

**Product Cluster Analysis  
for  
Online Grocery Solutions**

by YorkU Students

**CSML1000 Machine Learning in Business  
Context - Blended Live Online Fall 2024**

Project #2

2nd November, 2024

**Arslan Abakarov  
Daniel (Jee Hwan) Lee  
Karen Krucik, MBA(Oxon)  
Aayush Bhatt  
Mathi Mahalinga**

## Abstract

Supermarkets today face the challenge of analyzing their customer's product needs to offer a personalized shopping experience. With a wide range of products and diverse customer preferences, supermarkets often struggle to make relevant product recommendations that resonate with individual shoppers. We have developed a web application to help customers find products and then be offered with additional and relevant product choices based on the customer's own purchasing history.

## Background

A team from York University was tasked with developing a recommendation model that suggests products based on customers' past purchases, purchase frequency, and overall order history. By aggregating this data, they created a model capable of analyzing customer behavior to generate personalized product recommendations.

The team then deployed this model as a product recommendation website for online groceries. When customers search for items, the team's website, powered by modern machine learning and deep data analytics, dynamically displays relevant product recommendations on their screen. This in turn enhances customer shopping experiences and increases the shopping center's overall productivity.

## Objective

This project focuses on creating a product recommendation model for integration with existing eCommerce platforms. By analyzing customer data, it delivers product suggestions that enhance the shopping experience. Designed for easy adaptation, this add-on can be implemented across various retail sites, providing supermarkets and online stores with a valuable tool.

Further, the same data set can be used to cluster or group customers into those with similar purchasing characteristics, but delineated or distinct from each other. This cluster analysis enables effective marketing techniques based on customer purchase analytics.

For purposes of this study, the team focuses on grocery purchases and based its machine learning models upon historical purchasing behaviour of grocery purchase orders.

## Data Analysis

The dataset used to build the model comprises detailed order information from 2,019,501 online store purchase records, each containing 12 unique features. Sourced originally from Kaggle<sup>1</sup>, the extensive dataset provides essential and useful information, making it highly suitable for developing a highly accurate and effective predictive model.

---

<sup>1</sup> <https://www.kaggle.com/datasets/hunter0007/ecommerce-dataset-for-predictive-marketing-2023>

*Description of features:*

Column Name	Description
order_id	A unique number to identity the order
user_id	A unique number to identify the user
order_number	Number of the order
order_dow	Day of the Week the order was made
order_hour_of_day	Time of the order
days_since_prior_order	History of the order
product_id	Id of the product
add_to_cart_order	Number of items added to cart
reordered	If the reorder took place
department_id	Unique number allocated to each department
department	Names of the departments
product_name	Name of the products

## Data Exploration

The initial exploration of the dataset focuses on identifying trends and correlations between features. Key steps include analyzing the data for missing values and outliers, ensuring the dataset is clean, well-structured and ready for model development.

Given that clustering can be RAM and computationally demanding, the team adopted a fractional approach that takes a random subset of data for learning and training. This subset represents 50%, or 0.5, of the original dataset.

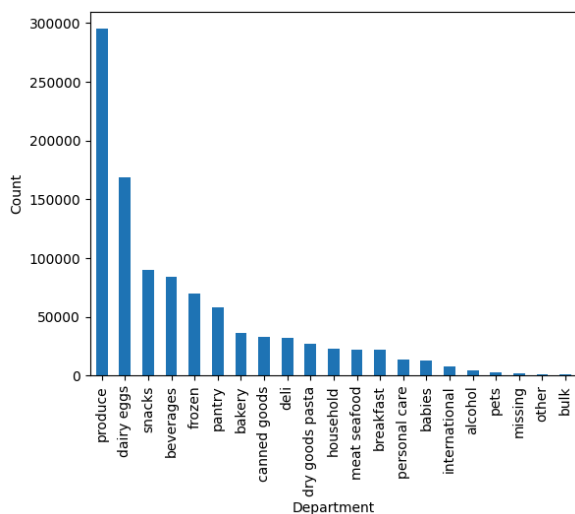
The following table provides statistical analysis of the data set and assists in a high level understanding of the distribution of values:

```
df.describe()
```

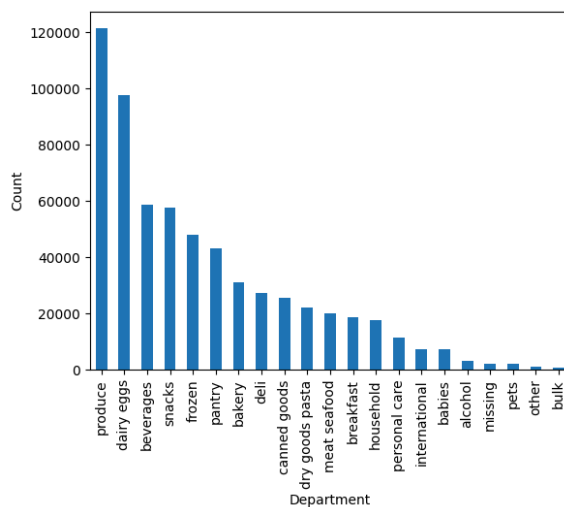
	order_id	user_id	order_number	order_dow	order_hour_of_day	days_since_prior_order	product_id	add_to_cart_order	reordered	department_id
count	2.019500e+04	20195.000000	20195.000000	20195.000000	20195.00000	18954.000000	20195.000000	20195.000000	20195.000000	20195.000000
mean	1.704271e+06	103694.308641	17.236791	2.734835	13.40619	11.479107	70.902154	8.276851	0.589304	9.976529
std	9.824355e+05	59651.029882	17.503683	2.089664	4.23877	9.008237	38.242132	7.020997	0.491972	6.273562
min	2.430000e+02	2.000000	1.000000	0.000000	0.00000	0.000000	1.000000	1.000000	0.000000	1.000000
25%	8.557035e+05	51616.500000	5.000000	1.000000	10.00000	5.000000	31.000000	3.000000	0.000000	4.000000
50%	1.700810e+06	103732.000000	11.000000	3.000000	13.00000	8.000000	83.000000	6.000000	1.000000	9.000000
75%	2.558420e+06	155813.500000	24.000000	5.000000	16.00000	16.000000	107.000000	11.000000	1.000000	16.000000
max	3.420952e+06	206187.000000	100.000000	6.000000	23.00000	30.000000	134.000000	104.000000	1.000000	21.000000

- Days since prior order: Days since prior order ranged from 0 to 30.
- Add to cart order: This represents the number of items in the cart , this ranges from 1 to 104 items

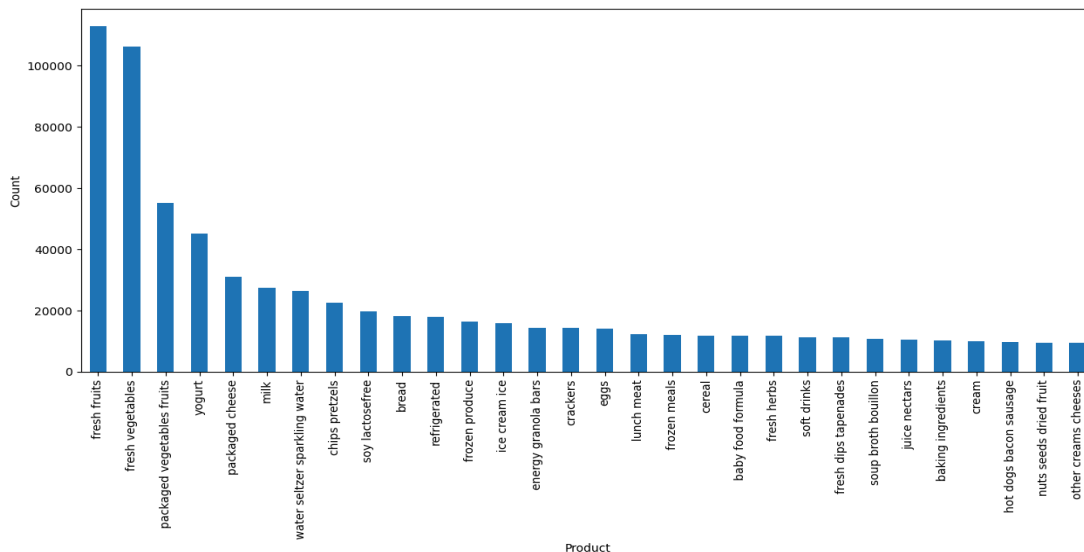
The following graphs indicate the representation of features across the data set. In this exploration, we cannot suggest strong correlation between each feature:



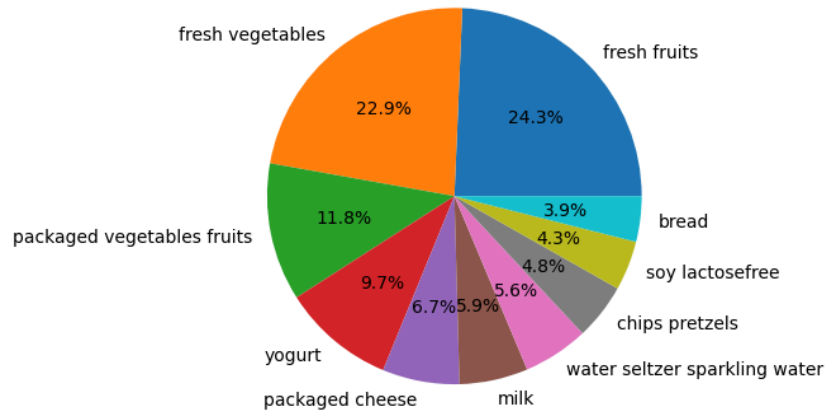
Class Distribution of Department with Duplicate Order



Class Distribution of Department with Unique Order



*Class Distribution of Top 30 Products*



*Pie Chart of Top 10 Products*

## Most and Least Purchased Items

For general data exploration at a high level, we can also look at the most and the least frequently purchased items.

This information can help for inventory purposes, for strategic repositioning of the companys' products and for competitor positioning.

### Ranking by Greatest and Least Purchased Item Frequency

product_name	
fresh fruits	0.111951
fresh vegetables	0.105255
packaged vegetables fruits	0.054515
yogurt	0.044762
packaged cheese	0.030601
...	
kitchen supplies	0.000274
baby bath body care	0.000250
baby accessories	0.000229
beauty	0.000180
frozen juice	0.000154

The information above indicates the highest frequency purchase items as being fresh fruits, fresh vegetables, packaged vegetables fruits etc. While the lowest demanded items were the second group of numbers underneath starting with kitchen supplies, baby bath body care etc.

## Departments with the Greatest Sales

In the data set we also explored which departments had the greatest proportion of overall sales.

In the data below, we present the top departments by sales. These are departments which the firm will do well to invest heavily in and to expand their product offering.

### Top Departments By Sales

department	
produce	294887
dairy eggs	168475
snacks	90050
beverages	84035
frozen	69832
pantry	58021
bakery	36400
canned goods	33026
deli	32452
dry goods pasta	27098

## Data Preparation and Feature Engineering

Feature engineering helps to enhance the quality and relevance of the data, which ultimately leads to better-performing models. For this diabetes prediction model, the goal is to achieve high predictability on unseen data and effective feature engineering plays a key role in that.

Key steps in this process include:

- Checking for missing values
- Checking for outliers
- Encoding categorical variables
- Managing duplicates
- Data correlation and observations

### Checking for missing values

To ensure the quality of the dataset, the `isnull()` function from the Pandas library will be used to check for missing values. Missing or empty values can significantly reduce the model's performance by introducing inaccuracies and bias in predictions.

We checked for null values by summing the number of nulls in each feature. This produced the following outcome:

```
# CHECK FOR NULLS
```

```
df.isnull().sum()
```

```
order_id      0
user_id       0
order_number   0
order_dow      0
order_hour_of_day  0
days_since_prior_order  62030
product_id     0
add_to_cart_order  0
reordered      0
department_id  0
department     0
product_name    0
dtype: int64
```

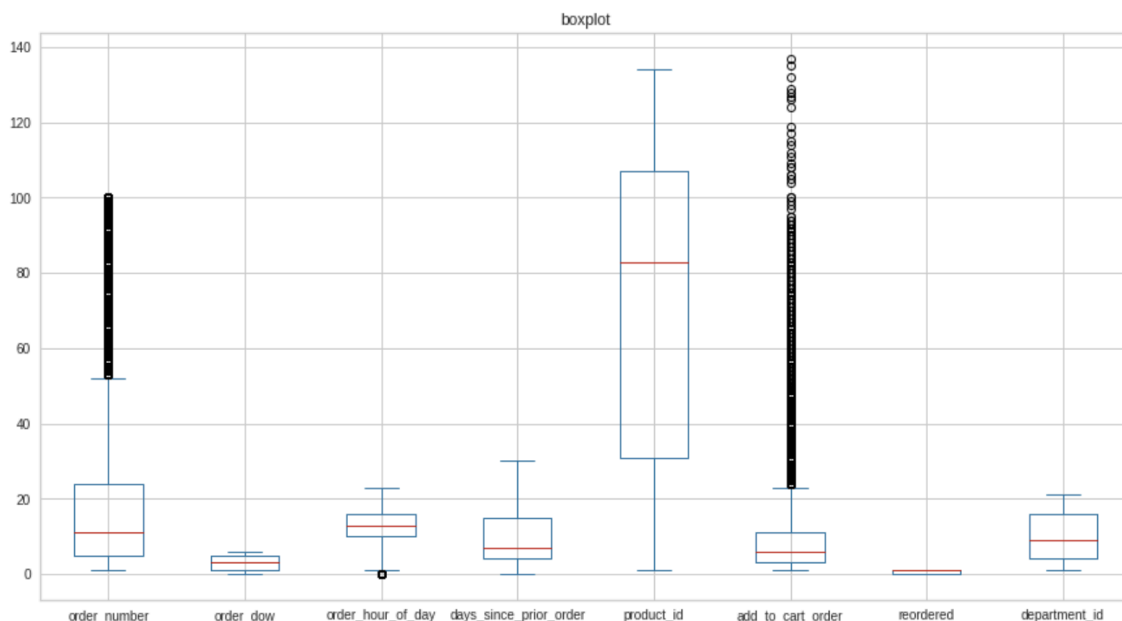
### Checking Orders for NAN days since prior order

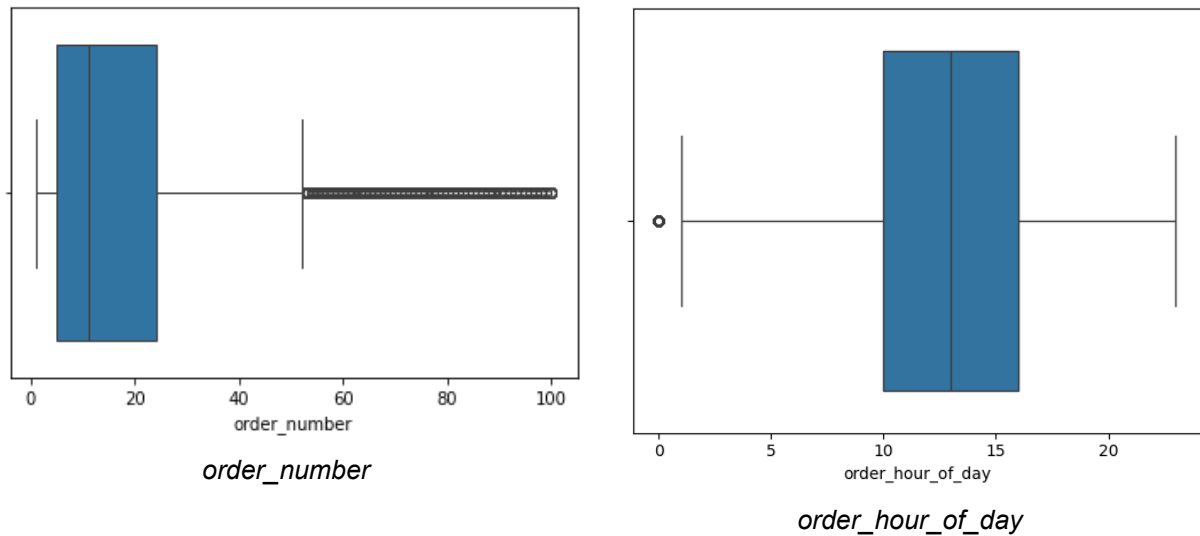
During data exploration, we noted that there were no missing values for most features. However, as may be seen above, there are 62030 entries where the `days_since_prior_order` had null values. So, we investigated to try to determine if they represent missing data or whether they were 'zero' values, i.e. the client had no days since prior order as it was the client's first order.

Upon further investigation, it was determined the null values were related to account history. In this instance, the numbers represented that they were the clients' first orders and, thus, there was no number of days since prior order. The null values were changed to zeros for modelling purposes.

### Checking for Outliers

As a next step, we focus on identifying outliers, which can negatively impact model training and performance by skewing predictions. As many of fields represent numerical representations of order numbers, or user numbers or product ids, these fields were not as useful for determining and removing outliers.

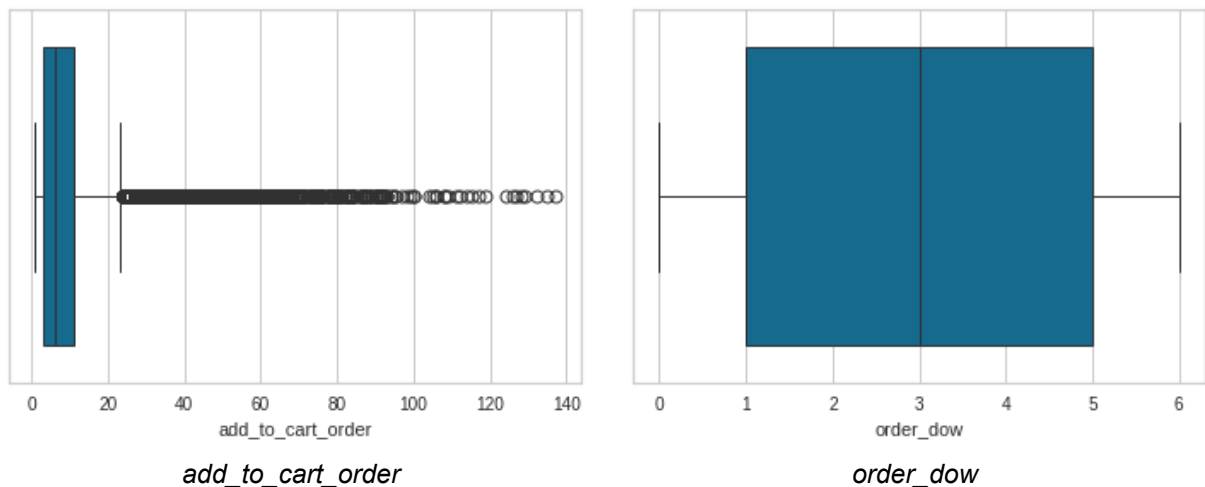




Thus, while order number and its outliers as a number when taken by itself may not provide much information due to it being a reference value, there were some results which were of interest. For example and as demonstrated above, order hour of day is helpful and as it indicates that the preponderance of orders were placed between 10 am and 3pm.

This time span result is relevant for a number of reasons. From a website support and scaling perspective, it indicates that the web servers need to be dynamically increased during this period and may be significantly reduced outside of these hours – thereby reducing overall spend dramatically by matching web server support to web server demand.

Further, if offline order support ( for example telephone support) or customer service support for online purchases are needed, the results indicate that staffing levels will need be significantly greater during these hours of the day to match demand.





Looking at other values, for example, add to cart order, we note that this value will also not provide significant information when viewed in isolation. However, a different value, the order day of week (order\_dow) will.

The order day of week, as demonstrated in the above right diagram, indicates that orders were placed during a Monday to Friday period. Like the order hour of day, this information provides business valuable insight. For example, knowing that orders are not placed on a weekend helps companies to focus on ensuring robust web server support levels, staffing and call center support during the regular Monday to Friday work week.

## One-Hot Encoding

The product\_id and department\_id were in the data both as names (string values) and as numerical id values. We carefully ensured that they were unique and matching, and chose to drop the string values as the numerical data was easier to manage for modelling purposes. This task was done with no degradation of data accuracy.

### *Checking if product id is unique for product name*

To ensure that there was no error in the matching of product\_name and product\_id and for prudence, we checked to confirm that product id was indeed unique for each product name.

We used coding to loop through the array and perform a **isUnique()** function to make product ID to product name. This confirmed that there was a one to one relationship between the two.

Thus, we were able to proceed with exclusively using product id and department id as numerical values rather than the string or named version of the feature.

## Feature Aggregation

In order to simplify and highlight patterns in customer behaviour, aggregation of features was performed.

Aggregated features assist in summarizing key behaviours. Some of the other benefits of aggregated features are dimensionality reduction, computational efficiency and clarity for recommendations.

Following are the aggregated features:

**purchase\_count=('order\_id', 'count'): *number of times each user has purchased each product***

Users clustered on the basis of purchase\_count are grouped with similar purchasing habits, assisting in recommending “Frequently bought together” products.

**reorder\_rate=('reordered', 'mean'): *reordering a product***

This cluster depicts customer’s loyalty to a product. It suggested consisted preferences of users

**avg\_days\_between=('days\_since\_prior\_order', 'mean'): *average time between purchases for user product pair***

Identifies purchasing cycles for users and recommend products based on their typical purchasing schedule

**avg\_cart\_position=('add\_to\_cart\_order', 'mean'):** *mean of each product in cart*

Commonly bought products, they can be shown first in suggested products. Low average means high priority for that product.

## Training The Model

We clustered the users using two K means clustering and DBSCAN clustering algorithms and compared the silhouette score for both.

	K-Means Algorithm	DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
Pros	Often used for customer segmentation due to its simplicity and interpretability	Useful if you want to identify core samples and detect noise (anomalous customers)
Cons	May require pre-scaling and could be sensitive to the number of clusters chosen	Can be computationally demanding for large datasets, depending on the distance metric

### Our Choice

K-Means clustering is often recommended for customer and product recommendation. As product segmentation followed by product recommendation, we have initially proceeded with the k means algorithm.

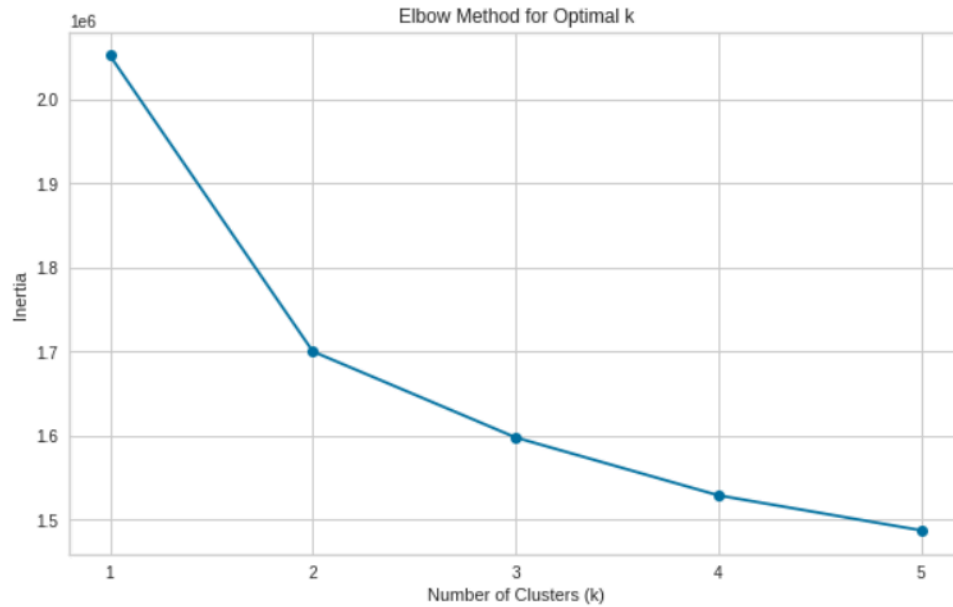
### Feature Importance:

Using a higher number of features can potentially provide more information for the K-Means algorithm to discover clusters based on complex relationships between features.

## K Means Algorithm

Notwithstanding the above, we wished to explore the data further and ensure that the k means algorithm did produce the optimal output.

Our graphical output follows on our first exploratory, k mean, is below:



The "elbow" or "kink" in the graph above suggests that the data supports two distinct clusters. This is further demonstrated by the dot graph, which illustrates the behavior, uniqueness, and spatial distribution of each cluster. Additionally, we used the silhouette score as a metric to evaluate cluster cohesion and separation.

Through an iterative process, we trained the model with various cluster counts and found that the silhouette score improved as the number of clusters decreased. Ultimately, we produced the optimal score in just two clusters.

As above, the silhouette score drops further as we increase the number of clusters.

Combining the elbow produced by the k-means algorithm and the silhouette score, we have found it to be a reasonable conclusion that 2 clusters are the best number of clusters for our current set of features and the dataset.

### *Data Output Details*

#### **Silhouette Score (0.4407)**

A silhouette score of 0.44 suggests a moderately defined clustering.

**Range:** Silhouette scores range from -1 to 1.

**0.44 Score:** In general, scores between 0.4 and 0.6 generally indicate clusters that are reasonably well-separated but not entirely distinct. While this is an improvement from lower scores, there may still be some overlap or data points near cluster boundaries.

Our clustering is likely capturing some structure in the data but might not perfectly segment it. This could mean there's still some overlap, but the clusters are beginning to take shape.

Thus, this score may indicate that the chosen number of clusters (**k**) is close to optimal, especially if our elbow plot supports this choice.

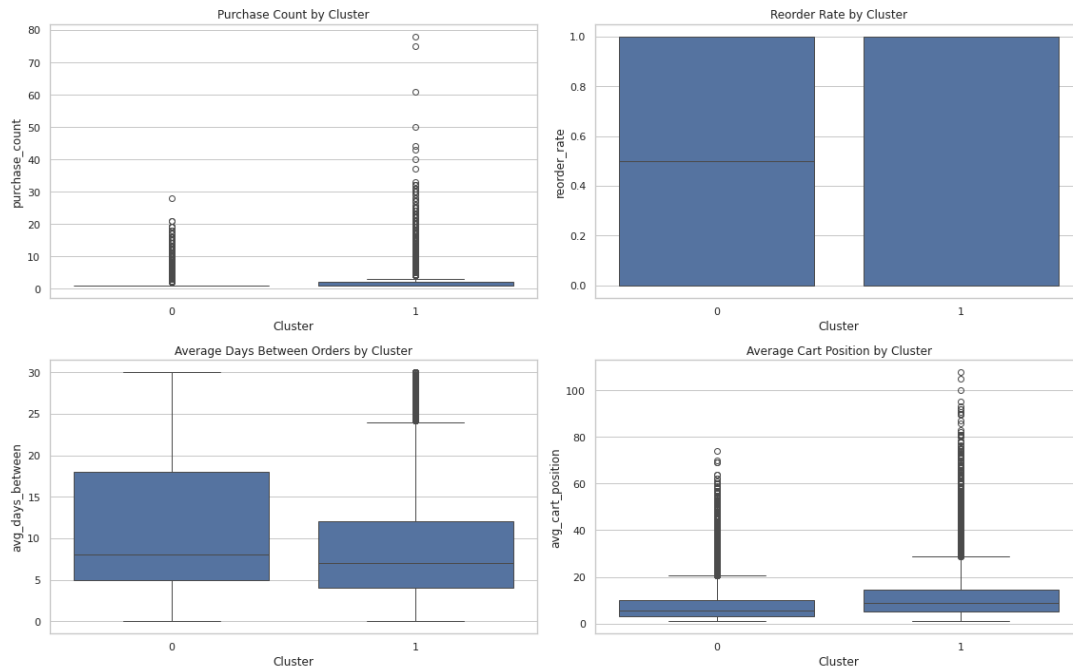
In sum, our silhouette score of **0.44** is moderately good, showing that our clusters are defined but not strongly separated.

### *Analysis of Clusters*

**Average Cart Position:** Higher average cart position means products are added later in shopping process, reflecting user behaviour

**Purchase Count:** Reflects total number of purchases made by the users in the dataset. User can be considered more prone to buy items if the users purchase count is higher

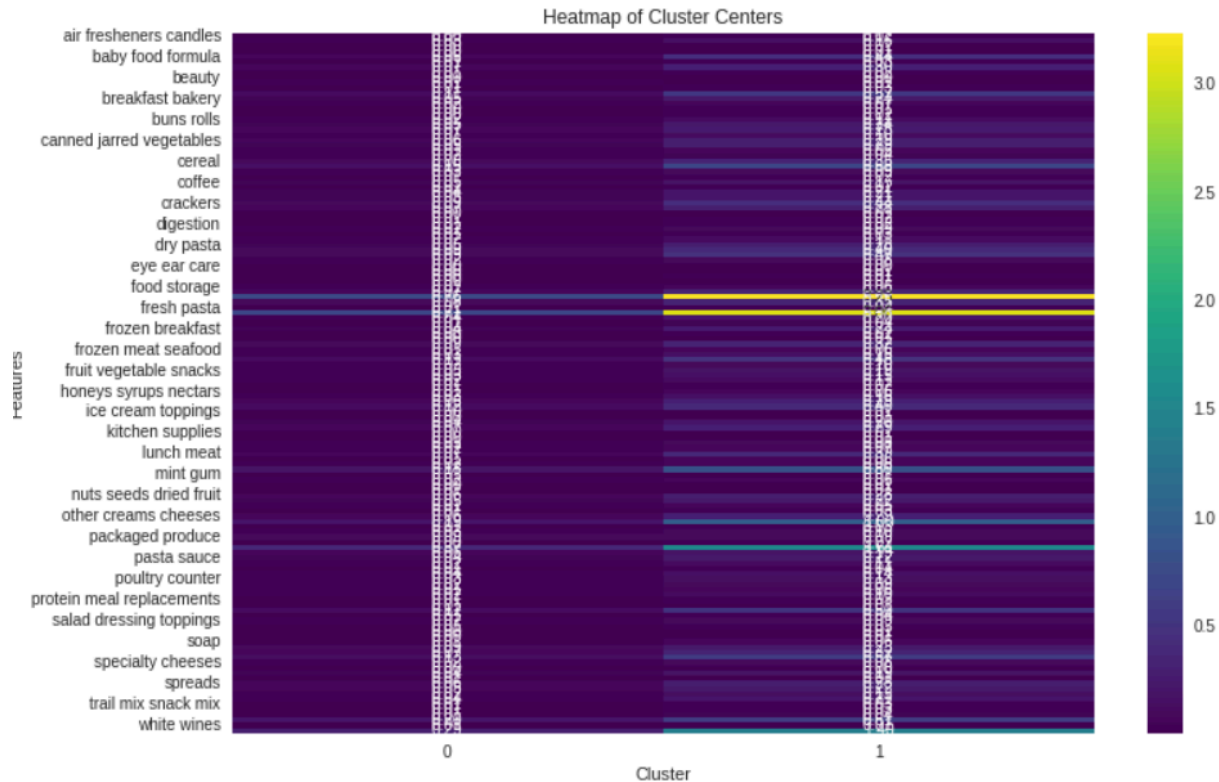
- **Cluster 0 (Occasional Buyers):** Users in cluster with lower purchase count and lower cart positions are on the lower left position. These users can be considered restricted buyers and may only purchase when necessary. These users can be targeted with discounts and newly arrived items.
- **Cluster 1 (Frequent Shoppers):** Users with high purchase count and high average cart position, located in the top and right side of the plot. These users can be considered impulsive buyers and can be targeted with promotions, and personalized products.



## When to Use Association Rule Mining

Association Rule Mining is especially useful when trying to find patterns in large datasets where traditional methods may not be as effective. Some common applications include:

- **Market Basket Analysis:** Discovering relationships between products frequently bought together (e.g., if a customer buys bread, they are likely to buy butter as well). This is used in retail for product placement and recommendation systems.
- **Recommender Systems:** For suggesting items (e.g., books, movies, products) based on user behavior and past preferences.



## Cluster Center Analysis

```
cluster_centers = pd.DataFrame(scaler.inverse_transform(kmeans.cluster_centers_), columns=clustered_customer_product_matrix.columns[1:-3])
```

cluster\_centers

product_name	air fresheners candles	asian foods	accessories	baby bath body care	baby food formula	bakery desserts	baking ingredients	baking supplies decor	beauty	beers coolers	...	spreads	tea	tofu meat alternatives	tortillas flat bread	trail mix snack mix	trash bags liners	vitamins supplements	water seltzer sparkling water	white wines	yogurt
0	0.004827	0.032947	0.001276	0.001391	0.062205	0.008205	0.066836	0.005022	0.001402	0.014801	...	0.057102	0.054276	0.023983	0.035268	0.008044	0.007608	0.009653	0.201174	0.008814	0.288121
1	0.015822	0.157807	0.008328	0.009091	0.442887	0.027481	0.297641	0.018806	0.004164	0.011659	...	0.278695	0.218876	0.131090	0.206315	0.029146	0.026856	0.044067	0.605552	0.012789	1.396669

2 rows x 134 columns

In the above code output, we can see the results of our cluster analysis. In this, each row represents a cluster and each column corresponds to a specific product category. Also, the values indicate the average purchase frequency (using normalised values) for each product category by cluster.

### Cluster 0 Analysis

- Low Values:** The values in this cluster are generally quite low across most product categories, with the highest values being for **baby food formula** (0.062205), **baking ingredients** (0.066836), and **beauty** (0.001402).
- Interpretation:** This cluster likely consists of customers who make relatively infrequent purchases across various categories. The presence of a few categories with slightly higher averages suggests that these users might occasionally purchase specific items, possibly when they need them, rather than as regular consumers. This indicates a more casual shopping behavior.

## Cluster 1 Analysis

- **High Values:** In contrast, the values for Cluster 1 are significantly higher, especially for **baby food formula** (0.442887), **baking ingredients** (0.297641), and **water seltzer sparkling water** (1.396669).
- **Interpretation:** This cluster is indicative of a more engaged customer base who regularly purchases a wide variety of products. The notably high value for **water seltzer sparkling water** suggests that this is a popular item among these users, potentially indicating a health-conscious or lifestyle-oriented demographic that prefers beverages over sugary options.

## Key Comparisons

- **Product Preferences:** The stark difference between clusters reveals distinct consumer segments. Cluster 1 is likely composed of more frequent shoppers or households with higher consumption needs (e.g., families buying baby food regularly), while Cluster 0 may consist of occasional shoppers or less frequent purchasers.
- **Shopping Behavior:** The shopping behavior of users in Cluster 1 suggests they are likely to buy in larger quantities or are more brand-loyal, leading to repeated purchases in certain categories, especially essential items like baby food and beverages.

## Implications for Marketing and Recommendations

### 1. Targeted Marketing Strategies:

- **Cluster 0:** Since these users purchase infrequently, marketing efforts could focus on creating awareness and promoting products that align with their occasional needs. We recommend targeted emails with promotions on specific items they've previously purchased could encourage more frequent shopping.
- **Cluster 1:** For this cluster, loyalty programs, bulk purchase discounts, or subscription services for high-frequency items (like baby food and beverages) could increase customer retention and satisfaction.

### 2. Product Recommendations:

- **Cluster 0:** Recommendations could focus on essentials that they might need but haven't purchased frequently, perhaps suggesting popular items within their interest range or seasonal products.

- **Cluster 1:** Since these users show a clear preference for certain categories, personalized recommendations based on their purchasing history could encourage them to try new products within the same category.

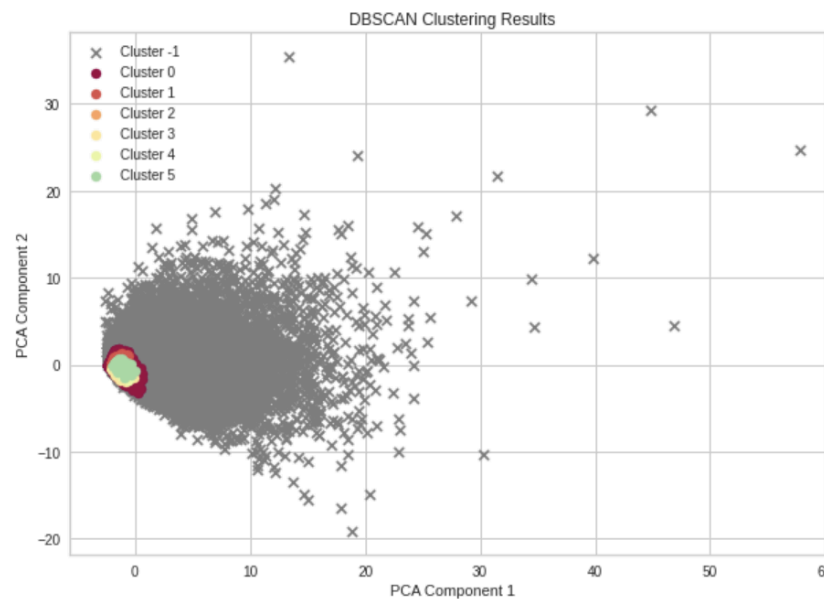
## DBSCAN

To ensure that k means was the best model to apply to the data, we chose to compare its output with an algorithm that is often used for cluster analysis : DBSCAN.

As with k means, we needed to reduce the data set as the amount of RAM required to run the algorithm exceeded our local capabilities.

We initially ran with an epsilon of 0.5 and a random sample of 10 which generated 320 clusters. However, this was not a realistic outcome. so we adjusted the epsilon value to 3.5 and random samples to 250 which generated 5 clusters.

While 5 clusters seem to be a better or more useful number than 2, the silhouette score was quite poor with an output 0.047



Thus, when comparing the two algorithms, it was clear that k-means was the better algorithm for this data set for our purposes - i.e. to generate distinctive clusters. This was demonstrated both graphically and using silhouette score values.



## Association Rule Mining

Association Rule Mining is a technique used in data mining to discover interesting relationships, patterns, or associations among a set of items in large datasets. The main goal is to identify rules that explain how certain items or features are related within transactions or data records. It is widely used in scenarios where we want to understand frequent co-occurrences or affinities between items or events.

### Key Components

1. **Support** – Measures how frequently a particular item or itemset appears in the dataset. Higher support indicates a commonly occurring itemset.
2. **Confidence** – Reflects the likelihood of an item (consequence) appearing in transactions containing a certain item (antecedent).
3. **Lift** – Shows how much the presence of one item increases the likelihood of another item appearing, compared to if they were independent.

Using association rule mining, we found items that were bought together. We could then use these findings to improve our recommendation system to recommend items that went together for other shoppers in the cluster.

So for example in this case when a shopper from Cluster 0 buys baby food formula, they also buy fresh fruits, so we can recommend fresh fruits to be added to the cart and so on.

Association Rules for Cluster 0:

antecedents	consequents	support	confidence	lift
baby food formula	fresh fruits	0.022829	0.735471	1.127315
baby food formula	fresh vegetables	0.016808	0.541483	0.961938
baby food formula	packaged cheese	0.005997	0.193186	1.706388
baby food formula	packaged vegetables fruits	0.008559	0.275752	1.292172
baby food formula	yogurt	0.007664	0.246894	1.720274

Association Rules for Cluster 1:

antecedents	consequents	support	confidence	lift
bread	milk	0.006566	0.106949	1.098158
bread	packaged cheese	0.007257	0.118207	1.250789

bread	packaged vegetables fruits	0.008305	0.135287	0.972958
bread	yogurt	0.007912	0.128882	1.052540
cereal	milk	0.005183	0.121474	1.247309

## Model Deployment

The final chosen model will be deployed for real-time recommendations based of user profile.

The model was deployed using Shinyapps.io ([www.shinyapps.io](http://www.shinyapps.io)). Its url may be found here:

[https://aabakarov.shinyapps.io/clustering\\_ecommerce/](https://aabakarov.shinyapps.io/clustering_ecommerce/)


### Select your Profile

User

Michael George - 44755


Predict

### Inspired by your shopping trends




fresh fruits  
\$36<sup>45</sup>  
\$19.99 shipping  
Today by 10:00 PM

Add to Cart




fresh vegetables  
\$32<sup>69</sup>  
\$19.99 shipping  
Today by 10:00 PM

Add to Cart




packaged vegetables fruits  
\$48<sup>86</sup>  
FREE One-Day  
Today by 10:00 PM

Add to Cart




yogurt  
\$34<sup>21</sup>  
\$19.99 shipping  
Today by 10:00 PM

Add to Cart




packaged cheese  
\$27<sup>96</sup>  
FREE One-Day  
Today by 10:00 PM

Add to Cart




water seltzer sparkling water  
\$34<sup>50</sup>  
FREE Shipping  
Today by 10:00 PM

Add to Cart



milk  
\$26<sup>81</sup>  
FREE Shipping  
Today by 10:00 PM

Add to Cart



chips pretzels  
\$39<sup>91</sup>  
FREE One-Day  
Today by 10:00 PM

Add to Cart

Its GitHub repo is public and may be found here:

[https://github.com/ArslanAbakarov/clustering\\_ecommerce](https://github.com/ArslanAbakarov/clustering_ecommerce)

This application is simple to understand and use by grocery store online shoppers.

The link to the video presentation of the document and device may be found [here](#)

## Conclusion

In conclusion, our project aimed to develop a personalized grocery recommendation model and a web application to provide tailored suggestions for distinct clusters of buyers. Through extensive data analysis and model training, we identified key purchasing patterns and behavioral trends, which allowed us to segment customers into meaningful clusters. Leveraging these insights, our recommendation system offers relevant grocery suggestions, enhancing user experience by promoting items aligned with each customer's preferences and needs.

Our project highlights the potential for machine learning to improve the grocery shopping experience, fostering customer satisfaction and encouraging brand loyalty. Moving forward, we anticipate refining our model with additional data and expanding the web applications capabilities to accommodate a broader audience. This project has provided us with a practical understanding of applying ML techniques to real-world business challenges and underscored the value of data-driven personalization in digital retail.