime_spend_videogames
Frequency
lifetime_spend_videogames

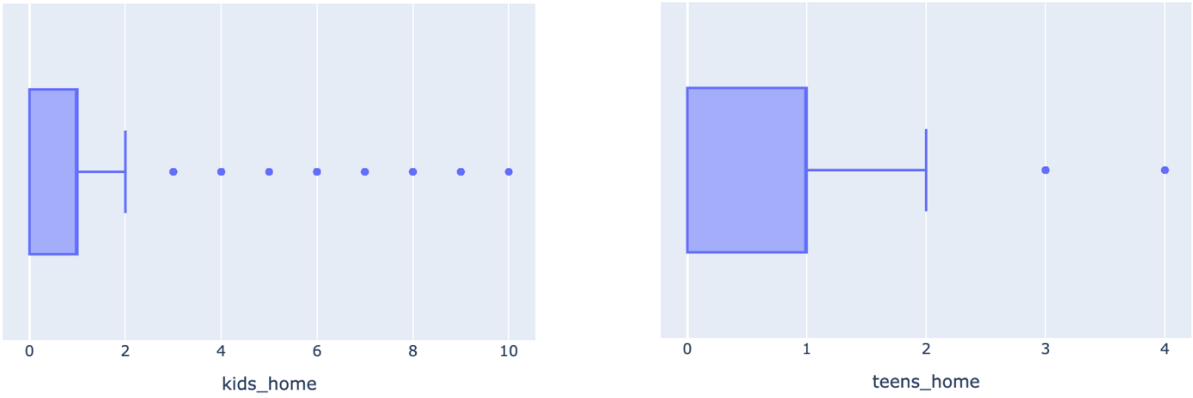

Histogram for lifetime_spend_petfood
Frequency
lifetime_spend_petfood

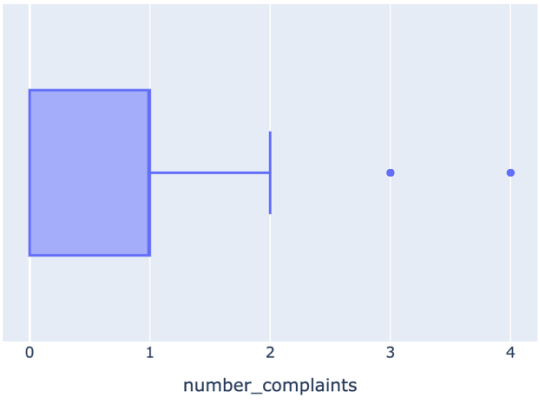

Histogram for lifetime_total_distinct_products
Frequency
lifetime_total_distinct_products


Histogram for percentage_of_products_bought_promotion
Frequency
percentage_of_products_bought_promotion

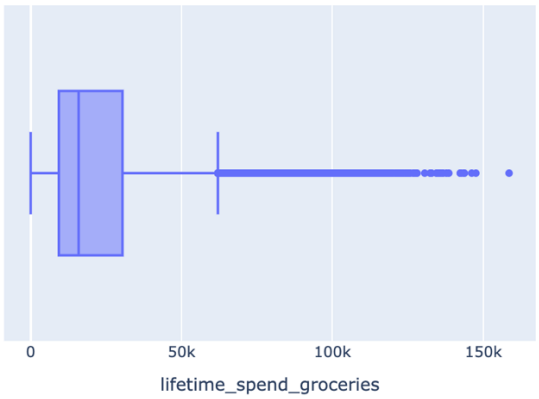


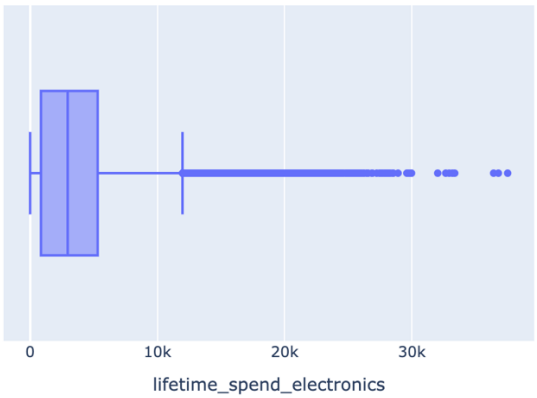
Annex 2: Remaining histograms from the Univariate Analysis in the EDA stage.

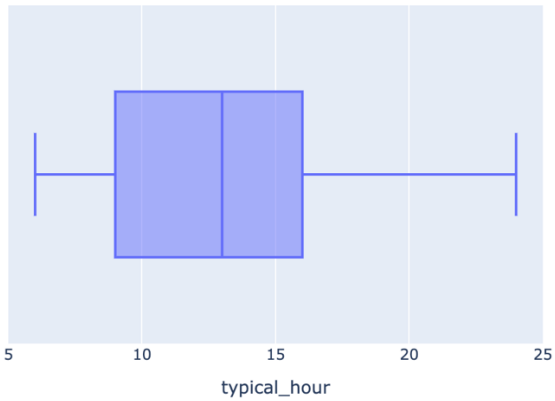


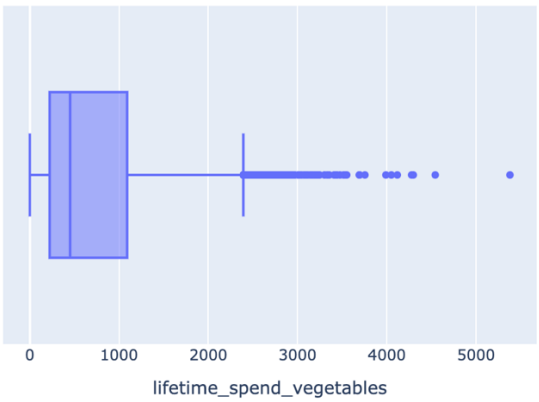

0
1
2
3
4
number_complaints

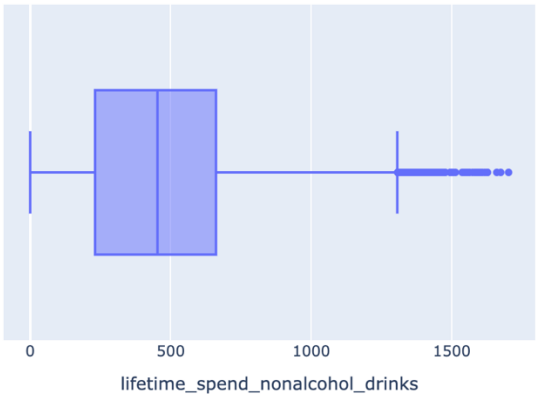

2
4
6
8
10
distinct_stores_visited

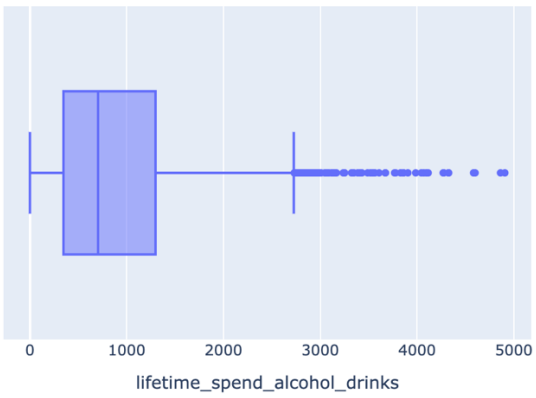

0
50k
100k
150k
lifetime_spend_groceries

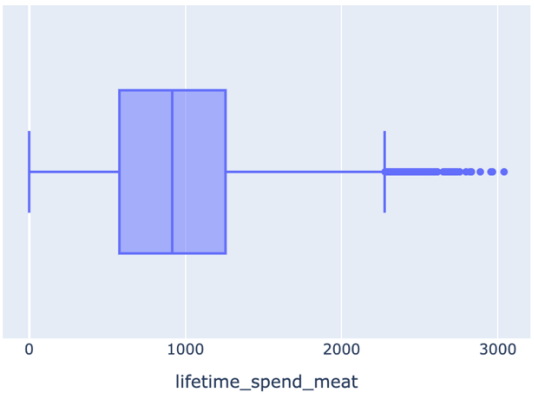

0
10k
20k
30k
lifetime_spend_electronics

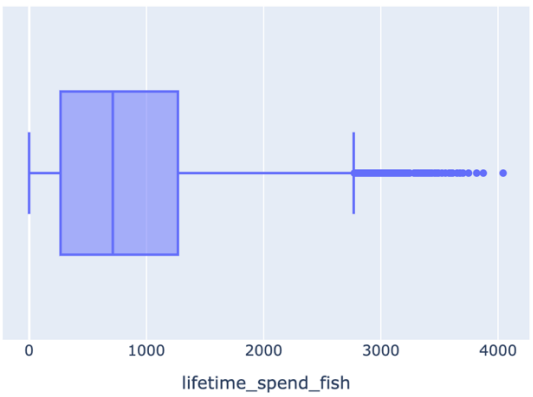

5
10
15
20
25
typical_hour

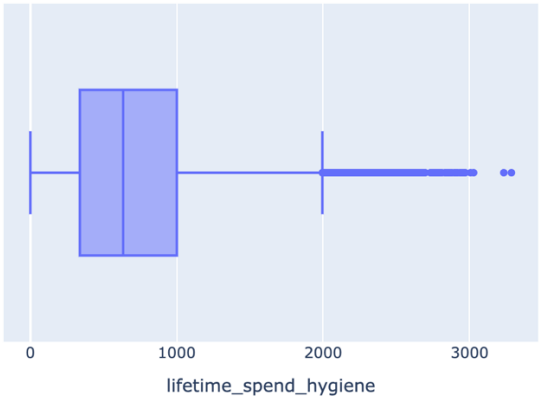

0
1000
2000
3000
4000
5000
lifetime_spend_vegetables

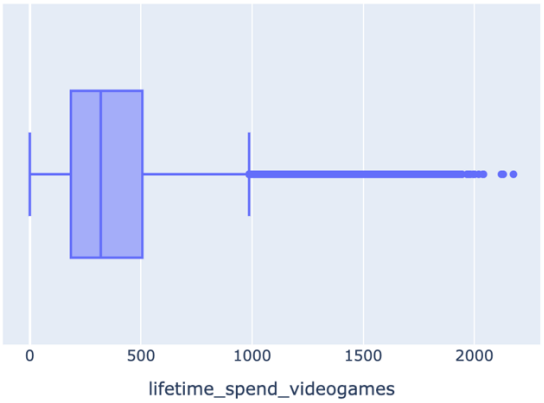

0
500
1000
1500
lifetime_spend_nonalcohol_drinks

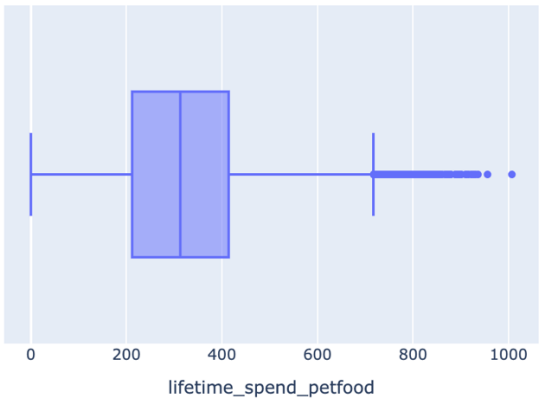

0
1000
2000
3000
4000
5000
lifetime_spend_alcohol_drinks

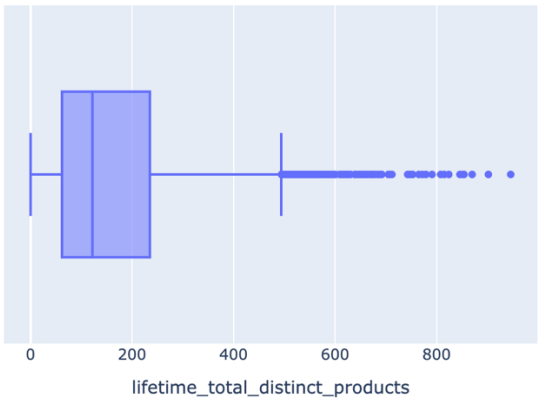

0
1000
2000
3000
lifetime_spend_meat

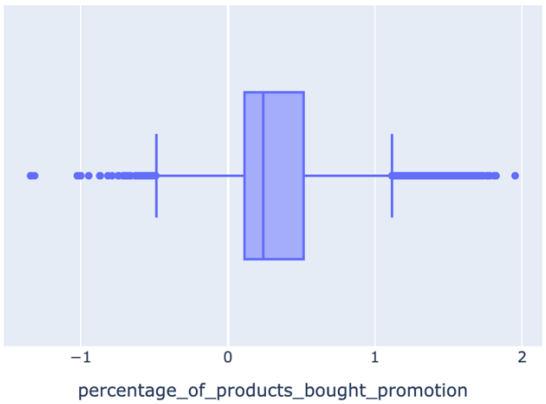

0
1000
2000
3000
4000
lifetime_spend_fish

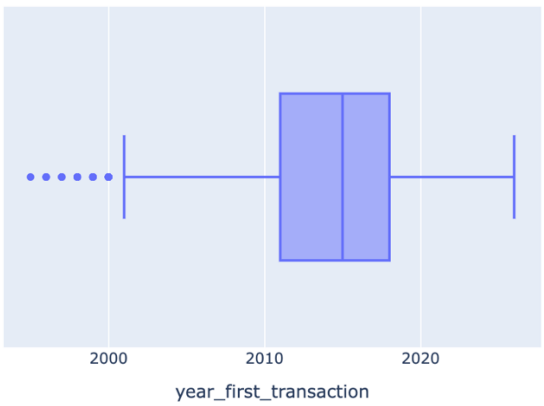

0
1000
2000
3000
lifetime_spend_hygiene

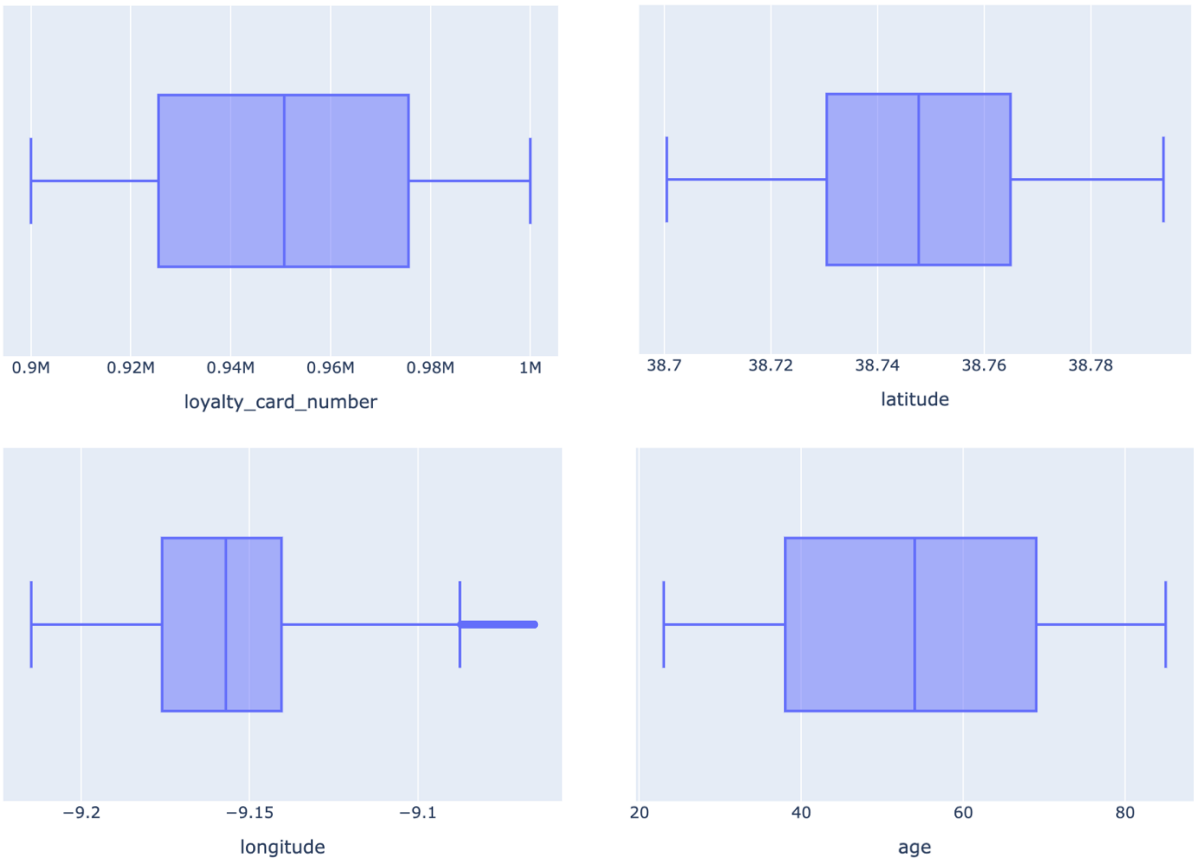

0
500
1000
1500
2000
lifetime_spend_videogames


0
200
400
600
800
1000
lifetime_spend_petfood

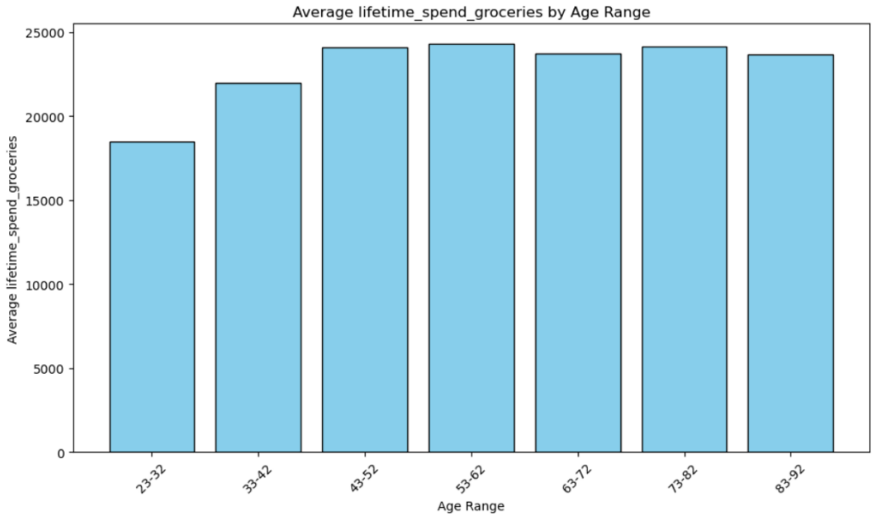

0
200
400
600
800
lifetime_total_distinct_products


−1
0
1
2
percentage_of_products_bought_promotion

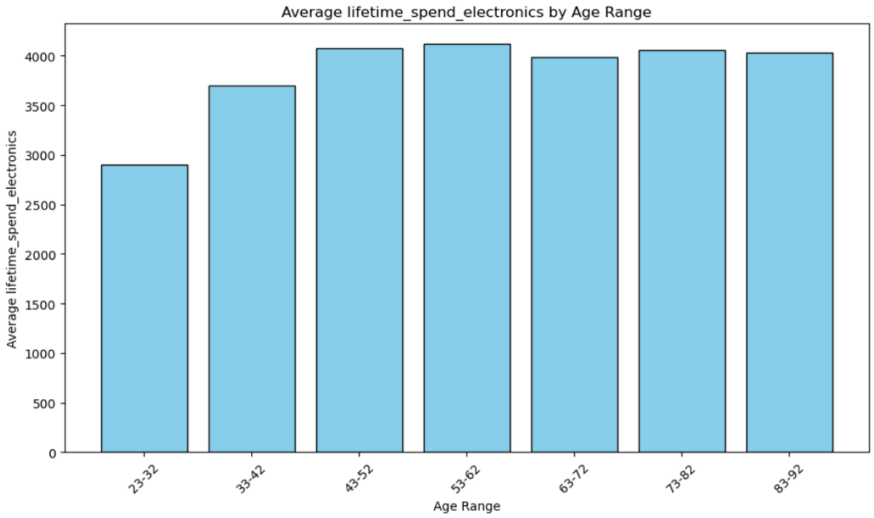

2000
2010
2020
year_first_transaction



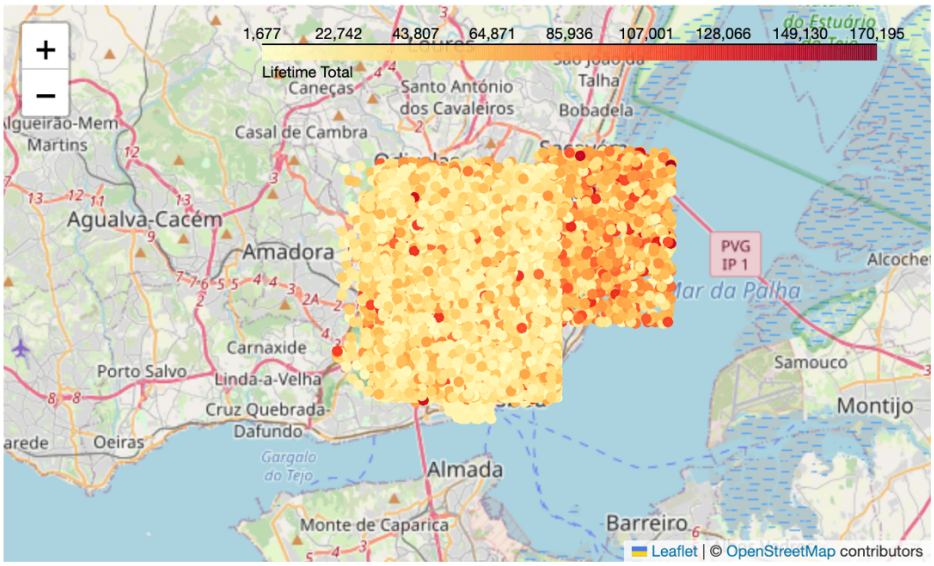
Annex 3: Boxplots from the Univariate Analysis in the EDA stage.



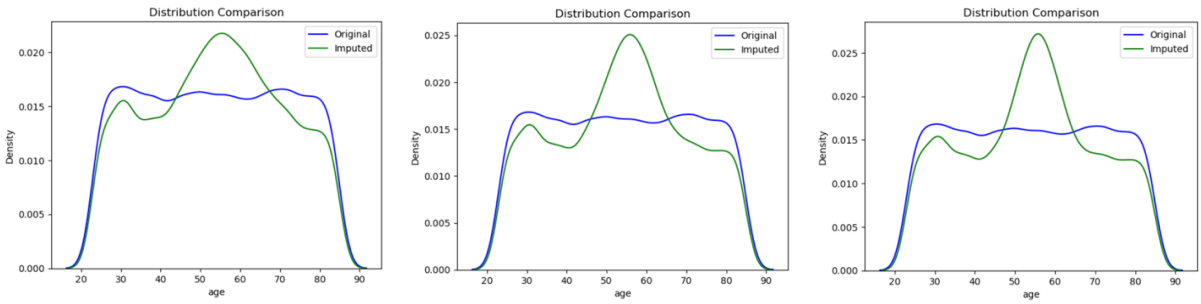
Annex 4: Average lifetime spend on groceries according to customers' age ranges.



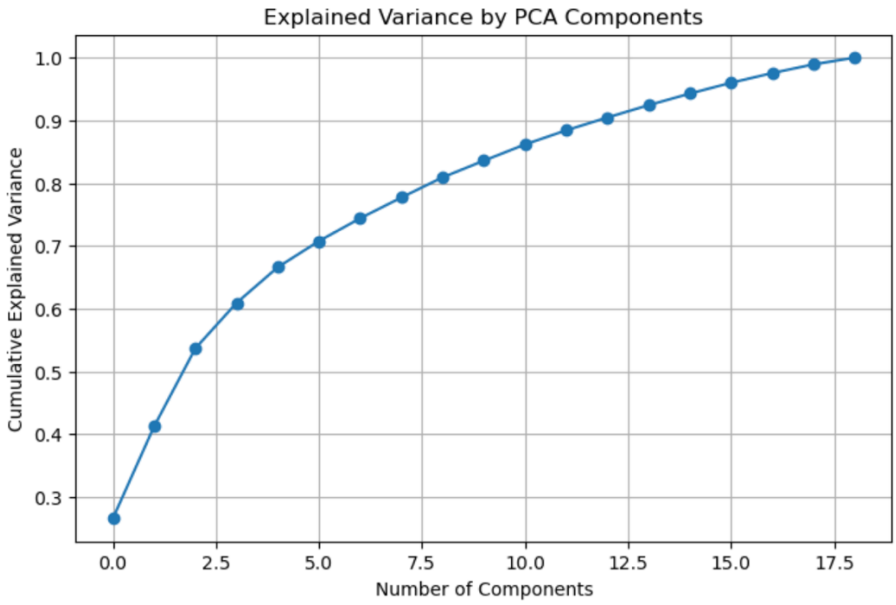
Annex 5: Average lifetime spend on electronics according to customers' age ranges.



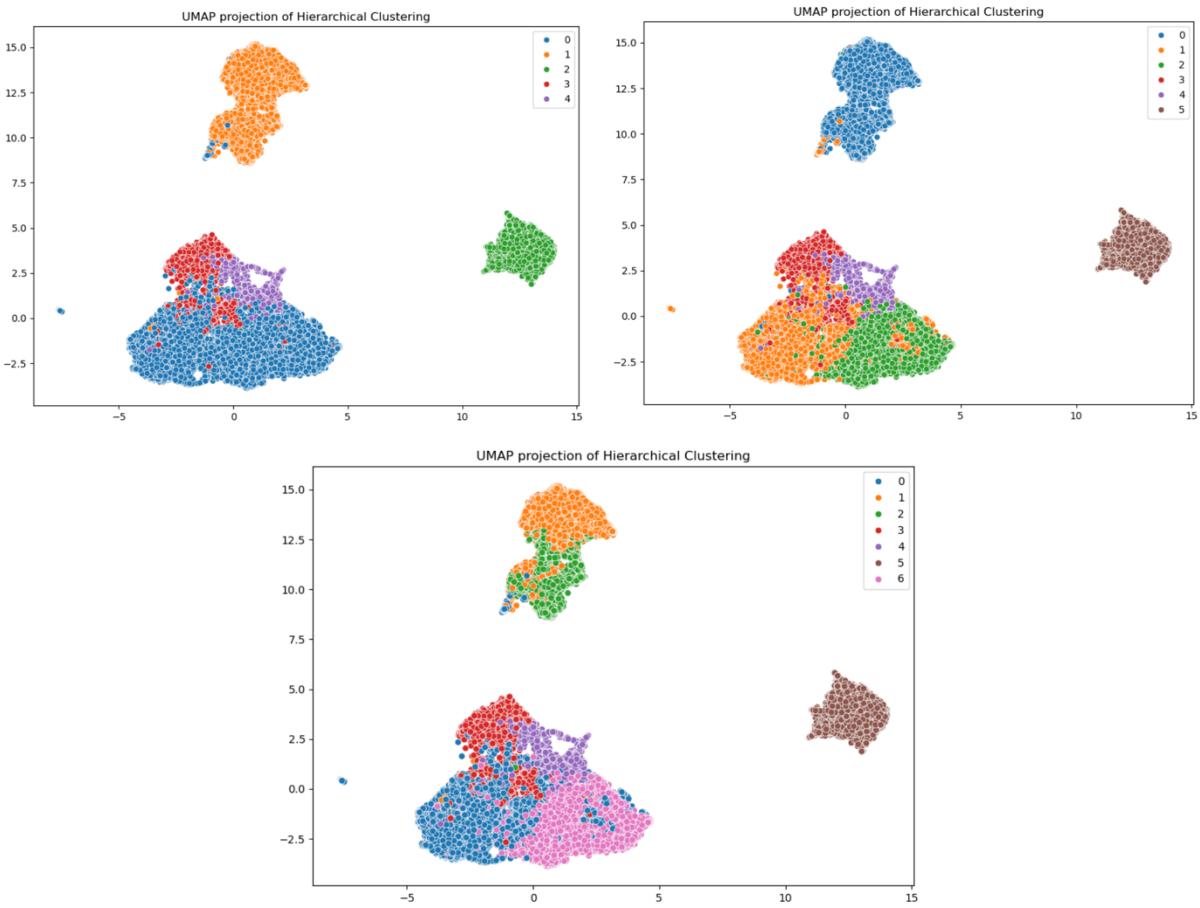
Annex 6: Coordinate map.



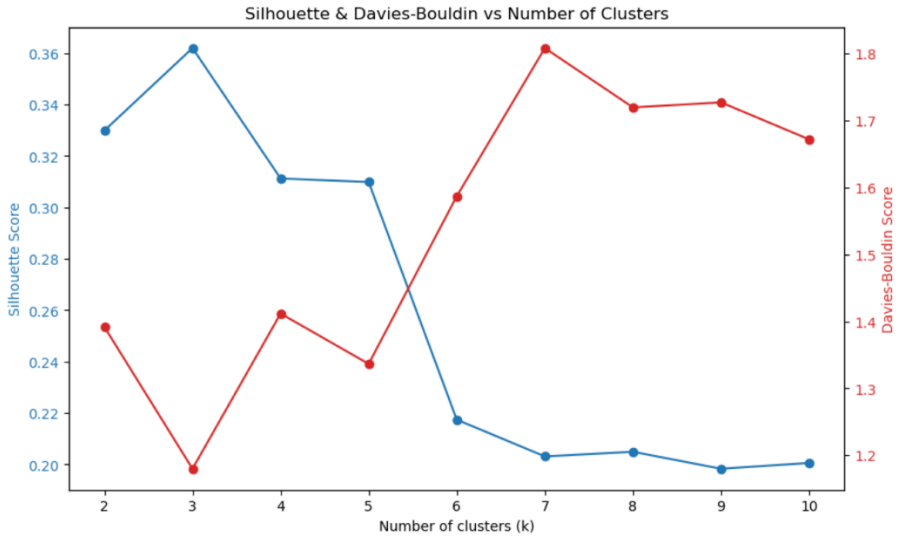
Annex 7: KNN imputation results for k=5, k=10 and k=15, respectively.



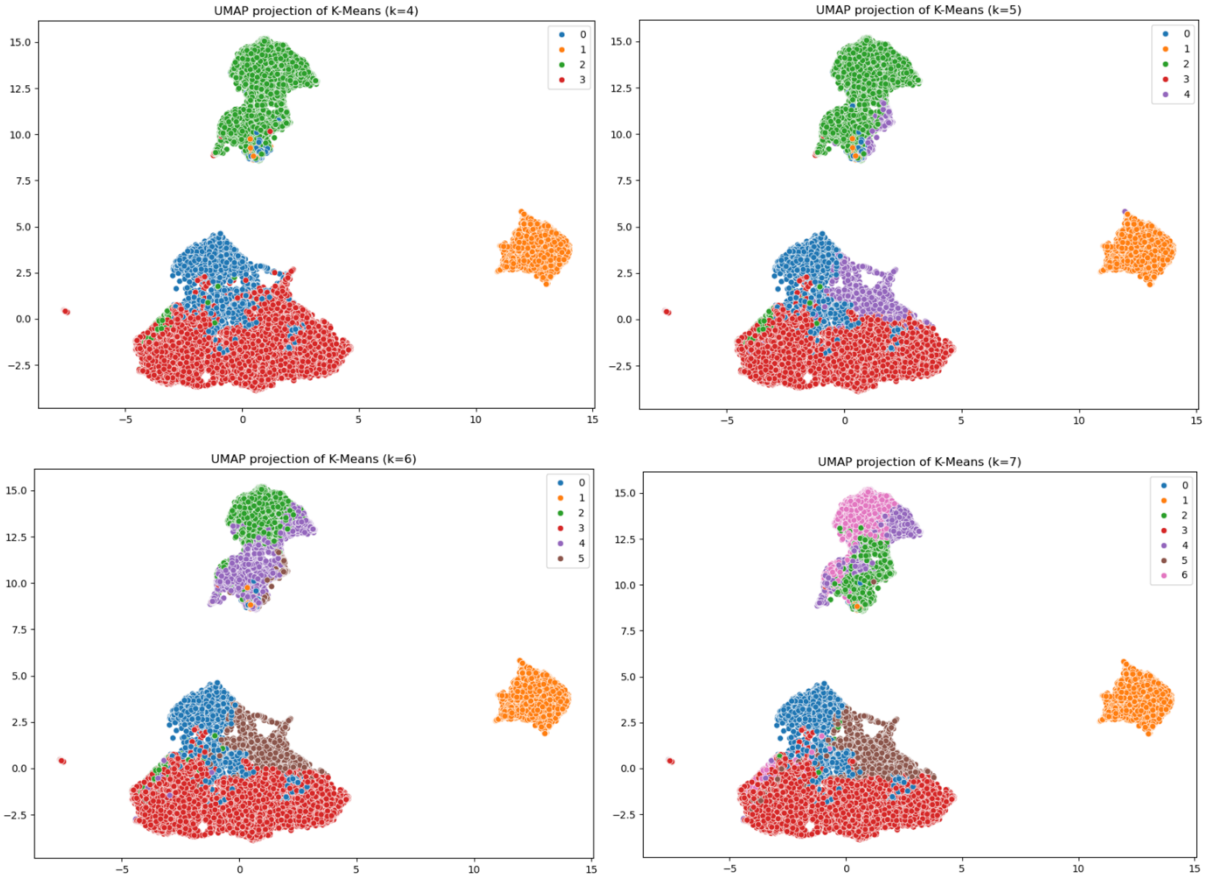
Annex 8: Cumulative Explained Variance plot for the Principal Component Analysis.



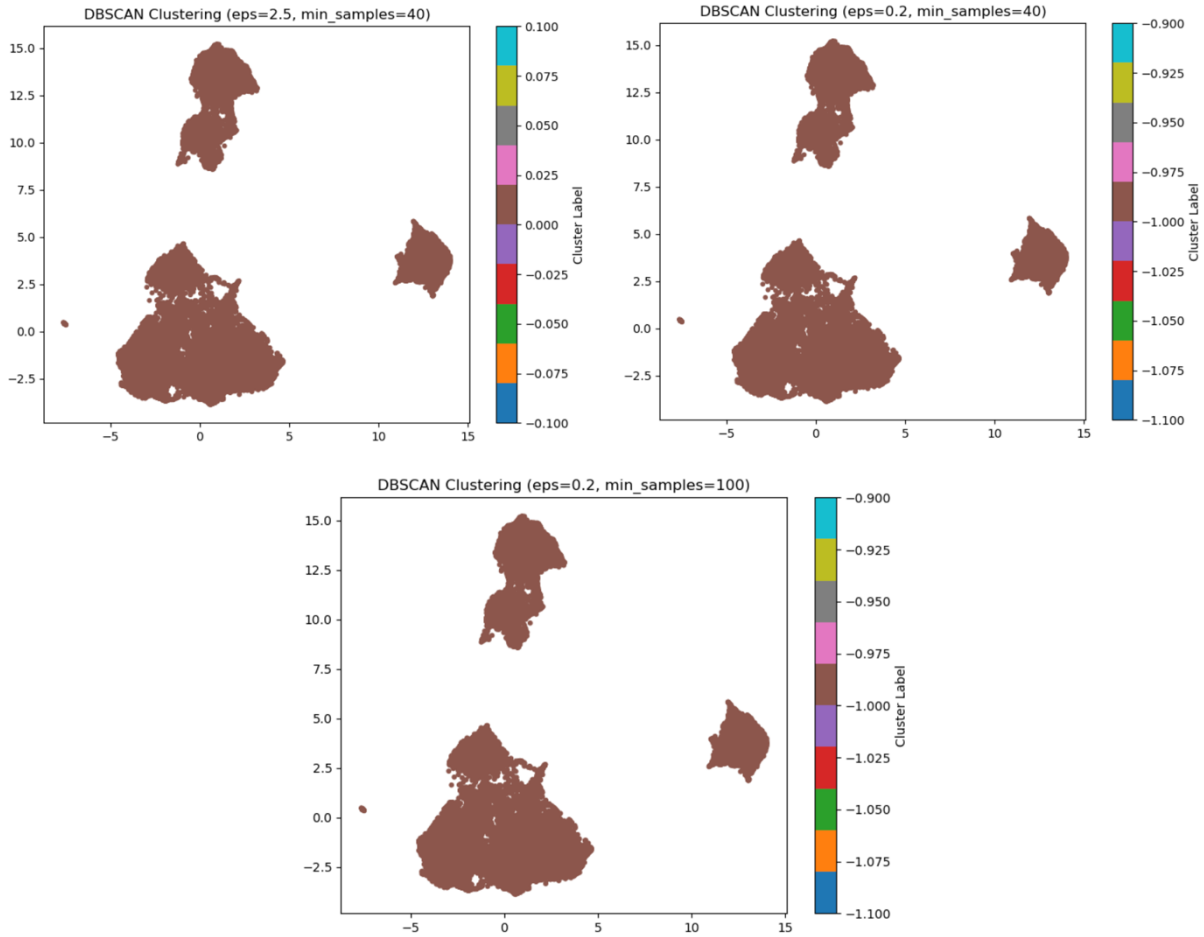
Annex 9: UMAP Projections for Hierarchical Clustering - 5, 6 and 7 clusters, respectively.



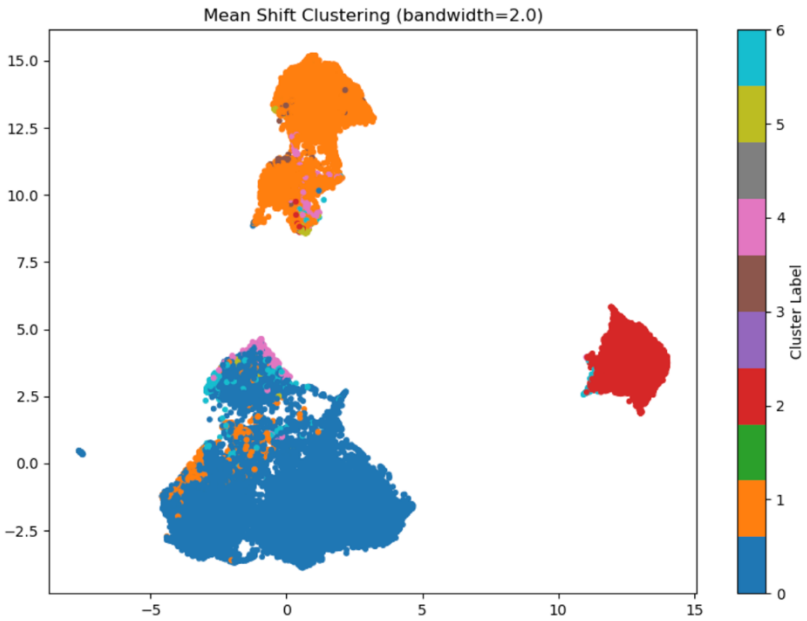
Annex 10: Silhouette and Davies-Bouldin scores for cluster number determination - K-Means.



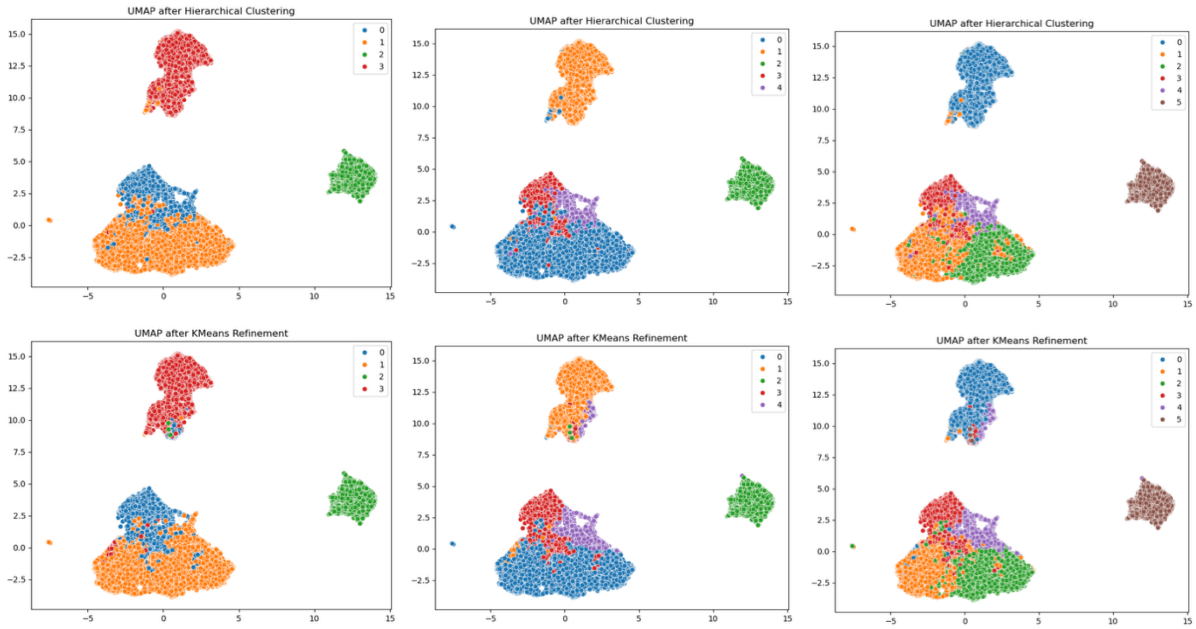
Annex 11: UMAP Projections for K-Means Clustering - 4, 5, 6 and 7 clusters, respectively.



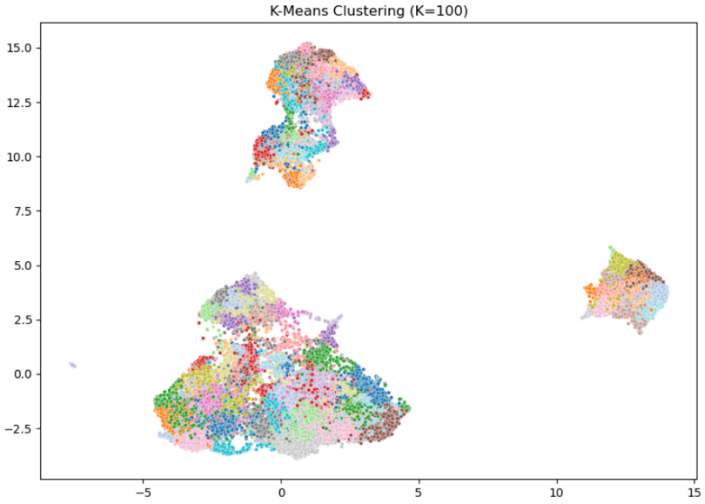
Annex 12: UMAP Projections for DBSCAN Clustering - eps=2.5 and min_samples=40, eps=0.2 and min_samples=40, and eps=0.2 and min_samples=100, respectively.



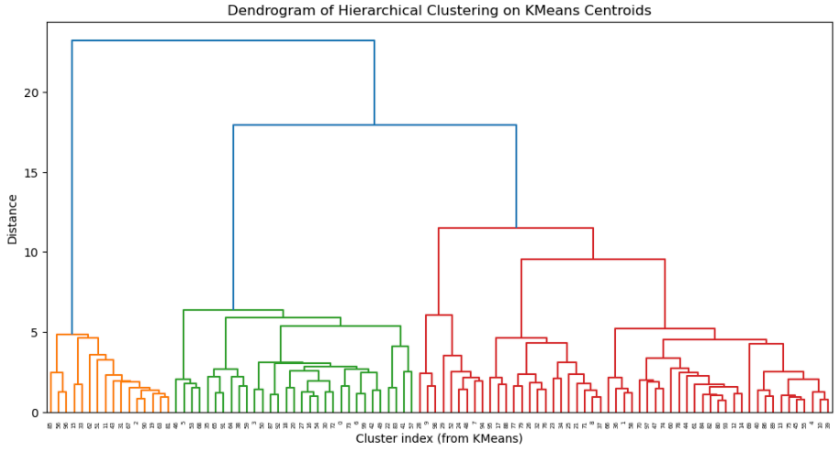
Annex 13: UMAP Projections for Mean Shift Clustering with bandwidth=2.0.



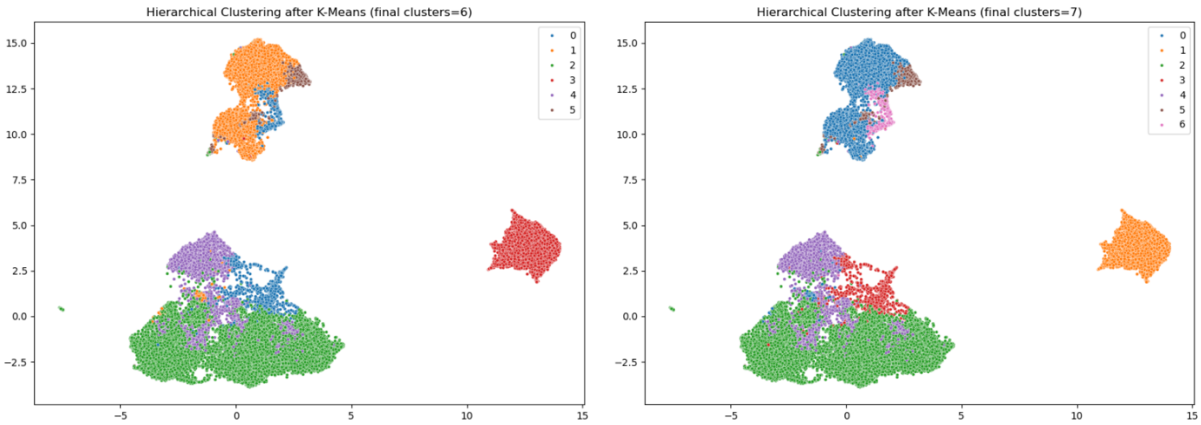
Annex 14: UMAP Projections for the Tandem Approach (Hierarchical + K-Means) – 4, 5 and 6 clusters, respectively.



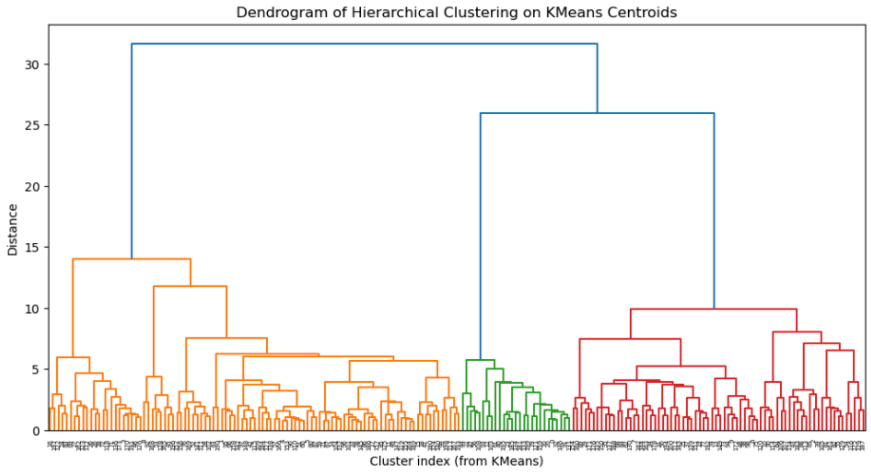
Annex 15: K-Means Clustering with k=100 for the Tandem Approach (K-Means (with large k) + Hierarchical Clustering).



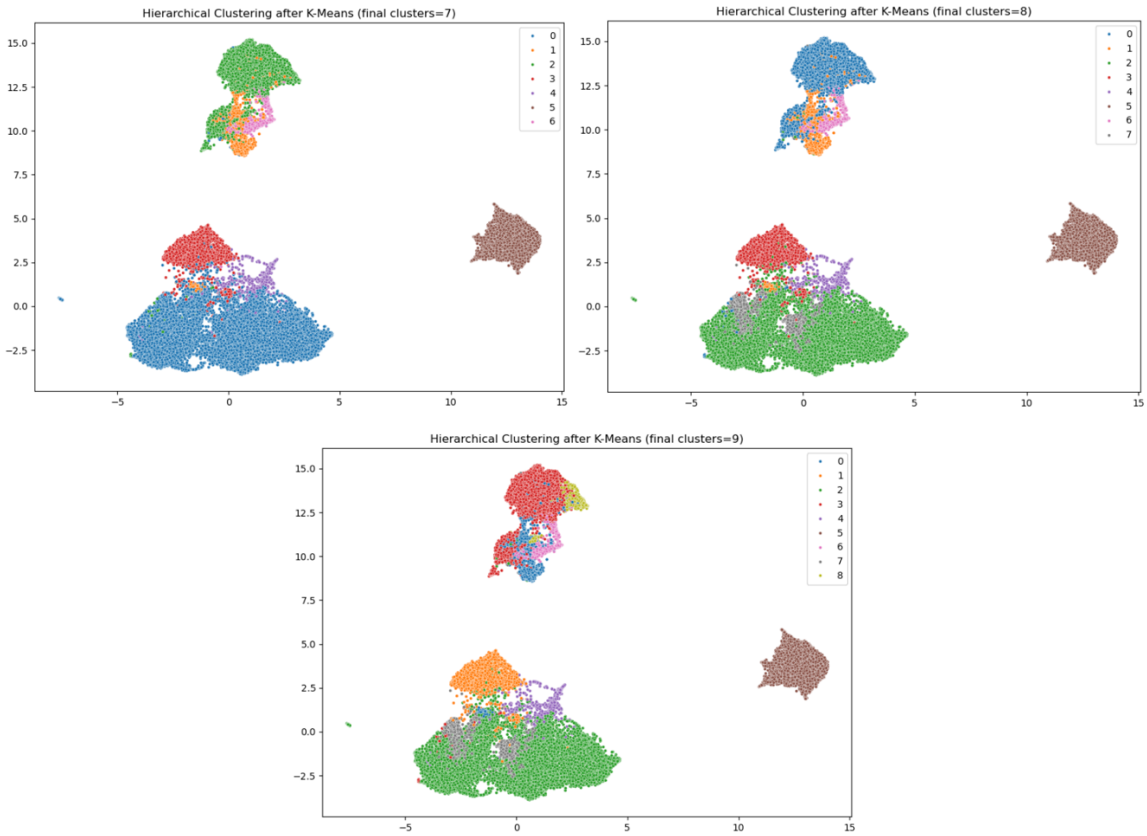
Annex 16: Dendrogram for the Tandem Approach (K-Means (with large k) + Hierarchical Clustering – k=100).



Annex 17: UMAPs for the Tandem Approach (K-Means (with large k) + Hierarchical Clustering) - k=100, 6 and 7 clusters, respectively.



Annex 18: Dendrogram for the Tandem Approach (K-Means (with large k) + Hierarchical Clustering - k=200).



Annex 19: UMAPs for the Tandem Approach (K-Means (with large k) + Hierarchical Clustering) - k=200, 7, 8 and 9 clusters, respectively.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	certainty	kulczynski
410	(oil, muffins)	(cake, tea)	0.178101	0.214206	0.051081	0.286811	1.338950	1.0	0.012931	1.101803	0.308001	0.149700	0.092397	0.262640
408	(cake, tea)	(oil, muffins)	0.214206	0.178101	0.051081	0.238468	1.338950	1.0	0.012931	1.079271	0.322153	0.149700	0.073449	0.262640
409	(cake, muffins)	(oil, tea)	0.100112	0.388143	0.051081	0.510242	1.314572	1.0	0.012224	1.249305	0.265918	0.116844	0.199555	0.320923
437	(oil, muffins)	(tea, cooking oil)	0.178101	0.275727	0.064256	0.360782	1.308474	1.0	0.015148	1.133060	0.286837	0.164939	0.117434	0.296911
438	(tea, cooking oil)	(oil, muffins)	0.275727	0.178101	0.064256	0.233040	1.308474	1.0	0.015148	1.071633	0.325500	0.164939	0.066844	0.296911
439	(cooking oil, muffins)	(oil, tea)	0.128076	0.388143	0.064256	0.501698	1.292560	1.0	0.014544	1.227884	0.259588	0.142170	0.185591	0.333622
412	(muffins)	(cake, oil, tea)	0.220234	0.180462	0.051081	0.231941	1.285262	1.0	0.011337	1.067025	0.284634	0.146107	0.062815	0.257500
404	(cake, oil, tea)	(muffins)	0.180462	0.220234	0.051081	0.283058	1.285262	1.0	0.011337	1.087628	0.270821	0.146107	0.080568	0.257500
399	(gums, cake)	(oil, tea)	0.113721	0.388143	0.056426	0.496175	1.278329	1.0	0.012285	1.214423	0.245667	0.126674	0.176564	0.320774
428	(gums, oil)	(tea, cooking oil)	0.207494	0.275727	0.073018	0.351902	1.276268	1.0	0.015806	1.117536	0.273141	0.178003	0.105174	0.308360

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	certainty	kulczynski
15	(beer)	(salt)	0.171701	0.096735	0.067081	0.390686	4.038730	1.0	0.050472	1.482429	0.908365	0.333149	0.325431	0.542071
16	(salt)	(beer)	0.096735	0.171701	0.067081	0.693456	4.038730	1.0	0.050472	2.702053	0.832975	0.333149	0.629911	0.542071
18	(white wine)	(beer)	0.134163	0.171701	0.088516	0.659768	3.842531	1.0	0.065480	2.434512	0.854381	0.407256	0.589240	0.587646
17	(beer)	(white wine)	0.171701	0.134163	0.088516	0.515524	3.842531	1.0	0.065480	1.787162	0.893102	0.407256	0.440454	0.587646
13	(beer)	(cider)	0.171701	0.095402	0.055420	0.322768	3.383244	1.0	0.039039	1.335729	0.850449	0.261805	0.251345	0.451838
14	(cider)	(beer)	0.095402	0.171701	0.055420	0.580908	3.383244	1.0	0.039039	1.976412	0.778717	0.261805	0.494033	0.451838
210	(asparagus, cooking oil)	(cake, oil)	0.127610	0.291537	0.057641	0.451697	1.549364	1.0	0.020438	1.292101	0.406440	0.159447	0.226067	0.324706
248	(napkins, cooking oil)	(oil, candy bars)	0.143159	0.227454	0.050200	0.350659	1.541669	1.0	0.017638	1.189739	0.410055	0.156672	0.159479	0.285681
249	(oil, candy bars)	(napkins, cooking oil)	0.227454	0.143159	0.050200	0.220703	1.541669	1.0	0.017638	1.099506	0.454798	0.156672	0.090500	0.285681
246	(napkins, oil)	(candy bars, cooking oil)	0.189249	0.172812	0.050200	0.265258	1.534952	1.0	0.017495	1.125821	0.429865	0.160969	0.111760	0.277873

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	certainty	kulczynski
20	(cider)	(white wine)	0.123529	0.156078	0.074608	0.603968	3.869646	1.0	0.055328	2.130944	0.846096	0.363941	0.530724	0.540992
21	(white wine)	(cider)	0.156078	0.123529	0.074608	0.478015	3.869646	1.0	0.055328	1.679111	0.878729	0.363941	0.404447	0.540992
22	(white wine)	(dessert wine)	0.156078	0.101373	0.055686	0.356784	3.519532	1.0	0.039864	1.397085	0.848268	0.275996	0.284224	0.453053
23	(dessert wine)	(white wine)	0.101373	0.156078	0.055686	0.549323	3.519532	1.0	0.039864	1.872564	0.796627	0.275996	0.465973	0.453053
231	(pancakes, energy bar)	(energy drink, protein bar)	0.128824	0.265588	0.057843	0.449011	1.690627	1.0	0.023629	1.332896	0.468910	0.171861	0.249754	0.333402
232	(energy drink, protein bar)	(pancakes, energy bar)	0.265588	0.128824	0.057843	0.217793	1.690627	1.0	0.023629	1.113741	0.556232	0.171861	0.102125	0.333402
211	(gadget for tiktok streaming, pancakes)	(energy drink, airpods)	0.153529	0.200196	0.051373	0.334610	1.671414	1.0	0.020637	1.202009	0.474564	0.169909	0.168059	0.295611
215	(energy drink, airpods)	(gadget for tiktok streaming, pancakes)	0.200196	0.153529	0.051373	0.256611	1.671414	1.0	0.020637	1.138665	0.502253	0.169909	0.121778	0.295611
240	(gadget for tiktok streaming, pancakes)	(energy drink, protein bar)	0.153529	0.265588	0.067647	0.440613	1.659008	1.0	0.026871	1.312887	0.469278	0.192469	0.238320	0.347660
244	(energy drink, protein bar)	(gadget for tiktok streaming, pancakes)	0.265588	0.153529	0.067647	0.254707	1.659008	1.0	0.026871	1.135755	0.540882	0.192469	0.119528	0.347660

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	certainty	kulczynski
32	(white wine)	(cider)	0.145778	0.113215	0.077054	0.528571	4.668753	1.0	0.060550	1.881060	0.919913	0.423517	0.468385	0.604587
31	(cider)	(white wine)	0.113215	0.145778	0.077054	0.680602	4.668753	1.0	0.060550	2.674475	0.886133	0.423517	0.626095	0.604587
42	(dessert wine)	(white wine)	0.093336	0.145778	0.057554	0.616633	4.229941	1.0	0.043948	2.228208	0.842197	0.316997	0.551209	0.505719
41	(white wine)	(dessert wine)	0.145778	0.093336	0.057554	0.394805	4.229941	1.0	0.043948	1.498136	0.893901	0.316997	0.332504	0.505719
22	(white wine)	(beer)	0.145778	0.089549	0.050549	0.346753	3.872200	1.0	0.037495	1.393732	0.868333	0.273566	0.282502	0.455618
21	(beer)	(white wine)	0.089549	0.145778	0.050549	0.564482	3.872200	1.0	0.037495	1.961393	0.814705	0.273566	0.490158	0.455618
29	(champagne)	(fresh tuna)	0.175123	0.110943	0.066073	0.377297	3.400826	1.0	0.046645	1.427739	0.855829	0.300344	0.299592	0.486430
30	(fresh tuna)	(champagne)	0.110943	0.175123	0.066073	0.595563	3.400826	1.0	0.046645	2.039569	0.794048	0.300344	0.509700	0.486430
24	(champagne)	(bluetooth headphones)	0.175123	0.107535	0.060772	0.347027	3.227107	1.0	0.041941	1.366772	0.836640	0.273891	0.268349	0.456084
23	(bluetooth headphones)	(champagne)	0.107535	0.175123	0.060772	0.565141	3.227107	1.0	0.041941	1.896883	0.773280	0.273891	0.472819	0.456084

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	certainty	kulczynski
358	(asparagus, shallot)	(tomatoes, carrots)	0.094832	0.459651	0.054596	0.575710	1.252492	1.0	0.011006	1.273535	0.222712	0.109216	0.214784	0.347243
305	(tomatoes, cauliflower)	(asparagus, carrots)	0.165433	0.390846	0.080622	0.487342	1.246890	1.0	0.015964	1.188227	0.237254	0.169497	0.158410	0.346809
304	(asparagus, carrots)	(tomatoes, cauliflower)	0.390846	0.165433	0.080622	0.206276	1.246890	1.0	0.015964	1.051458	0.325048	0.169497	0.048940	0.346809
306	(carrots, cauliflower)	(asparagus, tomatoes)	0.120859	0.539975	0.080622	0.667079	1.235390	1.0	0.015362	1.381786	0.216733	0.138953	0.276299	0.408193
368	(zucchini, carrots)	(asparagus, tomatoes)	0.100965	0.539975	0.066861	0.662222	1.226395	1.0	0.012343	1.361917	0.205334	0.116467	0.265741	0.393022
303	(asparagus, cauliflower)	(tomatoes, carrots)	0.143146	0.459651	0.080622	0.563218	1.225316	1.0	0.014825	1.237114	0.214604	0.154397	0.191667	0.369309
367	(zucchini, tomatoes)	(asparagus, carrots)	0.140154	0.390846	0.066861	0.477054	1.220569	1.0	0.012082	1.164852	0.210166	0.144054	0.141522	0.324061
359	(tomatoes, shallot)	(asparagus, carrots)	0.114651	0.390846	0.054596	0.476190	1.218359	1.0	0.009785	1.162931	0.202433	0.121081	0.140103	0.307938
296	(avocado, tomatoes)	(asparagus, carrots)	0.118166	0.390846	0.056241	0.475949	1.217742	1.0	0.010056	1.162396	0.202768	0.124215	0.139708	0.309923
300	(asparagus, tomatoes, carrots)	(cauliflower)	0.333034	0.198863	0.080622	0.242084	1.217339	1.0	0.014394	1.057026	0.267684	0.178654	0.053949	0.323750

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	certainty	kulczynski
153	(spaghetti, bluetooth headphones)	(fresh tuna)	0.080245	0.275775	0.051824	0.645822	2.341847	1.0	0.029694	2.044808	0.622977	0.170363	0.510956	0.416871
177	(fresh tuna)	(spaghetti, champagne)	0.275775	0.137592	0.088467	0.320795	2.331491	1.0	0.050523	1.269731	0.788553	0.272291	0.212432	0.481881
172	(spaghetti, champagne)	(fresh tuna)	0.137592	0.275775	0.088467	0.642967	2.331491	1.0	0.050523	2.028452	0.662204	0.272291	0.507013	0.481881
158	(cottage cheese, champagne)	(fresh tuna)	0.089100	0.275775	0.056884	0.638429	2.315036	1.0	0.032312	2.002994	0.623604	0.184693	0.500748	0.422349
162	(fresh tuna)	(cottage cheese, champagne)	0.275775	0.089100	0.056884	0.206269	2.315036	1.0	0.032312	1.147619	0.784344	0.184693	0.128630	0.422349
167	(frozen smoothie)	(champagne, fresh tuna)	0.146490	0.218385	0.073919	0.504606	2.310624	1.0	0.041928	1.577763	0.664569	0.254058	0.366191	0.421544
163	(champagne, fresh tuna)	(frozen smoothie)	0.218385	0.146490	0.073919	0.338482	2.310624	1.0	0.041928	1.290231	0.725698	0.254058	0.224945	0.421544
166	(fresh tuna)	(champagne, frozen smoothie)	0.275775	0.116213	0.073919	0.268043	2.306471	1.0	0.041871	1.207429	0.782129	0.232401	0.171794	0.452055
164	(champagne, frozen smoothie)	(fresh tuna)	0.116213	0.275775	0.073919	0.636067	2.306471	1.0	0.041871	1.989995	0.640921	0.232401	0.497486	0.452055
168	(laptop, champagne)	(fresh tuna)	0.078980	0.275775	0.050221	0.635878	2.305788	1.0	0.028441	1.988964	0.614871	0.164913	0.497226	0.408994

Annex 20: Tables resulting from the Association rules for cluster 0, 1, 2, 3, 4, 5 and 6, respectively.