Data Mining Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

CUSTOMER SEGMENTATION FOR XYZ SPORTS COMPANY

Group 72

Adriana Costinha, number: 20230567

Alícia Pinho Santos, number: 20230525

Mariana Cabral, number: 20230532

January, 2024

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

INDEX

1. INTRODUCTION

This report, developed for the final project of the Data Mining course within the Master's Degree in Data Science and Advanced Analytics at Nova IMS, focuses on the analysis of a provided database from the fictitious sports company, *XYZ Sports*.

The general objective of this project is to conduct a customer analysis by segmenting the database and identifying relevant customer clusters based on specific characteristics and behaviours. Keeping that in mind, our main purpose stands in using unsupervised learning algorithms to develop a data-driven marketing strategy that provides our company in study with the necessary insights to optimize and tailor their services for enhanced efficiency and customization.

Understanding customer segmentation's importance is crucial for a company's success. By looking at some studies on this topic we know that the key lies in personalization, which is proven to reduce customer acquisition costs by approximately 50%, lift revenues by 5% to 15%, and increase marketing return on investment by 10% to 30%.

2. DATA EXPLORATION

2.1. INITIAL ANALYSIS

After importing the provided dataset into a variable called *customers*, we tried to better understand the data we were working with. This original dataset consisted of thirty variables, excluding the *ID* that was set as the index, and 14.942 records.

In this section, we also identified the data types of the variables, as well as the existence of missing values, after looking for undesirable characters, and duplicate records.

Regarding the **feature's data types**, our approach was to look for the unique values that a certain feature of a specific type took and then see if they were according to the variable description. For instance, some assumptions that we consider: binary variables that were stored as 'int64', should only take the values '0', '1', or 'Nan' (in the case of missing values); all the features related to dates should be in the same format 'Year-Month-Day'. Additionally, we focused on the ones with the 'object' type, because it may indicate that those features would need some kind of particular transformation or treatment to convert to known Python data types. For instance, *Gender* only took the values 'Male' and 'Female' and was later encoded into a binary variable; also, the variables related to date *(EnrollmentStart, EnrollmentFinish, LastPeriodStart, LastPeriodFinsh, DateLastVisit)* were converted into 'datetime'.

In terms of **missing values**, all the variables in that situation presented a percentage of missing data of less than 5%, so no critical cases were identified [Table 1 – Missing Values Initial Analysis].

Duplicate records, however, had one piece of evidence in favour, that in our perspective most likely corresponded to twins who happen to share the same information, only differing in their customer *ID*, so we decided to keep them.

For further investigation, we decided to do a **statistical analysis** of all our features. We focused on analysing the results of numeric features, specifically, the differences in the values of some key metrics such as *mean*, *median*, *max*, and *min* to identify potential outliers. Some critical cases were found, but the presence of outliers was only corroborated by looking at the respective histograms, which had skewed data, and the box plots in the next section.

2.2. VISUAL EXPLORATION

To structure and simplify our analyses, for the visual exploration section, we first classified our features as either metric or non-metric [Table 2 – Metric and Non-Metric features].

Immediately thereafter, we produced a set of **histograms** [Figure 1 – Numeric Variables´ Histograms Before Outlier Removal] and a set of **boxplots** [Figure 2 – Numeric Variables´ Box Plots Before Outlier Removal] for each **metric feature**, which allowed us to visually identify significant outliers in all of them. Some features that stood out, by having more than 10 % of observation out of the limits of the box plots were *AttendedClasses* and *DaysWithoutFrequency*. [Table 3 – %Outlier according to the observations out of the Box Plot´s limits]. Moreover, we highlighted the *NumberOfReferences,* because it was an unbalanced feature presenting 98% of the data with the same value, so it didn't have a high explanatory value for our data. Finally, we produced a ***Pearson*'s Correlation Matrix** [Figure 3 – Correlation Matrix], through which we concluded that the features *Age* and *Income*, and *LifetimeValue* and *NumberOfRenewals* were significantly positively correlated, with the values of 0.88, and 0.71, respectively.

Moving on to **non-metric features,** we drove our analysis by excluding the variables related to dates. Since all the remaining variables only took two values, we focused on analysing how **balanced the distribution within the categories** of a certain feature was. By looking at some bar charts [Figure 4 – Categorical Variables' Absolute Frequencies] we extracted some insights about the relevance of the features: we identified that all the customers in our dataset didn't take *NatureActivities* and *DanceActivities,* additionally, almost all (99%) didn't participate in *OtherActivities* and *AthleticsActivities.* Some other cases of really unbalanced data should have been taken in consideration, such as *SpecialActivities, RacketActivities* and *HasReferences.*

3. DATA PREPROCESSING

3.1. COHERENCE CHECKING

To guarantee the reliability of our data for subsequent analysis, we conducted a structured coherence analysis. A **general assumption** considered was we were dealing with a **Portuguese company**, which meant that in most of the cases, our approach was geared toward justifying the data's validity, based on that assumption, to retain as much information as possible, thus avoiding data removal and loss.

Diving into our analysis details, we started by looking at the features ***Income*** along with ***Age,*** assuming that it was an incoherence when a customer with an age below sixteen has a value of income different than zero, since that is the legal age to work in Portugal. We found 360 records that violated this assumption, including missing values, so we decided to replace those *Income* values by '0'.

Additionally, we looked into the features related to the **visits to the gym**. Firstly, we assumed that in the relation between ***AllowedWeeklyVisitsBySLA*** and ***AllowedNumberOfVisitsBySLA,*** we couldn't have records with missing values in just one of them. However, we found 535 records that violated that assumption, corresponding to the missing data in the *AllowedWeeklyVisitsBySLA*. To justify them, we tried to capture the importance of that variable, by seeing if there was some pattern of relation with *AllowedNumberOfVisitsBySLA.* We concluded that, even though we knew the existence of a positive correlation (=0.67), we could not find any relevant pattern. Moreover, we assumed that the number of ***AllowedWeeklyVisitsBySLA*** should be less or equal to ***AllowedNumberOfVisitsBySLA,*** however, we found 65 cases against it. In this case, we didn't consider incoherence because we thought that there was a possibility that our fitness facility offers a special event pass, such as a 'week-long fitness boot camp', where there are unlimited weekly visits exceeding the total. Finally, we assumed that the ***RealNumberOfVisits*** should be less or equal to ***AllowedNumberOfVisitsBySLA.*** To make valid comparisons, we converted all the variables to the integer type and decided to assume that this situation was possible in the case that a certain customer was enrolled in a *UseByTime* contract. Taking that into consideration, we found 41 incoherences, which we decided to maintain and just change the value of the variable *UseByTime* to '1'.

We also considered coherence concerning the **dates**, converting all features into 'datetime'. Initially, we examined the **relationship between enrolment dates**, expecting the start date to precede the finish date. While we did not encounter inconsistencies, we identified instances where enrolment dates were equal, but exclusively observed in cases where a customer renewed their contract (*NumberOfRenewals* > 0). From this observation, a **valuable insight** emerged: some customers maintain the same contract over time, where the most recent contract is the first one ever initiated. Conversely, other customers change contracts, resulting in the most recent contract having an earlier date than their initial contract. Besides that, we considered an incoherence when ***DateLastVisit*** was after the ***EnrollmentFinish*** and there were no renewals, which means that, in this case, the end of the contract should be the last date of interaction with a customer. In this scenario, no incoherence was found. Additionally, we verified if the ***DateLastVisit*** was ever before the ***LastPeriodFinish,*** which happened, but only under the fact that was before *EnrollmentFinish*, which was not an incoherence. We also verified the **temporal relationship** between the **dates of the periods *LastPeriodFinish*** should precede ***LastPeriodStart*** and that is always the case. Finally, we assumed that the enrolment dates were independent of the period dates. The period dates were related to the activities that the gym had going on and they were either semestral or annual. If a customer subscribed to an activity, the starting date and finish date were fixed for all the customers that subscribed to that same activity for that semester or year, independently of the date of enrolment of the customer. So, the **following situations are valid** for a certain record:

- *LastPeriodFinish* after *EnrollmentFinish,* means that it ends the contract before the activity ends, and then drops out or just keeps renewing its contract. The only situation where this might not happen is for the date on *EnrollmentFinish* = '2019-10-31', corresponding to the last registered date in the dataset.
- *LastPeriodStart* being before *EnrollmentStart* means that a customer starts in the gym after the beginning of the activity.

Concerning the feature ***Age*** some other analysis was considered. Firstly, we started by looking into the range of values that *Age* took identifying some strange values near the minimum of zero. This made us think about looking for the presence of children in the gym activities. According to a similar gym in Portugal and some research, we considered that all the activities didn't have an age minimum, except ***CombatActivities, FitnessActivities*** and ***TeamActivities*** which should start at the age of 7 years. All the records that violated this were deleted.

Besides that, we verified if the relationship between ***HasRefences*** and ***NumberOfRefernces*** was well coordinated. We found twelve records with a number of references bigger or equal to one, that had missing data in *HasRefences,* so we filled the missing data with the value '1'. Also, we identified that all the customers that didn't have references, had a number of references equal to '0'. However, we found 13 cases with zero in the number of references that didn't have the correct value in *HasRefences*, so we corrected the value on those cases.

Finally, by examining ***LifeTimeValue*** we found only 3 cases of customers who visited the gym briefly initially but then completely ceased their visits, suggesting potential discontinuation after a trial period due to extended periods without any visitation.

3.2. OUTLIER REMOVAL

After finishing the coherence checking, some features still presented a significant amount of outliers.

In that matter, we applied several methods to deal with the outlier removal, including: **Manually**, ***IQR Method***, and ***LOF***[1], remaining with approximately 89.47%, 51.96%, and 93.1% of the data, respectively. Not satisfied by any of the individual methods because all of them individually violate the rule of thumb of removing more than 3% of the data, we opted for removing the outliers using a method that **combined all above**. We ended up with 98,71% of the original observations being preserved.

Immediately after, we checked the boxplots once more [Figure 5– Numeric Variables´ Box Plots After Outlier Removal], and saw that they had far fewer data points after the lower and upper limits, meaning we were able to reduce the outliers in a significant way.

3.3. DATA NORMALIZATION

We followed this order of procedure because we considered using *KNNImputer* to fill missing data, which is a distance-based algorithm, so it is sensible to outliers and scale of the data. Taking that into account, we scaled our data using *MinMaxScaler* between zero and one.

3.4. FILLING THE MISSING VALUES

In the previous steps we conducted some data transformation in that sense there were significant differences in the number of features with missing data and in the percentage of that data for each feature [Table 4 – Missing Values Preprocessing]. However, we still had all the features with less than 5 % of missing data, and the features identified in that situation were still metric and non-metric, so our approach was to address the problem in distinct ways.

Starting with **metric features**, our strategy was to fill the missing data by the application of two methods and then compare the differences, ending by choosing just one. First, we try the ***KNNImputer***, a method that uses the *k-Nearest Neighbours* (five, in our case) to calculate a probable value for the data that is missing based on the most similar records [Figure 6 – Alterations in Metric Variables' Distributions After KNN Imputer]. Alternatively, we decided to fill with the **median**, which is less sensitive to outliers than the mean [Figure 7 – Alterations in Metric Variables' Distributions After Median Imputer]. Although in terms of comparison, the differences were not so significant, which made sense given the low percentage of missing data, we decided to go with the first approach to preserve as much as possible the original distribution of the variables.

Regarding the non-metric, before doing any imputation, we decided to take into consideration the insights about the relevance of most of these features and remove the irrelevant ones mentioned before: *NatureActivities, DanceActivities, OtherActivities, and AthleticsActivities.* After that, in the remaining ones, our strategy was similar in terms of comparing alternative methods and deciding by one. In this case, we use **mode imputation** versus ***KNNClassifier***, which corresponds to the use of a supervised learning algorithm, in this case, *k-Nearest Neighbours*, with just the metric features as explanatory and a certain non-metric feature with the missing value as a target. The idea was to train the model to predict the values of the target and fill those that were missing with the predictions. The results weren't so different, so we decided to go with the second approach, to try to balance the data as much as possible.

[1] The ***Local Outlier Factor (LOF)*** is an unsupervised method that spots anomalies by comparing a data point's local density with its neighbours. It identifies outliers as points with significantly lower density, resulting in a notable density variation. The density of each point is estimated by computing the distances of nearby points (*k-nearest neighbours*) and defining a metric called "reachability distance" to enhance stability within clusters.

3.5. Feature Engineering

Our original dataset had a lot of features in data types that were difficult to work with, for instance, the ones concerning the dates. So, in order to simplify our analysis and extract more valuable information from our data, we **created the following features**:

- ***annual_subscription*:** Binary variable that stores the type of subscription of a certain customer associated with the last activities or two months if less. It's based on the fact that *LastPeriodStart* is always at the beginning of a year or semester, January or July, and *LastPeriodFinish* is always at the end of a year or at the end of a semester, December or June. So, if the value is one, it means that the customers subscribe to an annual activity, otherwise, it means the semestral. After this we dropped the variables regarding the periods.
- ***new_in*:** Binary variable that identifies the new customers, which we considered to be the ones that enrolled in the last year registered of the database (2019) and didn't drop out. If the value is one, it means that the customer is a new one, otherwise, means that's not. We considered that we should recognize the insights extracted from the behaviour of old customers are, in many cases, different from the ones based on new customers.
- ***years_as_customer:*** Besides being able to disguise the new customers from the old customers, we should be able to analyse with more detail the different classes of customers according to the years they had been enrolled at the gym. This variable tries to capture the years that a specific customer has been with the gym.
- ***total_activities:*** Categorical variable that divides our customers into three categories based on the number of activities they enrolled in during the last recorded period, distinguishing between **'None'**(in the case of no activities enrolled), **'Single'** (in the case of just one activity), and **'Multiple'** (if it is more than one).
- ***other_activities:*** To simplify our analysis and in order to maintain all the remaining activities that were really unbalanced, we aggregated them into a binary variable that describes other activities.

3.6. Encoding

At this step of the process, we just needed to convert the variable *Gender* into binary. For that purpose, we consider that if the value 'Male' appeared we converted to '1' otherwise, '0'.

Moreover, we also encoded the variable *total_activities* into numeric values considering: 'None' as '0', ' Single' as '1' and 'Multiple' as '2'. After that, since it is a good practice, we scaled it as well.

3.7. Feature Selection

In this segment, we removed various variables that didn't serve our analytical goals and presented inconsistencies, specifically *AllowedWeeklyVisitsBySLA* and *AllowedNumberOfVisitsBySLA*.

Additionally, upon examining the corresponding boxplot, we identified *NumberOfReferences* as a variable with extremely limited explanatory power and variance, primarily due to 98% of its values being zero. As a result, we opted to remove it from our analysis.

3.8. Redoing Data Exploration

Once we completed all the pre-processing steps, in order to ensure that our data was appropriately primed for the modelling phase, we chose to execute the ***ProfileReport*** function, which provided us with a comprehensive summary of our data. Additionally, to strengthen our pursuit of potential inconsistencies, we did another ***Pearson*** **Correlation Matrix** for the metric features, incorporating the newly created variables [Figure 8 – Correlation Matrix with newly created variables].

For the non-metric features, we constructed a **Contingency Table** [Figure 9 –Cramér's V Correlation Matrix for Categorical Features].

As a result of such analysis, we decided to drop some **highly unbalanced features** that we considered not relevant for our analysis, *HasReferences.* Additionally, we addressed **highly correlated features** to prevent redundancy in our models. For instance, we observed strong correlations of 0.91 between *NumberOfRenewals* and *years_as_customer*, and 0.88 between *Age* and *Income*. We retained *years_as_customer* and *Age*, considering these variables as more meaningful for our analysis. Specifically, in the case of *Age* versus *Income*, *Age* didn't exhibit artificially induced values.

4. MODELLING

4.1. SEGMENTATION

A very important step for cluster analysis is creating multiple segmentations to guarantee its quality. So, we chose multiple variables and created **two segmentations**:

- **Service Usage:** Evaluates the usage of the gym's services, by a certain customer. Usage means the total value expended, the total number of visits and activities enrolled. For this segmentation, we decided to use both categorical variables, binary (*WaterActivities*, *FitnessActivities*, *other_activities*) and ordinal (*total_activities);* and also metric variables *(LifetimeValue* and *NumberOfFrequencies)*.
- **Engagement Level**: Evaluates the customer's engagement with the gym, taking in consideration its recently activity in the facilities. We used only numerical variables (*AttendedClasses*, *RealNumberOfVisits,* and *DaysWithoutFrequency)*.

Before further analysis, it is important to mention that we systematically explored numerous variable combinations for these segmentations, ultimately deriving multiple results based on what yielded the best outcomes. Throughout our trials, we were mindful to avoid incorporating highly correlated variables to prevent redundancy. Additionally, we excluded variables showcasing extreme values — whether extremely low or high — as these could either minimally impact the results or introduce bias into our analysis. Despite identifying a correlation of 0.62 between *WaterActivities* and *FitnessActivities*, we opted to retain both variables. Our decision was informed by the observation that Service Usage remained unbiased despite this correlation.

4.2. SERVICE USAGE

Given the types of variables in the **Service Usage** segmentation, we decided to use only one model, ***K-Prototypes***[2], that combines *K-Means* and *K-Modes*, providing an algorithm for clustering data that contains both numerical and categorical features.

In our implementation, we started by dividing the numerical and the binary features and making a graph for the optimal number of clusters with *K-Prototypes*. By applying the ***Elbow Method***, we decided on three clusters. Then we characterized the clusters in terms of the mode for each binary feature and basic statistics for the numerical features. Finally, we created a ***t-SNE*** to check the status of our clusters [Figure 10 –Service Usage: t-SNE].

[2] ***K-Prototypes*** combines *K-Means* for numerical data with *K-Modes* (a clustering algorithm for categorical data) to handle mixed data types. It defines a dissimilarity measure that considers both numerical (*Euclidean* distance) and categorical (matching dissimilarity) features when clustering, and minimizes a cost function that considers the sum of dissimilarities for both types.

4.3. Engagement Level

For the **Engagement Level** segmentation, we decided to test multiple models such as ***Hierarchical Clustering***, ***K-Means***, ***Mean-Shift***, ***DBSCAN***, ***Gaussian Mixture Model (GMM)*** and ***Self-Organizing Maps (SOM)***. We kept track of the ***R-squared*** and ***silhouette scores*** for each model, so we could choose the best one to merge in the end with the Service Usage segmentation for the cluster analysis.

4.3.1. Hierarchical Clustering

In ***Agglomerative Hierarchical Clustering Methods***, each data point is considered a single cluster. Then, it's calculated the pairwise distances between all clusters merging the two closest based on a distance matrix, and so on repeating these same steps until there is only one cluster left. To calculate these distances, it's needed to choose a linkage method. We calculated a *R-squared* plot for each method and concluded the best one was always ***Ward***, independently of the number of clusters we'd want.

After that, we created a **dendrogram** that illustrated the merging process and the relationships between clusters. Upon further inspection, we determined the best number of clusters varied between five and eight, and calculated the ***R-squared*** and ***silhouette scores*** for each number. We chose eight clusters for this model, prioritizing the *R-square* score.

4.3.2. K-Means Clustering

In this algorithm, the dataset is divided into 'K' distinct clusters while minimizing the ***inertia*** for each number. We used 'k-means++' to calculate the centroids of each cluster since it's a smarter seed initialization than 'k-means'. It uses random points of the dataset to be centroids, instead of using random points of the space.

We calculated and plotted the *inertia* score for different numbers of clusters, and used the ***Elbow Method*** to determine the best possible number, which ended up being either three or four.

In order to complement the analysis made with the *inertia*, we calculated and plotted the ***silhouette score*** for different numbers of clusters. We know that a high *silhouette score* means the point is well-matched to its assigned cluster. With this in mind and the ***R-squared*** values, we decided the best number of clusters would be seven for this method, even though the *inertia* recommended three or four.

4.3.3. Mean-shift Clustering

Unlike *K-means*, **Mean-shift** doesn't require the number of clusters beforehand, but it does require a specification of ***bandwidth***, the size of a window around each point. This algorithm iteratively shifts points towards the mode of the data distribution and continues until no further shifts change the cluster assignments significantly.

We started by estimating the *bandwidth*, while adjusting the ***quantile*** parameter since it impacts the number and sizes of clusters generated by the *Mean Shift* algorithm, and the number estimated of clusters was six. Then again, we calculated the ***R-squared*** and ***silhouette scores***.

4.3.4. DBSCAN Clustering

DBSCAN stands for ***Density-Based Spatial Clustering of Applications with Noise***, meaning it's a density-based algorithm. It groups together data points that are close to each other in the feature space and separates regions of lower point density. Just as *Mean-shift*, *DBSCAN* also doesn't require the number of clusters beforehand.

The most important parameters of this algorithm are the epsilon (***eps***) – the maximum distance between two points to be considered neighbours and the minimum points (***min_samples***) – the threshold to form a dense region.

After fine-tuning those parameters and with the help of a ***K-distance graph*** to find out the right *eps* value, the number estimated of clusters was seven.

4.3.5. GAUSSIAN MIXTURE MODEL (GMM)

GMM assumes the existence of various *Gaussian distributions*, where each represents a different cluster. By plotting the ***Akaike Information Criterion (AIC)*** and the ***Bayesian Information Criterion (BIC)***, which are measures of model performance that account for model complexity, we were able to determine the ***number of components***, which is a parameter equivalent to the number of clusters, and then use that to get our clusters. We tested the ***R-squared*** and ***silhouette score*** for four, six and seven clusters, deciding that seven was the better option.

4.3.6. SELF-ORGANIZING MAPS (SOM)

SOM are unsupervised *neural networks* that can be used for producing a low-dimensional representation of input data in the form of a grid. The data points are represented by *nodes* and the map is initialised with a random set of connections, that are adjusted with the *learning rate* and updated through training.

We initialised a 20x12 **grid**, meaning a 240-unit **map**, and after training, we plotted the ***Component Planes***, ***U-Matrix,*** and ***Hit-Map*** to gain some information on how *SOM* was adapting to the data. For instance, in the *U-Matrix*, we were capable of having an idea of the clusters that could be formed and, in the *U-Map* we could detect a potential outlier, a *neuron* whose average distance from the others in the input space was really high, being coloured in red.

1) HIERARCHICAL CLUSTERING OVER SOM

   Once our *SOM* was defined, and its respective units, we applied the ***Hierarchical Clustering* over** them, repeating the steps we had done before in *Hierarchical Clustering*, but we passed the nodes of the SOM matrix instead. We were finally able to visualise the clusters we had created. It is important to note that this approach combines the strengths of both *Hierarchical Clustering* and *SOMs*, but it also has some limitations. The resulting clusters may be difficult to interpret, as they are based on the *SOM* units rather than the original data points, and the clusters produced by the *Hierarchical Clustering* algorithm may not align well with the topological structure of the *SOM* map.

2) K-MEANS CLUSTERING OVER SOM

   The principle of ***K-Means* over *SOM*** is the same as the one applied when doing *Hierarchical Clustering* over *SOM*.

4.3.7. METHOD CHOICE

After analysing all these methods and recording their ***R-squared*** and ***silhouette scores*** [Table 5 – Engagement Level: Comparing All the Methods], we concluded that the most effective methods were *Hierarchical Clustering*, *K-means*, *Mean-shift*, and *K-Means* over *SOM*. Consequently, we excluded *DBSCAN* due to its low *R-square* value, and *GMM* due to its low *silhouette score*. Additionally, we disregarded *Hierarchical* over *SOM* because it demonstrated an average performance in both metrics.

To further analyse these four methods, we plotted their ***t-SNE*** graphs and concluded that ***K-means*** was the best one, but instead of using seven clusters, we decided to use four, since it had a better *silhouette score* and it was easy to interpret [Figure 11 –Engagement Level: t-SNE (K-means)].

5. Merging the Perspectives

In order to merge both segmentations, we used the ***Hierarchical Clustering*** method so we could get a better understanding of our data, as well as fewer clusters with a significant number in them [Figure 12 – Merging Perspectives Hierarchical Clustering: Ward's Dendrogram]. The ideal number of clusters should be either four, five or six, and after some analysis, we concluded that four was the ideal.

6. Cluster Analysis

Finally, after getting the final clusters of the merged segmentations, we could analyze the clusters of the individual segmentations, as well as the merged segmentations. In order to do that we used **line graphs** to understand the mean distribution of each cluster observations across the different metric variables used for the separation, and also **bar charts** to characterized the population per cluster [Figure 13 – Clustering Profiling].

In the **Service User Clustering**, there was a balanced distribution among the clusters, with a strong abundance in Cluster 0. On the other hand, **Engagement Level Clustering** had a non-uniform distribution, with Cluster 1 concentrating most results with distinct dominant characteristics in each cluster, likely due to a significant bias. Finally, the **Merge Segmentation** aligned with the distribution of Engagement level, as depicted in both the bar chart and line graph. The dominance of the same features persisted, with Cluster 1 once again emerging as the most populated one.

For a two-dimensional visualization of cluster distribution, we used ***t-SNE*** [Figure 14 – Merging Perspectives: t-SNE] The results aligned with previous analyses, showing overlap between clusters.

7. Feature Importance

To comprehend the structure of our clusters, we studied the **importance of each feature** i.e., we aimed to assign a value that reflects the impact of each feature in defining the clusters. In general, features with higher importance contribute with more information about the clusters and may be more useful for understanding the underlying structure of the data.

Therefore, our initial step was to calculate the ***R-square*** for each feature, where values closer to one indicate greater importance. From this analysis, we found that the variables *AttendantClasses, DaysWithoutFrequency* and *RealNumberOfVisits* are the most important.

Next, we used a **Decision Tree** [Figure 15 – Feature Importance: Decision Tree]to identify the initial features used for classification since these are considered as the most important. Once again, the same three variables surfaced, although in a different sequence. This approach achieved an outstanding precision rate of 99.02%, which is pretty good.

8. Marketing Plan[3]

According to the principles of marketing, we are able to identify **Customer Relationship Groups** based on Potential Profitability and Projected Loyalty [Figure 16– Customer Groups]. Therefore, we will consider these groups to analyse our final four clusters, resulting from merging the two perspectives, aiming to propose marketing actions tailored to better satisfy each distinct group.

Starting with **Cluster 2**, with 275 observations, it represented the **'True Friends'** category, marked by both high loyalty and profitability for the gym. Their loyalty was evident through prolonged gym memberships, indicated by the highest mean value of *year_as_customer*. These customers were

[3] Note: We focus our analysis in these values [Table 6 – Merging Perspectives: Relevant Feature Values]

exclusively long-term members, *new_in* is always zero, and exhibit minimal dropout rates. Additionally, they demonstrated high attendance in classes and frequent gym visits. In terms of profitability, they showcased higher lifetime values. Their service usage pattern distinguishes them as the cluster with greater participation in water activities, fewer engagements in fitness activities, and a higher percentage of semestral subscriptions.

Our **marketing approach** for this cluster was based on the idea that, given their high-value and loyal customer base, leveraging their engagement could enhance the gym's less frequented activities. We considered offering a campaign reward—allowing additional hours for free in other activities—for the time spent visiting the gym.

Following to **Cluster 3**, with 778 observations, we categorized it as the **'Butterflies'**. It presented the highest percentage of new customers with the most substantial average visits to the facilities during the last period (high profit), yet it also exhibited a relatively high percentage of dropouts (low loyalty). In terms of service usage, it had the highest percentage of engagement in fitness activities among all clusters and was enrolled in a time-based package.

To prevent these 'Butterflies' from flying away, our **strategy** involved focusing on activities that contribute most to satisfaction and retention, namely water-related activities. We aimed to offer new customers a discounted package if they engaged in fitness activities alongside water activities, thus avoiding additional charges for using the pool facilities.

Regarding **Cluster 0** (2158 instances) it comprised old customers who haven't recently engaged in gym activities and were not yielding the expected profit considering their membership duration—they were categorized as **'Barnacles**.' This group exhibited the highest number of days without frequency, engages the most in *other_activities*, and was also the only cluster with annual memberships that did not utilize the time-based package. This insight revealed that these activities tend to be more long-term, leading to customer demotivation. To re-engage them, we considered introducing semestral plans for these activities and incentivizing their participation by rewarding them with free time to utilize the pool facilities.

Finally, **Cluster 1**, with 11,522 observations, represented the **'Strangers**.' It was the most challenging to classify, but we observed the highest percentage of dropouts within this cluster. To address this, we considered promoting free companionship for these customers—either through training sessions with personal trainers or by offering a monthly opportunity to bring a friend along to keep them motivated.

9. Conclusion

Through the development of this project, we were able to both consolidate the materials learned during the Data Mining course, and also apply knowledge obtained through self-study.

We encountered obstacles that hindered the project's progress and the formulation of conclusions, such as the absence of external information regarding the case study, which took a significant influence on the construction of clusters, as evidenced in our examination. Moreover, the dataset itself was diverse and complex, requiring careful treatment for information extraction.

For future analyses, we suggest testing alternative data scaling methods, such as *Standard Scaling*, to gain a better understanding of their impact on cluster construction. Additionally, exploring alternatives to certain assumptions made during coherence analysis could potentially lead to different outcomes.

Furthermore, our lack of prior experience presented a considerable challenge. This project marked our initial foray into demanding cluster analysis projects. Despite these hurdles, we derived valuable insights.

10. References

(1) McKinsey & Company (2023). What Is Personalization? [Online] Available at: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-personalization [19/12/2023].

(2) Estádio Universitário de Lisboa. Descubra os Ginásios do Estádio Universitário. [Online] Available at: https://www.estadio.ulisboa.pt/noticia/descubra-os-ginasios-do-estadio-universitario [19/12/2023].

(3) scikit-learn. n.d. LocalOutlierFactor. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html [06/01/2024].

(4) k-prototypes. n.d. API Reference. [online] Available at: https://kprototypes.readthedocs.io/en/latest/api.html [06/01/2024].

(5) Kotler, P., & Armstrong, G. (Year). Principles of Marketing (18th ed.). Pearson.

11. Appendix

11.1. Figures

*Figure 1 – **Numeric Variables´ Histograms Before Outlier Removal***

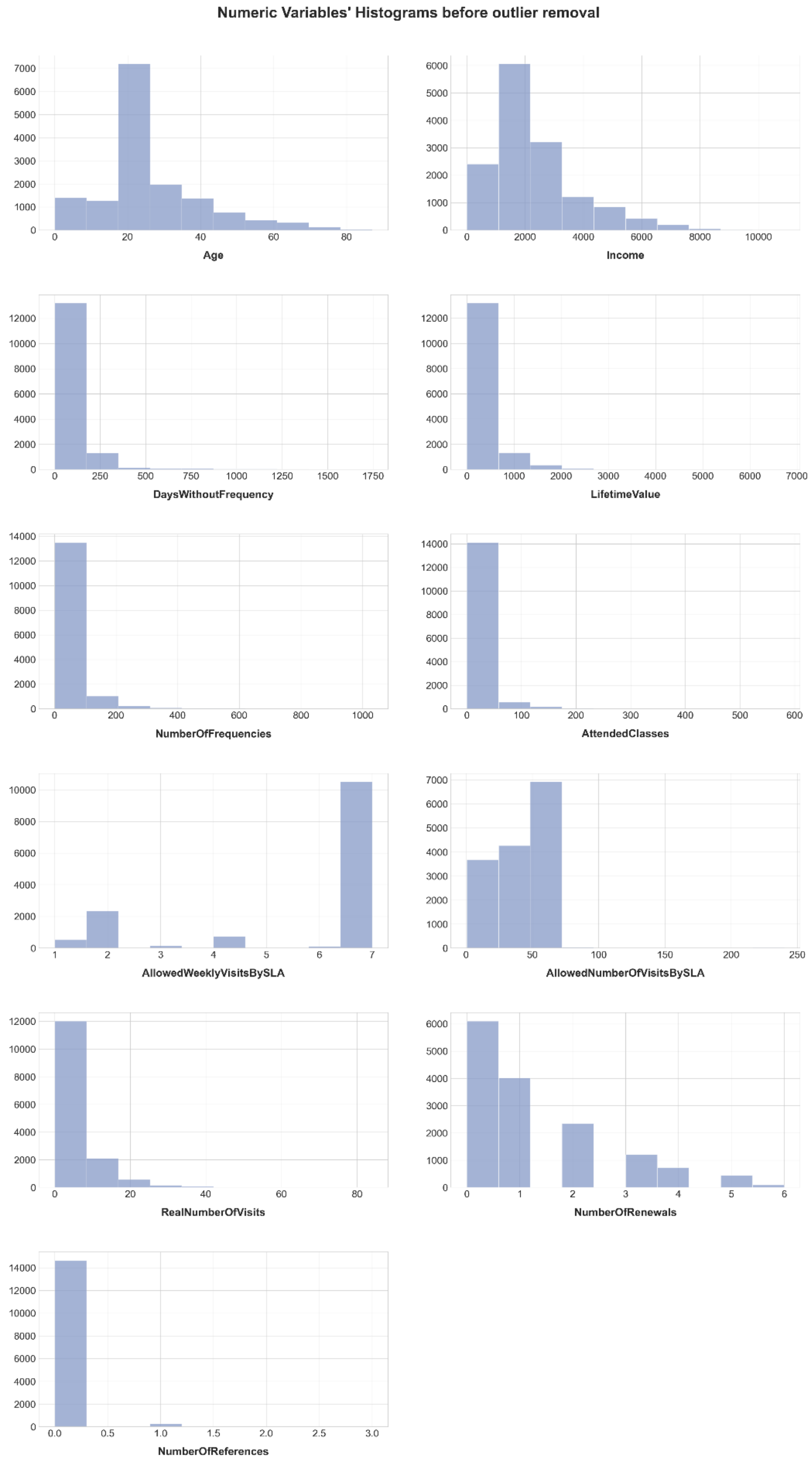


Figure 2 – ***Numeric Variables´ Box Plots Before Outlier Removal***

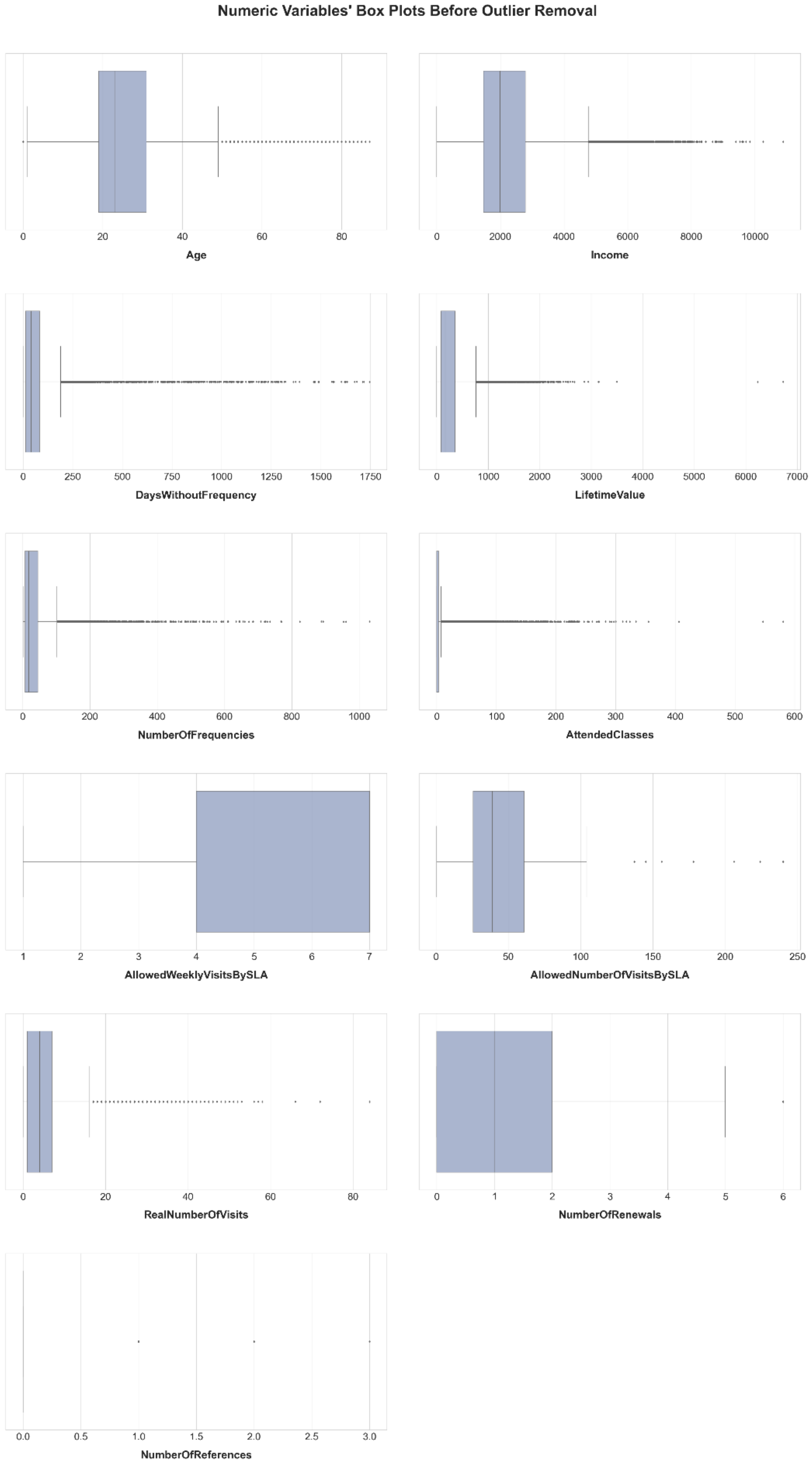


*Figure 3 – **Correlation Matrix***

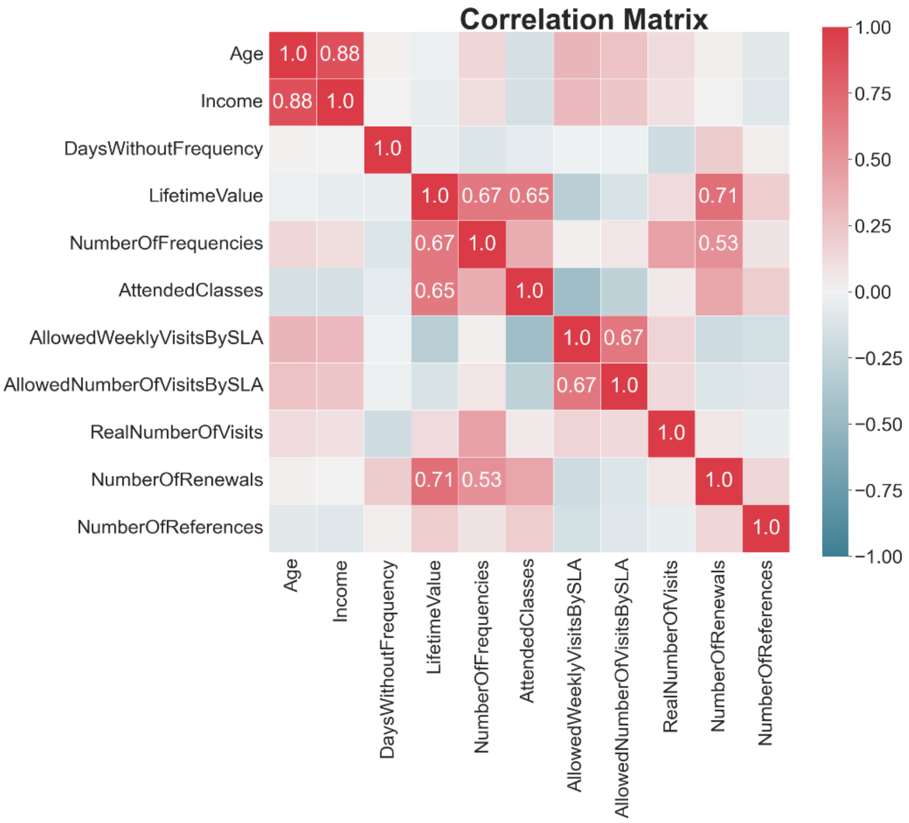


Figure 4 – ***Categorical Variable's Absolute Frequencies***

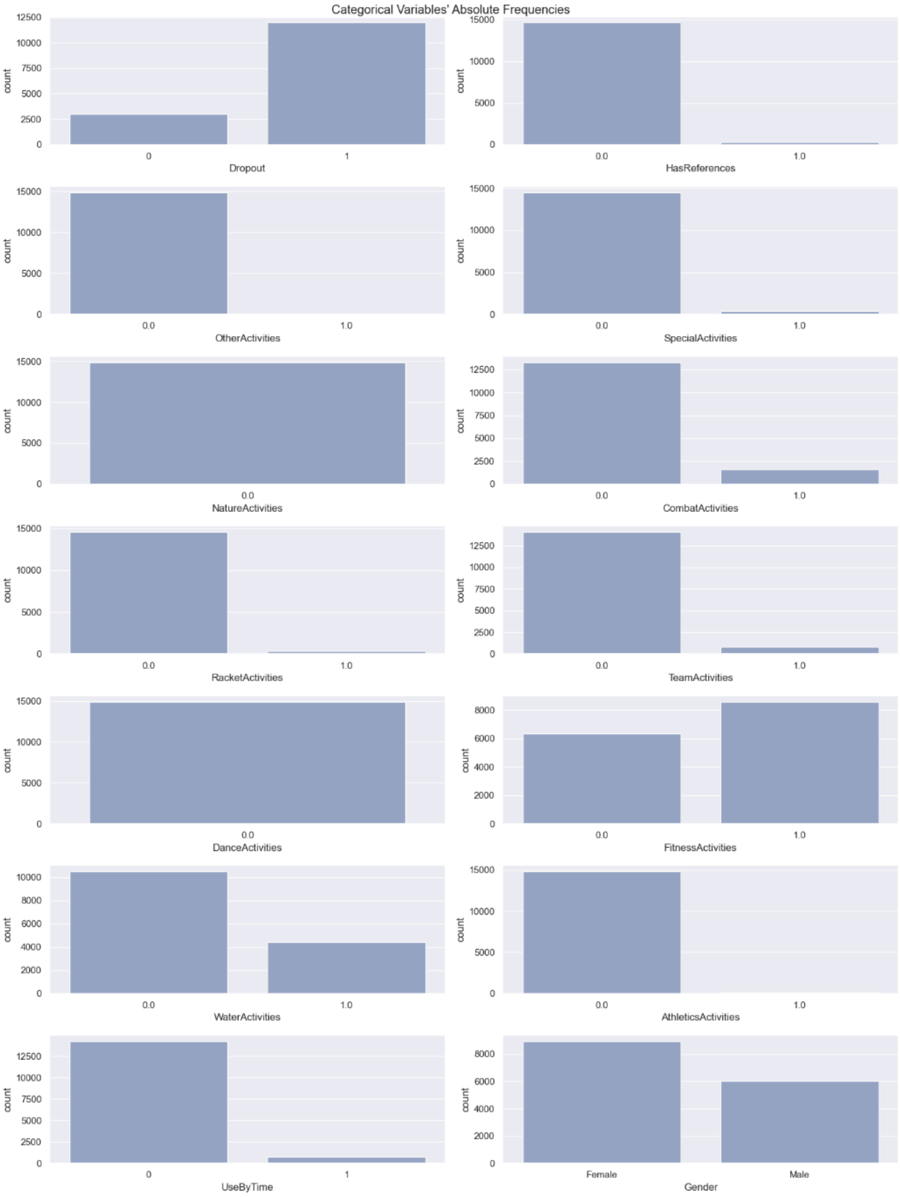


*Figure 5 – **Numeric Variables´ Box Plots After Outlier Removal***

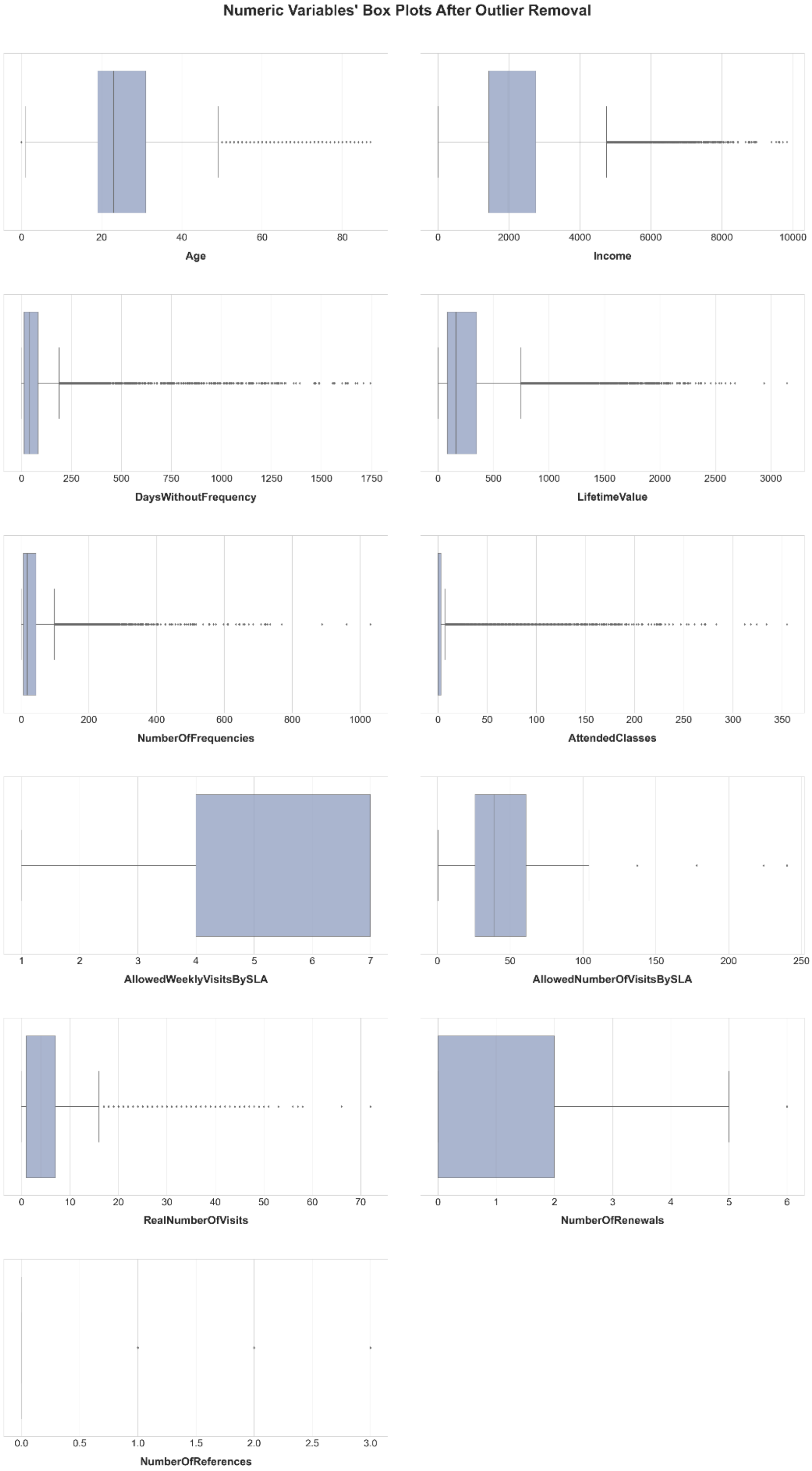


Figure 6 – ***Alterations in Metric Variables' Distributions After KNN Imputer***

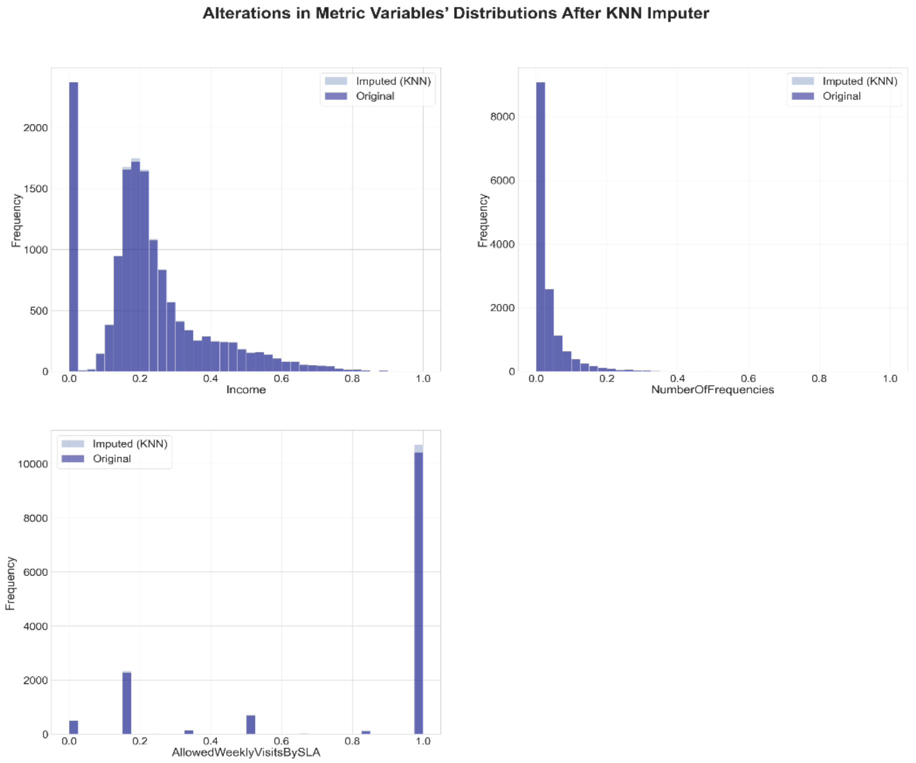


Figure 7 – ***Alterations in Metric Variables' Distributions After Median Imputer***

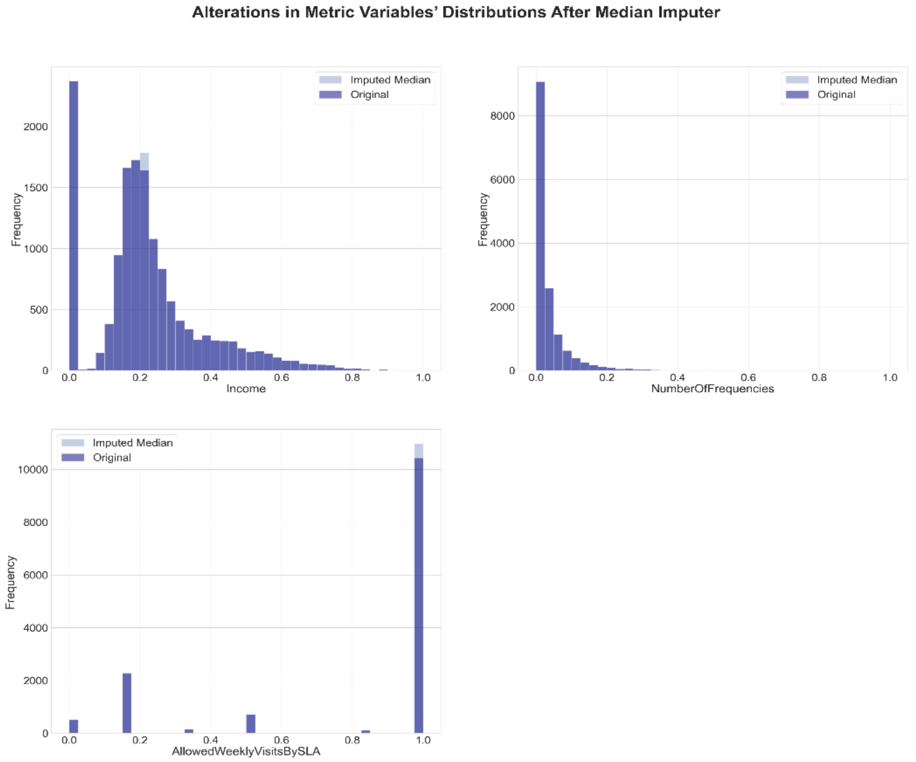


*Figure 8 –**Correlation Matrix with newly created variables***

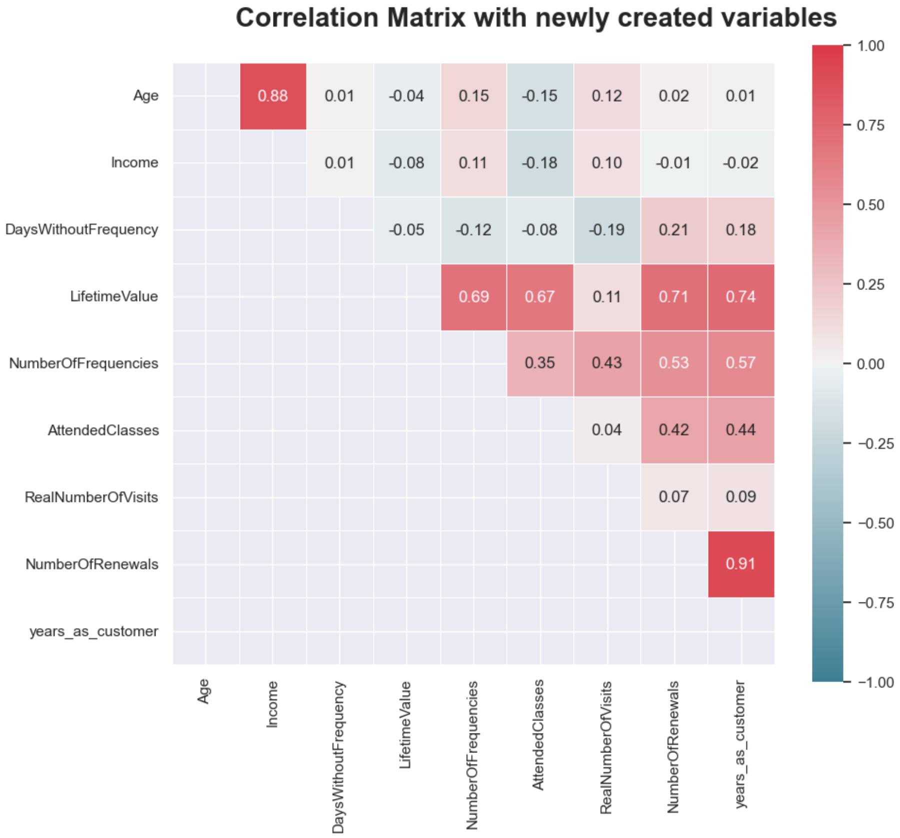


*Figure 9 –**Cramér’s V Correlation Matrix for Categorical Features***

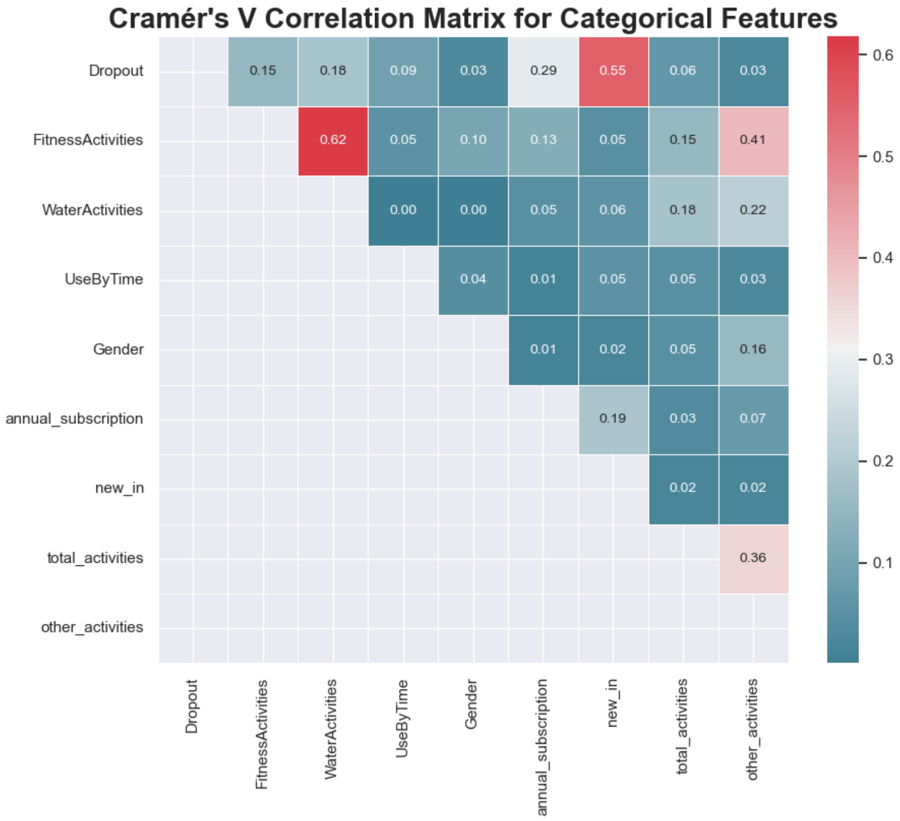


*Figure 10 –**Service Usage: t-SNE***

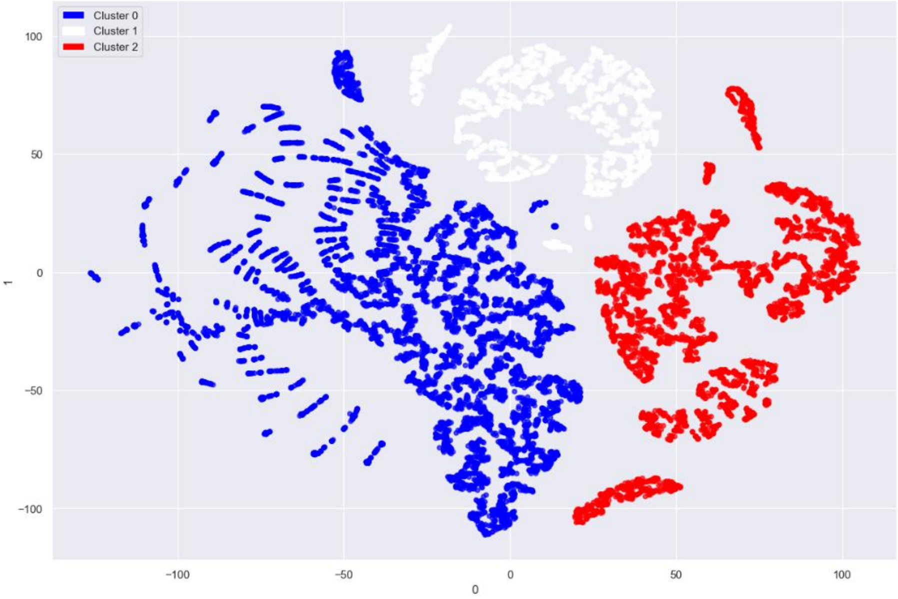


*Figure 11 –**Engagement Level: t-SNE (K-means)***

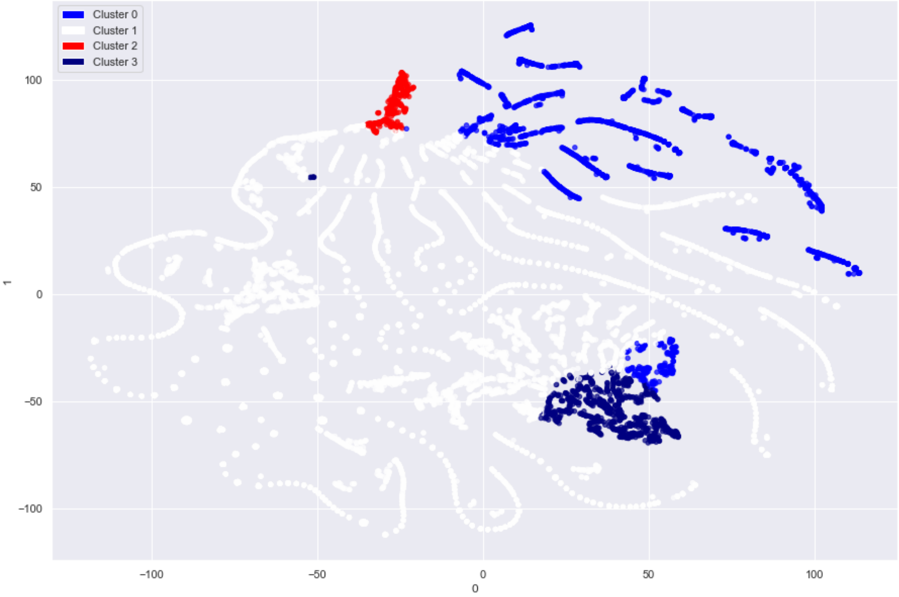


*Figure 12 – **Merging Perspectives Hierarchical Clustering: Ward's Dendrogram***

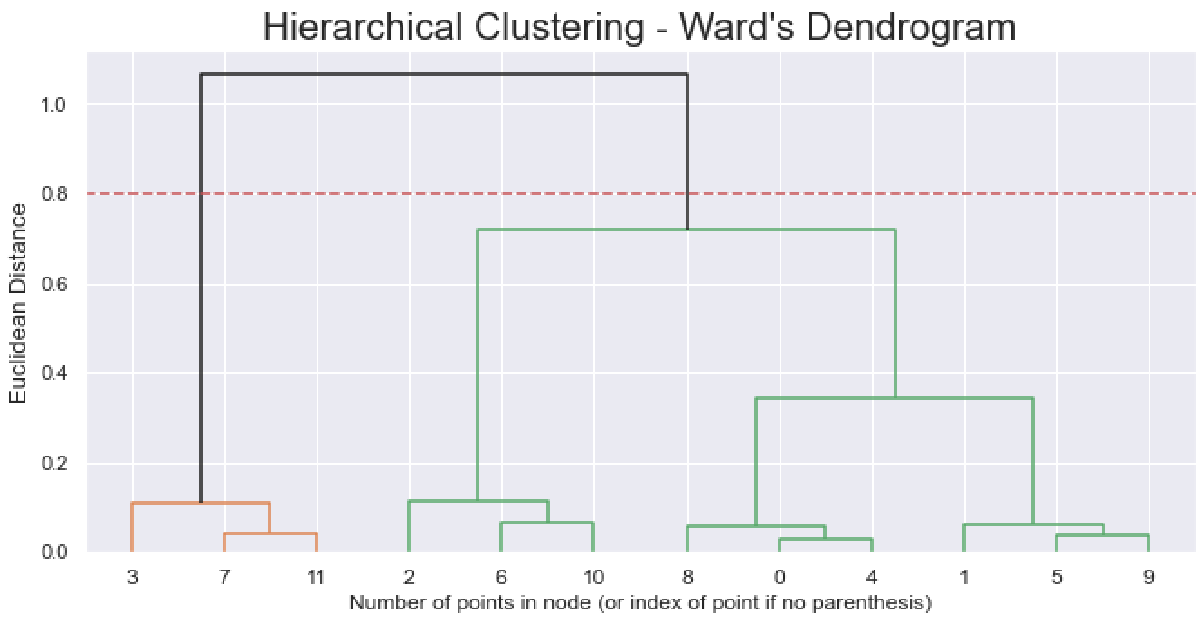


*Figure 13 – **Clustering Profiling***

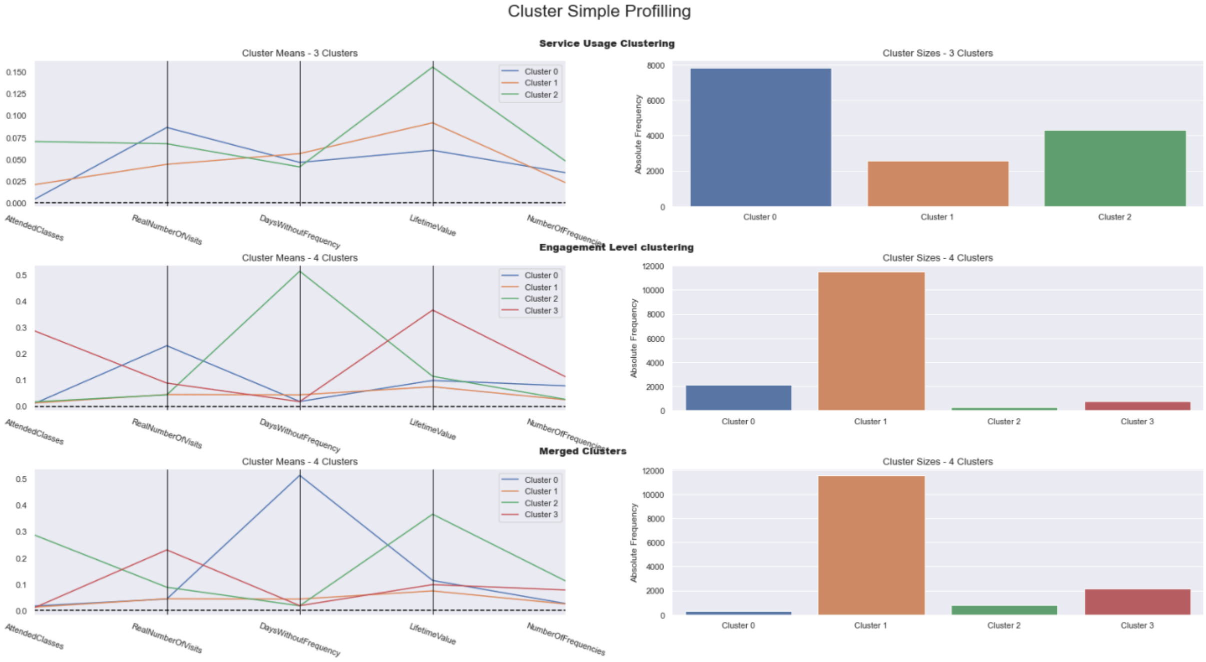


Figure 14 – ***Merging Perspectives: t-SNE***

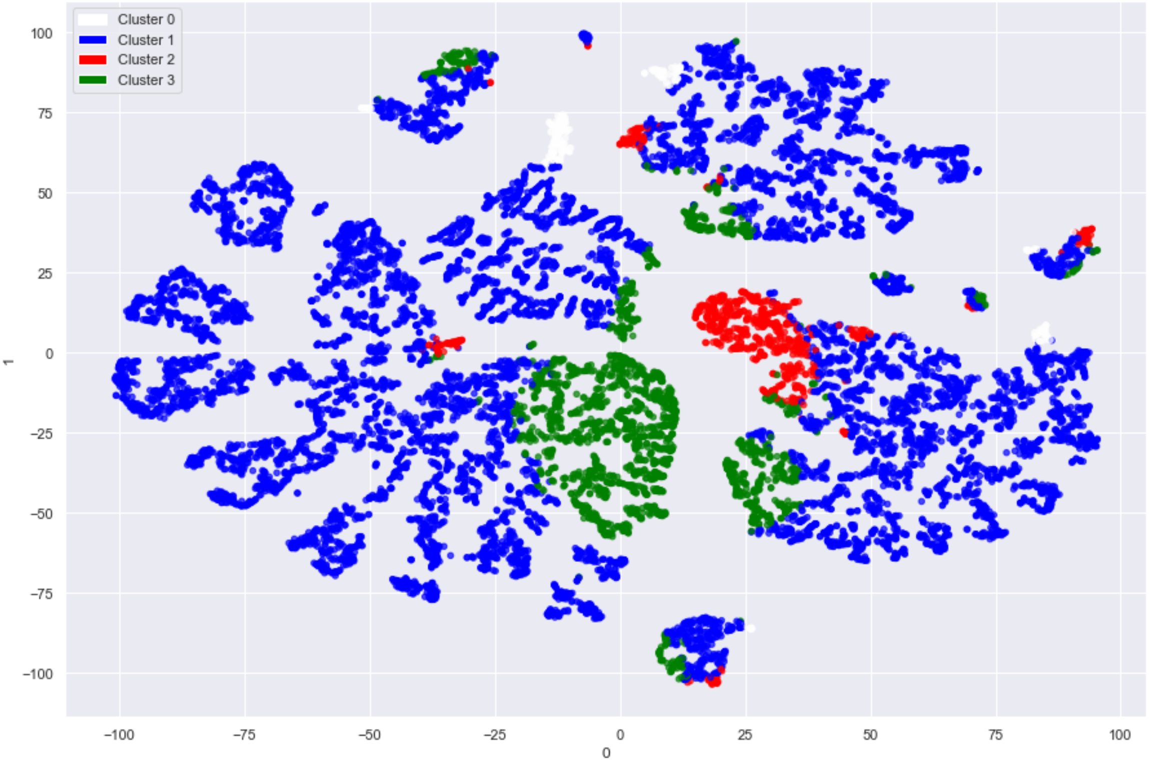


Figure 15 – ***Feature Importance: Decision Tree***

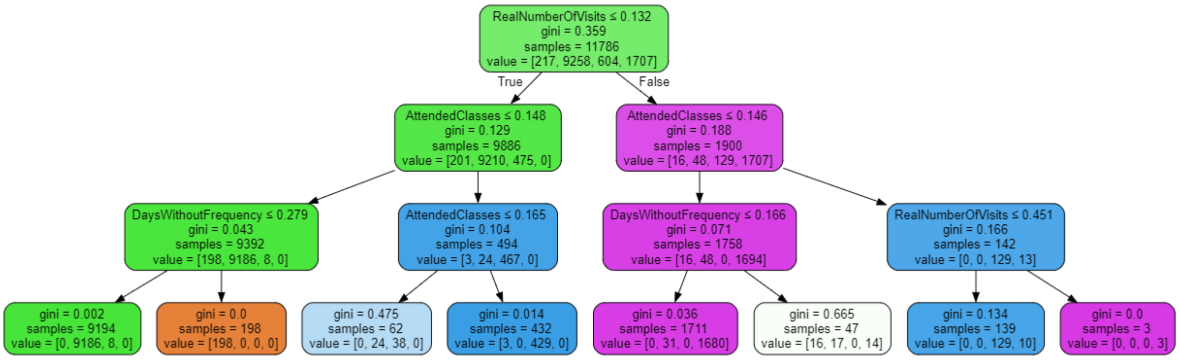
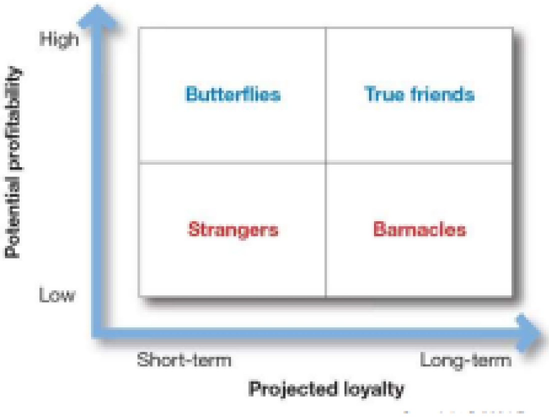
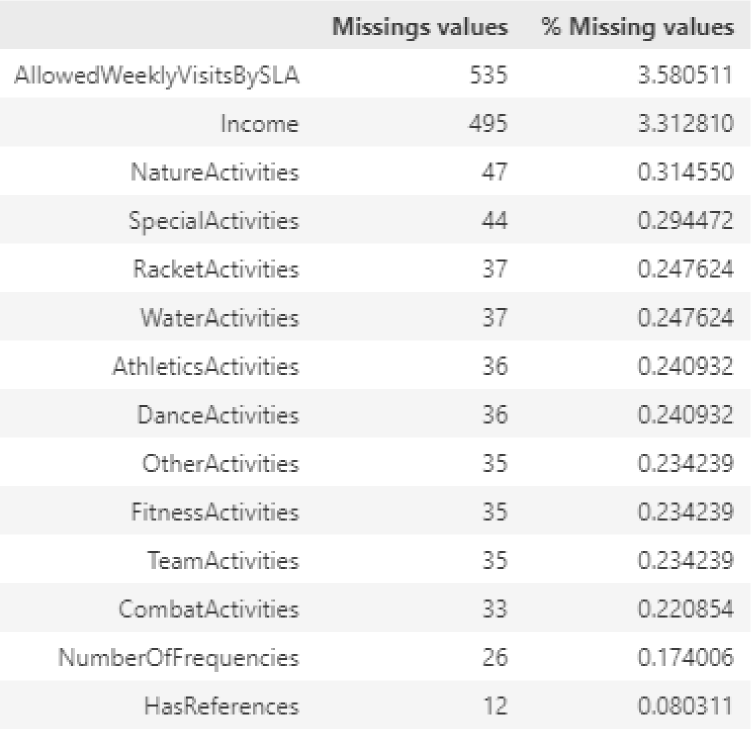


Figure 16– ***Customer Groups***



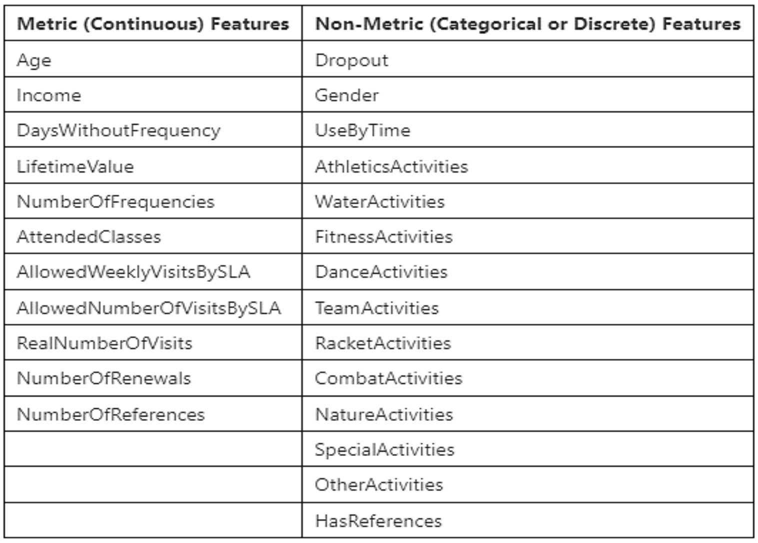
11.2. TABLES

*Table 1 – **Missing Values Initial Analysis***



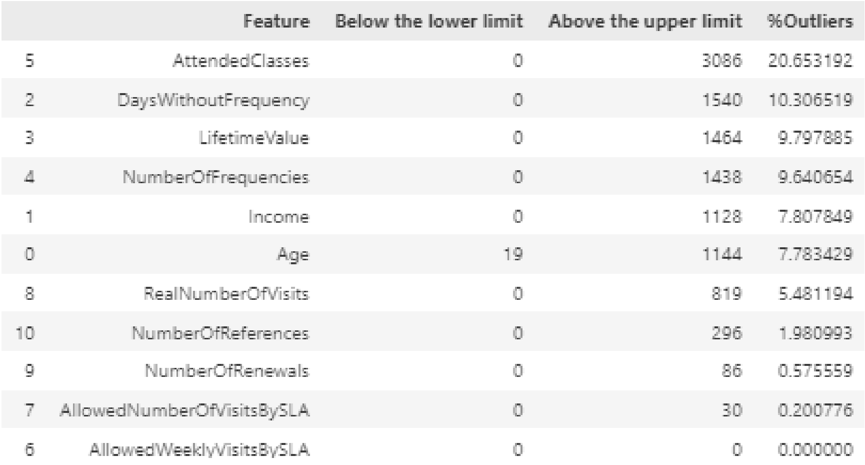
	Missings values	% Missing values
AllowedWeeklyVisitsBySLA	535	3.580511
Income	495	3.312810
NatureActivities	47	0.314550
SpecialActivities	44	0.294472
RacketActivities	37	0.247624
WaterActivities	37	0.247624
AthleticsActivities	36	0.240932
DanceActivities	36	0.240932
OtherActivities	35	0.234239
FitnessActivities	35	0.234239
TeamActivities	35	0.234239
CombatActivities	33	0.220854
NumberOfFrequencies	26	0.174006
HasReferences	12	0.080311

*Table 2 – **Metric and Non-Metric Features***



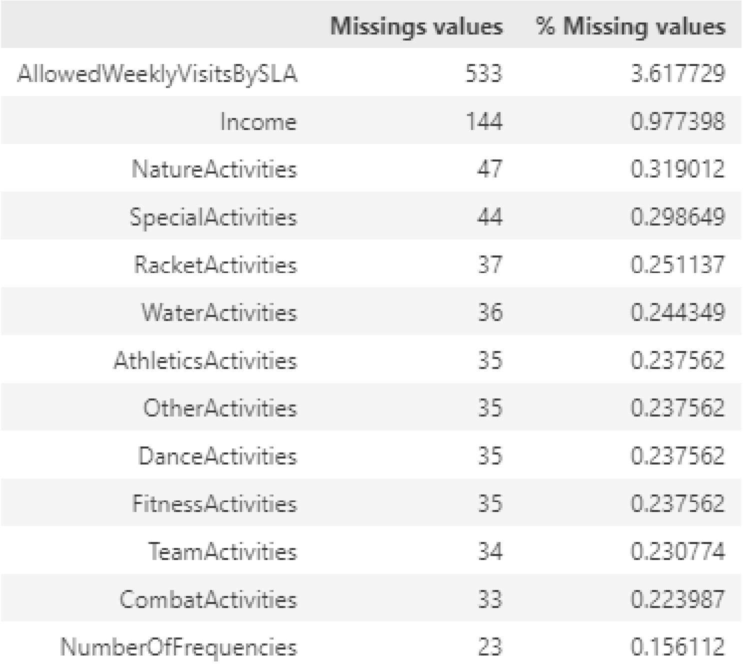
Metric (Continuous) Features	Non-Metric (Categorical or Discrete) Features
Age	Dropout
Income	Gender
DaysWithoutFrequency	UseByTime
LifetimeValue	AthleticsActivities
NumberOfFrequencies	WaterActivities
AttendedClasses	FitnessActivities
AllowedWeeklyVisitsBySLA	DanceActivities
AllowedNumberOfVisitsBySLA	TeamActivities
RealNumberOfVisits	RacketActivities
NumberOfRenewals	CombatActivities
NumberOfReferences	NatureActivities
	SpecialActivities
	OtherActivities
	HasReferences

*Table 3 – **%Outlier according to the observations out of the Box Plot´s limits***



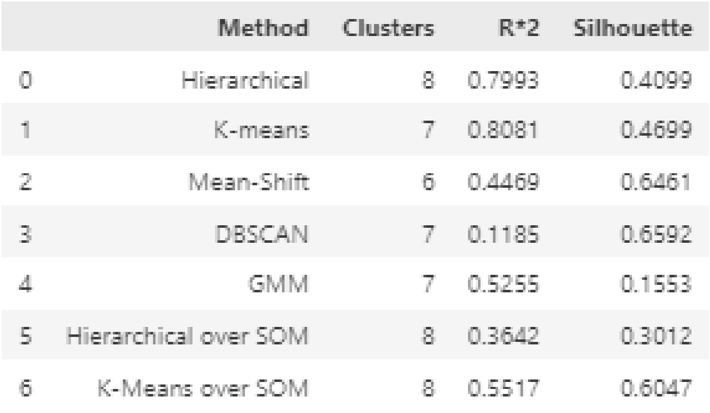
	Feature	Below the lower limit	Above the upper limit	%Outliers
5	AttendedClasses	0	3086	20.653192
2	DaysWithoutFrequency	0	1540	10.306519
3	LifetimeValue	0	1464	9.797885
4	NumberOfFrequencies	0	1438	9.640654
1	Income	0	1128	7.807849
0	Age	19	1144	7.783429
8	RealNumberOfVisits	0	819	5.481194
10	NumberOfReferences	0	296	1.980993
9	NumberOfRenewals	0	86	0.575559
7	AllowedNumberOfVisitsBySLA	0	30	0.200776
6	AllowedWeeklyVisitsBySLA	0	0	0.000000

*Table 4 – **Missing Values Preprocessing***



	Missings values	% Missing values
AllowedWeeklyVisitsBySLA	533	3.617729
Income	144	0.977398
NatureActivities	47	0.319012
SpecialActivities	44	0.298649
RacketActivities	37	0.251137
WaterActivities	36	0.244349
AthleticsActivities	35	0.237562
OtherActivities	35	0.237562
DanceActivities	35	0.237562
FitnessActivities	35	0.237562
TeamActivities	34	0.230774
CombatActivities	33	0.223987
NumberOfFrequencies	23	0.156112

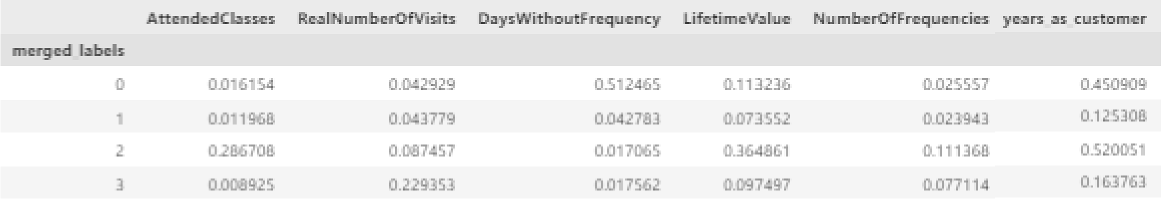
*Table 5 – **Engagement Level: Comparing All the Methods***



	Method	Clusters	R^2	Silhouette
0	Hierarchical	8	0.7993	0.4099
1	K-means	7	0.8081	0.4699
2	Mean-Shift	6	0.4469	0.6461
3	DBSCAN	7	0.1185	0.6592
4	GMM	7	0.5255	0.1553
5	Hierarchical over SOM	8	0.3642	0.3012
6	K-Means over SOM	8	0.5517	0.6047

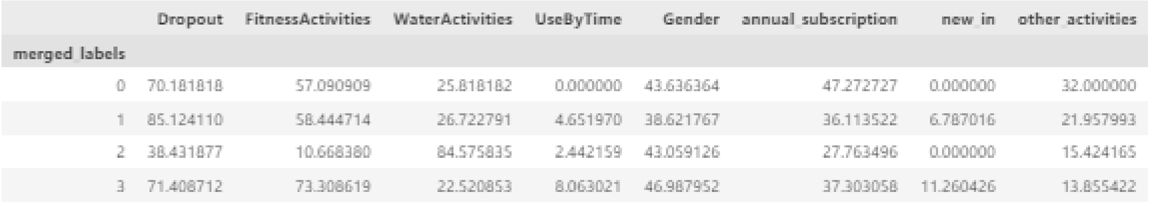
Table 6 – ***Merging Perspectives: Relevant Feature Values***

Metric



	AttendedClasses	RealNumberOfVisits	DaysWithoutFrequency	LifetimeValue	NumberOfFrequencies	years_as_customer
merged_labels						
0	0.016154	0.042929	0.512465	0.113236	0.025557	0.450909
1	0.011968	0.043779	0.042783	0.073552	0.023943	0.125308
2	0.286708	0.087457	0.017065	0.364861	0.111368	0.520051
3	0.008925	0.229353	0.017562	0.097497	0.077114	0.163763

Non-Metric



	Dropout	FitnessActivities	WaterActivities	UseByTime	Gender	annual_subscription	new_in	other_activities
merged_labels								
0	70.181818	57.090909	25.818182	0.000000	43.636364	47.272727	0.000000	32.000000
1	85.124110	58.444714	26.722791	4.651970	38.621767	36.113522	6.787016	21.957993
2	38.431877	10.668380	84.575835	2.442159	43.059126	27.763496	0.000000	15.424165
3	71.408712	73.308619	22.520853	8.063021	46.987952	37.303058	11.260426	13.855422