Data Mining Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

XYZ Sports Company - Customer Segmentation

Group 92

Alexandre Spagnol, number: 20230434

Filipe Rodrigues, number: m20201866

Hugo Alves, number: 20230438

January, 2024

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

INDEX

INDEX OF FIGURES

INDEX OF TABLES

1. Introduction

Customers are the core of any business. They are the ones who buy the product or service, evaluate it, and, in the end, the best promoters it can have. Thus, understanding how to acquire clients should be a key component in the development of any business strategy, and that starts by getting to know the current ones. Market segmentation emerges as a vital concept in this context. It involves aggregating customers with characteristics that are more or less homogeneous among themselves, and from that go on to develop different marketing strategies - which aim to attract and retain the clients - tailored to each of the identified groups (Tarver, 2023). Data Mining can be viewed as a tool for performing market segmentation, mainly through the form of clustering - a subcategory of grouping algorithms that refers to a collection of patterns based on similarity, where those in a cluster are more similar than they are to a different pattern belonging to a different cluster (Jain et al., 1999).

This project will focus on the application of these concepts to the sports center XYZ Sports Company, where multiple activities are available for its members. The company is interested in understanding and merging the demographic characteristics of their customers and the value they bring to the business, as well as the activities they choose to participate in. Over the following sections of this report, we will utilize a dataset spanning over five years (between the 1st of June 2014 and the 31st of October 2019), consisting of current and previous users of the facility to identify clusters of similar clients, which will then serve as a base for the development of more specific data-driven marketing strategies desired to increase customer attraction, loyalty, and, ultimately, the revenue of XYZ Sports Company.

2. Data Exploration

The initial approach for every data mining process is data exploration, which provides basic insights regarding the nature of the data. Through data exploration and characterization, we will have access to detailed information about the data structure and its distribution, as well as relationships between variables. With the help of tools like descriptive statistics, we can identify problems and, consequently, the respective techniques to solve them, so as to facilitate the analysis of the dataset on the following steps (ScienceDirect, 2023).

We started by collecting information about the dataset's dimensions (14942 instances and 30 features), the existing variable types, missing values (and its predominance - **figure 1**), duplicated rows and unique values for each variable. As previously stated, we also conducted descriptive statistics which proved to be very insightful, since we identified a minimum *Age* of 0 years, possible outliers in the *Income*, *DaysWithoutFrequency* and *NumberOfFrequencies* variables, as well as a lack of value diversity in both *Nature* and *DanceActivities* (both had only values of 0 for all the entries). We also identified 2 422 cases where customers had an *EnrollmentStart* date equal to the *EnrollmentFinish* date, which was curious and worthy of handling at a future stage. Furthermore, we performed a preliminary correlation testing for the metric features (**figure 2**), in addition to several other explorations carried through **heat maps** (in the search of the most prevalent weekday for *EnrollmentStart*, for instance), **scatter plots** (intervariable relationship exploration), **box plots** and **bar charts** (in order to better understand data distribution and value proportions for each variable).

Examples for each visualization can be found in **figures 3-6**. This last part constituted our visual exploratory data analysis (EDA).

3. Data Preprocessing

Preprocessing is fundamental procedure in our task, since it is a step that encompasses dealing with all the missing values, duplicates and any other inconsistencies or noisy data there are. During this stage, the dataset was transformed to enhance its cleanliness for subsequent cluster analysis (ScienceDirect, 2023). Our strategy comprised three key steps: duplicate values deletion, missing values management, and outlier removal.

In the initial phase, a singular duplicate row was identified and promptly removed. Subsequently, missing values were addressed systematically. For the *Age* variable, new columns, *BirthYear* and *YearOfEnrollment*, were introduced to ensure accurate age determination. Inconsistencies, such as enrolling before birth, led to their imputation as missing values, later replaced by the median age of customers under 18. *Age* values of 0 were treated using KNN imputation, following categorical variable handling and MinMaxScaling. Concerning the *Income* variable, missing values for individuals under 16 were set to 0 (since, in Portugal, they can't earn income until that age) and the same was done if they had income within that age group (we assumed the registered income in that case would be coming from one of the parents, so there was no real income coming from children in reality). The remaining missing values were imputed using KNN after addressing categorical variables and scaling. *AllowedWeeklyVisitsBySLA* missing values were partially replaced based on mean values from *AllowedNumberOfVisitsBySLA* groups (this was possible since SLA contracts are relatively standard). Unaccounted values were then addressed through KNN. The *HasReferences* column, whose missing values all had positive entries for *NumberOfReferences*, was straightforwardly handled by replacing the NaN with 1. For other categorical features, missing data were replaced with the mode, as they only constituted around 0.3% of the total data. Finally, and as previously stated, KNN imputation was employed for the remaining missing data, since it's a method that has been considered in the literature to be easy to implement and understand, as well as efficient for numerical variables (Chen & Shao, 2000), after applying MinMaxScaler (we did not use RobustScaler, since it could lead to loss of information, nor StandardScaler, because of its reliance on the existence of data with Gaussian Distribution).

In the final stage, five approaches were considered for outlier removal: manual inspection, Interquartile Range (IQR) method, Isolation Forest (tested for its low complexity computational algorithm to detect outliers in large-scale data, which focuses on anomalies' identification, considering the assumption that those instances are rare and distant from the center of normal clusters, through tree-based isolation (Tan et al., 2022)), and combinations of manual with IQR and manual with Isolation Forest. The combination of manual and IQR methods was selected to filter outliers, as it allowed for the keeping of most of the data when comparing to the other alternatives, without maintaining 100% of the original dataset (which was happening when combining manual filtering with the Isolation Forest). The outliers were stored separately for potential use in future cluster prediction. This comprehensive preprocessing strategy resulted in a cleaner dataset, setting the stage for more

meaningful and accurate cluster analysis. Histograms and boxplots after outlier removal are illustrated in **figures 7-11**.

4. Feature Engineering and Selection + Scaling

Before we tackled this part of the analysis, we had to guarantee that all variables had the correct datatype and that they were properly optimized (e.g. variables with small integer values using datatype "int8").

Once corrections were made, coherence within time-related variables was examined. The variables *LastPeriodStart* and *LastPeriodFinish* were identified to consistently represent the beginning of January and July and the end of June and December, respectively. As they added noise without bringing in valuable information, both variables were dropped. Then, in order to create the new *EnrolledDays* variable, a comprehensive approach was taken involving *EnrollmentStart*, *EnrollmentFinish*, and *DateLastVisit*. Cases where *EnrollmentStart* equaled *EnrollmentFinish* were addressed by considering the *EnrollmentStart* as the effective start date, while using *DateLastVisit* as the most recent date to calculate the enrollment interval. New variables, such as *TotalNumberActivities* (computed as the number of activities a user was participating in), *MonthsEnrolled*, *TotalIncomeDuringEnrollment* and *PercentageIncomeSpent* were created to provide additional insights.

Expendable variables were then removed, including redundant date variables, uninformative univariate variables like *NatureActivities* and *DanceActivities*, features evaluated as non-essential for clustering analysis. Nevertheless, we decided for more testing to identify highly correlated features that could negatively impact our research by introducing redundancy into our analysis, as well as not reducing the number of variables could bring the curse of dimensionality upon our dataset (prohibiting a correct use of clustering algorithms) (Linsen & Molchanov, 2018). Further testing involved Pearson and Spearman correlation tests (**figures 12** and **13**), identifying highly correlated pairs. PCA was applied to assess the impact of removed features on data variability, but results were disregarded in favor of other tests.

Ultimately, two more features were eliminated - *Age* and *NumberOfReferences* - based on correlation tests, resulting in a final set of 26 variables. Despite the dataset's high dimensionality, additional feature removal was avoided, aligning with the planned use for clustering processes. The metric features were scaled using StandardScaler instead of MinMaxScaler, considering algorithm sensitivity (Mohamad & Usman, 2013) and the commonality of standardization in cluster analysis preprocessing (Firmin, 2023).

Finally, having undergone meticulous preprocessing steps, the dataset stood ready for comprehensive analysis.

5. DBSCAN for Outlier Removal

Before proceeding to the clustering task, the final preparation of our data involved the removal of multidimensional outliers with the aid of DBSCAN. By being an algorithm which finds clusters based on the density of the points' neighborhood, we were also able to use it with the purpose of identifying extreme values in our dataset that we couldn't previously recognize - since the methods previously used were only suited for one-dimensional outliers. To that extent, we started by computing the k-distance graphic - fitted solely to our metric features -, which allowed us to identify 3 as the optimal radius (epsilon or "eps") of the neighborhood when using 10 as the minimum number of points belonging to it (including the point itself). Then, we created an instance of the DBSCAN algorithm - again, fitted only to our metric features - and it retrieved, in a separate cluster, the observations that it identified as being outliers. In the end, we removed 245 records, adding to the 84 that had already previously been taken out of our dataset.

6. Definition of the Perspectives

Having defined the range of features that we were interested in, we defined different perspectives with the objective of understanding the distribution of XYZ Sports Company's customers regarding the most crucial aspects for market segmentation, which would then allow us to develop more specific marketing strategies. Taking the main purposes of the project - gaining insights on the value, demographics and activities preferred by the users - into account, we defined three main perspectives and selected the most relevant attributes to integrate each one:

- **Activities Perspective** - containing the *AthleticsActivities*, *WaterActivities*, *FitnessActivities*, *OtherActivities*, *SpecialActivities*, *CombatActivities*, *RacketActivities*, *TeamActivities* and *TotalNumberActivities* features, it aimed to group customers according to the activities they tend to participate in;
- **Value Perspective** - containing the *Income*, *LifetimeValue*, *PercentageIncomeSpent*, *TotalIncomeDuringEnrollment* and *NumberOfRenewals* features, it aimed to group customers according to the amount they spend in the gym, and its relationship with their financial capabilities overall;
- **Time Perspective** - containing the *DaysWithoutFrequency*, *NumberOfFrequencies*, *AttendedClasses*, *RealNumberOfVisits* and *EnrolledDays* features, it aimed to group customers according to the time they spend in XYZ's facilities, and the frequency with what they visit them.

Due to the presence of binary features in the Activities perspective, it required the utilization of algorithms that were suited for these categorical variables. Therefore, we employed k-modes and k-prototypes to define the best number of clusters and find the optimal grouping in what concerns the activities offered by XYZ Sports Company (a theoretical framework of these two algorithms can be found in **sections A** and **B** of the appendix, respectively). For the Value and Time perspectives, we tried using k-means, self-organizing maps (with k-means and hierarchical clustering), mean-shift clustering and DBSCAN and selected, for each perspective, the algorithm which was found to produce the best clusters.

7. Clustering

7.1. K-Modes

7.1.1. Activities Perspective

For the k-modes algorithm, we started by computing the sum of squares within clusters when using one to seven clusters and plotting it to define the optimal number of clusters by utilizing the "elbow" method. The resulting plot (check **figure 14**) suggested that using either two or three clusters would be the more correct approach. Additionally, we calculated and plotted the silhouette scores for each number of clusters within the previously defined interval - this time, the outcome indicated five as the best number of clusters. Considering these inputs, we tested k-modes for two, three, four and five clusters.

Analyzing the outputs of each clustering attempt - both the resulting means for each variable (shown in **figure 15**) and the respective R-Squared scores (visible in **table 1**) - the utilization of four clusters in this perspective emerged as the most suited approach for our objectives. However, we also noticed that, while *Water*, *Fitness* and *CombatActivities* consistently stood out, the remaining activities didn't seem to have a major impact in cluster definition. Therefore, we grouped *Athletics*, *Other*, *Special*, *Racket* and *TeamActivities* into a single binary feature, *OtherActivities*, which returned 1 if the user took at least one of those activities and 0 otherwise. Then, we recalculated our clusters, and both the characterization of each cluster (**figure 16**) and the respective R-Squared results (**table 1**) suggested that this procedure of grouping less relevant activities would lead to better results. Still, using four clusters seemed to be the best option to take.

7.2. K-Prototypes

7.2.1. Activities Perspective

Similarly to what was done for k-modes, we started the utilization of the k-prototypes algorithm by evaluating the "elbow" of the sum of squares within clusters (visible in **figure 17**), which indicated two to four as the ideal number of clusters. Nevertheless, considering the previous insights obtained through k-modes, we also tested for five clusters. Again, identically to what was previously observed, the outputs (check **figure 18** and **table 2**) pointed out four clusters as the best solution, but they also led to the same conclusion regarding the merge of less relevant activities into a single category. After testing for this new combination (the outputs can be seen in **figure 19** and **table 2**), the utilization of this new variable instead of the previous ones led to better conclusions regarding the profile of XYZ's customers. Comparing this solution with the best obtained through the k-modes algorithm, the final choice for the activities perspective lies in the use of k-prototypes, four clusters, and a new grouping of *Athletics*, *Other*, *Special*, *Racket* and *TeamActivities* into one new feature.

7.3. K-Means

7.3.1. Value Perspective

Equivalently to what was done with the other models, we started the utilization of the k-means algorithm by evaluating the "elbow" of the sum of squares within clusters (visible in **figure 20**), where

we were able, with the help of the KneeLocator function, to find that the elbow was at six clusters. To have a second opinion, we also evaluated the silhouette scores between the same range of clusters (which can be seen in **table 3**), where the best values were two, three and six clusters. However, even though two and three had higher scores and were closer to the standard "good" score, after considering the "elbow" method, six was, again, the value of the cluster to choose. Therefore, we decided for the creation of six clusters using k-means.

After running the model, we evaluated it based on the R-Squared and got a score of 0.70, which was accepted as being a high value. By analyzing the cluster profiles, we could see that cluster 4 contained most of the observations, while the rest were relatively evenly distributed (check **figure 21** for the visualization of the cluster profiles).

7.3.2. Time Perspective

Again, the first step was to apply the "elbow" method of the sum of squares within clusters (as shown in **figure 22**), getting an ideal number of six, using also KneeLocator to help us find it. Then, we calculated the silhouette scores for the same range of possible clusters (displayed in **table 4**), where the best values were found when using two to five clusters. Once again, despite two to five clusters having higher scores, considering the "elbow" method, six was concluded to be the optimal number of clusters to choose. Nonetheless, we also tested with five clusters to confirm our decision.

After running the model with five and six clusters, we got R-Squared scores of 0.58 and 0.65, respectively. We also analyzed the distribution of the dataset through the clusters, seeing that cluster 4 contained the majority of the observations (**table 23**).

7.4. Self-Organizing-Maps

Self-organizing maps (SOM) can be difficult to work with small amounts of data, as was the case with our dataset. It is said that, to have good statistical accuracy, 100 000 steps would be required, and we were far off that number - we would need to cycle through all the samples several times (T. Kohonen, 2001). Also, in both our perspectives, the use of SOM would result in a very large number of clusters that are not good on their own and do not provide valuable insights considering the goals of this project. Consequently, based on this, we decided to apply SOM along with k-means and also with hierarchical clustering.

7.5. Self-Organizing Maps and K-Means

7.5.1. Value Perspective

The first step was to apply the SOM algorithm and try to reach convergence on the quantization error. Being a very long and time-consuming step, we decided to run the model until the change of quantization error was very small and, after 1 298 seconds, we reached a quantization error of around 0.2328. We proceeded to plot both the component planes and the U-matrix (visible in **figures 24** and **25**, respectively). Using the previous knowledge from k-means, we applied it again with six clusters, resulting in six well-defined clusters.

After analyzing the cluster profile, most of the samples were found to be in cluster 3, while the rest were evenly distributed along the remaining five clusters (the visual profiling can be seen in **figure 26**) and an R-Squared of 0.67.

7.5.2. Time Perspective

For the time perspective, we followed the same steps, applying SOM and achieving a quantization error of approximately 0.3031 after 1 438 seconds. After plotting both the components planes and the U-matrix (visible in **figures 27** and **28**), we were able to make the clusters using k-means and six clusters as we did previously.

After profiling the cluster, we had, once again, a main cluster where most of the observations were contained, with the rest of them being evenly distributed (**figure 29**).

7.6. Self-Organizing Maps and Hierarchical Clustering

7.6.1. Value Perspective

Using the same SOM instance created for SOM with k-means, we also applied hierarchical clustering. We then applied the “elbow” method of the sum of squares within clusters, for the different types of hierarchical methods, where the “Ward” method along with two or five clusters yielded the best results. We decided that two clusters would be too little, so we proceeded with five clusters. With the help of the dendrogram, we were able to clearly visualize the five clusters created (as seen in **figure 30**).

After profiling, we were able to see that most of the samples belonged to cluster 1, while the rest were fairly distributed along the other clusters.

7.6.2. Time Perspective

Based on what was done previously, we used the same SOM instance formulated for SOM with k-means and applied hierarchical clustering. To decide the ideal number of clusters, we used a combination of the “elbow” method and its dendrogram (visible in **figure 31**), and we decided on the utilization of five clusters.

The end profiling resulted in a main cluster, number 3, containing most of the observations from our dataset, and the rest being evenly split between the remaining four clusters.

7.7. Mean-Shift Clustering

7.7.1. Value Perspective

Another model used for clustering was Mean-Shift clustering. Here we applied different values of bandwidth and, for each, computed the R-Squared and the ideal number of clusters (**table 5**). In the end, the decision was a bandwidth of 1.2 that resulted in an R-Squared of 0.52 and a total number of 12 clusters. Due to the large number of different groups being created, we decided that we wouldn't continue with this model, so no profiling report was made, nor t-SNE plotting.

7.7.2. Time Perspective

Regarding this perspective, the process was similar, as were the conclusions. After trying several values of bandwidth, we decided that 2 was the best value, resulting in eight clusters with an R-Squared of 0.35 (as shown in **table 6**). Again, due to the large amount of clusters suggested, we decided that we wouldn't continue with this model, so no profiling report was made, nor t-SNE plotting.

7.8. DBSCAN

7.8.1. Value Perspective

DBSCAN was used previously for outlier detection and removal, but it can also be used for clustering. As mentioned before, it has two main hyper-parameters, "eps" and "min_samples", where "eps" is "the maximum distance between two samples for one to be considered as in the neighbourhood of the other. This is not a maximum bound on the distances of points within a cluster. This is the most important DBSCAN parameter to choose appropriately for your data set and distance function." (Scikit-Learn). We used a common method of finding the "min_samples", using the double of the number of features, which, in our case, resulted in ten minimum samples. Regarding eps, we "tried" a range between 0.1 and 1 with a step of 0.05 (with the results shown in **table 7**). Taking conclusions from it, we decided on an "eps" of 0.30 and after applying it to our model, it resulted in a R-Squared of 0.62 with a total of 18 clusters. As was the case with mean-shift clustering, due to this large number of clusters, we decided that we wouldn't continue with this model. Therefore, no profile report was made, nor t-SNE plotting.

7.8.2. Time Perspective

For this perspective, we applied a similar logic. We ran the model with an "eps" ranging from 0.1 to 3 with a step of 0.05. By analyzing the outputs (summarized in **table 8**), we decided that an "eps" of 0.30 would be the optimal choice. Additionally, we also tried a range between 1 and 30 for "min_samples", resulting in 13 as being the best value. This would result in an R-Squared of 0.25 and a total of six clusters. After fitting the final model, we got a total of 3 587 new "outliers" that DBSCAN left out. We shared the opinion that we would not use this algorithm since we had already used it for outlier removal and wanted to try other ones. Because of this, no further cluster profiling was made.

8. Perspectives' Merge

After thorough analysis, we understood the best clustering method for the Activities Perspective was the k-prototypes (with four clusters) and, for the Value and Time Perspective, we opted for SOM with k-means (with six clusters) and SOM with hierarchical clustering (with five clusters), respectively. The following step was to decide which approach was better to merge these three perspectives. Initially, we considered using k-means or manual merging, but we decided against both since the first technique didn't make sense to apply when one of the perspectives only had categorical variables and the latter was deemed too complicated and more prone to make mistakes. Consequently, we opted for merging through hierarchical clustering, for it is a good and simple method of understanding relationships

between clusters in a visual way, and also because it's a technique known for its flexibility in terms of applicability for numeric and categorical data (fitting perfectly in our situation) (Datarundown, 2023).

The dendrogram allowed us to understand that the most efficient number of clusters to be formed was four, as seen in **figure 32**. Having more than four clusters would mean that most of the clusters would be extremely small, which would translate to a less significant segmentation and, consequently, less significant analysis - we observed this in the beginning, as we tried six final clusters and found the clusters to be less significant and relevant for analysis.

9. Customer Segmentation and Marketing Approaches

Cluster 0 - This cluster encompasses people who mostly focus on fitness and, to a lesser degree, in water activities, while being the group involved in less activities overall (tied with cluster 3). The customers in this cluster were identified as being the ones who earned and spent less during their enrollment. Additionally, they renewed their membership less frequently in comparison with other clusters, as they also enrolled for fewer days and attended fewer classes. A good approach for this segment of customers could be to create a cheaper plan of attendance that would include only fitness and water activities.

Cluster 1 – This clusters includes the vast majority of the users in our database, and it is characterized by having the most participative customers in terms of number of activities taken, in particular water activities. Although they are not the ones who earned the most, they are one of the groups who spent the most at the facility, having a great number of renewals as well as class attendance and frequency. A good approach for this segment of customers could be to offer a price reduction after a six-month enrollment if they attended a minimum number of classes for a frequency of visits of at least four days per week. If the minimums were not met, the monthly fee would return to normal, with the possibility of having the discount again the following month if the conditions were met.

Cluster 2 - This cluster includes attendees who mostly go to fitness and, to a lesser degree, to water or other activities. These customers have both the higher income and amount of invested money in the facility, as well as the highest attendance and renewals. A good approach for this segment of customers could be to reduce the number of promotion newsletters sent to their mail, but instead the advertisement, both *in loco* and by e-mail, of attractive premium services (for instance, for an extra fee, they could have access to laundry for all their sports clothing).

Cluster 3 - The last cluster has customers who tend to attend a similar proportion of water and fitness activities, even though they do not go to a lot of classes. Like the customers in cluster 1, the people in this segment don't have a lot of income to spend on XYZ Sports Company. Nevertheless, these attendees have a great number of renewals and are frequent visitors. A good approach for this segment of customers could be a mix of strategies between the one defined for cluster 0 and for cluster 1.

Taking into consideration all the approaches discussed in each cluster, it would be interesting to define a general strategy which could be a flexible plan that would allow customers to choose to do just one, two or all activities, with each of these versions having a progressively higher price as the number of

activities included increases. It would also be interesting to allow this 3-modal plan to be changed monthly if the client desired, without the need to sign a new contract (making it as easy as possible for the client to interact with the XYZ Sports Company, and equally easy for XYZ to process these changes), removing permanent restrictions and encouraging "contract renewal".

10. Reclassification of Outliers

The final step of this project was to re-include the 329 outliers previously removed. As we had already removed some before the feature engineering stage, we were forced to replicate the feature engineering steps for the outliers we removed. We converted data types, made a data coherence check, created the new features present on the main dataset, proceeded with feature selection and applied the same scaling instance we had already used to scale data before clustering.

We applied supervised learning techniques to classify these outliers and assign them to their respective clusters. To do this, we opted for a decision tree as our model, with which we were able to correctly predict 93.19% of the customers' clusters. With this, we predicted the values and successfully classified all outliers.

11. Conclusion

Clustering, as we saw, is dependent on and presents different results for each strategy and method employed. Taking that into consideration, we cannot say for sure our outcomes are the best possible, especially when we achieve clusters like cluster 1, which includes over 73% of our customers (without considering outliers) while having segments with 4% of our data (cluster 3). Nevertheless, using t-SNE (**figure 33**) it was possible to identify well-defined clusters after the segmentation merging, translating confidence in the significance of the results that were obtained. We consider that the merging approach proved to be insightful about the types of customers of XYZ Sports Company, but it is important to point out that, in a less generic situation, an isolated analysis of each cluster perspective (in our case, activities, value and time) could be more helpful (and less time-consuming) since we would also have a more specific problem/question to answer. Additionally, the process of feature selection for the clusters could be improved, in the case of a well-established problem, which did not happen in this project because of its exploratory nature. Due to time restrictions, it was not possible to compare differences between using Euclidean and Manhattan distances while performing the perspectives' merge. Nevertheless, for a future approach, we believe that this could be interesting to test.

12. References

Chen, J., & Shao, J. (2000). Nearest neighbor imputation for survey data. *Journal of Official Statistics, 16*(2), 113-131. https://www.google.com/url?q=https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/nearest-neighbor-imputation-for-survey-data.pdf&sa=D&source=docs&ust=1704488928922225&usg=AOvVaw3DqIAYzuTbap3UaKKjQgH5

Firmin, S. (2023). Standardization in Cluster Analysis. *Alterix*. https://knowledge.alteryx.com/index/s/article/Standardization-in-Cluster-Analysis-1583461087248

J, E. (2023). When to Use Hierarchical Clustering: A Guide for Data Analysts. *Datarundown*. https://datarundown.com/hierarchical-clustering/

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys, 31*(3), 264–323. https://doi.org/10.1145/331499.331504

Kohonen, T. (2001). The Basic SOM. *Springer Series in Information Sciences, 30*, 105–176. https://doi.org/10.1007/978-3-642-56927-2_3

Linsen, L. & Molchanov, V. N. (2018). *Overcoming the Curse of Dimensionality When Clustering Multivariate Volume Data*. https://doi.org/10.5220/0006541900290039

Mohamad, I. B., & Usman, D. (2013). Standardization and Its Effects on K-Means Clustering Algorithm. *Research Journal of Applied Sciences, Engineering and Technology, 6*(17), 3299–3303. https://doi.org/10.19026/rjaset.6.3638

ScienceDirect. (2023). *Data Exploration*. https://www.sciencedirect.com/topics/mathematics/data-exploration

ScienceDirect. (2023). *Data Preprocessing*. https://www.sciencedirect.com/topics/engineering/data-preprocessing

Scikit-Learn. (n.d.). *sklearn.cluster.DBSCAN*. Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

Scikit-Learn. (n.d.). *sklearn.preprocessing.StandardScaler*. Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

Tan, X., Yang, J., & Rahardja, S. (2022). Sparse random projection isolation forest for outlier detection. *Pattern Recognition Letters, 163*, 65–73. https://doi.org/10.1016/j.patrec.2022.09.015

Tarver, E. (2023). *Market Segmentation: Definition, Example, Types, Benefits*. Investopedia. https://www-investopedia-com.translate.goog/terms/m/marketsegmentation.asp?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt-PT&_x_tr_pto=sc

13. Annexes

```
Index: 14942 entries, 10000 to 24941
Data columns (total 30 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Age                         14942 non-null  int64
 1   Gender                      14942 non-null  object
 2   Income                      14447 non-null  float64
 3   EnrollmentStart             14942 non-null  datetime64[ns]
 4   EnrollmentFinish            14942 non-null  datetime64[ns]
 5   LastPeriodStart             14942 non-null  datetime64[ns]
 6   LastPeriodFinish            14942 non-null  datetime64[ns]
 7   DateLastVisit               14942 non-null  datetime64[ns]
 8   DaysWithoutFrequency        14942 non-null  int64
 9   LifetimeValue               14942 non-null  float64
 10  UseByTime                   14942 non-null  int64
 11  AthleticsActivities         14906 non-null  float64
 12  WaterActivities             14905 non-null  float64
 13  FitnessActivities           14907 non-null  float64
 14  DanceActivities             14906 non-null  float64
 15  TeamActivities              14907 non-null  float64
 16  RacketActivities            14905 non-null  float64
 17  CombatActivities            14909 non-null  float64
 18  NatureActivities            14895 non-null  float64
 19  SpecialActivities           14898 non-null  float64
 20  OtherActivities             14907 non-null  float64
 21  NumberOfFrequencies         14916 non-null  float64
 22  AttendedClasses             14942 non-null  int64
 23  AllowedWeeklyVisitsBySLA    14407 non-null  float64
 24  AllowedNumberOfVisitsBySLA  14942 non-null  float64
 25  RealNumberOfVisits          14942 non-null  int64
 26  NumberOfRenewals            14942 non-null  int64
 27  HasReferences               14930 non-null  float64
 28  NumberOfReferences          14942 non-null  int64
 29  Dropout                     14942 non-null  int64
dtypes: datetime64[ns](5), float64(16), int64(8), object(1)
memory usage: 3.5+ MB
```

Figure 1: Information about Existing Variables, Respective Number of Non-Null Values and Datatypes

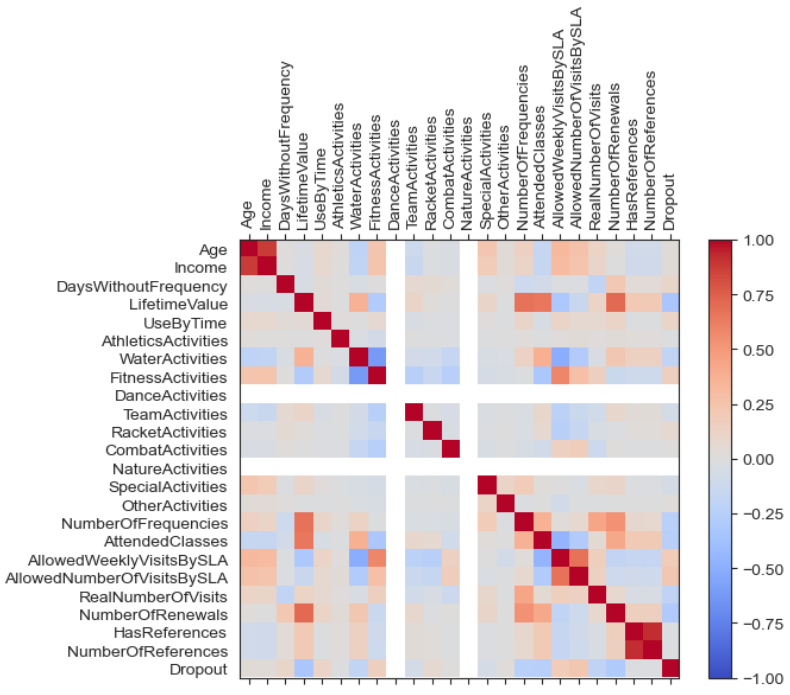

Figure 2: Correlation Matrix for Non-Date Variables

Figure 3: Number of Customers Whose Enrollment Started on Each Day of the Week
(Example of one of the heat maps performed during Visual EDA. Enrollments starting on Monday had a higher absolute frequency in comparison with the other weekdays.)

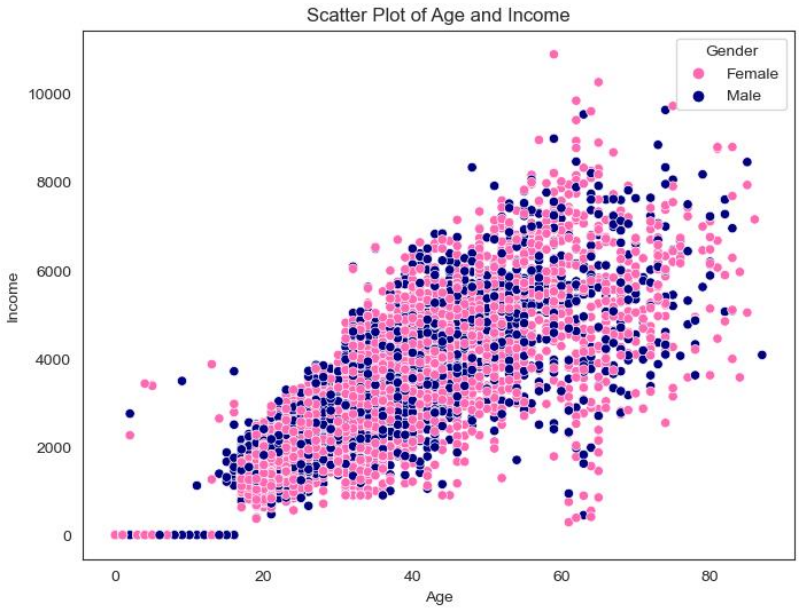

Figure 4: Variation of *Income* of Female and Male Customers over *Age*
(Example of one of the scatter plots performed during Visual EDA. Female and male customers seem to have a similar variation of income in the several age stages.)

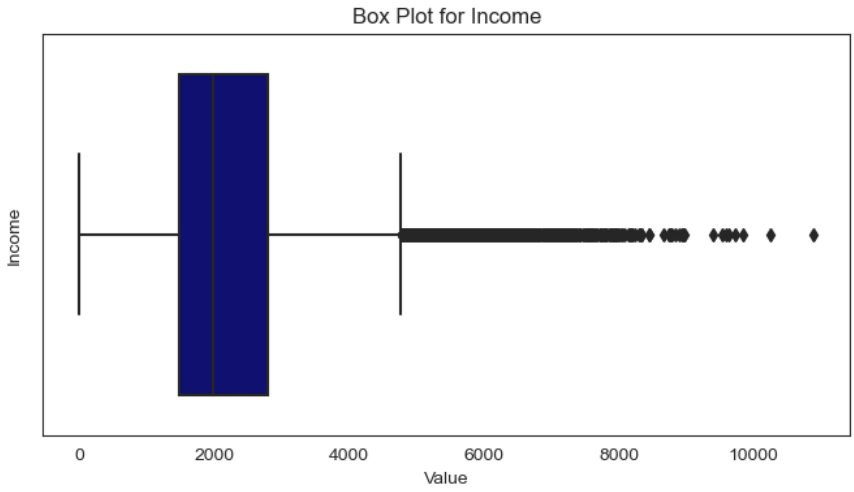

Figure 5: Box Plot for *Income*
(Example of one of the box plots performed during Visual EDA. According to the usual application of the interquartile range, incomes over 5000 can be interpreted as outliers.)

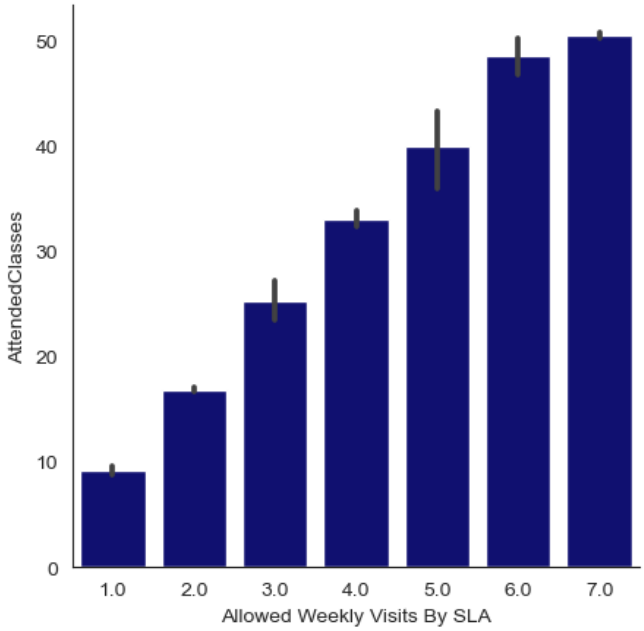

Figure 6: Bar Chart of *AttendedClasses* in Terms of Number of *AllowedWeeklyVisitsBySLA*
(Example of one of the bar charts performed during Visual EDA. There is a higher number of attended classes for customers who are allowed to visit every day.)

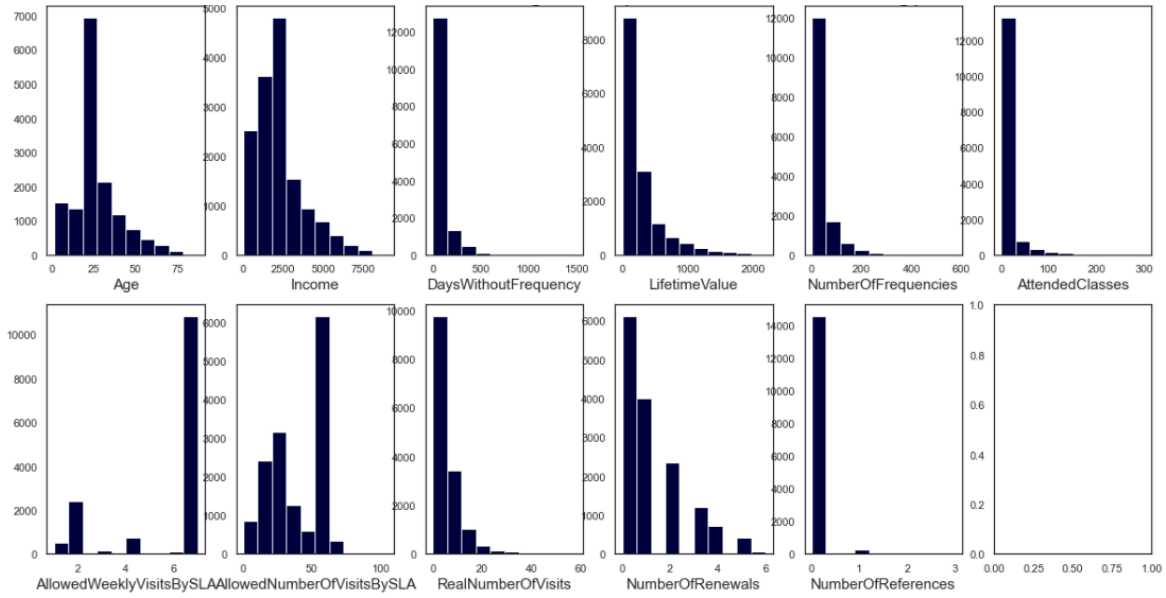

Figure 7: Metric Features' Histograms (After Manual Filtering)

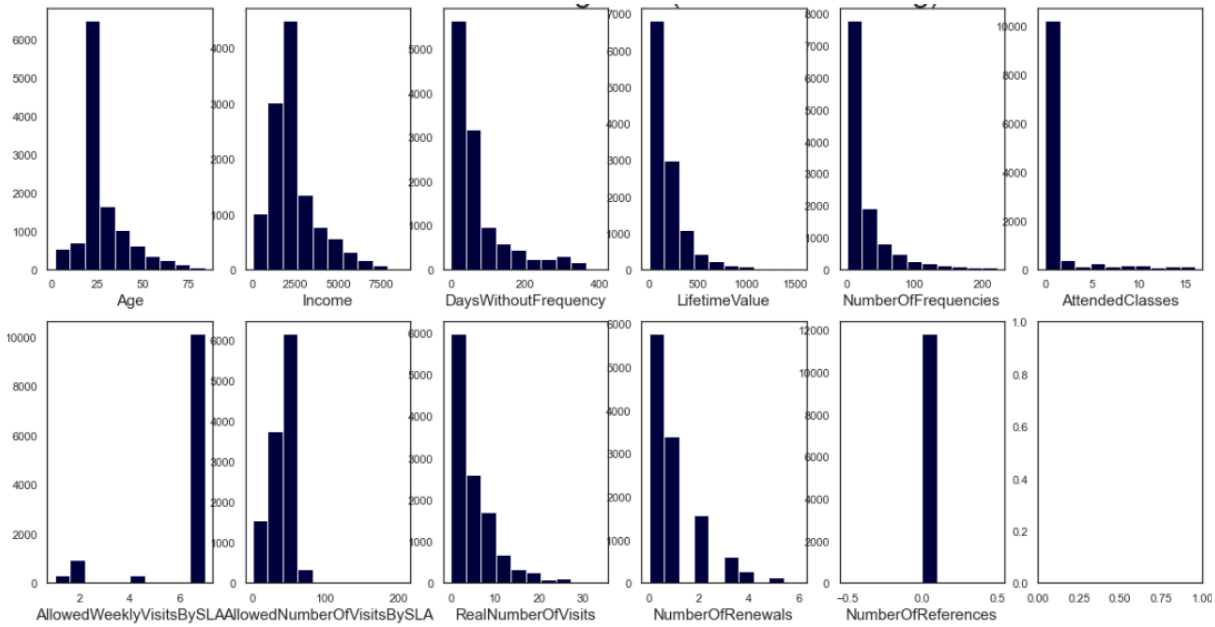

Figure 8: Metric Features' Histograms (After IQR Filtering)

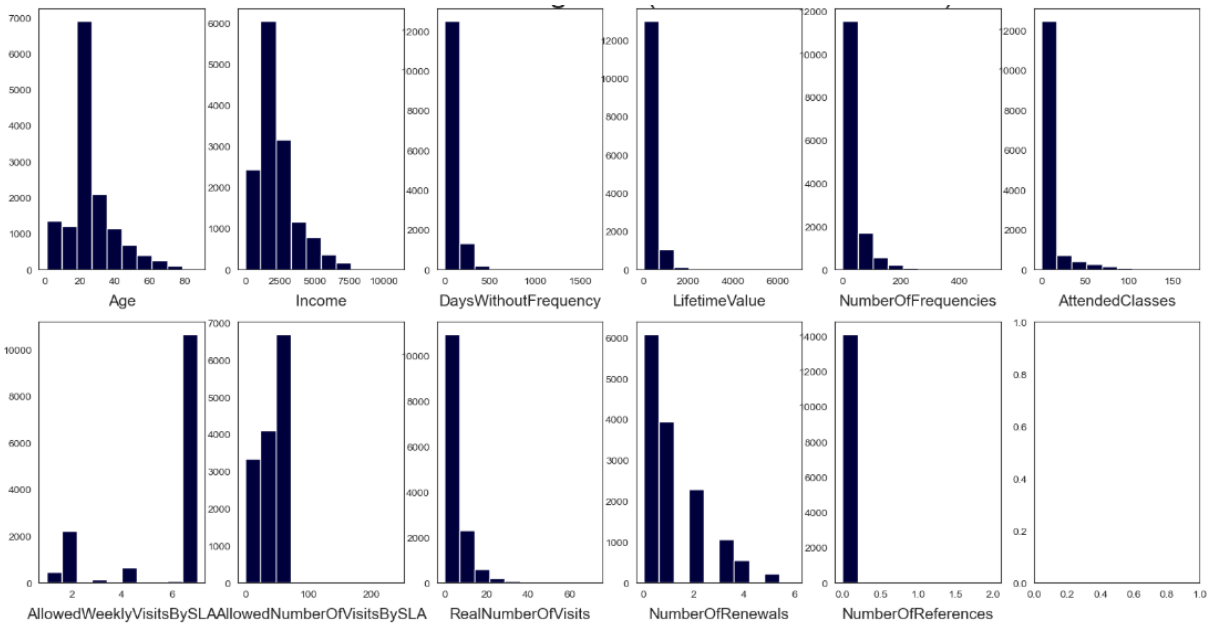

Figure 9: Metric Features' Histograms (After Filtering with Isolation Forest)

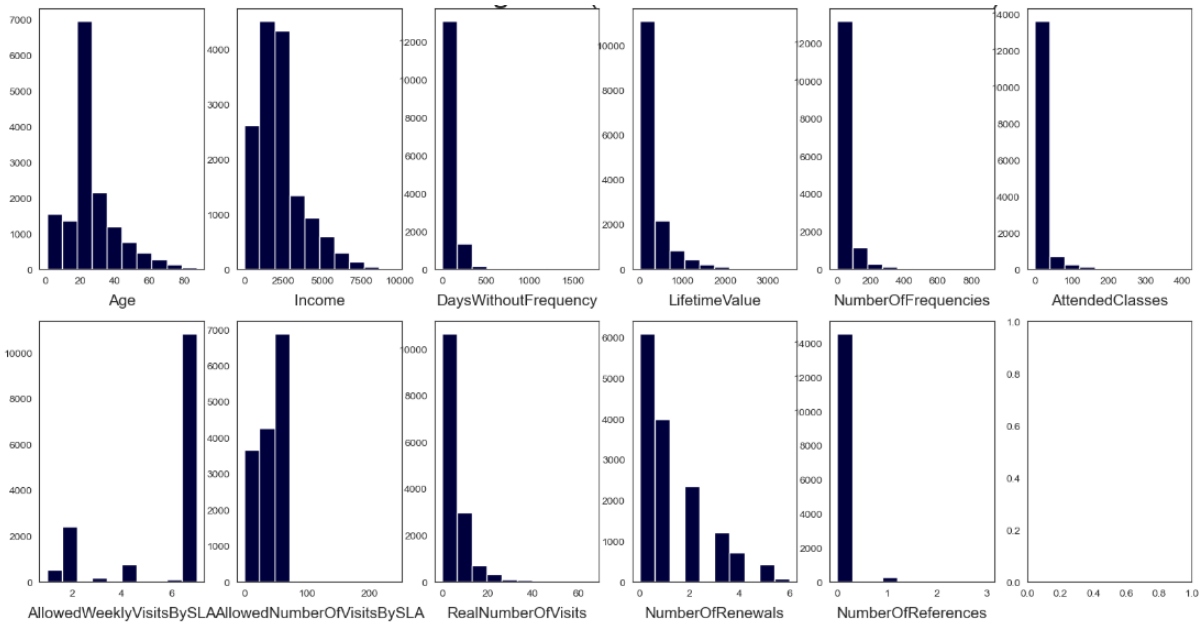

Figure 10: Metric Features' Histograms (After Combining Manual and IQR Filtering)

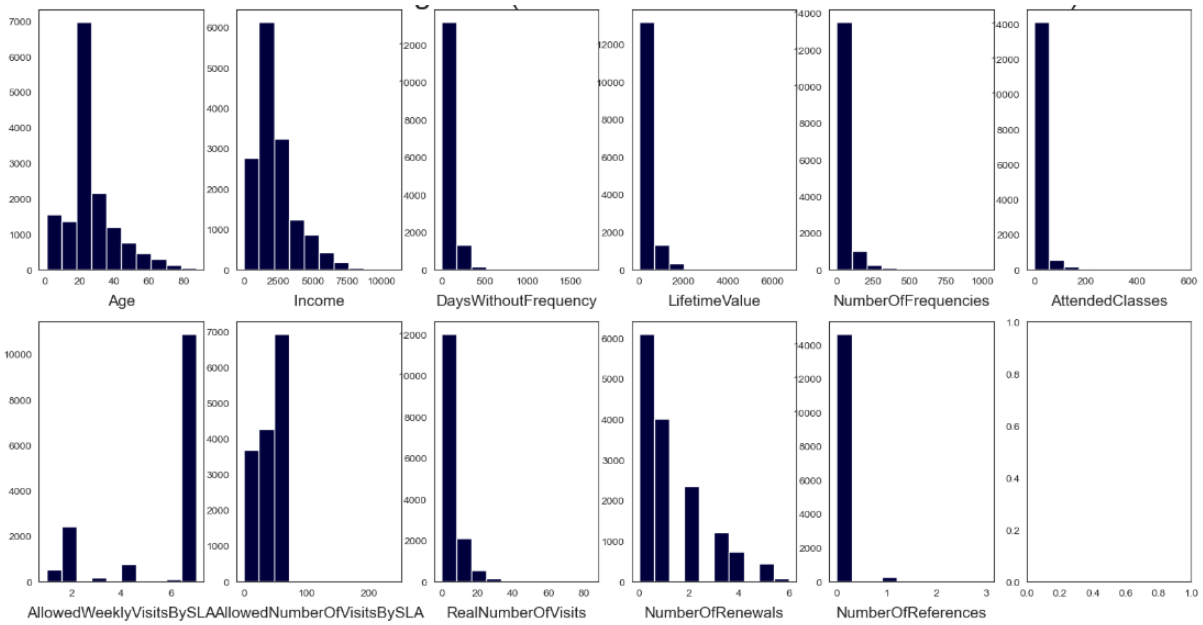

Figure 11: Metric Features' Histograms (After Combining Manual Filtering and Isolation Forest)

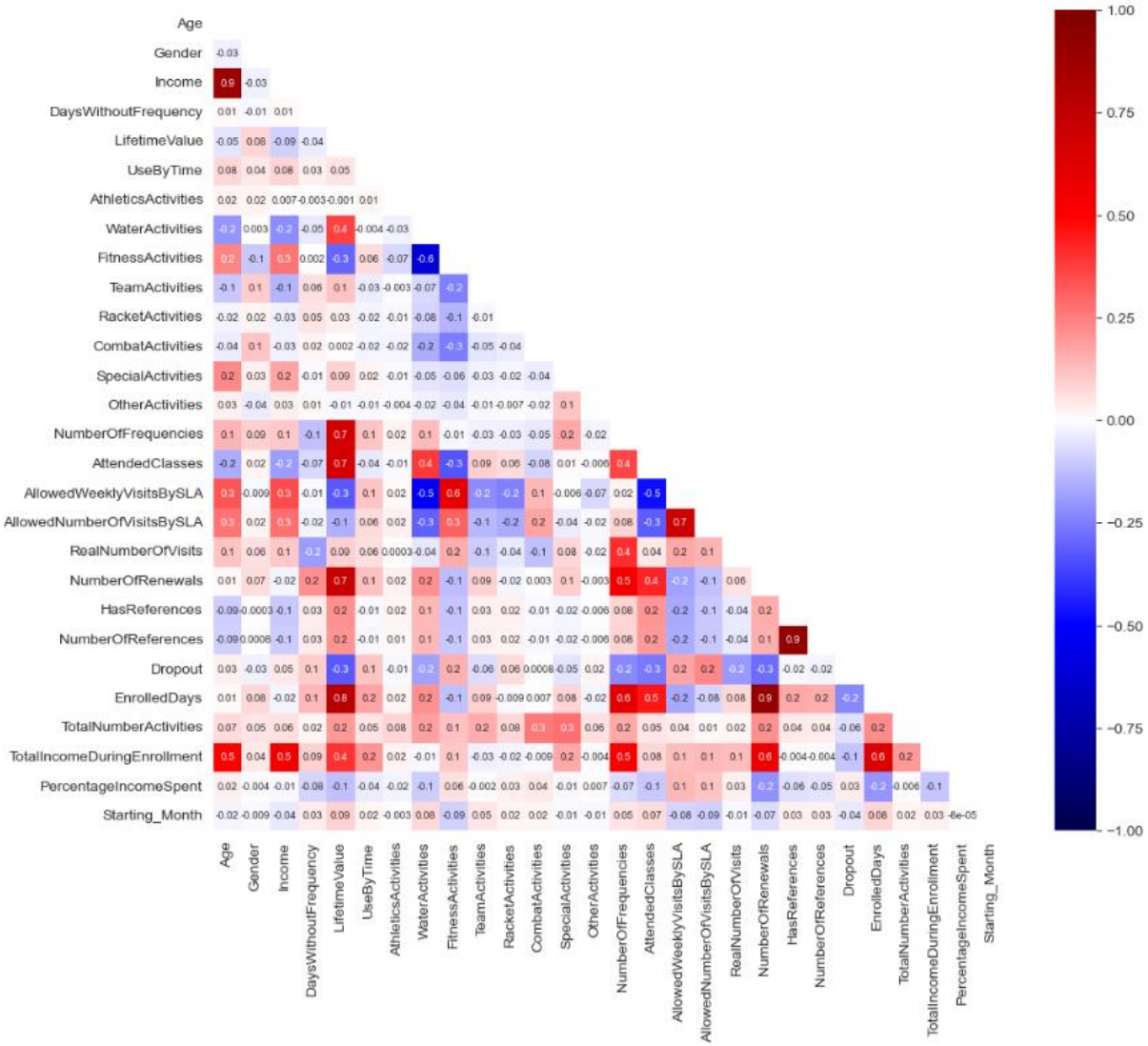

Figure 12: Correlation Matrix (Pearson Correlation) Performed During Feature Selection
(Squares with darker blues or darker red indicate higher correlation between variables (negatively or positively, respectively).)

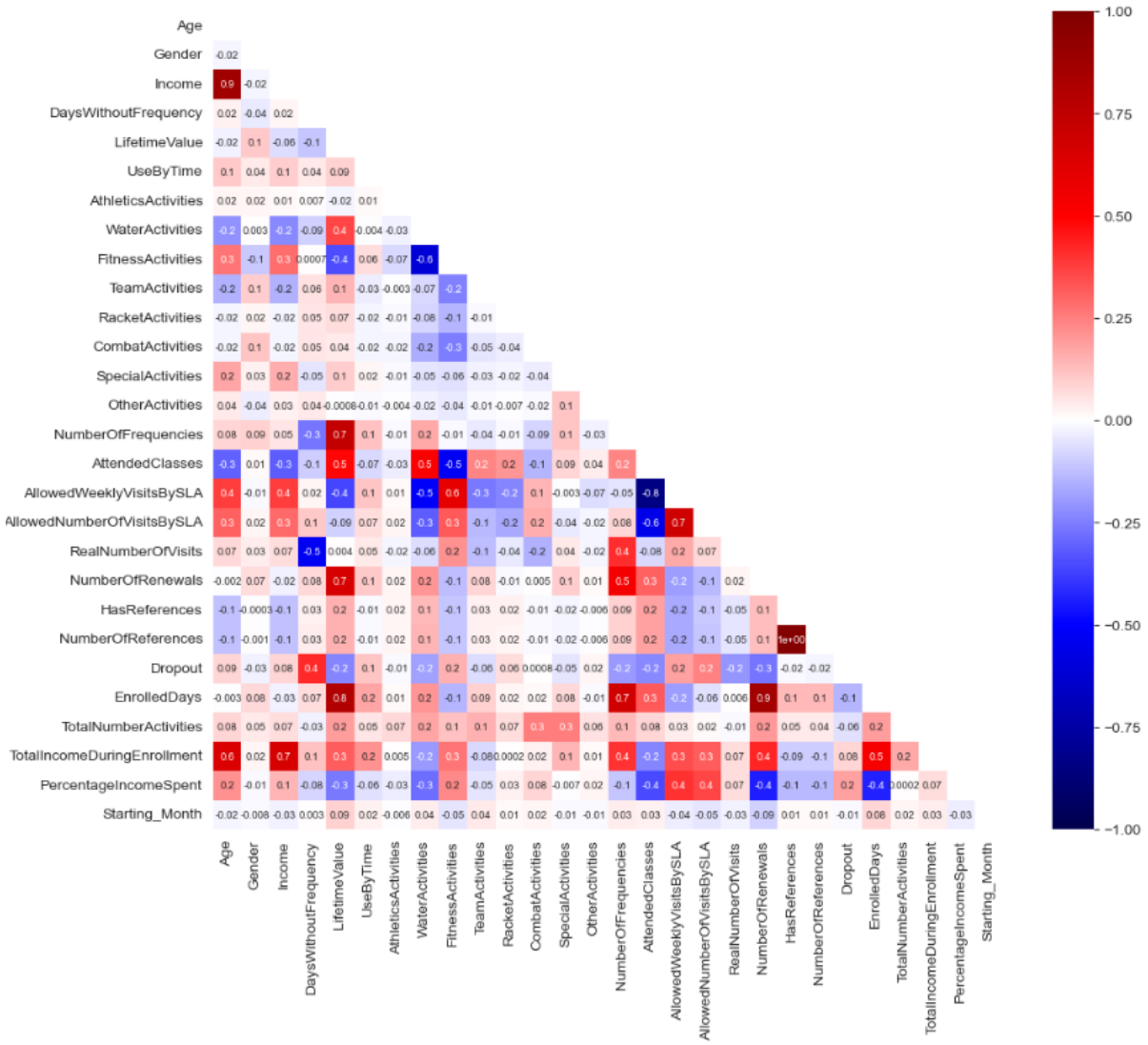

Figure 13: Correlation Matrix (Spearman Correlation) Performed During Feature Selection
(Squares with darker blues or darker red indicate higher correlation between variables (negatively or positively, respectively).)

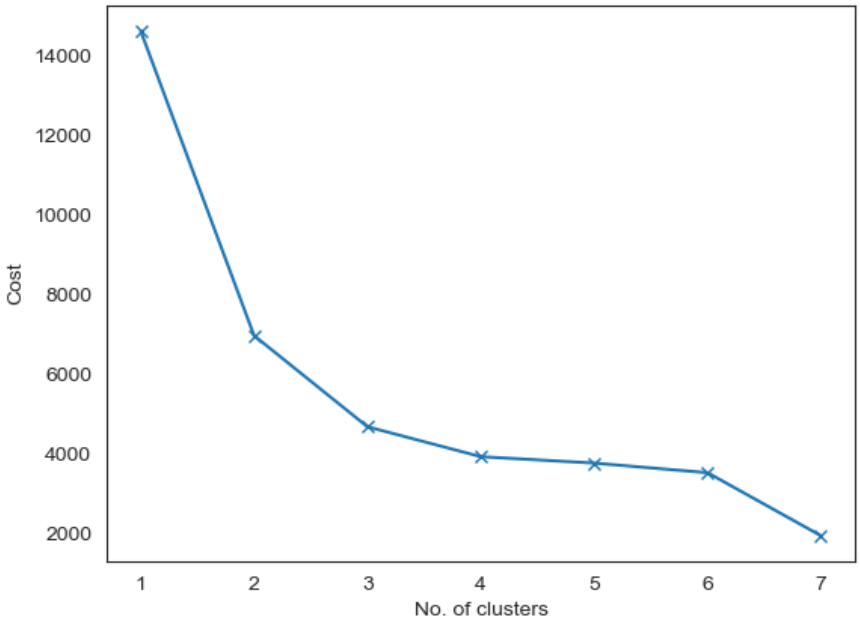

Figure 14: Elbow Curve for the K-Modes Algorithm

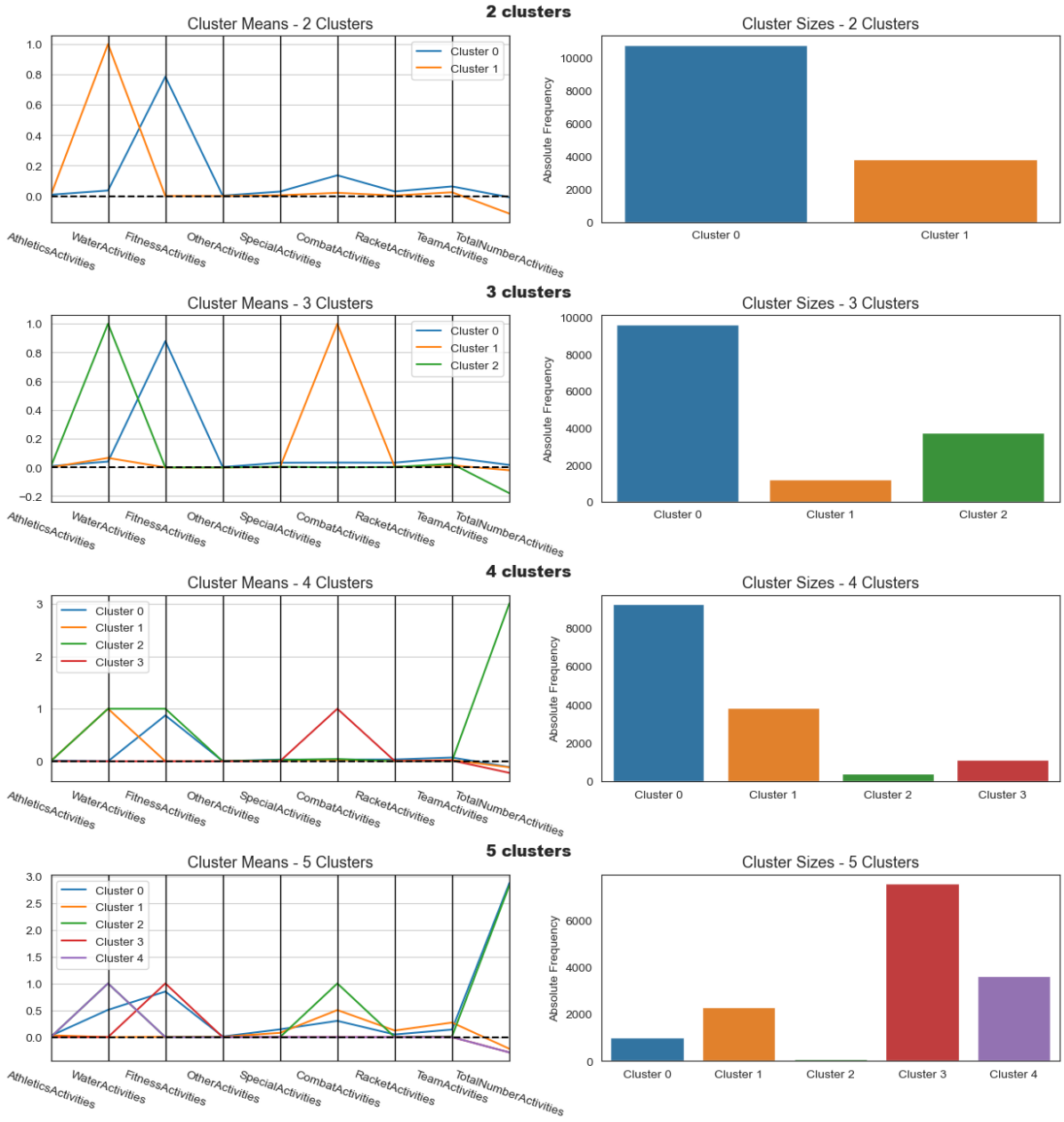

Figure 15: Visual Cluster Profiling for the K-Modes Algorithm

Number of Clusters	Before Merging Activities	After Merging Activities
2	0.309	0.354
3	0.545	0.459
4	0.676	0.731
5	0.912	0.872

Table 1: R-Squared Scores for the K-Modes Algorithm

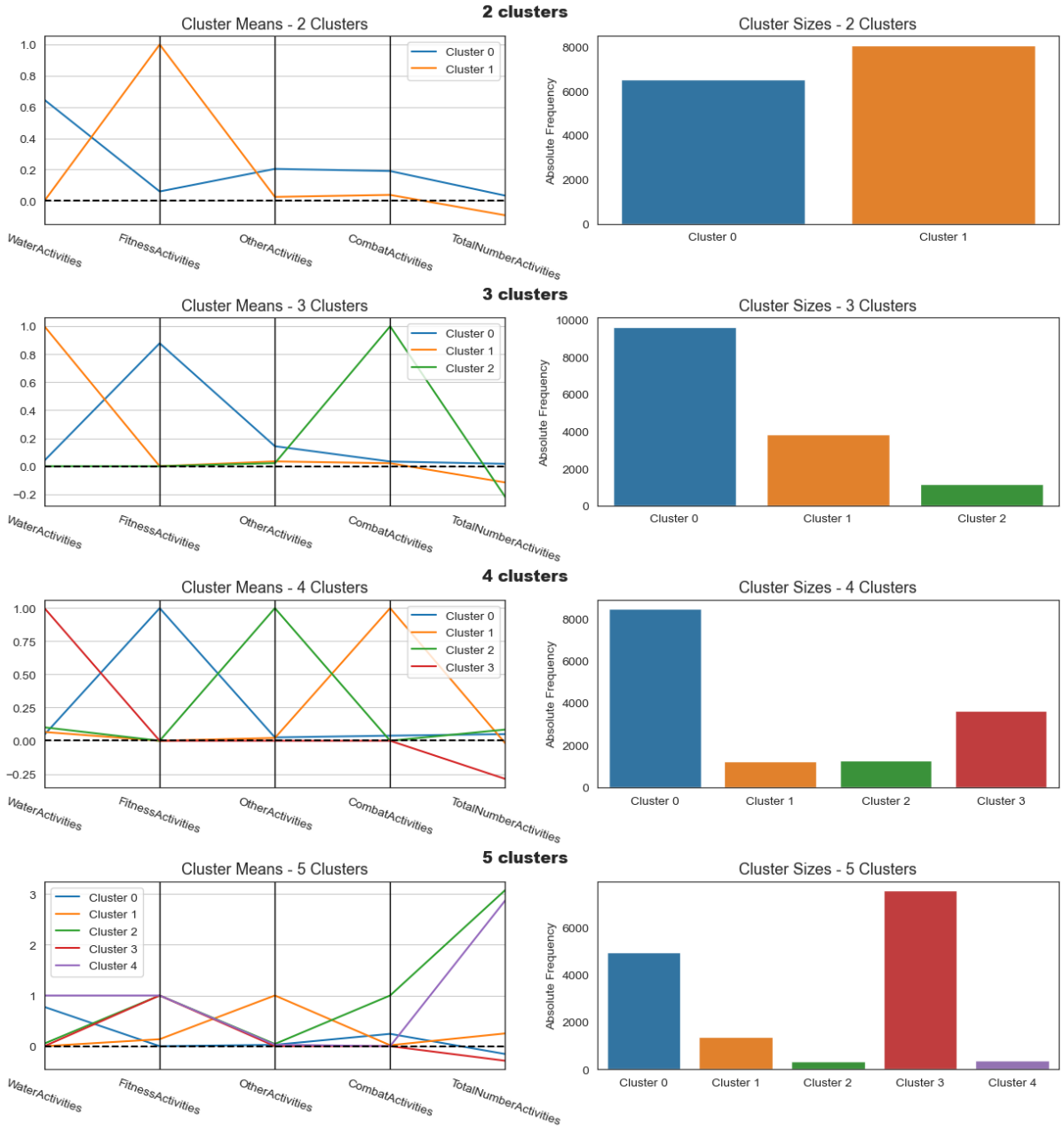

Figure 16: Visual Cluster Profiling for the K-Modes Algorithm Using Grouped Activities

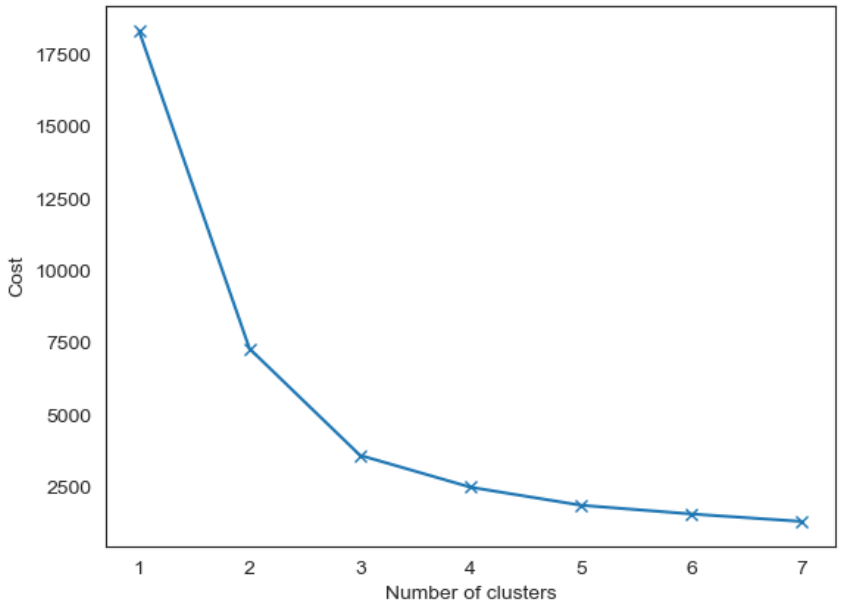

Figure 17: Elbow Curve for the K-Prototypes Algorithm

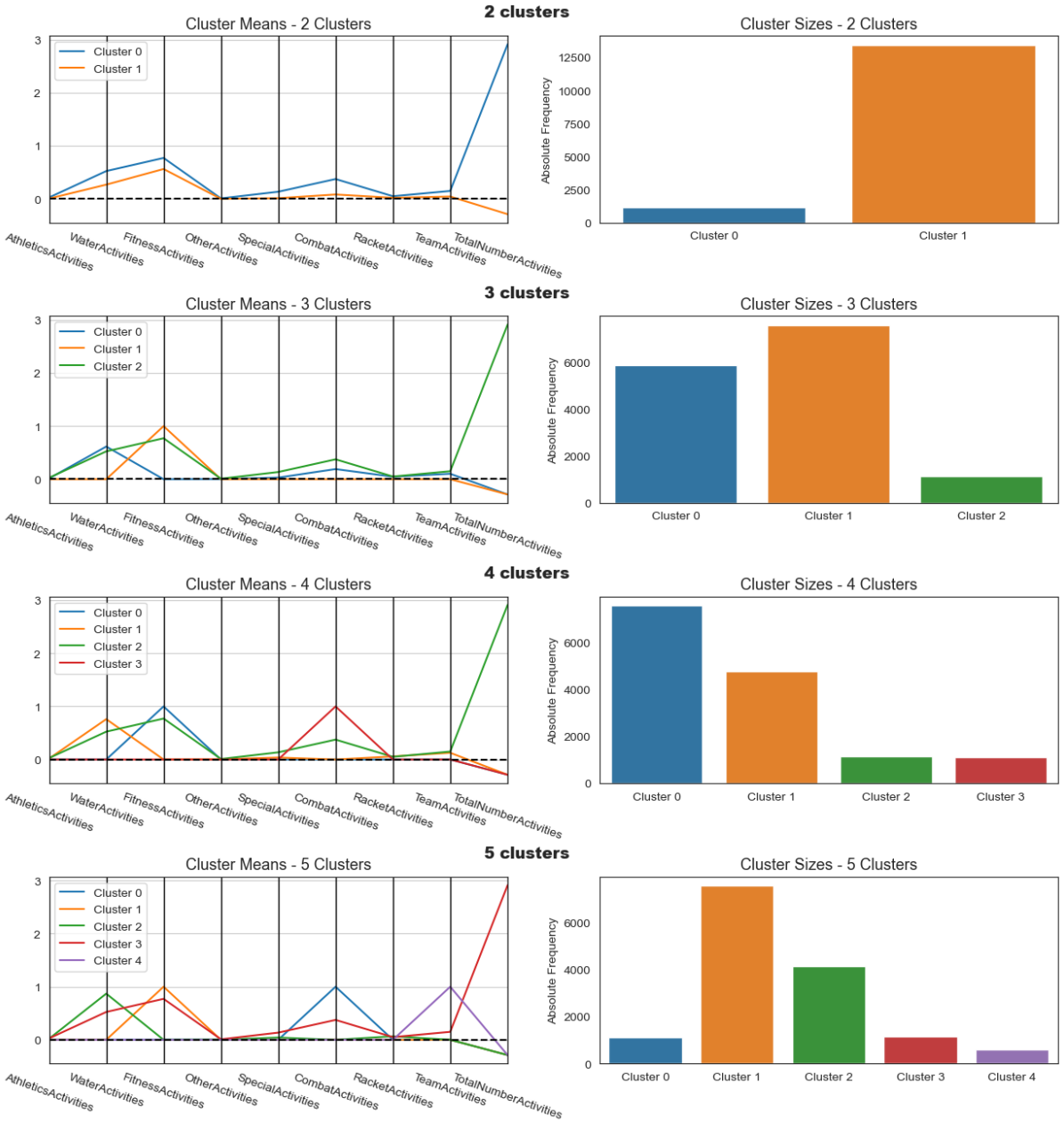

Figure 18: Visual Cluster Profiling for the K-Prototypes Algorithm

Number of Clusters	Before Merging Activities	After Merging Activities
2	0.557	0.562*
3	0.812	0.824*
4	0.894	0.942
5	0.922	0.973

Table 2: R-Squared Scores for the K-Prototypes Algorithm

*Despite having different R-Squared Scores, the clusters were equal to the ones achieved before merging the activities.

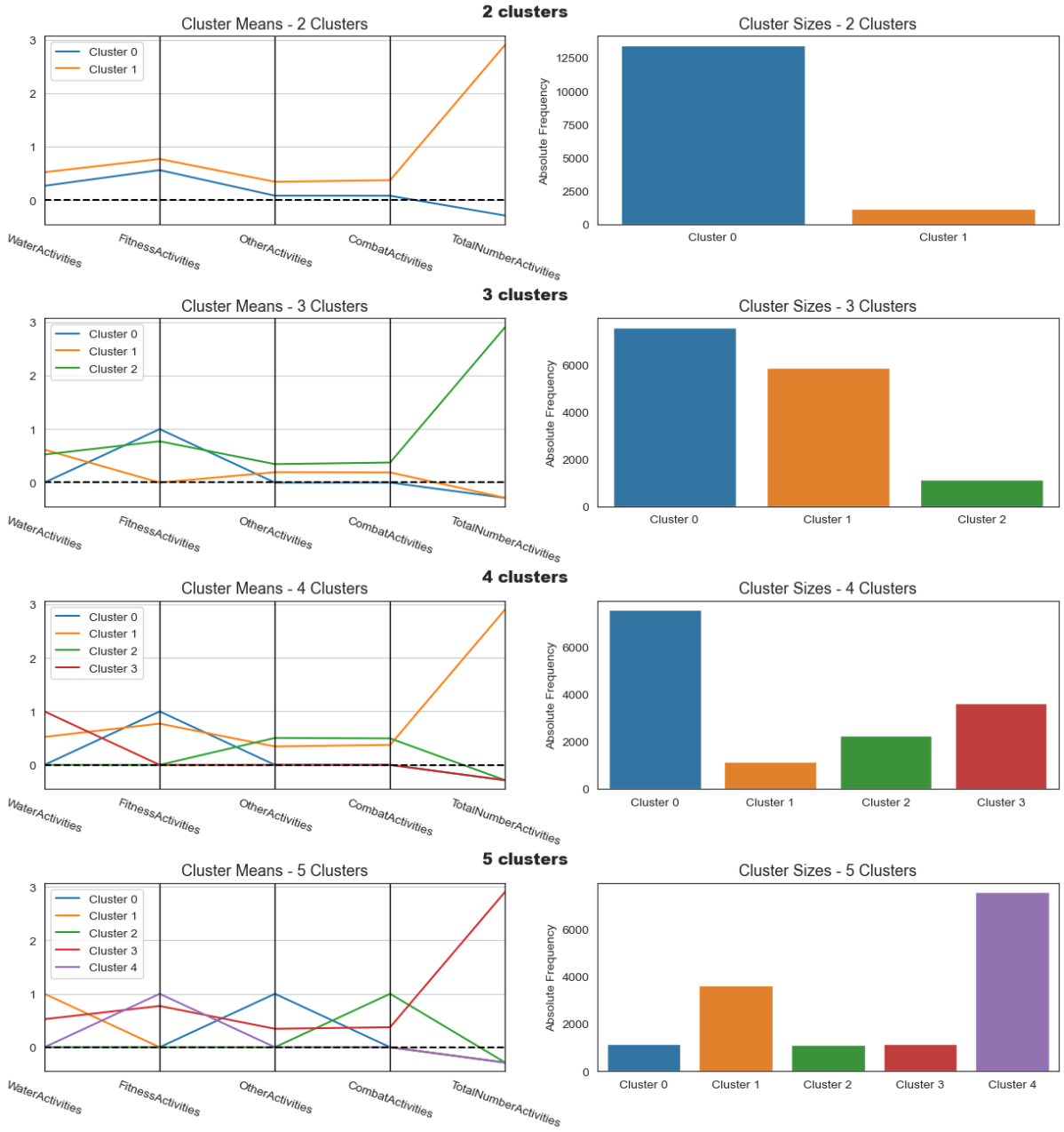

Figure 19: Visual Cluster Profiling for the K-Prototypes Algorithm Using Grouped Activities

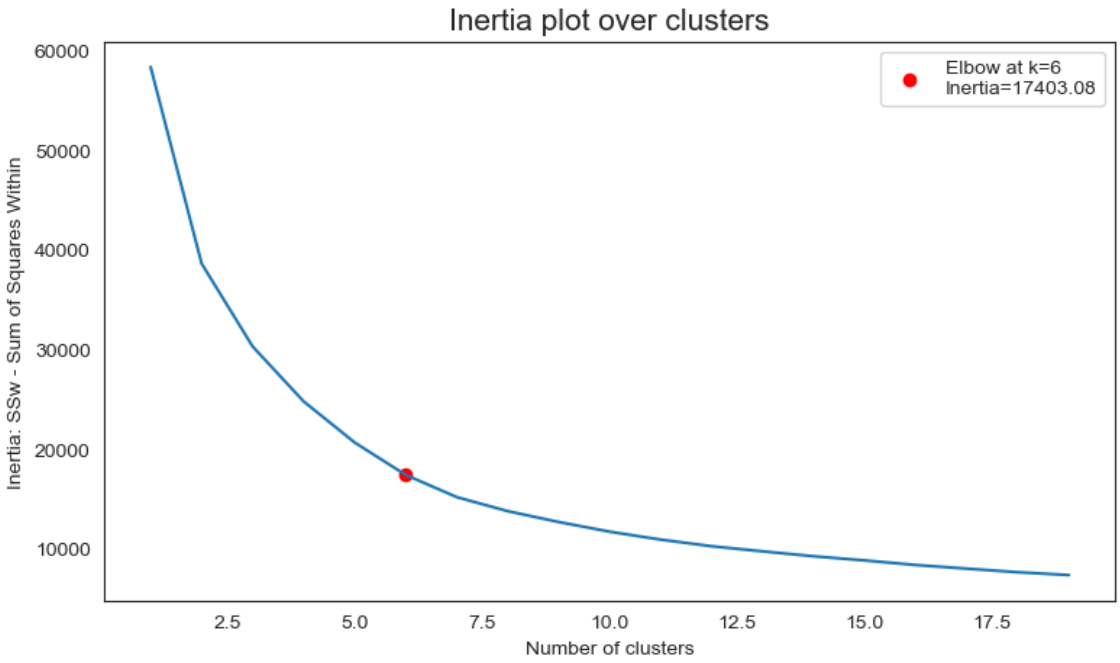

Figure 20: Visual Inertia for a Range of Clusters Between 1 And 19 (Value Perspective)

Number of Clusters	Silhouette Score
2	0.467
3	0.448
4	0.345
5	0.341
6	0.370
7	0.335
8	0.321
9	0.326
10	0.331
11	0.319
12	0.326
13	0.317
14	0.321
15	0.327
16	0.325
17	0.309
18	0.312
19	0.313

Table 3: Silhouette Score for Each Number of Clusters (Value Perspective)

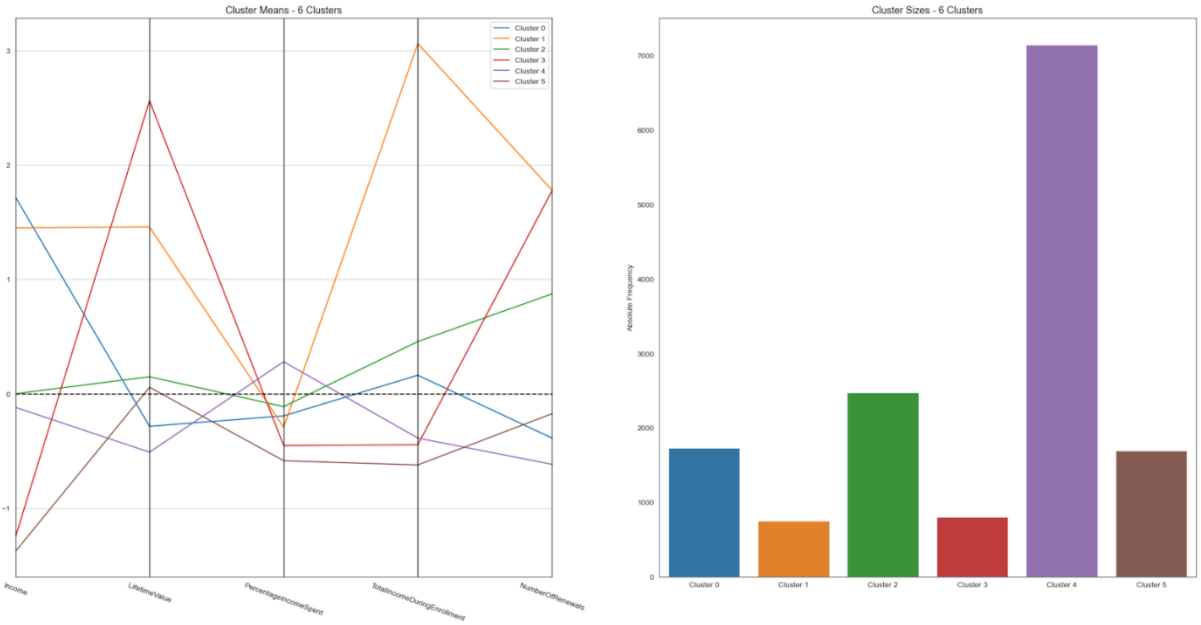

Figure 21: Visual Cluster Profiling for the K-Means Algorithm (Value Perspective)

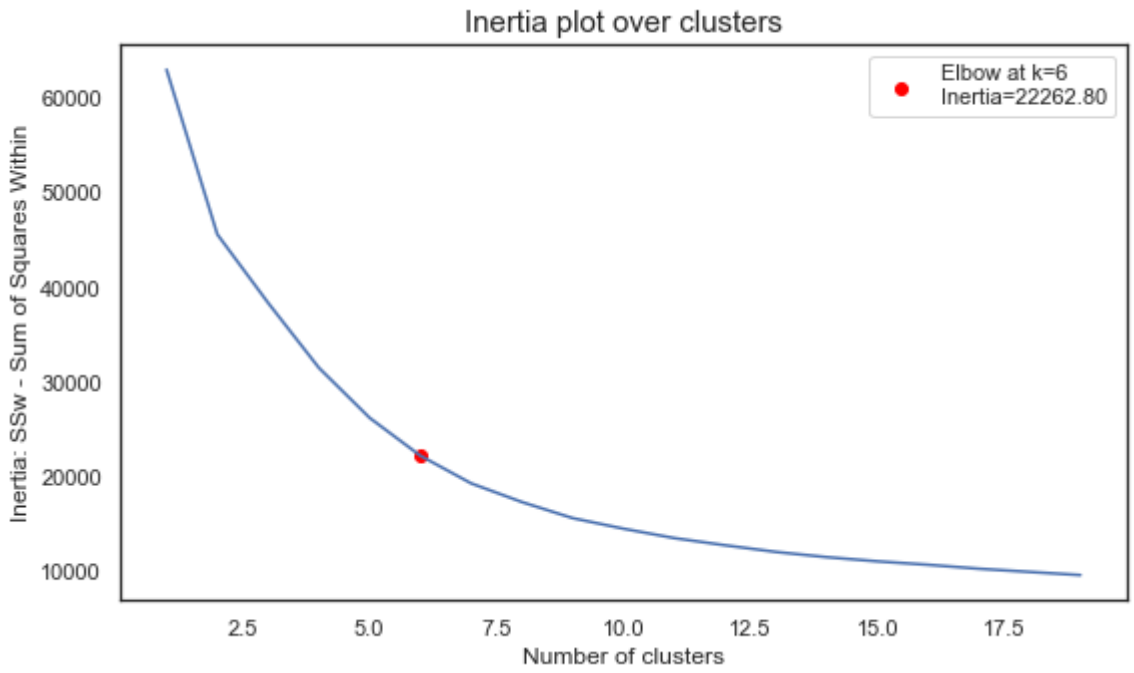

Figure 22: Visual Inertia for a Range of Clusters Between 1 And 19 (Time Perspective)

Number of Clusters	Silhouette Score
2	0.504
3	0.434
4	0.453
5	0.417
6	0.390
7	0.376
8	0.334
9	0.344
10	0.328
11	0.324
12	0.290
13	0.279
14	0.275
15	0.274
16	0.277
17	0.273
18	0.273
19	0.262

Table 4: Silhouette Score for Each Number of Clusters (Time Perspective)

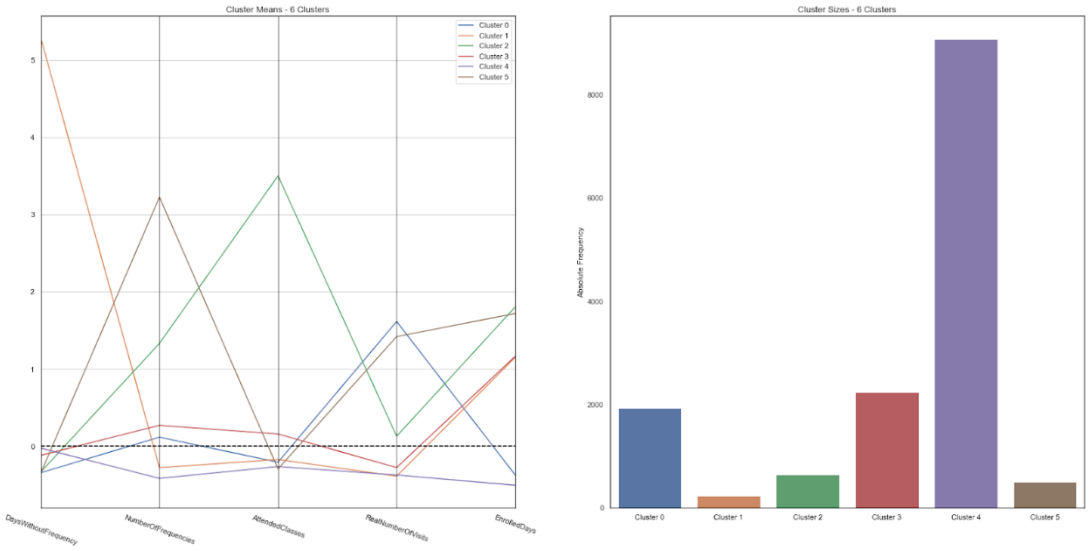

Figure 23: Visual Cluster Profiling for the K-Means Algorithm (Time Perspective)

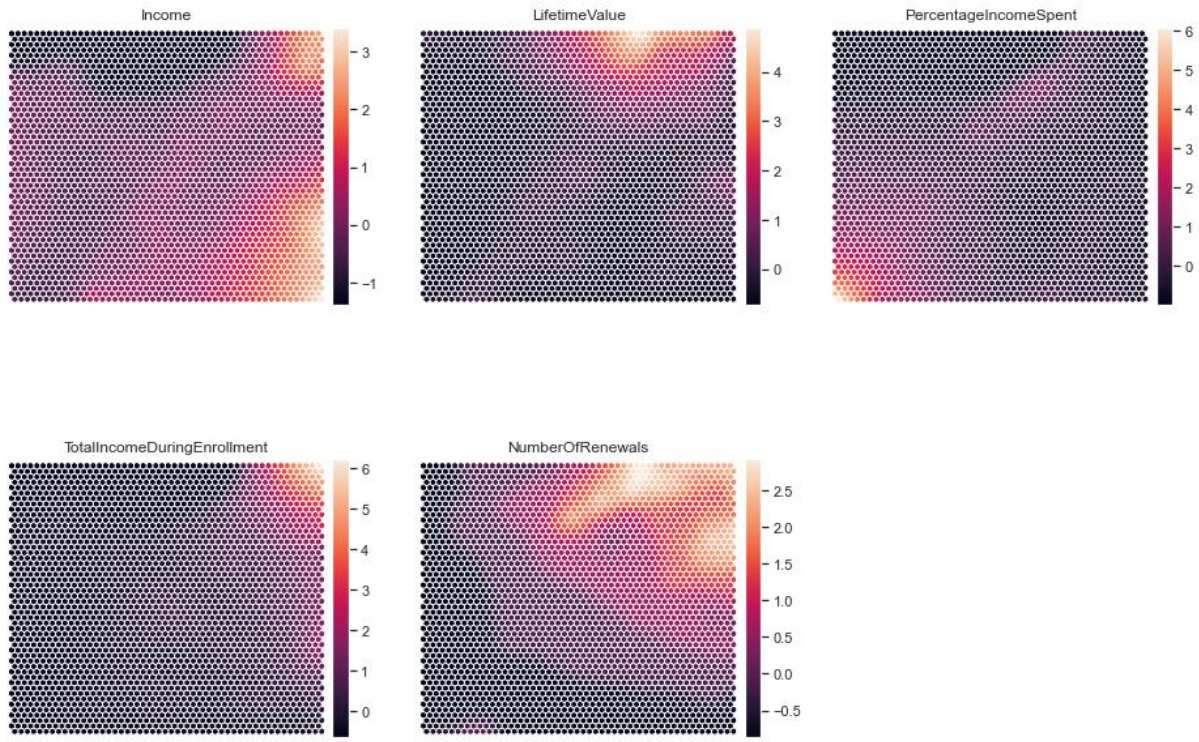

Figure 24: Component Planes for the Self-Organizing Maps (Value Perspective)

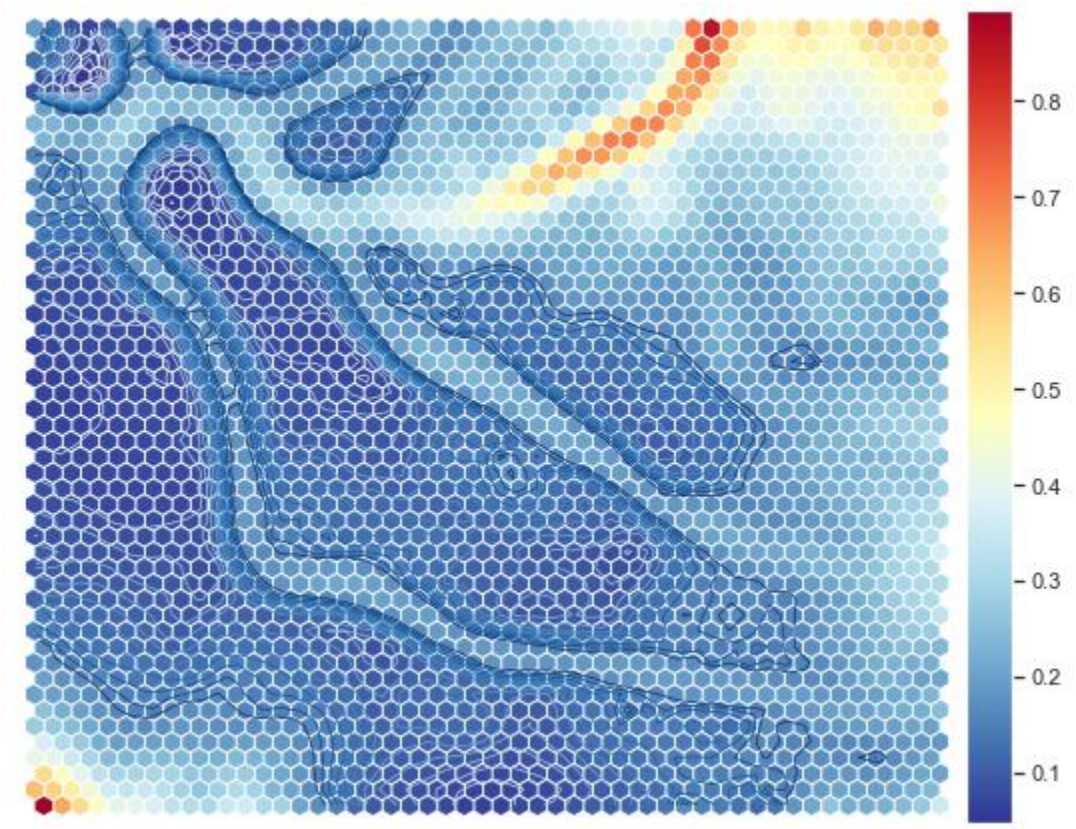

Figure 25: U-Matrix for the SOM Algorithm (Value Perspective)

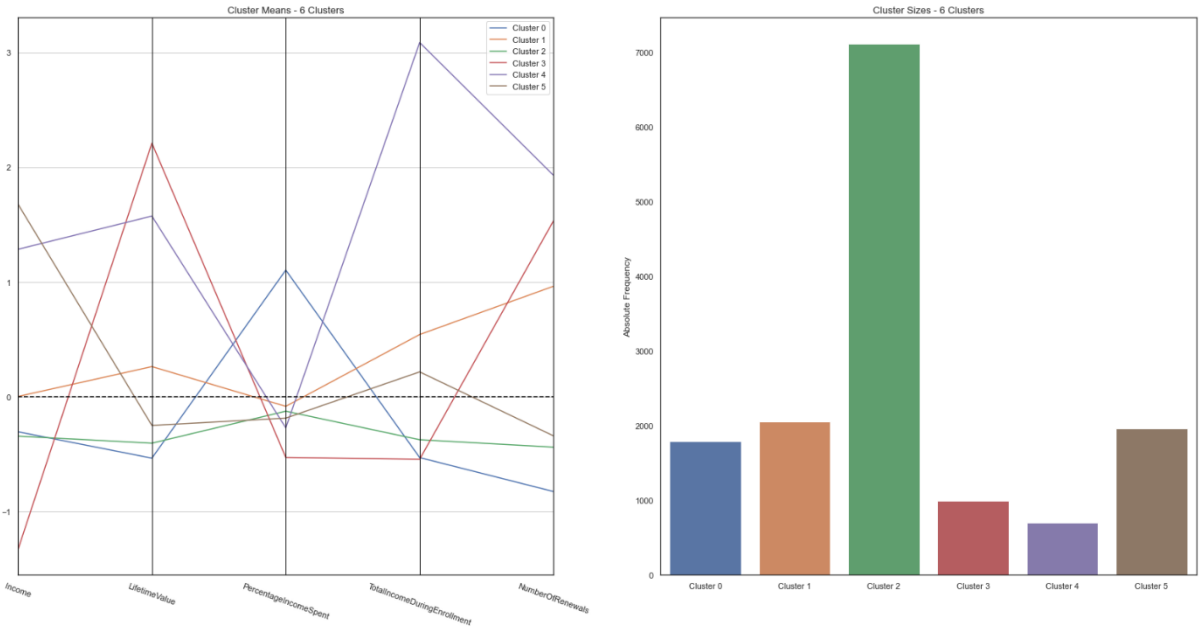

Figure 26: Visual Cluster Profiling for the SOM + K-Means Algorithm (Value Perspective)

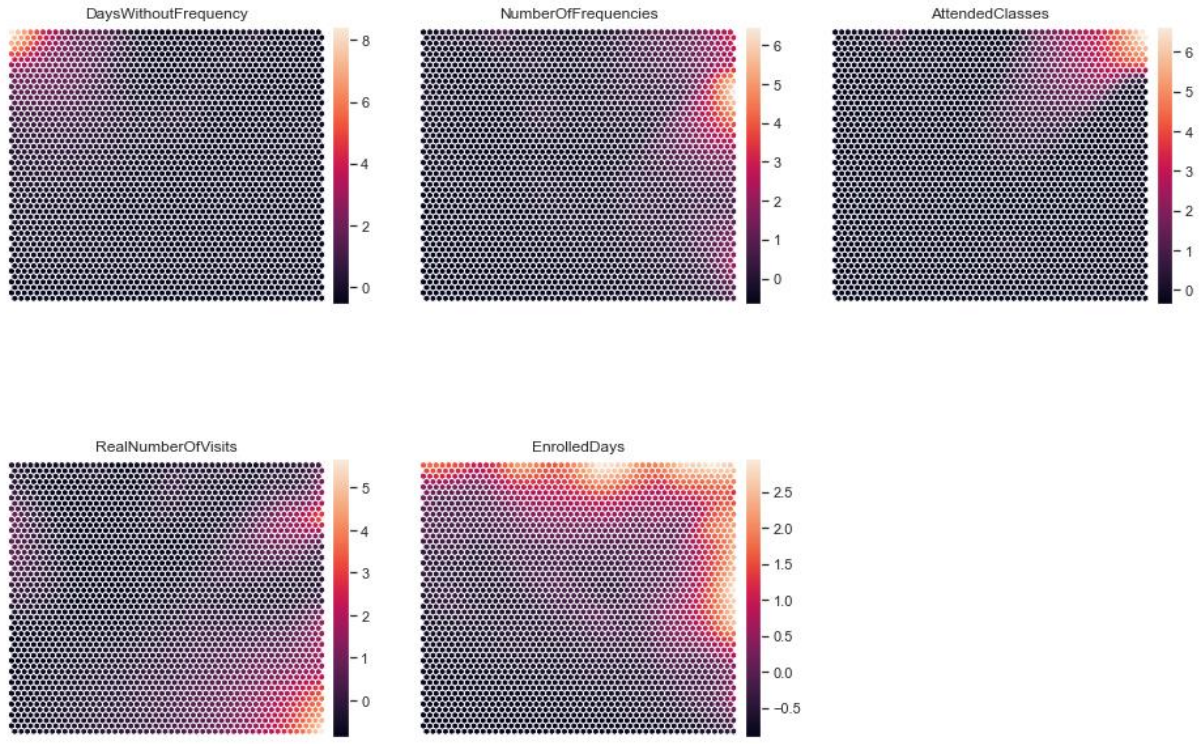

Figure 27: Component Planes for the Self-Organizing Maps (Time Perspective)

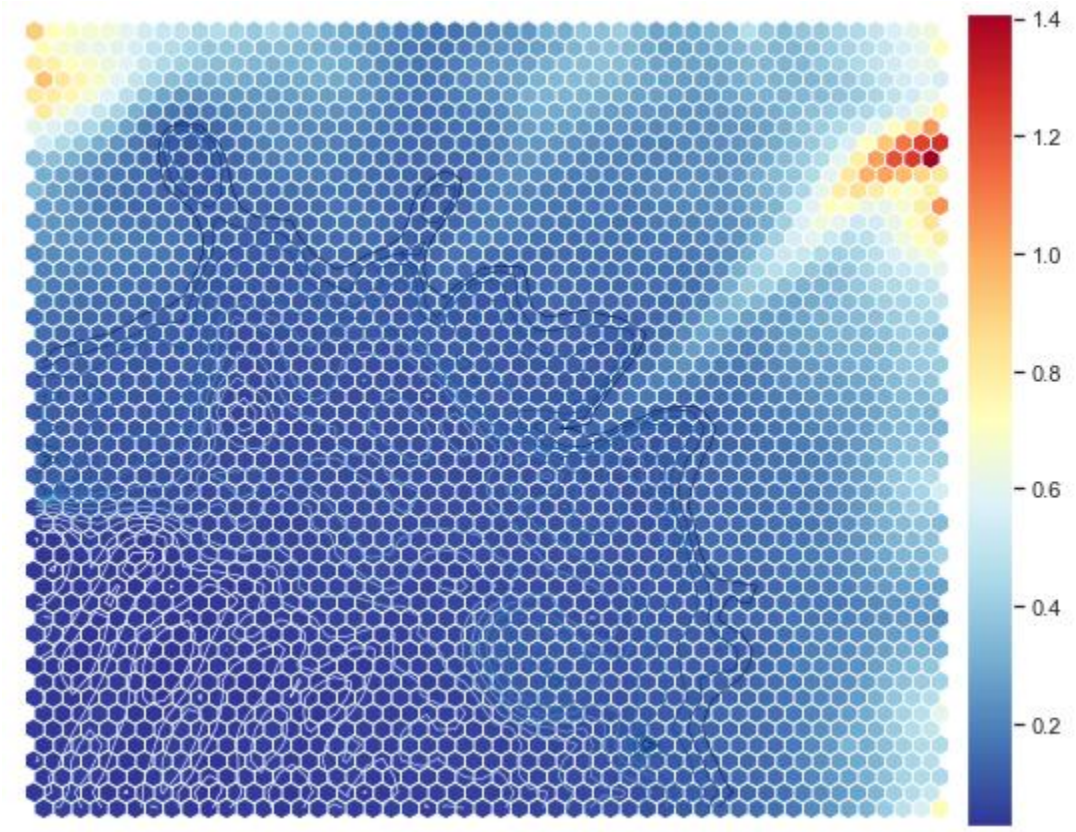

Figure 28: U-Matrix for the SOM Algorithm (Time Perspective)

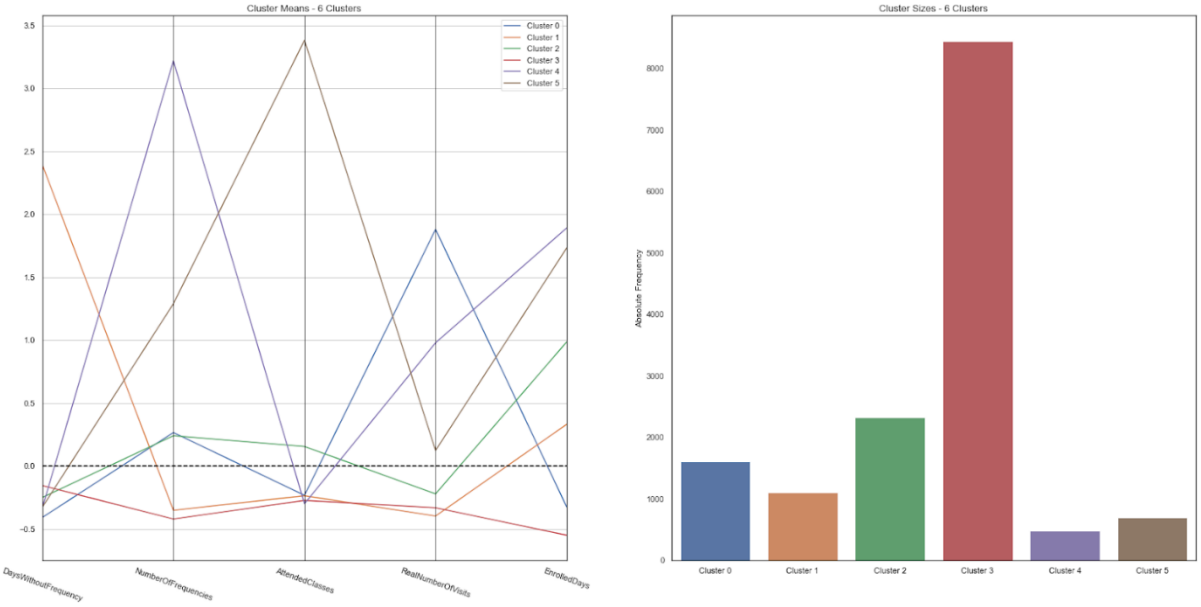

Figure 29: Visual Cluster Profiling for the SOM + K-Means Algorithm (Time Perspective)

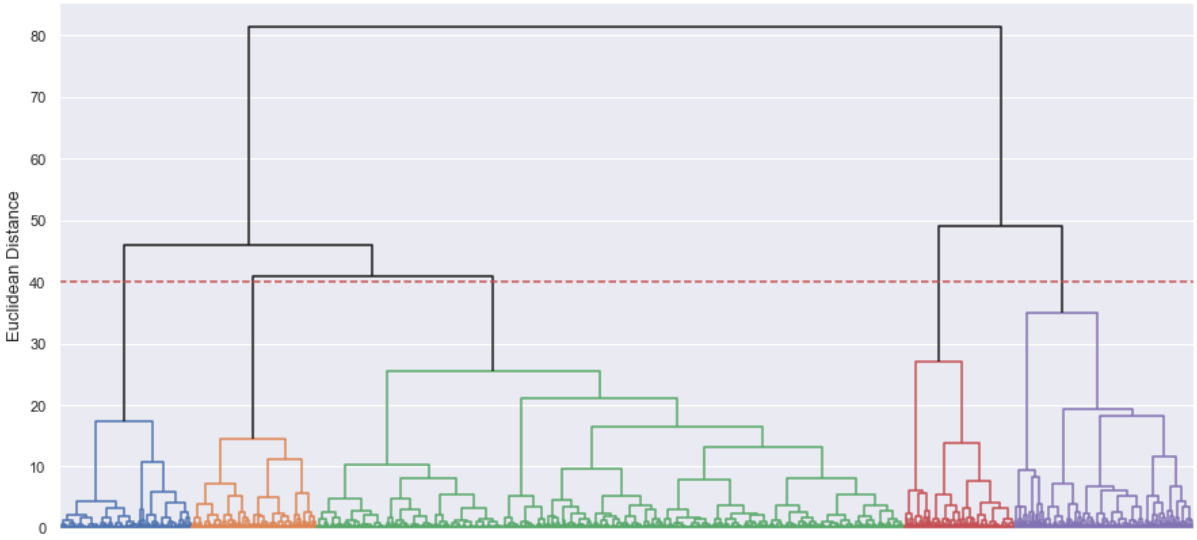

Figure 30: Hierarchical Clustering Dendrogram (Value Perspective)

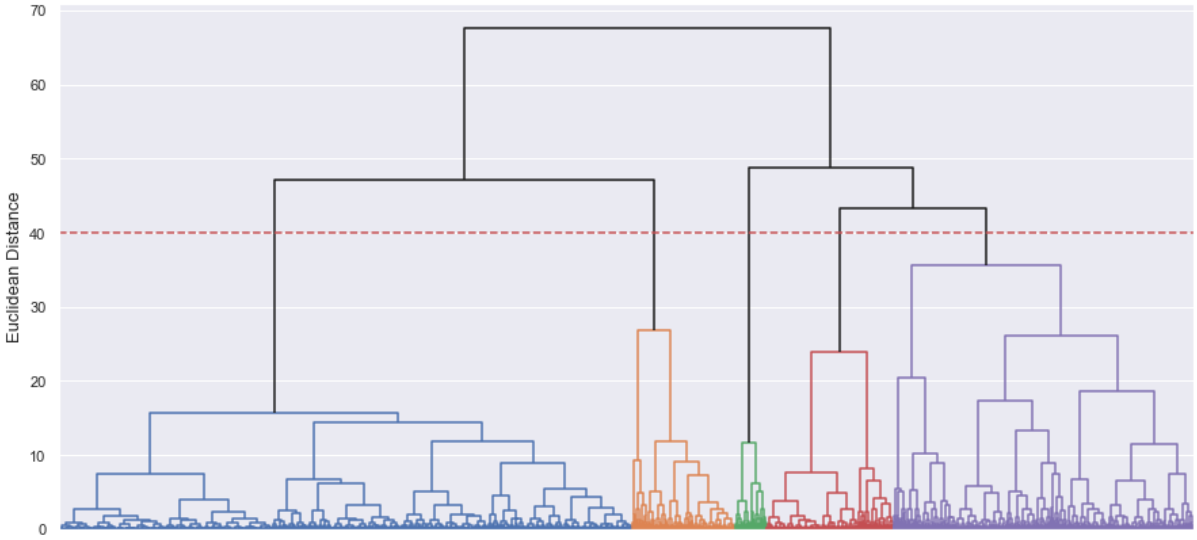

Figure 31: Hierarchical Clustering Dendrogram (Time Perspective)

Bandwidth Value	Number Of Clusters	R-Squared
0.5	334	0.930
1	38	0.746
1.2	12	0.516
1.5	5	0.212
2	1	0

Table 5: Bandwidth Value Testing (Value Perspective)

Bandwidth Value	Number Of Clusters	R-Squared
0.5	734	0.937
1	105	0.743
1.2	63	0.678
1.5	26	0.552
2	8	0.350
2.2	2	0.1
2.5	1	0

Table 6: Bandwidth Value Testing (Time Perspective)

"Eps"	Clusters	R-Squared
0.10	39	0.380
0.15	34	0.492
0.20	24	0.556
0.25	19	0.580
0.30	19	0.604
0.35	18	0.622
0.40	17	0.595
0.45	19	0.610
0.50	16	0.609
0.55	15	0.609
0.60	14	0.576

Table 7: "Eps" Value Testing (Value Perspective)

"Eps"	Clusters	R-Squared
0.10	44	0.155
0.15	44	0.222
0.20	13	0.221
0.25	14	0.248
0.30	6	0.236
0.35	14	0.248
0.40	6	0.222
0.45	5	0.210
0.50	9	0.204
0.55	6	0.182
0.60	4	0.173
0.65	4	0.139
0.70	2	0.116
0.75	4	0.106
0.80	3	0.097
0.85	2	0.081

Table 8: "Eps" Value Testing (Time Perspective)

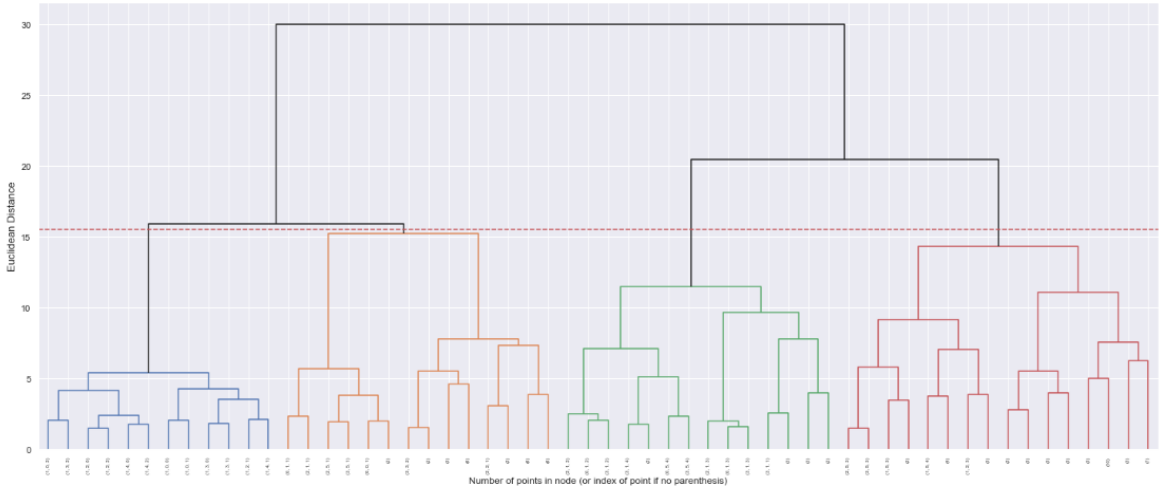

Figure 32: Dendrogram for Merging Solution

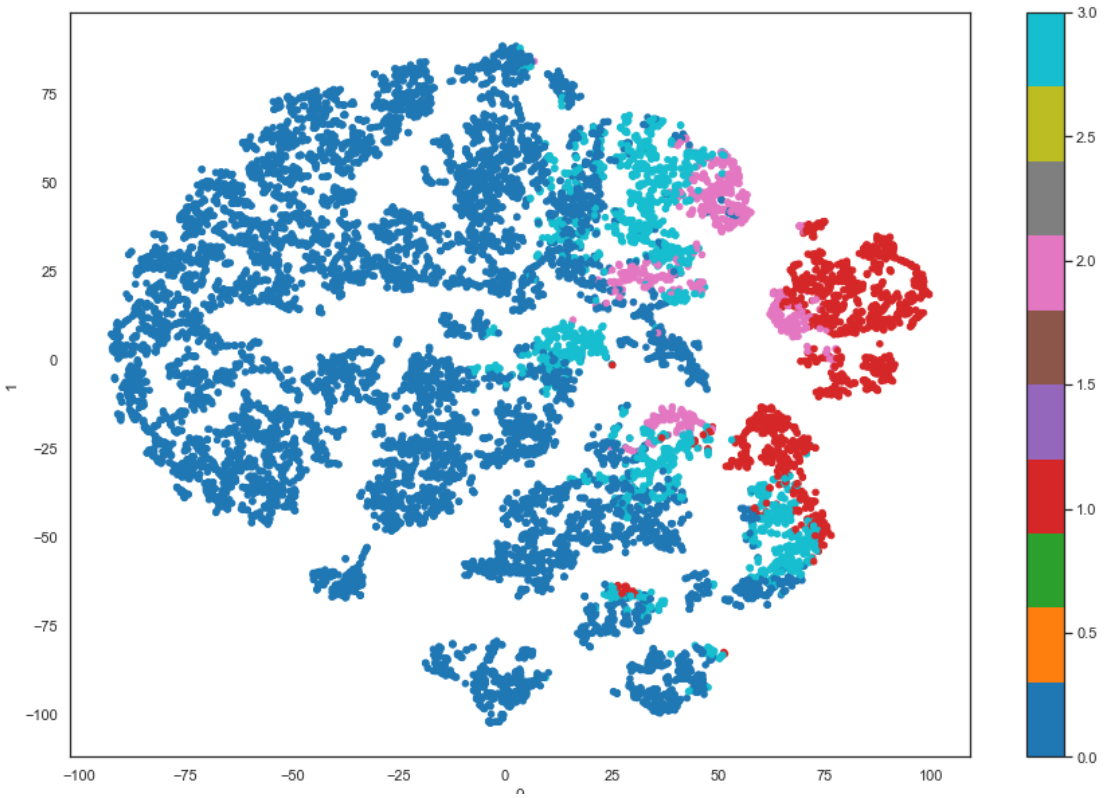

Figure 33: t-SNE Cluster Visualization (Merged Solution)

14. Appendix

Section A

The main drawback of the k-means algorithm is its inability to properly handle categorical variables, mainly when using Euclidean distances for computing distances between data points. Taking this into account, k-modes was developed as an extension of k-means, particularly built and prepared to deal with non-numerical attributes by replacing the clusters' means by their modes, using the frequencies of each value within a variable to update the centroids' positions (Han & Kamber, 2018).

Similarly to k-means, k-modes begins by randomly selecting "k" centroids, with "k" being the desired number of clusters. Then, using the Hamming metric - which calculates distance as the number of different characters or elements in strings or vectors (Waggener, 1995, p. 206) -, each point in the dataset is allocated to its nearest center, and the center is updated. These steps are repeated until a certain number of iterations is reached, or the clusters do not change significantly (Surya T. G. et al., 2023).

References

Han, J., & Kamber, M. (2018). *Data Mining : Concepts and Techniques (3rd ed.)*. Elsevier.

Waggener, W. N. (1995). *Pulse code modulation techniques : with applications in communications and data recording*. Van Nostrand Reinhold.

Surya T. G., Karthik C. S., & Sharath P. (2023). Clustering Categorical Data: Soft Rounding k-modes. *Information & Computation, 296*, 105115–105115. https://doi.org/10.1016/j.ic.2023.105115

Section B

Having k-means built to handle numeric variables and k-modes prepared to deal with categorical features, the k-prototypes algorithm combines both methods into a single one that can receive both numeric and categorical attributes. By utilizing a new dissimilarity measure calculated using a combination of both previous cost functions, it allows for the clustering of observations that are described by mixed datatypes (Huang, 1998).

References

Huang, Z. (1998). Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery, 2*(3), 283–304. https://doi.org/10.1023/a:1009769707641