MDS and Cluster Analysis on online customers behavior

Despoina Iapona & Gerard Palomo

2024-12-14

Contents

1 Introduction

The dataset provides insights into user behavior on an online shopping site. It features various types of data, including continuous, binary, and categorical variables. These variables encompass the count of administrative actions (`Administrative`), interactions related to products (`ProductRelated`), and binary markers indicating whether a visit took place on a weekend (`Weekend`) or led to a purchase (`Revenue`). Furthermore, it includes categorical information such as the user's browser (`Browser`), location (`Region`), and type of visitor (`VisitorType`).

1.2 Goals of the analysis

The primary objective of this analysis is to explore the dataset applying **Multidimensional scaling (MDS)** and **Cluster Analysis** techniques. This study aims to identify patterns, clusters and the influence of specific variables on the data structure. The analysis focuses on understanding the when the customers end up making a purchase and if this customers have a specific behavior that can be identified using the techniques mentioned above.

2 Distance Metrics

2.2 Why Are Distance Metrics Important?

Distance metrics are essential to numerous machine learning algorithms, as they influence how data points are categorized or clustered. Selecting the appropriate metric is crucial for achieving accurate and significant outcomes, especially when dealing with different types of data. The choice of metric can significantly impact the results of the MDS or clustering analysis, making it essential to understand the characteristics and requirements of each metric. Depending on the data type, different distance metrics are more suitable for capturing the underlying patterns and relationships within the dataset.

2.3 Distance Metrics for Different Data Types

Continuous Data

For continuous variables, the following metrics are commonly used:

- **Euclidean Distance**:
  - The simplest measure, determining the direct distance between two points in a multi-dimensional environment.
$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$
  - **Use Case:** Works well for normalized continuous data.
- **Manhattan Distance**:
  - Calculates the total of the absolute differences between matching features.
$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$
  - **Use Case:** Appropriate for data with lower dimensions or when linear variations are more significant.

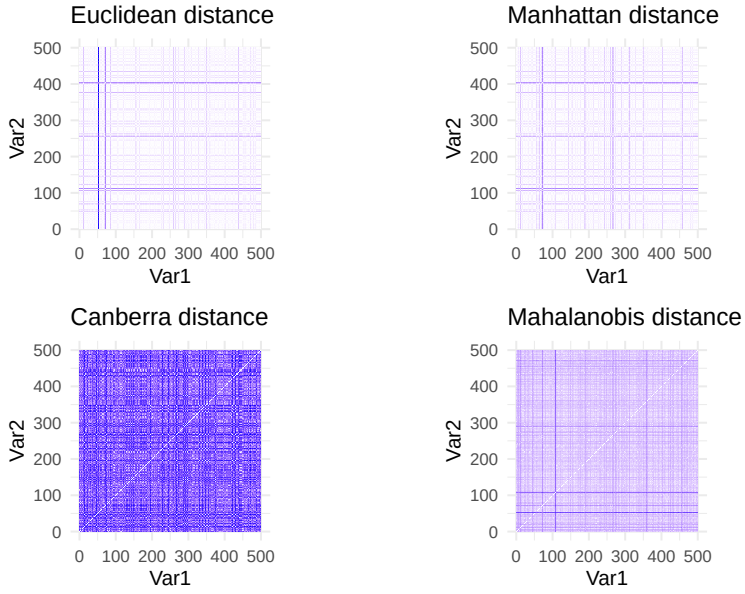
- **Canberra Distance**:
  - Places greater emphasis on minor differences, particularly for values that are low in sizes.

$$d(x, y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

  - **Use Case:** Ideal for datasets with variables of small sizes.
- **Mahalanobis Distance**:
  - Considers the relationships between variables and adjusts distances based on variance.

$$d(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)}$$

    Where S is the covariance matrix.
  - **Use Case:** Ideal for related continuous variables that have different scales.



Binary Data

For binary variables, the following measurements are commonly utilized:

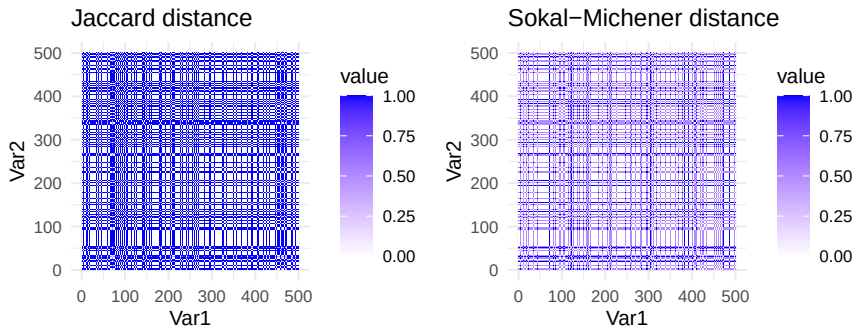
- **Jaccard Distance**:
  - Focuses on mismatches while ignoring shared absences.

$$d(x, y) = 1 - \frac{a}{a + b + c}$$

  - **Use Case:** Appropriate for sparse binary datasets.
- **Sokal-Michener Distance**:
  - Takes into account both matches and mismatches.

$$d(x, y) = 1 - \frac{a + d}{p}$$

  - **Use Case:** Ideal for datasets where matches are as important as mismatches.



Categorical Data

For categorical variables, the following metrics are effective:

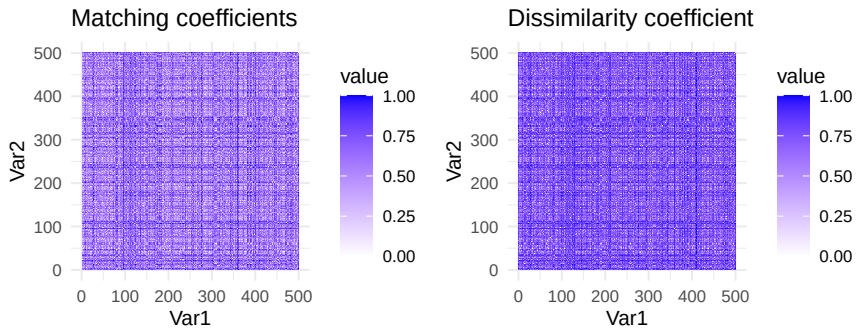
- **Matching Coefficients**:
  - Measure the similarity between categorical variables by counting exact matches.

$$d(x, y) = \frac{\text{Number of Matches}}{\text{Total Observations}}$$

  - **Use Case:** Suitable for datasets where similarity in categories matters.
- **Dissimilarity Coefficients**:
  - Focus on differences between categories.

$$d(x, y) = \frac{\text{Number of Mismatches}}{\text{Total Observations}}$$

  - **Use Case:** Useful when the focus is on categorical diversity.



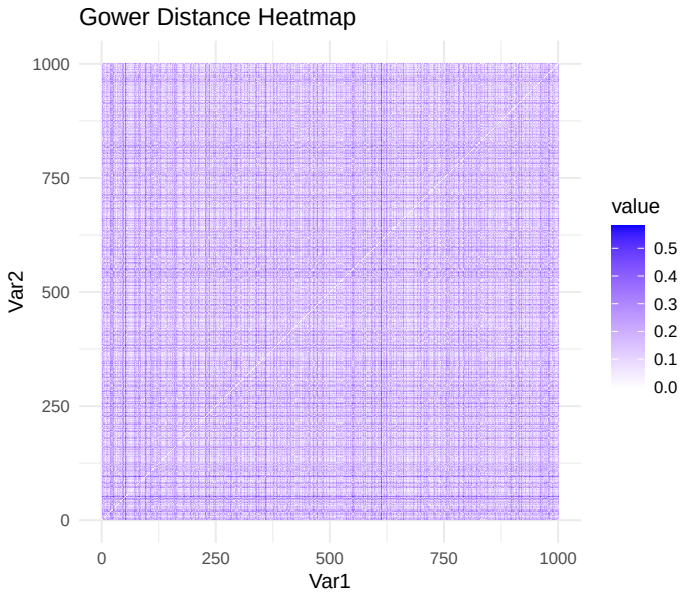
Mixed Data

For datasets with mixed variable types (continuous, binary, and categorical), a flexible metric is required:

- **Gower Distance**:
  - Combines different metrics depending on the data type of each feature.
  - Formula:

$$d(x, y) = 1 - \frac{\sum_{h=1}^{p} s_h(x_h, y_h)}{p}$$

    Where s_h(x_h, y_h) is a similarity score for each variable h.
  - **Use Case:** Ideal for datasets with mixed data types.



2.3 Choice of distance metric

As seen in the plots above, each distance metric provides very different results, that's why it is important to choose the right distance metric for the data at hand. In this case, we needed to choose a mixed type distance metric that could handle the different types of data in the dataset. The Gower distance was chosen as it is a flexible metric that can handle mixed data types, such as continuous, binary, and categorical variables.

There are more robust metrics like the Related Metric Scaling (ReIMS) by Cuadras and Fortiana (1998) or robust implementations of the Gowers distance, however, due to the size of our dataset and the computational complexity of some of this implementations, we decided to use the Gower distance as it is a good compromise between computational complexity and robustness.

3 Multidimensional Scaling (MDS)

3.1 Use of Distance Matrices in MDS

Multidimensional Scaling (MDS) is a technique that reduces dimensions by using distance matrices to show high-dimensional data in a simpler form. Its aim is to keep the distances between pairs of observations as close as possible in the new space.

The steps are:

- 1. Create a distance matrix (D) for all observation pairs using a selected distance method (Gower).
- 2. Adjust the points in the lower-dimensional space so that the distances between them reflect the original distance matrix.

MDS is particularly useful for visualizing patterns, clusters, and relationships in high-dimensional data.

3.2 Multidimensional Scaling (MDS) Using Gower's Distance

Before implementing the MDS technique it is essential to check that the distance matrix fulfills the Euclidean property. This property ensures that the distances between points in the reduced space are consistent with the original data. To check this, we can compute the Gram matrix. If the Gram matrix is positive semi-definite, the distance matrix satisfies the Euclidean property.

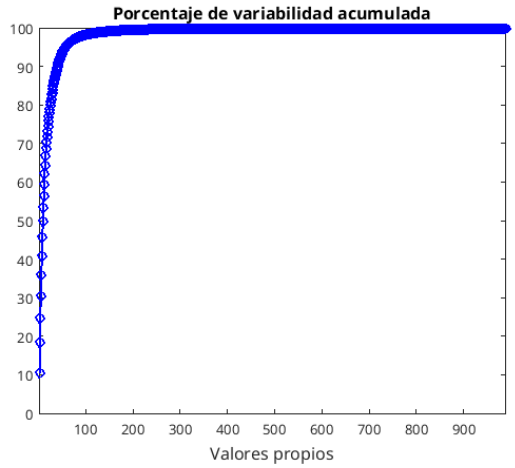
In our case, the distance matrix fulfilled the Euclidean property and the MDS yielded the following results:

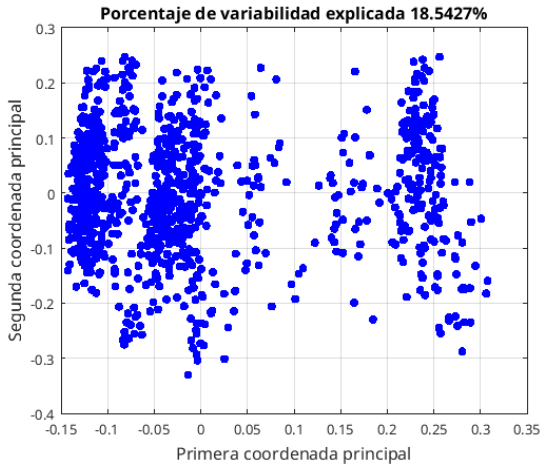
Cumulative Variance Explained

- The cumulative variance plot indicates that the first two dimensions account for 18.54% of the overall variability.
- This suggests that with only 2 dimension we may no be able to distinguish all the patterns in the data.

MDS Configuration

- The scatterplot shows the first two dimensions (PC1 and PC2).
- Clusters and groups suggest that some observations share similar traits.



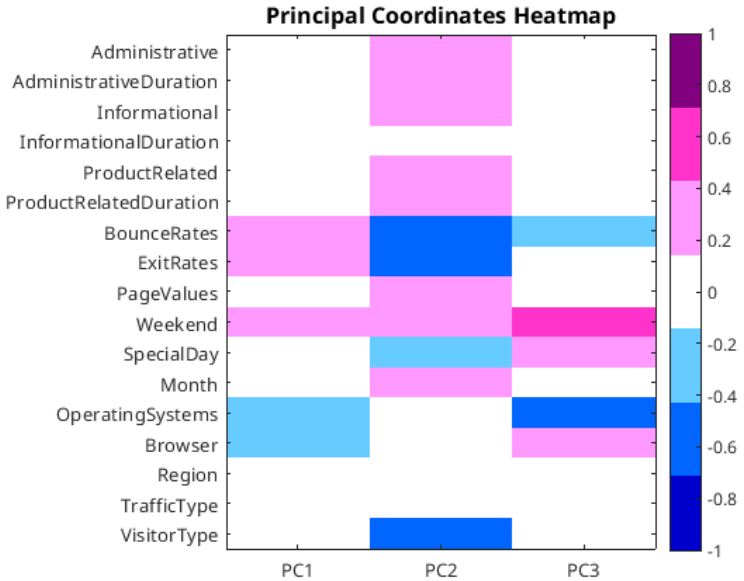


Influence of the Original Variables in the MDS Maps

The influence of the original variables on the MDS configuration is analyzed by computing the **cross-correlations/associations** between the original dataset variables and the first three principal coordinates.

The heatmap visualizes the correlations of each original variable with the first three principal coordinates (PC1, PC2, and PC3).

The heatmap highlights the importance of both quantitative and categorical variables in shaping the structure of the MDS maps. Quantitative variables, such as BounceRates and ExitRates, exhibit strong correlations with the first and second principal coordinates (PC1 and PC2), indicating their substantial contribution to the observed variability in the dataset. At the same time, categorical variables like VisitorType show a strong association with the second principal coordinate (PC2), OperatingSystems and Browser with the first principal component (PC1), and Weekend with both PC1 and PC2.



3.3 Variable Partial Influence on the Principal Coordinates

The study of the influence of the original variables on the MDS map was performed using Gower's interpolation formula. This method, allows us to assess how much each variable influences the principal components. The results for qualitative and quantitative variables are presented separately.

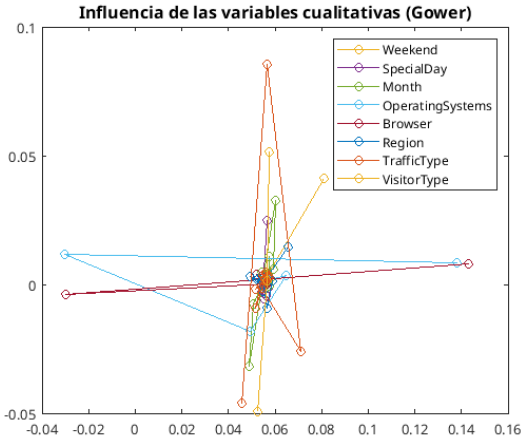
Influence of Qualitative Variables

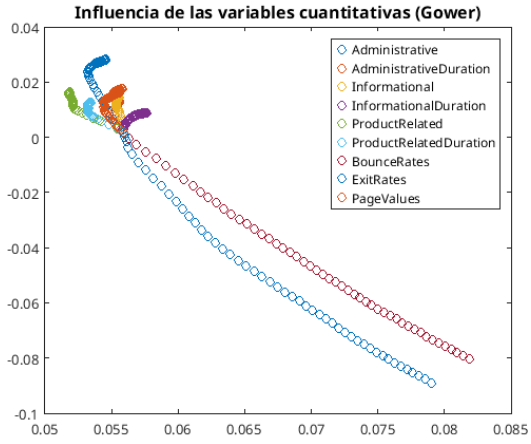
The analysis of the qualitative variables is shown in the plot below on the left. It can be observed that the first principal component (PC1) is mainly influenced by the variable TrafficType and in lesser extent by the VisitorType.

The second principal component (PC2) is most influenced by the OperatingSystems and Browser variables. These variables specify technical details, such as the operating system or browser used by users. Their contribution to the second axis shows that technical preferences play a secondary but important role in differentiating observations.

Influence of Quantitative Variables

The influence of the quantitative variables is shown by the plot on the right. By far, the BounceRates and ExitRates variables have the most significant impact on the first and second principal components (PC1 and PC2). This result indicates that the bounce and exit rates are crucial factors in shaping the structure of the MDS map.



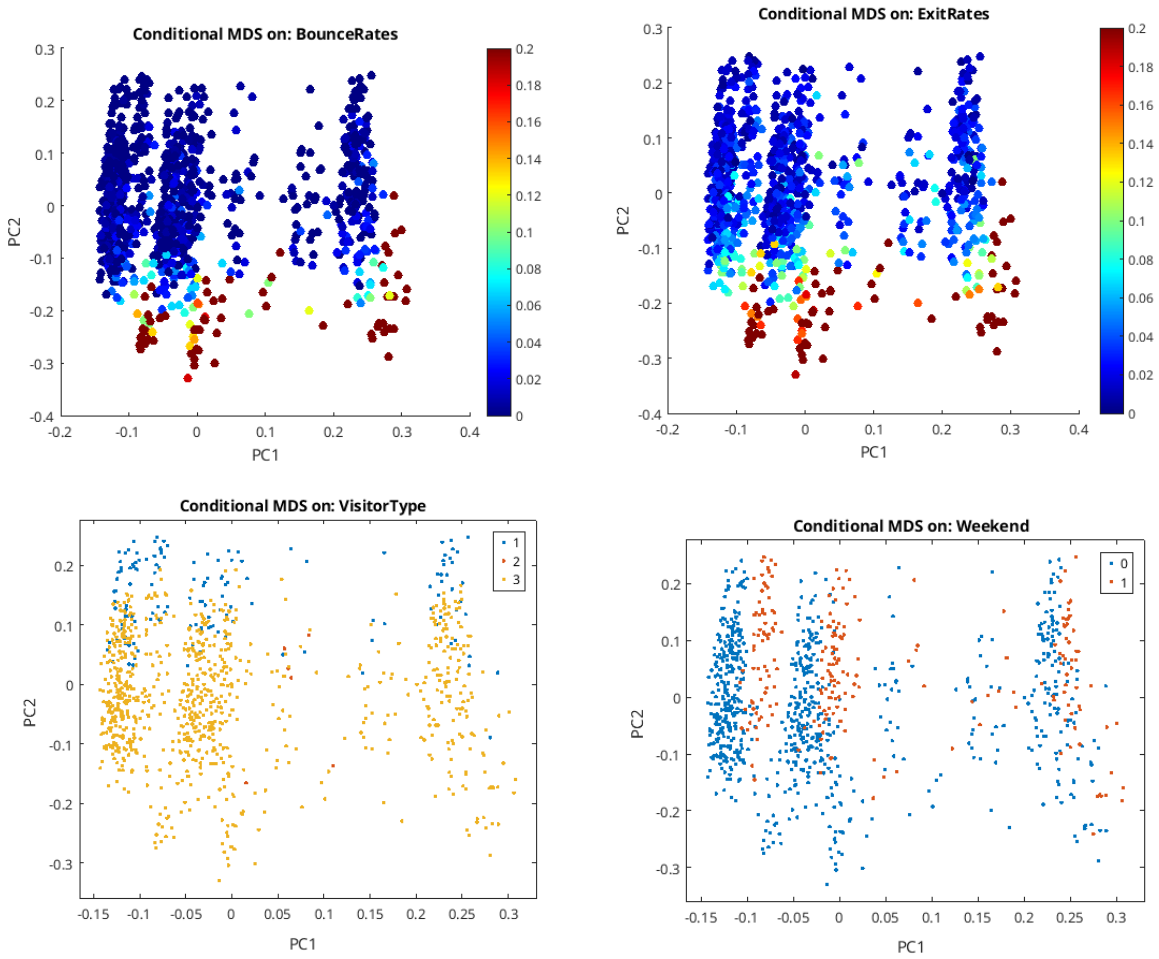


3.4 Conditional Scatterplots for PC1 and PC2

We can also map the original observations in the MDS configuraion, colored by some of the most correlated variables and check if we can observe the patterns that the correlation heatmap suggested previously.

The results of the scatterplots show that the BounceRates and ExitRates variables show a strong correlation specifically with PC2. We can observe that the highes values are at the bottom and the lowest ones on top, which is consistent with the correlation heatmap showing a negative correlation with both. Similarly, the VisitorType variable is mainly associated with the second component (PC2), and category number 1 is mainly on top while 2 and 3 are down below.

Weekend binary variable is mainly correlated with the first component (PC1), and we can see 3 different groups in the scatterplot, where each of them is clearly separated by the weekend variable.

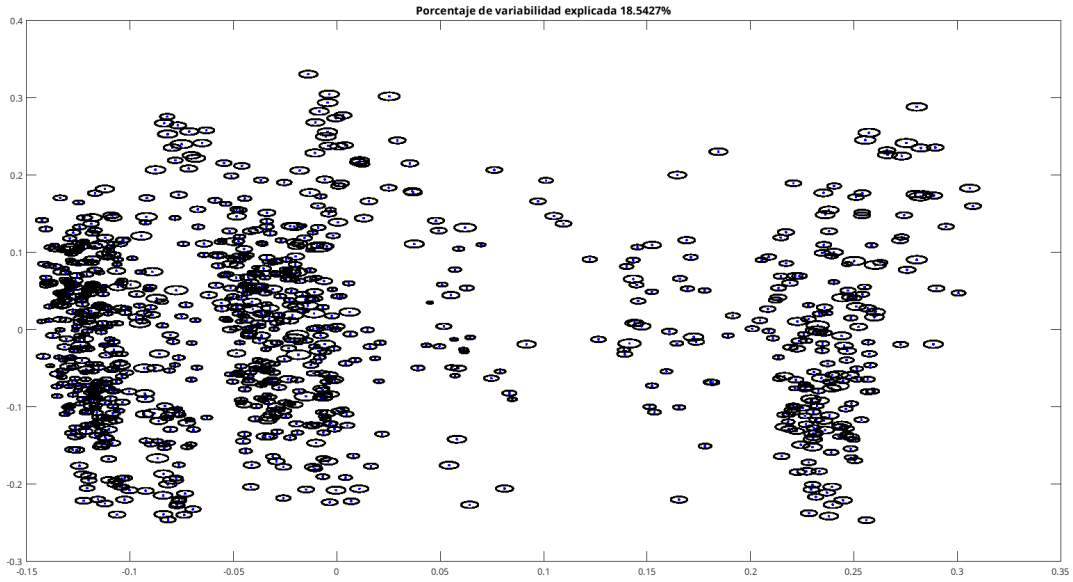


Sensitivity of the MDS Configuration

The jackknife technique evaluates the stability of an MDS configuration by systematically excluding one observation at a time and recalculating the MDS. The variability of each point's position across all jackknife iterations is analyzed to assess how sensitive the configuration is to individual data points. Visualizing these variations highlights stable points (minimal change) and unstable points (large deviations), providing insights into the robustness of the MDS solution.

In the graph below, the scatter of points shows the variability of their positions in the MDS array. The size of the points represents the variability of each observation across the jackknife iterations. The wider the spread, the more sensitive the point is to the exclusion of other observations.

We observe areas with higher variability but in general the points are stable, which indicates that the MDS configuration is robust and not highly sensitive to individual observations.



4 Cluster analysis

Clustering is an unsupervised learning technique used to group data based on similarities. It does not require prior knowledge of the number of groups and is useful for discovering patterns, understanding relationships, and summarizing data. In this section, we will apply several clustering methods, including hierarchical and non hierarchical clustering, to identify meaningful clusters in the dataset.

- **Hierarchical Clustering:**
  This method builds a hierarchy of clusters, either by starting with individual data points and merging them (agglomerative) or by starting with one large group and splitting it (divisive). Different linkage criteria (single, complete, and average) were used to calculate the distance between clusters, with results shown in dendrograms.
- **K-Medoids Clustering:**
  K-medoids is similar to k-means, but it uses actual data points as the center of each cluster, making it more flexible with different distance measures. The PAM (Partitioning Around Medoids) algorithm was used to find the best clusters.
- **K-Prototypes Clustering:** K-prototypes is an extension of k-means that can handle mixed data types (categorical and numerical). It combines k-means for numerical data and k-modes for categorical data to find the best clusters.

The selection of this clustering methods was based on the data types present in the dataset. Since the dataset contains a mix of continuous, binary, and categorical variables, we chose clustering methods that can handle these different data types effectively. K-means, for example, is not suitable for categorical data, so we opted for K-prototypes instead.

4.1 Hierarchical Clustering

The results of the hierarchical clustering analysis are presented in the form of dendrograms, which visually display the relationships between data points and clusters. The height of the branches in the dendrogram

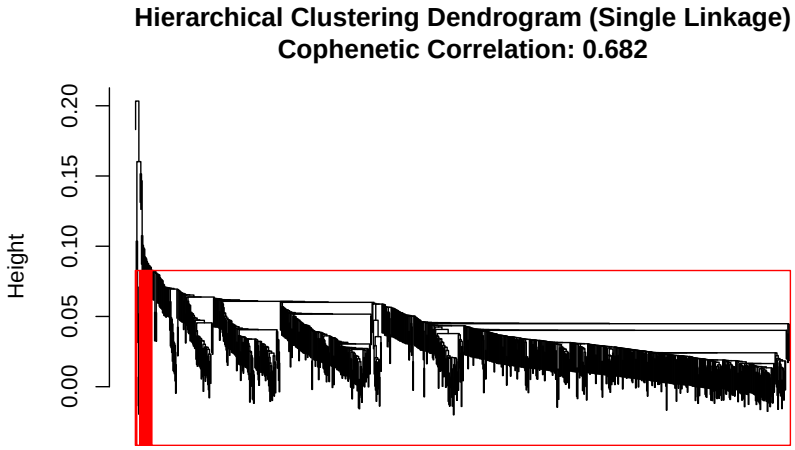
represents the dissimilarity between clusters or data points. By cutting the dendrogram at a certain height, we can identify the number of clusters that best represent the data.

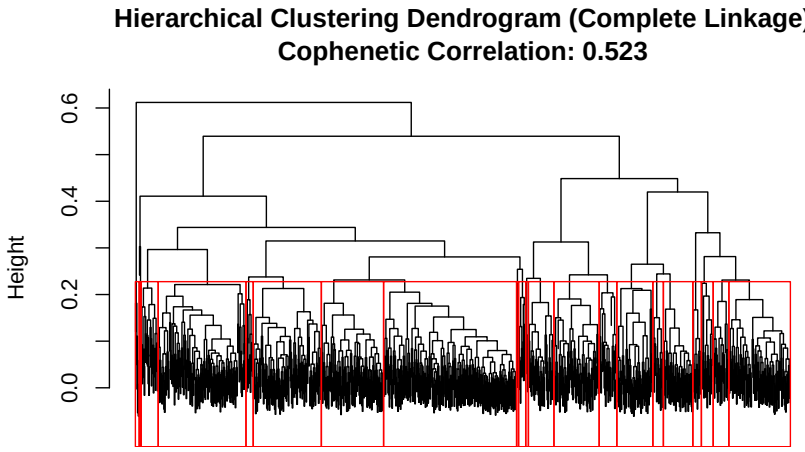
There are different linkage methods used to calculate the distance between clusters, including: - **Single Linkage:** The distance between two clusters is defined as the shortest distance between any two points in the two clusters. - **Complete Linkage:** The distance between two clusters is defined as the longest distance between any two points in the two clusters. - **Average Linkage:** The distance between two clusters is defined as the average distance between all pairs of points in the two clusters.

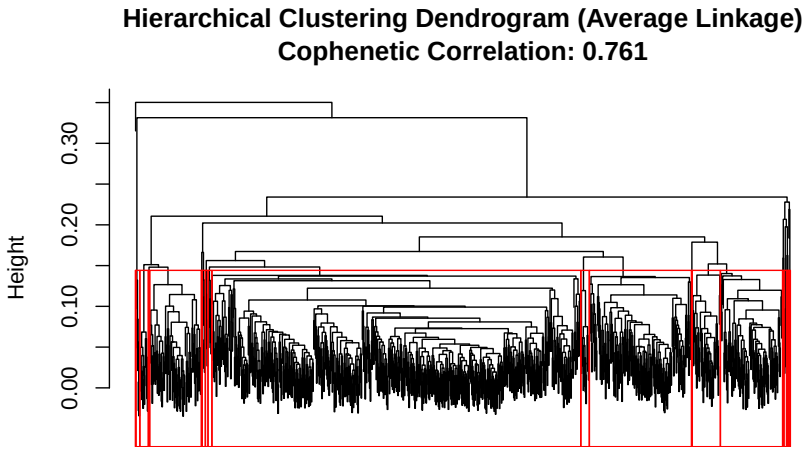
The choice of linkage method can significantly impact the clustering results, as it determines how the dissimilarity between clusters is calculated. The way to choose the best linkage method is to check the cophenetic correlation coefficient, which measures how well the dendrogram preserves the original pairwise distances between data points.

In order to decide the number of clusters, we've used a rule of thumb introduced by Mardia et al. (1989) which states that the number of clusters should be the square root of the number of observations divided by 2.

Clustering also requires the selection of a distance metric to calculate the dissimilarity between data points. In this analysis, we used the Gower distance, which is suitable for mixed data types and can handle continuous, binary, and categorical variables. The choice of this distance metric is explained in more detail in the Multidimensional Scaling section.







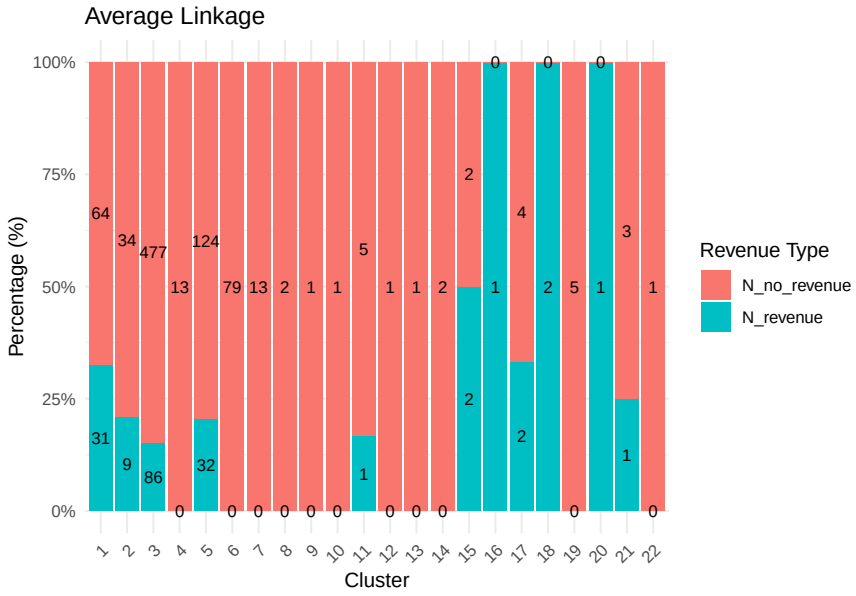
As we can see from the dendograms, each linkage function yields really different results. The single linkage method has created one cluster that includes almost all the data points, leaving all the other clusters with very few observations. The complete linkage function has created the most balanced clusters, with a more even distribution of data points. The average linkage method has created around 5 big clusters while the rest of the clusters have very few observations.

In order to choose the best linkage function we can check the cophenetic correlation coefficient. The cophenetic correlation coefficient measures how well the dendrogram preserves the original pairwise distances between data points. A higher cophenetic correlation coefficient indicates that the dendrogram is a good representation of the original data.

In our case, the highest cophenetic correlation coefficient is achieved by the average linkage method (0.775), which indicates that the dendrogram preserves the original pairwise distances better than the other methods.

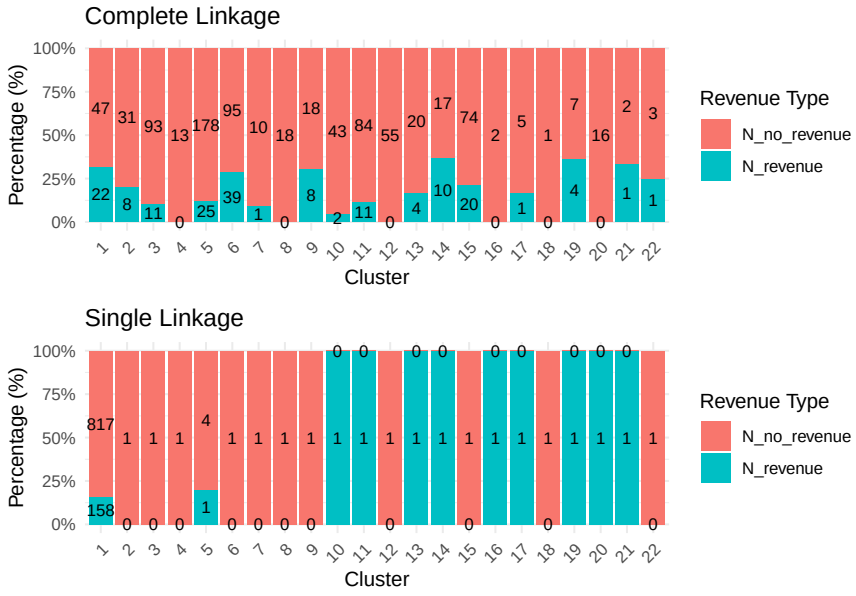
Revenue Analysis

An interesting analysis we can perform is to check how the clusters differentiate between customers that have made a purchase (revenue = 1) and those that have not (revenue = 0). We can summarize the proportion of revenue and no revenue customers in each cluster and visualize the results in a bar plot.



In the plot above we can see that this clustering method does not differentiate well between revenue and no revenue customers. The clusters have a mix of both types of customers, with no clear separation between them. The clusters that seem to be pure have only very few observations.

As an exercise we can try to use other linkage metrics to check if the resulting clusters differentiate more between the revenue and no revenue customers.



The rest of the linkage methods yield similar unsatisfactory results. The clusters that contain a significant amount of observations have a mix of revenue and no revenue customers, with no clear separation between the two. The clusters that seem to be pure have very few observations.

4.2 Non-Hierarchical Clustering

Non-Hierarchical Clustering also called partitioning, divides data into a predetermined number of clusters (k) without creating a hierarchical structure. These techniques focus on finding the optimal partition of data based on a specific objective function, such as minimizing within-cluster distances or maximizing between-cluster distances. Unlike hierarchical methods, non-hierarchical methods don't generate a dendrogram.

In order to choose the right amount of clusters, we can use 2 different techniques: the silhouette method and the elbow method. The silhouette method measures how similar an object is to its own cluster compared to other clusters. The silhouette width ranges from -1 to 1, with higher values indicating better clustering. The elbow method, on the other hand, looks at the within-cluster sum of squares (WCSS) and identifies the point where the rate of decrease slows down, indicating the optimal number of clusters. This techniques are implemented by running the clustering algorithm for a range of k values and then plotting the silhouette width and WCSS for each k.

There are several non-hierarchical clustering algorithms, including K-means, K-medoids, and K-prototypes. The choice of algorithm depends on the data types present in the dataset. K-means is suitable for continuous data, while K-medoids and K-prototypes can handle mixed data types, including categorical variables. As our dataset contains mixed data types, we will use K-medoids and K-prototypes for the non-hierarchical clustering analysis.

K-prototypes Clustering

The **k-prototypes algorithm** is a clustering method designed for mixed-type data, handling both numerical and categorical variables. It combines dissimilarity measures:

- **Squared Euclidean distance** for numerical variables.
- **Hamming distance** for categorical variables.

Prototypes, which represent clusters, are calculated as:

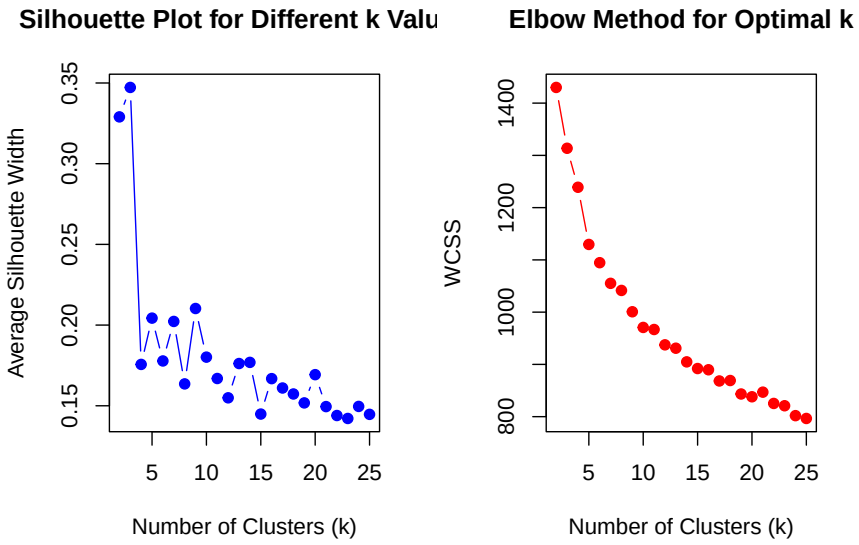
- **Mean/Median** for numerical variables.
- **Mode** for categorical variables.

The algorithm minimizes the within-cluster sum of squares and consists of two main steps:

1. Assigning data points to the nearest prototype.
2. Updating prototypes based on the assigned data points.

The number of clusters (**k**) must be defined before running the algorithm.

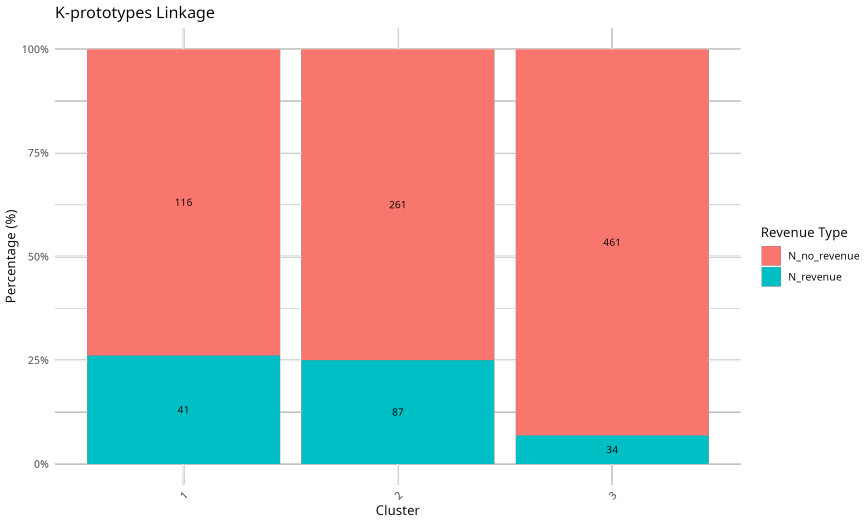
In order to implement this method, we will use the `kproto` function from the `clustMixType` package with the option `type = "gower"` to handle mixed data types, developed by Grané and Sow-Barry (2021).




```
#> Optimal k based on Silhouette plot: 3
```

The siluette plot shows that the optimal number of clusters is 3. This is the number of clusters that maximizes the average silhouette width. The elbow method suggests that the optimal number of clusters is a bit higher but no clear elbow is observed. We will use the optimal number of clusters based on the silhouette plot to implement the k-prototypes algorithm, as the jump in the silhouette width is more pronounced after 3 clusters.

We can also analyze the distribution of the binary variable "Revenue" within each cluster to understand how the clusters differ in terms of the target variable.

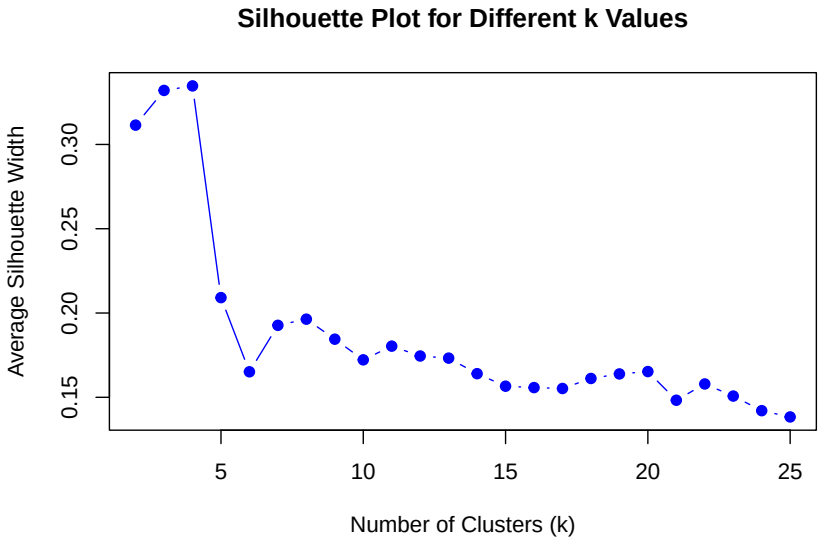


The results of the clustering analysis show that the clusters have a mix of revenue and no revenue customers, with no clear separation between the two. We could only argue that customers classified in cluster 1 and 2 may have slighlty higher revenue rates than the ones from cluster 3.

K-medoids Clustering

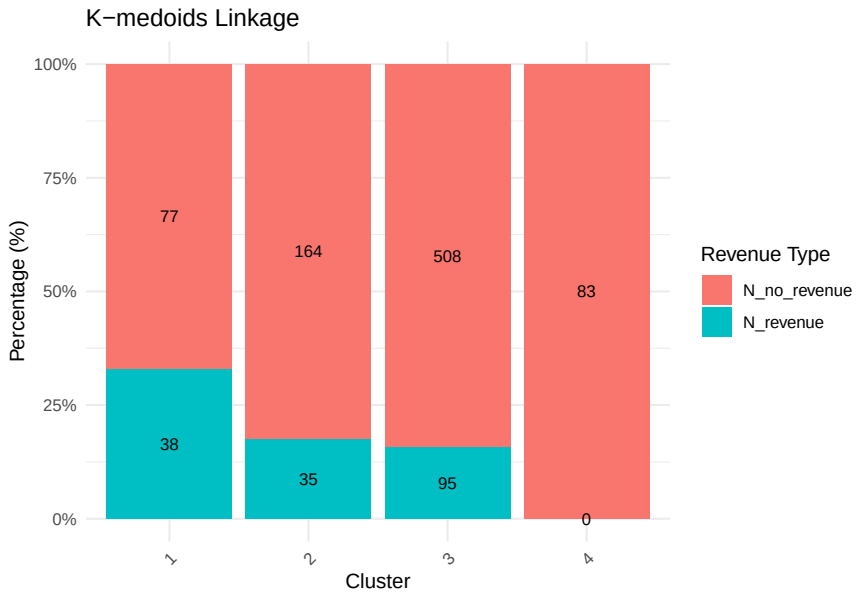
K-medoids is a clustering algorithm that partitions a dataset into k groups using actual data points as cluster centers (medoids). It iteratively assigns data points to their closest medoid and updates the medoids to minimize the total dissimilarity within clusters. The Partitioning Around Medoids (PAM) algorithm is a popular implementation of K-medoids that is more robust to noise and outliers than K-means.

We will use the `pam` function from the `cluster` package to perform K-medoids clustering on our dataset, also based on our Gower distance metric.



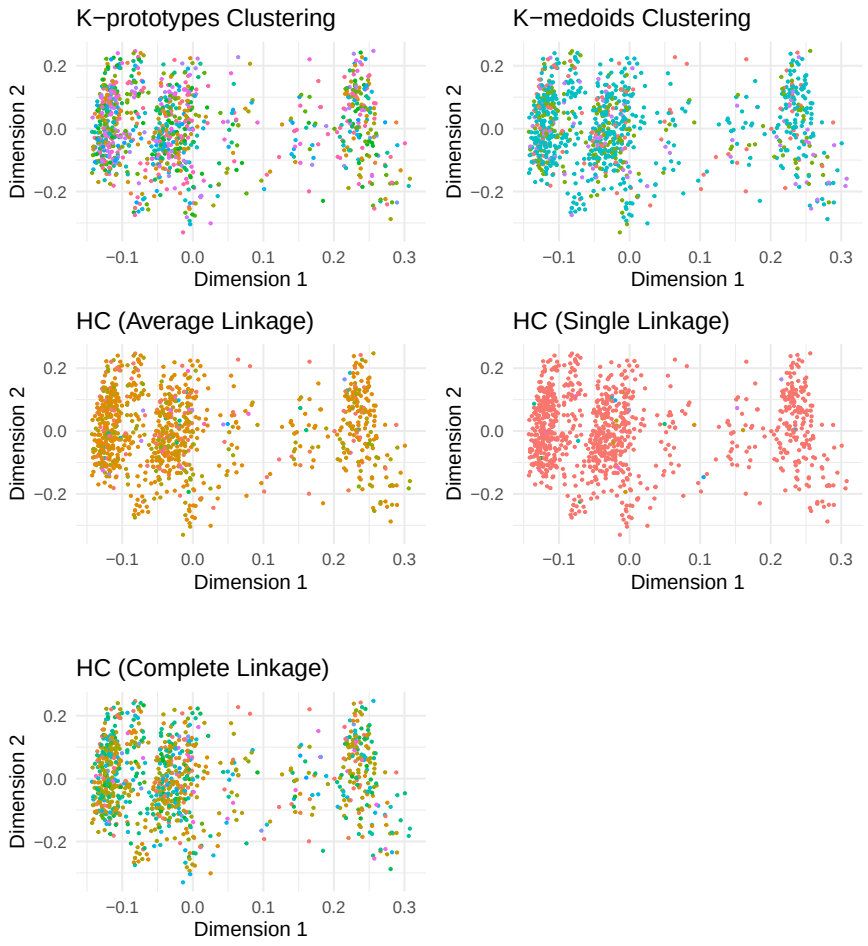

```
#> Optimal k based on Silhouette plot: 4
```

The siluette plot shows that the optimal number of clusters is 4. This is the number of clusters that maximizes the average silhouette width. We now implement the pam algorithm with 4 clusters.



4.2.1 Visualization of clusters in 2 dimensions using MDS

By using the Multidimensional Scaling (MDS) technique applied in the previous section, we can try to visualize the clusters in a 2-dimensional space. This will help us understand the relationships between the clusters and how they are separated in the low-dimensional space.



The visualization does not show a clear separation between the clusters, which can be due to the low variability explained by the first two dimensions of the MDS(~18%). If we could be able to visualize the data in a higher dimensional space, we could potentially see a better separation between the clusters.

References

UCI Machine Learning Repository. (n.d.). *Online Shoppers Purchasing Intention Dataset*. Retrieved from https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset.

Grané, A. (2024). *Multivariate Analysis: 3. Distances and Joint metrics* . Master in Statistics for Data Science, Universidad Carlos III de Madrid. Retrieved from aurea.grane@uc3m.es. Licensed under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0.

Grané, A. (2024). *Multivariate Analysis: 4. Multidimensional Scaling (MDS)*. Master in Statistics for Data Science, Universidad Carlos III de Madrid. Retrieved from aurea.grane@uc3m.es. Licensed under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0.

Grané, A. (2024). *Multivariate Analysis: 5. Cluster Analysis*. Master in Statistics for Data Science, Universidad Carlos III de Madrid. Retrieved from aurea.grane@uc3m.es. Licensed under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0.

Yihui, X. (2018). *R Markdown Cookbook*. Retrieved from https://bookdown.org/yihui/rmarkdown-cookbook/.

R Graph Gallery. (n.d.). *R Graph Gallery: A collection of graphs made with R*. Retrieved from https://www.r-graph-gallery.com/.

Mardia, K.V., Kent, J.T., & Taylor, C.C. (2024). *Multivariate Analysis*. Wiley Series in Probability and Statistics. Wiley. ISBN: 9781118738023. https://books.google.es/books?id=Mw0LEQAAQBAJ.x º