Data Mining Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

Customer Segmentation strategy for Enhanced Marketing and Engagement

<Group 88>

<Jaime Simoes> Number <20230522>

<Maximilian Laechelin> Number <20230979>

<Ilyass Jannah> Number <20230598>

<December>, <2023>

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

INDEX

1. Introduction

XYZ Sports Company wants to set up advertisements that are more exciting and match the special needs of many users. With time, the company learned a lot. It understood that it's very important to know who their customers are. For this, the company began a big and total customer grouping plan. They used their many details about customers from their ERP system to do so. The information starts from June 1, 2014 and ends on October 31, 2019. It helps to serve the customers better by really getting to know how they act and what they like.

The main objective of this project is to make segments of customers and provide valuable insights by applying different techniques. By sorting customers into simple groups, XYZ wants to get better at giving personalized services. This will also help them make good decisions on marketing activities and grow bigger in the fitness business. This program can be very good for the company. It helps them understand how much each customer is worth, but it needs to be done well.

Embracing market segmentation as a guiding principle, this project employs several clustering techniques and algorithms (K-Means, Heat-Maps, Hierarchical clustering, etc.) to help identify the customer segments. But just knowing the clusters isn't enough. We need a deeper and more complex analysis to understand the differences between each group. This will lead us to offer better business suggestions to help the company make more informed strategic decisions concerning the opportunities, product definition, positioning, promotions, pricing, and target marketing . The following report dives into the complexity of this customer segmentation strategy, helping the XYZ Company to navigate the ever-evolving landscape of customer preferences and industry dynamics

2. Data Pre-Processing and Exploration

2.1. Presentation of the dataset

The ERP system of XYZ sports company is effectively providing us information regarding consumer behavior. The dataset obtained holds many informations and details about how the customer behaves and how much he is engaged. With the information piled up during the large period of time from June 2014 to October 2019, the dataset provides complete details about the fitness facility summed up in 30 different columns with the 14,942 rows that shows the different customer encounters. The customer base is made up of people from diverse backgrounds. And this can be established by looking at demographic indicators such as *age* and *gender*. The financial background of the client can be assumed by their *monthly income* and how much they spent at the company (*lifetime value*). The enrollment details such as the *start* and *end dates* provide a way to measure customer engagement.

The dataset also can help understand customer activities, from *athletics* and *water sports* to *fitness classes* and more, offering some insight on customer preferences. Loyalty of the customer can be measured by the *number of visits*, *attended classes* and *renewals.*

Through the utilization of this dataset, we hope to help XYZ Sports Company give better services and target marketing according to the needs of their customers.

2.2. Data Pre-Processing and Exploration

The success of every clustering task relies heavily on the quality of the data it is trained on, and data preprocessing and exploration is the key to ensuring that the data is in the best possible state for the clustering algorithms to work on. As the dataset contains 30 columns, in this part we are going to focus on the features that were deemed worth mentioning.

Starting with *enrollment start* and *finish*, we began by casting their type to date time and then we proceeded to make some plots. These plots gave us an idea about the general trend of enrollments and what are usually the busier months. But why not make it simpler? Instead of working with 2 separate columns, we can get the same information with only 1 column which we will call *enrollment time* (*enrollment finish* - *start*). We notice that a non negligible percentage of people have enrolled for 0 days although they haven't dropped out so we can change their *enrollment finish* to the last date (31/10/2019) and after recalculating the *enrollment time* one last time, we can drop the other 2 columns. (Fig. A1)

For the *last period start* and *finish*, we followed the same reasoning for the *enrollment start* and *finish* but we noticed that it wasn't giving us much insight so we ended up deleting the columns without any replacement.

The column *date last visit*, as the name indicates, is in date format. Instead of working with this type of data, we opted for the creation of a new column *days since last visit* (last date of the data collected - *date last visit*). This new column contains the exact same information as its predecessor but is much easier to work with and understanding what the numbers mean is more intuitive. In the graph below we notice that 20% of our customers came very recently.

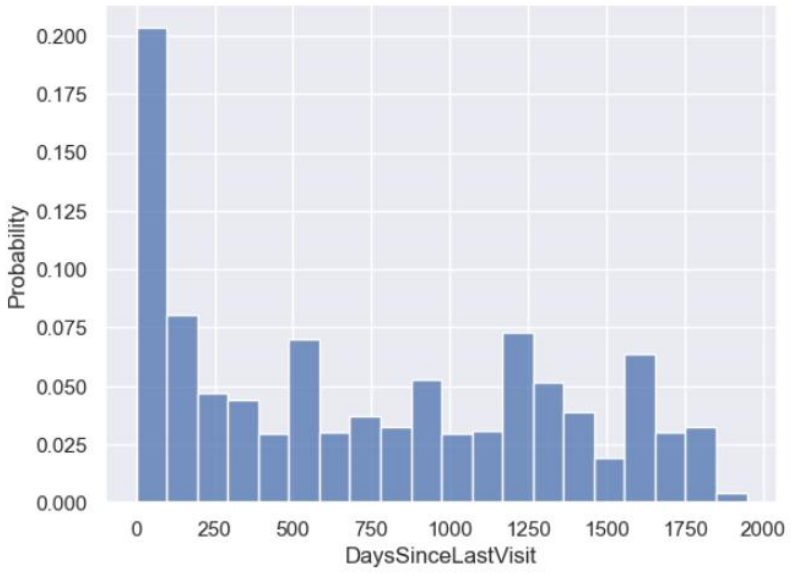


Fig.1: Distribution of days since last visit column

We have multiple features about the number of visits; *allowed weekly visits by SLA*, *allowed number of visits by SLA*, and *real number of visits*. The first one seems to be the least informative because the *real number of visits* gives us information about the activity level and dedication of the customer and *allowed number of visits by SLA* includes the information contained in *allowed weekly visits by SLA*: if you know the total number of visits allowed in a period of time, you can easily deduce the weekly number. We can once again merge the information contained in these 2 columns into 1; *laziness index*. This index is created by the following formula: 1 - *real number of visits* / *allowed number of visits by SLA*. This new feature is a good indicator of *dropout*; the higher the index is, the more likely the customer to dropout(Fig. A2). After we delete the 3 columns as we have no use for them anymore.

The activities features are important and we can use them to see which activity is the most popular amongst our clientele. We notice that whilst the *fitness activities* are the most privileged by far (57.47%), there are 2 activities that have a 0% participation rate which are *dance activities* and *nature activities* that we are going to drop. We then proceed to fill the missing values in these columns using the mode (value that appears the most often) of each category. We can do this in this case because they only have a small amount of missing values.

Some of the other features underwent some light transformation and are going to be covered together in this paragraph. The *income* column got its missing values filled with the median of each age group, while the *number of frequencies* missing values were filled with the mode. The values in *gender* were submitted to some binary encoding (transforming values from strings to integers in this case for Male/Female to 1/0) . Since the features *has references* and *number of references* are highly correlated as we can see in the table below; we can drop the *has references* column because the *number of references* contains more information.

	HasReferences	NumberOfReferences
HasReferences	1.000000	0.918348
NumberOfReferences	0.918348	1.000000

Fig. 2:Correlation between the has references and number of references

Another method to simplify things and to reduce the total number of features is looking at the variance. Doing this we notice that some features have a very low variance which means that they have a low impact in clustering making them less informative for the algorithm. For example, in the case of k-means, clusters are formed based on the mean value. If the features have a low variance, that means that they are concentrated around a central point which makes the clustering process less influential. So with that being said, we decided to drop the features that have a variance less than a certain threshold in our case 0.01.

There is one last method to reduce the dimensionality of our dataset which is the correlation. It is pretty useful because it helps deal with the curse of dimensionality by identifying redundant information. Highly correlated features transfer the same information. So applying this reasoning in our data we start by plotting a correlation matrix. After looking at the matrix, 0.8 seems to be a good threshold because it helps reduce the dimensionality without losing too much information in the process which practically translated to dropping *age* (highly correlated with *income* which we value more as we are more interested in financial gains) and *number of renewals* (which is highly correlated with *enrolled time* which we value more because it gives more information about it encapsulate the latter and more).

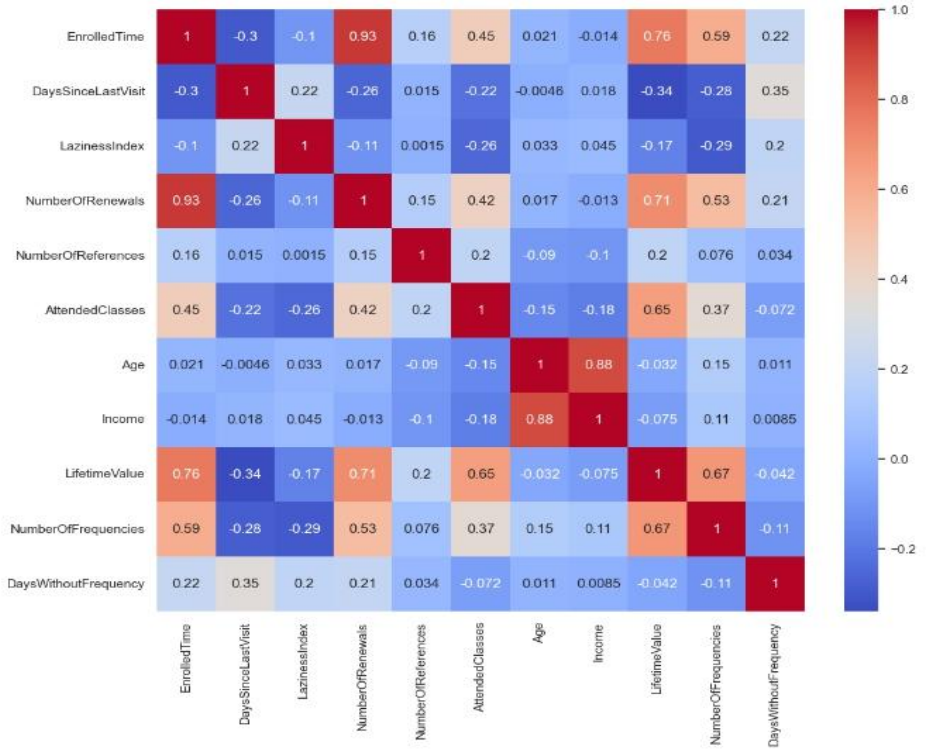


Fig. 3: Correlation matrix of the meaningful features

The last part of the pre-processing phase is dealing with outliers. In order to spot these outliers, we can start by using the describe method to get an idea about which features might have outliers and then we plot the suspected features (Fig. A3) to get a better idea.

	count	mean	std	min	25%	50%	75%	max
Gender	14942.0	0.402289	0.490376	0.000000	0.000000	0.0000	1.000000	1.0
Income	14942.0	2180.503279	1582.550391	0.000000	1420.000000	1970.0000	2760.000000	10890.0
DaysWithoutFrequency	14942.0	81.224936	144.199576	0.000000	13.000000	41.0000	83.750000	1745.0
LifetimeValue	14942.0	302.561871	364.319566	0.000000	83.600000	166.2000	355.075000	6727.8
UseByTime	14942.0	0.047116	0.211893	0.000000	0.000000	0.0000	0.000000	1.0
WaterActivities	14942.0	0.295476	0.456272	0.000000	0.000000	0.0000	1.000000	1.0
FitnessActivities	14942.0	0.577031	0.494047	0.000000	0.000000	1.0000	1.000000	1.0
TeamActivities	14942.0	0.055414	0.228795	0.000000	0.000000	0.0000	0.000000	1.0
RacketActivities	14942.0	0.023357	0.151040	0.000000	0.000000	0.0000	0.000000	1.0
CombatActivities	14942.0	0.107683	0.309990	0.000000	0.000000	0.0000	0.000000	1.0
SpecialActivities	14942.0	0.026436	0.160432	0.000000	0.000000	0.0000	0.000000	1.0
NumberOfFrequencies	14942.0	40.054210	65.428767	1.000000	7.000000	18.0000	45.000000	1031.0
AttendedClasses	14942.0	10.152456	29.154202	0.000000	0.000000	0.0000	3.000000	581.0
NumberOfReferences	14942.0	0.022286	0.166777	0.000000	0.000000	0.0000	0.000000	3.0
Dropout	14942.0	0.800964	0.399289	0.000000	1.000000	1.0000	1.000000	1.0
EnrolledTime	14942.0	428.836100	432.144322	8.000000	118.000000	282.0000	578.000000	1977.0
DaysSinceLastVisit	14942.0	737.066457	591.105873	0.000000	154.000000	651.0000	1254.000000	1946.0
LazinessIndex	14942.0	0.843053	0.180794	-1.540176	0.770379	0.9001	0.967752	1.0

Fig. 4: Descriptive statistics of the remaining features

After that was done, we noticed that the features that have outliers to be dealt with are: *income*, *days without frequency*, *lifetime value*, *number of frequencies*, *attended classes*, *enrolled time*, and *laziness index*. Now to deal with them. We started by using the interquartile approach which consists of deleting data that is either past the upper limit (third quartile + 1.5 interquartile range) and lower than the low limit (first quartile - 1.5 interquartile range). Using that method results in deleting way too much data so we decided to create custom by hand filters which results in keeping 95% of the data which is a good sign. In the end we opted for merging the 2 methods which meant removing the data that is considered outliers for both sides.

Now our data is clean and ready for the following steps which is the creation of clusters by applying different algorithms to it.

3. Clustering and Customer Segmentation:

Now that the data is all cleaned and pre-processed, we can start thinking about how to approach the real problem at hand: what is the best approach to create meaningful clusters for this data and what can we actually learn from them.

That being said, we can't just go on and apply clustering algorithms already; we need some minor adjustments first. For that we start by scaling our data in 2 different ways to compare results: standard scaler (*data std*) and minmax scaler(*data mm*). Then we go to the feature selection step where we opt for manually selecting the features because if we leave that to the model; they will prioritize features that are best for the clustering but not necessarily features that give valuable insights. After a number of trials we settle for the 5 feature categories: metric, demographic, behavior, activities, and subscription.

```
metric_features = ['Income', 'DaysWithoutFrequency', 'NumberOfFrequencies', 'LifetimeValue',
                   'AttendedClasses', 'NumberOfReferences', 'EnrolledTime', 'DaysSinceLastVisit', 'LazinessIndex']
```

```
demographic_features = ["Gender","Income","LifetimeValue"]
```

```
behaviour_features = ["DaysWithoutFrequency","UseByTime","NumberOfFrequencies","AttendedClasses","NumberOfReferences",
            "DaysSinceLastVisit","LazinessIndex"]
```

```
activities_features = ["WaterActivities","FitnessActivities","TeamActivities","RacketActivities","CombatActivities",
                       "SpecialActivities"]
```

```
subscription_features = ["EnrolledTime","Dropout"]
```

Fig. 5: Different features categories with their respective columns

We can finally get to the clustering part. In this step, we will apply a number of different clustering algorithms to our cleaned and scaled data and compare the results.

Starting off with the simplest yet most widely used clustering algorithm: **K-means**. This will serve as a baseline method to compare against. But the K-means algorithm takes the number of clusters as input and not a result of the model so we start off by using the elbow method and the silhouette score(*Fig. A4*) to figure out the best number of clusters to give as input.

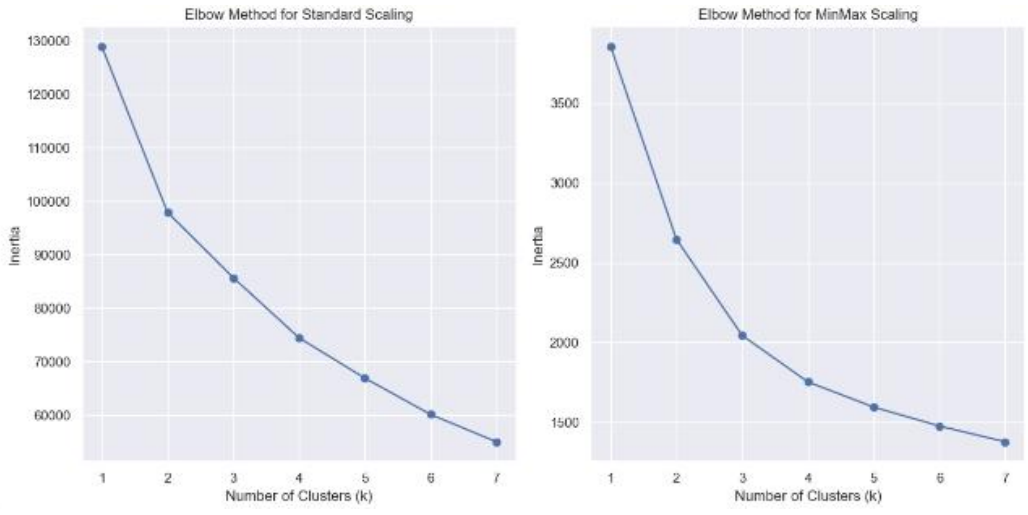


Fig. 6: Elbow Method applied to 2 different scaled data to figure out the best number of cluster for the K-means algorithm

After looking at the graphs, we can safely say that 3 clusters is a safe pick for the K-means so after applying the algorithm we get the following clusters and R^2 scores(*Fig. A5*).

Another good clustering algorithm is the **Self Organizing Maps**(**SOM**). It is a valuable tool for understanding the structure of complex datasets and dimensionality reduction. Before going into the elbow method and the clustering, we can start by gaining more insights into the SOM's initial state, initial parameter tuning, and assessing the potential effectiveness of the algorithm for the given dataset. This can be done by plotting the Hit Map(*Fig. A6*) and the SOM Matrix(*Fig. A7*). The hit map shows the frequency or the count of how many times each unit(neuron) in the SOM was the winner during the training process. Whilst the SOM matrix simply represents the distances between the units. To measure the quality of the SOM, we can use the quantization error which is the average distance between each input vector and its best-matching unit; the lower the error is the better the SOM represents the data so that's what we are going for. In our case it was with minmax scaled data (quantization error of 0.31 against 1.6 for the standard scaled).

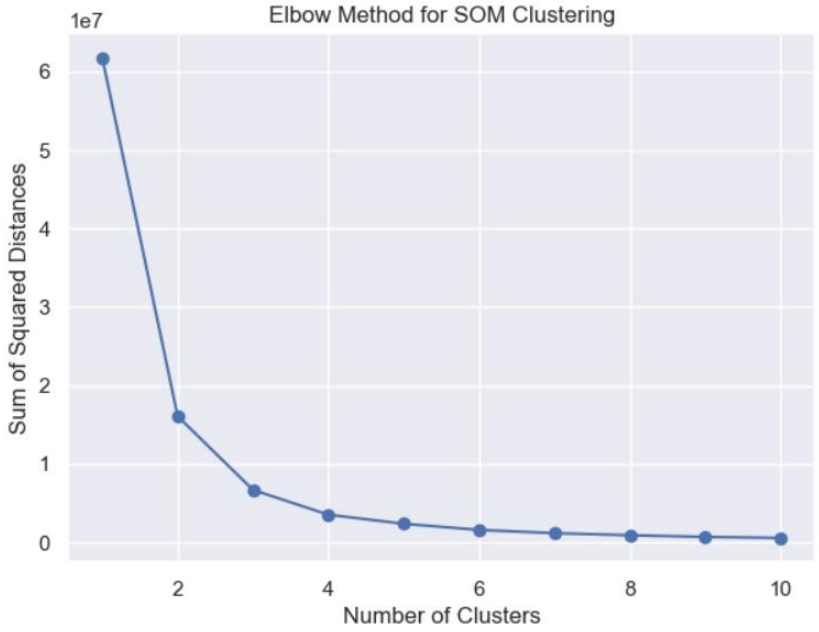
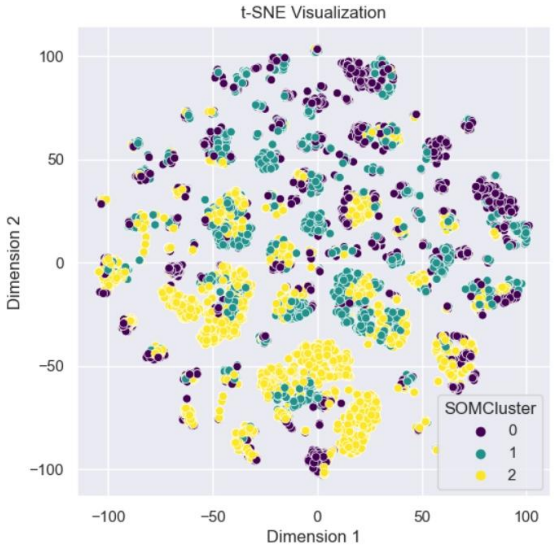


Fig. 7: Elbow Method to determine the number of clusters for the SOM algorithm

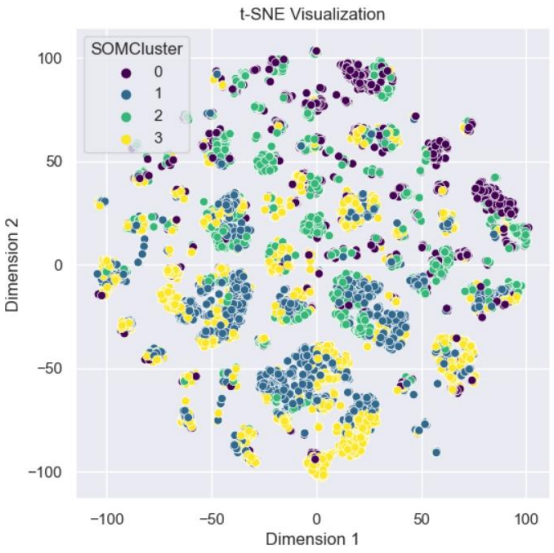
So applying the elbow method reveals a close call between 3 and 4 clusters so we choose to try with both of them and compare at the end (*Fig. A8*). The graph below shows the difference between the 2 numbers using the t-SNE(t-Distributed Stochastic Neighbor Embedding) to capture the local relationships between data points.

We notice that in our case choosing 4 clusters would be wiser as it gives a better R^2 score.

t-SNE visualization for 3 clusters



t-SNE visualization for 4 clusters



R^2 for SOM Clustering with 3 clusters: 0.1788

R^2 for SOM Clustering with 4 clusters: 0.2517

Getting into the **Density Based Clustering**, we use 2 algorithms: **Mean Shift** and **Gaussian Mixture Model**(**GMM**). Contrary to the partitioning methods, density-based clustering uses the density to create the clusters which makes it more robust to outliers. Whilst Mean Shift is more simple and adaptable, GMM is more powerful in capturing data structures that we can notice in the difference of R^2 scores between the 2 of them(*Fig. A9*).

	R^2 score	
	Standard scaled data	**MinMax scaled data**
Mean Shift	0.1273	0.1102
GMM	0.1998	0.1602

Fig. 8: Table showing the R^2 scores of the 2 methods for the 2 different scaling techniques

So far, we have only been creating clusters using the *metric features* which are kind of general. What if we try to create and optimize clusters based on the type of the variables? In the following paragraph, instead of using the *metric features* as a whole, we will use the more specific types of features (*demographic*, *behavior, activities*, and *subscription*) and see if we can get more information.

Now we apply K-means, hierarchical with different linkages and GMM to the different categories of features and with different numbers of clusters and then we compare their R^2 scores. We also used it on DBScan and Mean Shift but the results weren't worth mentioning.

R² scores for Demographic:

	kmeans	gmm	complete	average	single	ward
2	0.830860	0.830860	0.830860	0.830860	0.830860	0.830860
3	0.870085	0.849354	0.861366	0.831266	0.831217	0.863869
4	0.895922	0.880611	0.876764	0.846339	0.831623	0.886607
5	0.917377	0.901939	0.895608	0.861497	0.831815	0.907423
6	0.935899	0.915554	0.910173	0.878399	0.831963	0.926597
7	0.945266	0.920026	0.913724	0.914139	0.831965	0.936793
8	0.952967	0.922503	0.914830	0.914653	0.832101	0.945404
9	0.958073	0.932113	0.917301	0.917452	0.832169	0.953022

R² scores for Behaviour:

	kmeans	gmm	complete	average	single	ward
2	0.354209	0.130571	0.207003	0.000718	0.207003	0.302261
3	0.546401	0.209473	0.524727	0.207722	0.207269	0.509070
4	0.637079	0.252741	0.527414	0.210868	0.207988	0.608751
5	0.696116	0.418767	0.573022	0.247976	0.208262	0.658465
6	0.730572	0.514671	0.655790	0.357273	0.208524	0.706213
7	0.757557	0.516503	0.670404	0.358021	0.208793	0.729930
8	0.781360	0.515368	0.672754	0.383672	0.209329	0.752094
9	0.797597	0.542329	0.701557	0.384574	0.209741	0.767160

R² scores for Activities:

	kmeans	gmm	complete	average	single	ward
2	0.517463	0.172591	0.078345	0.001470	0.000236	0.460177
3	0.719940	0.549364	0.557659	0.003346	0.000892	0.679719
4	0.808143	0.768143	0.608834	0.003740	0.002214	0.785811
5	0.850741	0.812610	0.619664	0.100341	0.002490	0.841865
6	0.892513	0.877224	0.828473	0.275861	0.002772	0.878541
7	0.918774	0.904632	0.831012	0.771556	0.003310	0.911732
8	0.940593	0.940100	0.840551	0.847950	0.004034	0.936126
9	0.949181	0.951326	0.848472	0.868639	0.004531	0.946703

R² scores for Subscription:

	kmeans	gmm	complete	average	single	ward
2	0.790673	0.790673	0.790673	0.790673	0.790673	0.790673
3	0.889841	0.885096	0.879483	0.826374	0.804076	0.886130
4	0.942285	0.899380	0.928814	0.916899	0.839777	0.937244
5	0.961774	0.949325	0.941556	0.942521	0.840017	0.958808
6	0.971627	0.955907	0.945692	0.946670	0.840028	0.969184
7	0.981161	0.974032	0.973559	0.974375	0.840272	0.976614
8	0.985422	0.980861	0.975575	0.977921	0.840327	0.983958
9	0.988987	0.982843	0.978721	0.979860	0.855413	0.987316

Fig. 9: Different results of the R^2 scores for different features and number of clusters

From the above table, we choose K-Means for our clustering with 3 clusters all across the board. Now these different perspectives can be merged so that we get the mean of each feature for every combination of perspectives. For example, if customer A fits in Demographic cluster 1 and Behaviour cluster 2, the combination (1,2) becomes a cluster. In order to have decent interpretability the perspectives were merged 2 by 2. This way the number of clusters in each combination is 3*3 = 9, some a lot more represented than others as can be seen by the crosstab commands.

In order to get a more interpretable result we performed K-Means for the features of each combination of perspectives. This way, the number of clusters is reduced. Only for the combination of “Demographic” with “Activities” there is a noticeable difference in the different clusters so that conclusions can be taken as we can see in the figure down below.

Cluster_labels_demo_act	Gender	Income	LifetimeValue	WaterActivities	FitnessActivities	TeamActivities	RacketActivities	CombatActivities	SpecialActivities
0	0.413358	0.164051	0.198961	1.000000	0.000000	0.026019	0.003487	0.021459	0.005097
1	0.000000	0.273852	0.086814	0.046995	0.835909	0.043872	0.026073	0.098985	0.026386
2	1.000000	0.262899	0.108535	0.032435	0.709039	0.088242	0.036251	0.197710	0.039828

Fig. 10: Resulting clusters from the combination Demographic - Activities (MinMax scaler)

4. Insights Gained

Taking into account all the clustering models tried, K-Means seems to be the one that best retains the original dataset, making it the main one to focus on. The main clusters to make this analysis are the ones obtained with K-Means on all features and the ones obtained just for the combination Demographic-Activities. The clusters from the other models are used to confirm initial ideas, if the same trend is generally seen across the board.

The main activities practiced and the ones with a clear distinction between the different clusters are Water and Fitness. Water has a slightly higher proportion of males than females and also more associated with *usebytime* subscription option. But most importantly this activity has a higher *LifetimeValue*, in the form of a higher *NumberOfFrequencies* and *AttendedClasses*" and lower *Dropout* rates. *EnrolledTime* is also higher for Water which means that alongside having less customers dropping out, they do it later in relation to Fitness.

There also seems to be a difference in *Income*, with Fitness being associated with a slightly higher value, which means these are the customers we want to make sure are retained.

Other important observations are that the number of enrollments has been decreasing in the years of these observations, the months with most enrollments made are September and October (after summer) and the months with most enrollments finishing are July and september.

With all this information we suggest advertising to attract new customers for Water activities, since this is the most profitable and important feature of the company. For Fitness Activities the problem seems to be retaining customers. Since these are the ones associated with higher income it's very important to avoid the trend seen of lower retention. For that, consider analyzing these programs to improve customer satisfaction.

Aside from the Activities, there needs to be a closer look at seasonal trends. During months of higher flux of customers, either in or out, plan marketing campaigns to capitalize on this fact.

5. Conclusion

In order to provide a comprehensive study of customer segmentation for XYZ Sports Company, several clustering algorithms were employed in order to understand consumer attitude, demand and interest in the company. The model we focused the most on was K-means because it was best in conserving information from the initial dataset. Clusters generated based on all features and the mix of Demographic-Activities were very instrumental in forming our awareness of the customer's dynamics.

This project can help establish a number of key findings such as activities and customer retention, seasonal trends and enrollment patterns.

For activities and customer retention, the analysis determined Water and Fitness activities as the major characteristics determining the target market segmentation. Male participation in water activities was greater with a greater likelihood to the use-by-time subscription, and most importantly, the higher lifetime value. Customers involved in these kinds of activities visit more frequently, take more classes and have a better rate than their fitness oriented counterparts. The link between fitness and a higher income casts light on the need to not just hold onto this segment, but also make it happier.

As for the seasonal patterns and trends of enrollment, it was noted that the number of enrollments decreased over time. Peak enrollment months following the summed period were September and October, with July and September leading in the number of completions of enrollment. Thus, marketing campaigns throughout these spikes can enable you to benefit from the flow of customers, and attracting and holding onto more people.

In view of these findings, the following recommendations can be made applicable by the company.

Targeting the marketing for Water activities: Because of the high profitability and customer retention in water activities a focused marketing campaign can assist to gather new customers. By showing advantages of this sort of activity we can gain more customers.

Retention strategies for fitness activities: Address the drop in retention in fitness, especially since it's usually associated with higher income. Assess the programs to determine what needs improvement in order to boost customer satisfaction.

Seasonal Marketing Campaigns: Develop strategic marketing campaigns during periods characterized by increased customer movement such as September and October to consolidate the efforts of acquiring and retaining customers.

Customer Satisfaction Analysis: To identify weak points and areas to enhance, regularly check customer satisfaction.

In conclusion, these results from the customer segmentation provide XYZ Sports Company with a map that can be helpful for them to improve their marketing strategies and boost customer engagement in order to develop prospects for growth. Through customizing their approaches, they can be able to develop a more satisfied and loyal group of customers.

6. Appendix

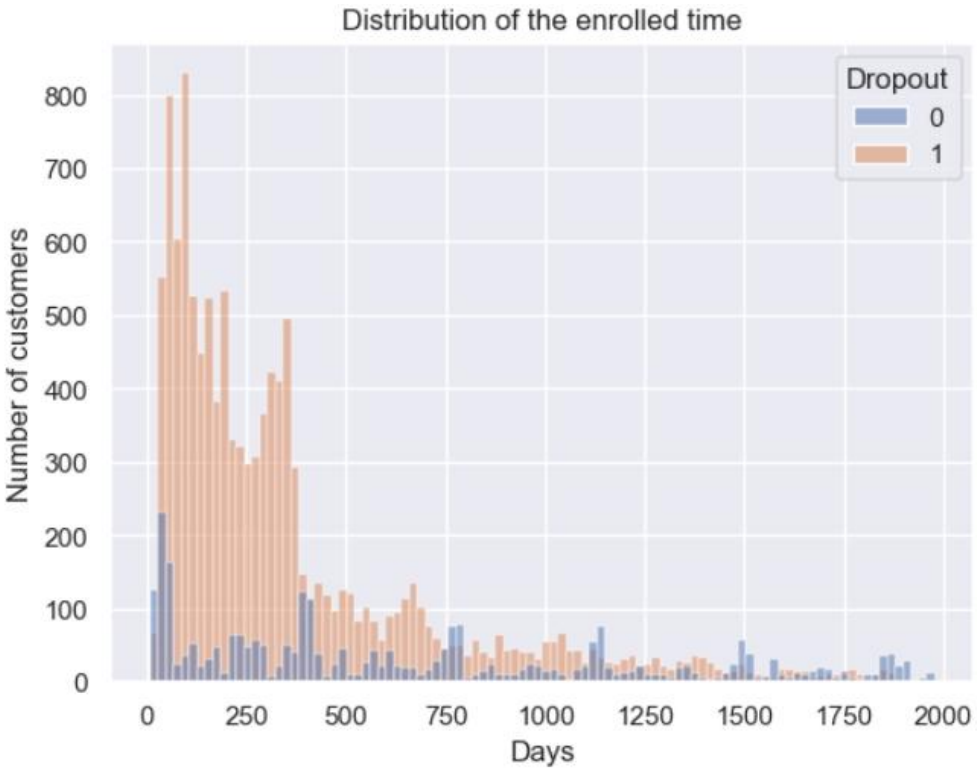


Fig. A1: Distribution of the enrolled time depending on whether the customer has dropped out or not

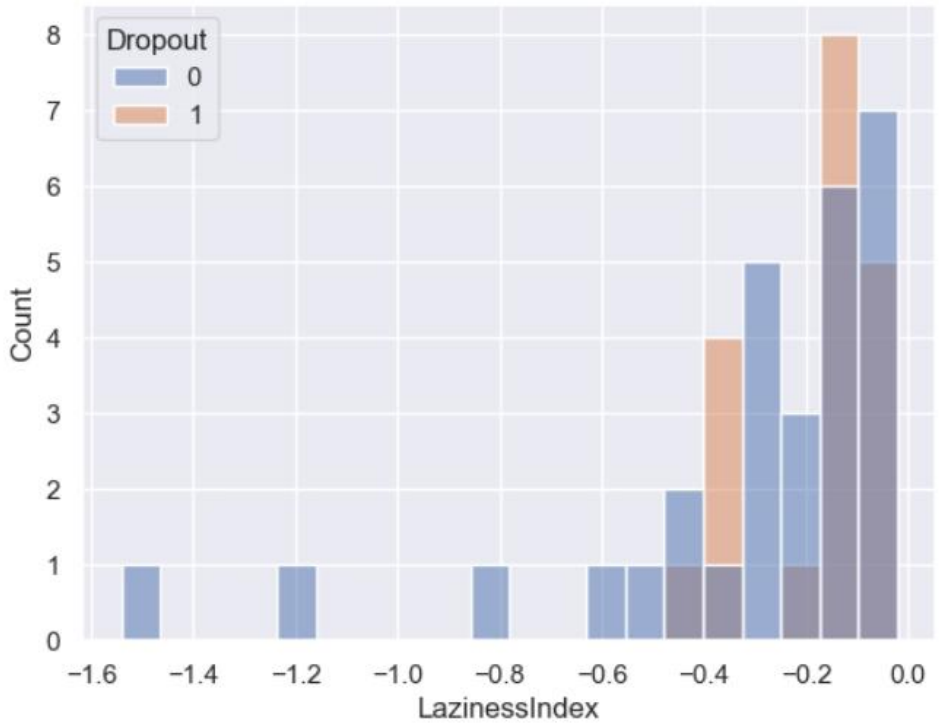


Fig. A2: Histogramme of the number of customer to dropout or not depending on the laziness index

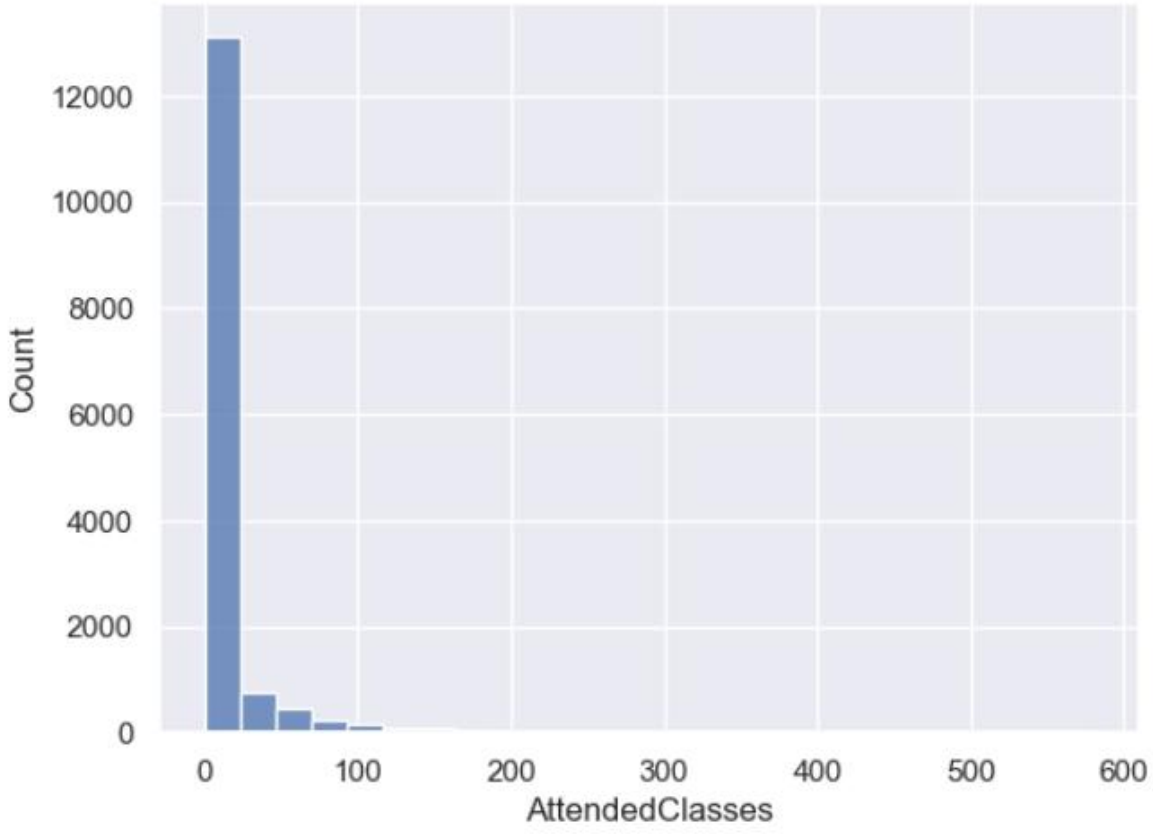


Fig. A3: Plot for outlier detection for the attended classes feature

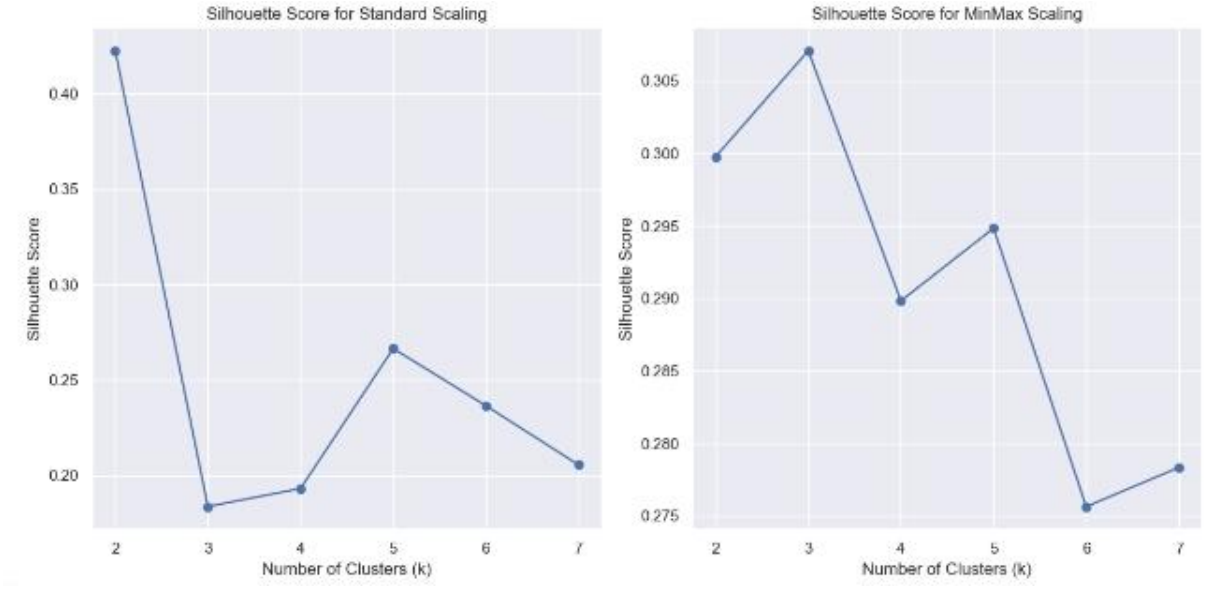


Fig. A4: Silhouette score for the 2 types of scaling for the data for the K-means algorithm

Standard Scaled Data

KMeanCluster	0	1	2
Gender	-0.02	0.17	-0.04
Income	0.04	-0.16	0.01
DaysWithoutFrequency	-0.46	-0.37	0.78
LifetimeValue	-0.30	2.00	-0.33
UseByTime	-0.06	0.12	0.03
WaterActivities	-0.07	0.73	-0.17
FitnessActivities	0.10	-0.48	0.04
TeamActivities	-0.04	0.14	0.01
RacketActivities	-0.07	-0.01	0.10
CombatActivities	-0.01	-0.09	0.05
SpecialActivities	-0.01	0.22	-0.06
NumberOfFrequencies	-0.20	1.76	-0.37
AttendedClasses	-0.22	1.52	-0.25
NumberOfReferences	-0.11	0.32	0.04
Dropout	-0.11	-0.74	0.43
EnrolledTime	-0.29	1.76	-0.26
DaysSinceLastVisit	-0.50	-0.82	0.99
LazinessIndex	-0.15	-0.58	0.43

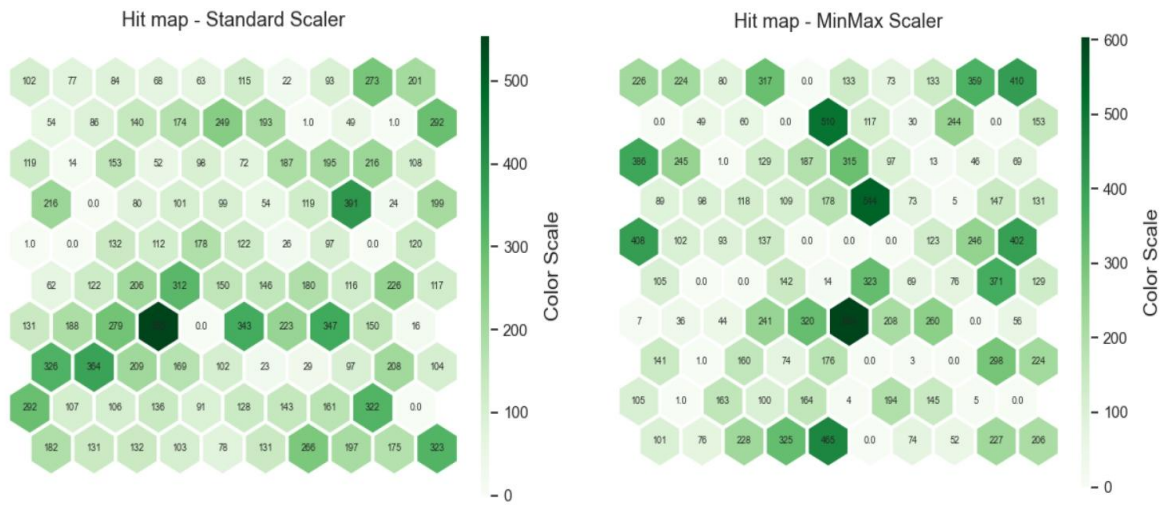
R^2 = 0.3355

MinMax Scaled Data

KMeanCluster	0	1	2
Gender	0.38	0.49	0.39
Income	0.25	0.23	0.24
DaysWithoutFrequency	0.10	0.11	0.24
LifetimeValue	0.08	0.36	0.08
UseByTime	0.03	0.10	0.04
WaterActivities	0.29	0.54	0.20
FitnessActivities	0.59	0.43	0.63
TeamActivities	0.05	0.09	0.04
RacketActivities	0.00	0.01	0.05
CombatActivities	0.11	0.10	0.11
SpecialActivities	0.02	0.05	0.02
NumberOfFrequencies	0.06	0.28	0.05
AttendedClasses	0.02	0.19	0.02
NumberOfReferences	0.00	0.02	0.01
Dropout	0.71	0.56	1.00
EnrolledTime	0.13	0.60	0.14
DaysSinceLastVisit	0.16	0.12	0.69
LazinessIndex	0.90	0.87	0.92

R^2 = 0.4698

Fig. A5: The clusters and the R^2 score for the standard and minmax scaled data for the K-means algorithm



Quantization Error = 1.5965654322952028 **Quantization Error** = 0.312218664226867

Fig. A6: Hit maps of the 2 different scaling method for the SOM clustering and their quantization error

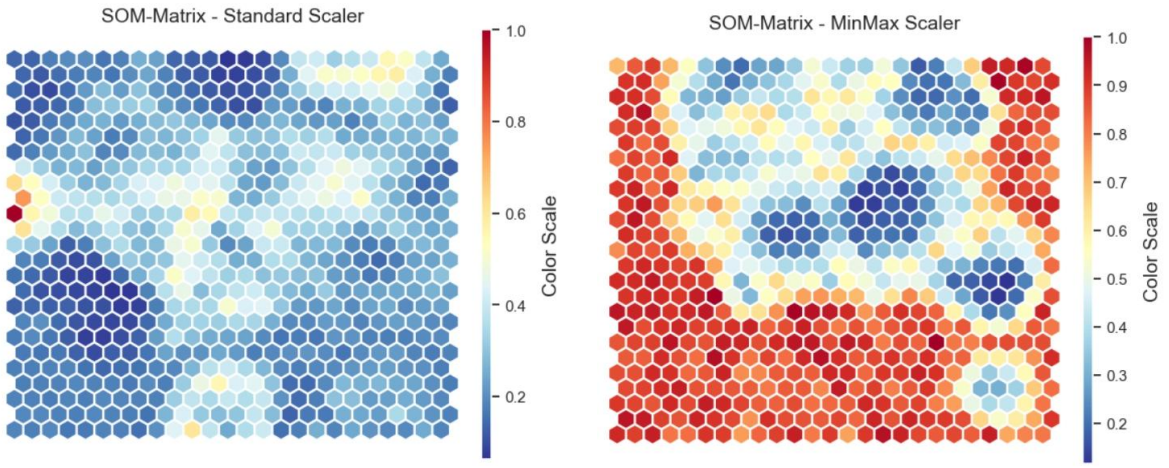


Fig. A7: SOM matrix of the 2 different scaling method for the SOM clustering

SOM clustering with 3 clusters

SOMCluster	0	1	2
Gender	0.42	0.42	0.37
Income	0.20	0.27	0.25
DaysWithoutFrequency	0.28	0.07	0.16
LifetimeValue	0.22	0.13	0.06
UseByTime	0.07	0.06	0.02
WaterActivities	0.50	0.25	0.18
FitnessActivities	0.35	0.67	0.65
TeamActivities	0.10	0.04	0.03
RacketActivities	0.03	0.01	0.03
CombatActivities	0.09	0.11	0.12
SpecialActivities	0.02	0.04	0.02
NumberOfFrequencies	0.12	0.14	0.04
AttendedClasses	0.15	0.01	0.01
NumberOfReferences	0.02	0.00	0.00
Dropout	0.66	0.77	0.95
EnrolledTime	0.33	0.26	0.09
DaysSinceLastVisit	0.32	0.23	0.55
LazinessIndex	0.89	0.89	0.92
Demographic_labels	0.69	0.68	0.58
Behaviour_labels	0.72	0.85	0.33
Activities_labels	0.68	0.87	0.99
Subscription_labels	0.83	0.67	0.06
Cluster_labels_demo_beh	0.58	0.58	0.63
Cluster_labels_demo_act	0.75	1.13	1.14
Cluster_labels_demo_sub	0.58	0.58	0.63

SOM clustering with 4 clusters

SOMCluster	0	1	2	3
Gender	0.44	0.37	0.45	0.37
Income	0.17	0.24	0.27	0.26
DaysWithoutFrequency	0.24	0.10	0.07	0.28
LifetimeValue	0.27	0.06	0.15	0.08
UseByTime	0.08	0.02	0.08	0.03
WaterActivities	0.59	0.17	0.29	0.25
FitnessActivities	0.30	0.71	0.63	0.56
TeamActivities	0.10	0.02	0.05	0.06
RacketActivities	0.01	0.03	0.01	0.04
CombatActivities	0.09	0.10	0.12	0.11
SpecialActivities	0.03	0.01	0.04	0.02
NumberOfFrequencies	0.15	0.05	0.17	0.04
AttendedClasses	0.20	0.01	0.02	0.02
NumberOfReferences	0.02	0.00	0.00	0.01
Dropout	0.54	0.95	0.69	0.94
EnrolledTime	0.40	0.09	0.31	0.14
DaysSinceLastVisit	0.20	0.49	0.16	0.56
LazinessIndex	0.86	0.90	0.88	0.95
Demographic_labels	0.66	0.52	0.71	0.71
Behaviour_labels	0.90	0.41	0.96	0.34
Activities_labels	0.57	0.97	0.84	0.96
Subscription_labels	1.13	0.09	0.89	0.14
Cluster_labels_demo_beh	0.56	0.63	0.55	0.63
Cluster_labels_demo_act	0.64	1.17	1.11	1.04
Cluster_labels_demo_sub	0.56	0.63	0.55	0.63

Fig. A8: The clusters for the minmax scaled data for 3 and 4 clusters for the SOM algorithm

Mean Shift

ClusterLabels	0	1	2	3
Gender	0.40	0.36	0.43	0.49
Income	0.25	0.09	0.16	0.13
DaysWithoutFrequency	0.16	0.64	0.21	0.04
LifetimeValue	0.11	0.10	0.57	0.60
UseByTime	0.05	0.00	0.03	0.03
WaterActivities	0.28	0.50	0.86	0.83
FitnessActivities	0.59	0.17	0.23	0.15
TeamActivities	0.05	0.15	0.14	0.06
RacketActivities	0.02	0.11	0.03	0.02
CombatActivities	0.11	0.11	0.00	0.06
SpecialActivities	0.02	0.02	0.00	0.04
NumberOfFrequencies	0.09	0.02	0.37	0.43
AttendedClasses	0.03	0.04	0.71	0.53
NumberOfReferences	0.00	0.25	0.39	0.02
Dropout	0.83	1.00	0.54	0.24
EnrolledTime	0.19	0.20	0.74	0.72
DaysSinceLastVisit	0.39	0.87	0.14	0.04
LazinessIndex	0.91	0.98	0.90	0.75
Demographic_labels	0.64	0.44	0.71	0.71
Behaviour_labels	0.60	0.01	1.00	1.02
Activities_labels	0.89	0.84	0.17	0.24
Subscription_labels	0.45	0.10	1.54	1.23
Cluster_labels_demo_beh	0.60	0.64	0.57	0.51
Cluster_labels_demo_act	1.05	0.69	0.34	0.36
Cluster_labels_demo_sub	0.60	0.64	0.57	0.51

GMM

ClusterLabels	0	1	2	3
Gender	0.09	0.00	0.11	-0.10
Income	0.27	-0.45	-0.05	0.08
DaysWithoutFrequency	-0.08	-0.08	-0.07	0.15
LifetimeValue	0.11	0.51	1.54	-0.63
UseByTime	0.23	-0.15	0.20	-0.13
WaterActivities	-0.25	0.75	0.67	-0.38
FitnessActivities	0.28	-0.86	-0.23	0.39
TeamActivities	-0.14	0.43	0.11	-0.19
RacketActivities	-0.14	0.38	0.09	-0.15
CombatActivities	0.17	-0.23	-0.11	0.02
SpecialActivities	-0.01	0.05	0.84	-0.12
NumberOfFrequencies	0.41	-0.02	1.48	-0.52
AttendedClasses	-0.37	0.85	1.03	-0.37
NumberOfReferences	-0.12	-0.12	3.24	-0.12
Dropout	0.01	-0.33	-0.28	0.26
EnrolledTime	0.40	0.22	1.43	-0.69
DaysSinceLastVisit	-0.13	-0.24	-0.29	0.33
LazinessIndex	0.09	-0.35	-0.37	0.22

Fig. A9: The clusters for the density based approach between GMM and Mean Shift