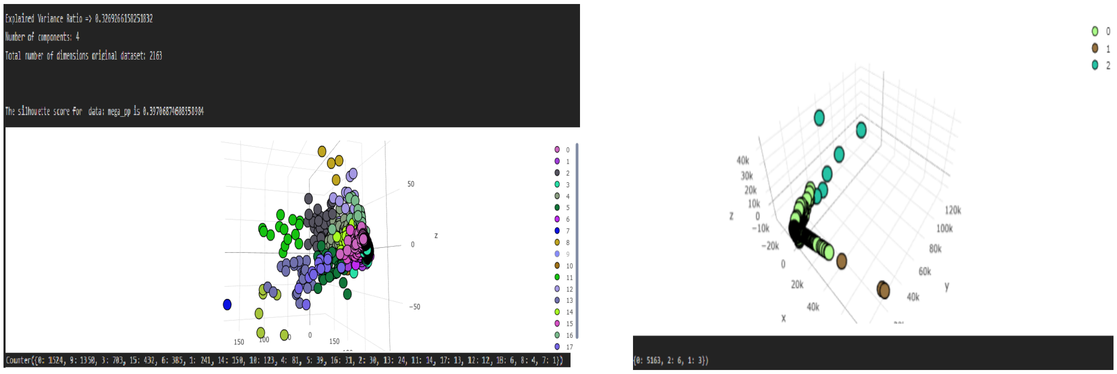
Customer segmentation Audax

Dusko Trajkov
d25trajkov@gmail.com



```
Explained Variance Ratio => 0.3269266158251832
Number of components: 4
Total number of dimensions original dataset: 2163



The silhouette score for  data: mega_pp is 0.39706874608358984
```

```
Counter({0: 1524, 9: 1350, 3: 703, 15: 432, 6: 385, 1: 241, 14: 150, 10: 123, 4: 81, 5: 39, 16: 31, 2: 30, 13: 24, 11: 14, 17: 13, 12: 12, 18: 6, 8: 4, 7: 1})
```

```
{0: 5163, 2: 6, 1: 3})
```

Figure 1: On the left a display teaser of 19 clusters accompanied with some statistics : separation score of 0.4 obtained from clustering on 4 components, which explain 32 percent of the variation in the original dataset with 2163 dimensions.On the right a display of how K-means clustering can be used on scaled data to detect the 9 extreme retailers in the dataset.

Abstract

In order to uncover differences among the 5611 unique retailers whom Audax supplies magazines to certain algorithms were used such as: **K-means , K-Medoids, Agglomerative Clustering, Gaussian mixture model(GMM) and the Hierarchical Density-Based Spatial Clustering (HDBSCAN) [1]**. After many iterations with different features two main discoveries were uncovered : the first focuses on the overall characteristics of retailers containing features such as mean number of articles sold across all types of magazines and categories (e.g. mean number of articles sold, revenue, mean percentage of returned articles etc), while the second is going deeper and contains many features individually distributed on a *Rubriek_Taal*” level where we look for clusters of retailers who indicate to have similar higher level feature distribution (e.g. Revenue) for a certain *Rubriek_Taal*” combination.The first part was quite intuitive and successful, because the number of dimensions remained low, thus the algorithms converged quite fast and little dimensionality reduction was needed. The second approach was less intuitive and far harder to uncover many well distributed clusters even after dimensionality reduction techniques such as the Principle Component Analysis was initiated. The main reason for this is believed to be the large number of outliers in the dataset. this results in the majority of the retailers(ca 3000) being grouped together and the rest of points are either grouped in a lot smaller clusters or they cannot be grouped in any cluster. After dealing with the outliers by removing the most extreme ones from the dataset, filtering out redundant variables and just focusing on the first components with the largest variances, which represent around 50% of the total variance in the original dataset the resulting clusters were large in number, the separation coefficient, the Silhouette score was equal to 0.28 and the total number of clusters equaled 32. This result was achieved using K-means by trial and error and the number of clusters were chosen in such a manner as to maximize the number of clusters while keeping the Silhouette score as high as possible in comparison to the optimal threshold indicated by both the Silhouette test and the Elbow-method, which often ranged between 2 and 8 with a very uneven distribution.

1 Introducing Assignment

The goal of this assignment is given a number of features(aka ‘VerkoopID’) find similarities and cluster all retailers based on these similarities. This document excludes all background information about the subject matter(Audax as a business) and focusses only on summarizing the best findings uncovered during the analysis and a short description of the steps taken to enable an effective and reliable clustering. It is highly recommended to research the bold terms in this document so to make reading the document more sensible, since the terminology of the technical bits and unsupervised learning algorithms are not discussed here. Note : sources of descriptions and in-depth information to algorithms and certain terms are added as footnotes.

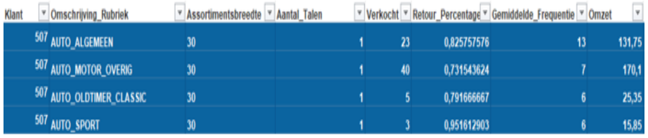
2 Methodology

2.1 Preprocessing

The first step of the whole process consists of pre-processing of the data and data exploration. missing values, outliers (using the **IQR-method [2]**), negative values are detected and then removed. After this is done thoroughly the data is transformed in such a manner(standardization) as to make comparison between variables with different unit measures (euros, counts , percentages etc.) possible. In cases where the number of columns, variables grows in the hundreds or thousands the transformed data may be used to uncover exactly which of these variables differ from one another across all entries. This is done by looking at a correlation matrix or better yet a model can be designed like the **DecisionTreeRegressor [3]** whom helps explain how relevant a certain variable is given a certain combination of the other variables. If the excluded variable can be predicted with the other variables with high certainty then that variable can also be excluded from the clustering analysis. The main reason for reducing the data is so the clustering algorithms can converge fast and thus improve efficiency. Moreover a phenomenon called **the curse of dimensionality [4]** can occur when the number of variables/features becomes too large. In such cases the distance between two points keeps getting smaller which leads to nonexistent relationships and false conclusions. Thus it is imperative that the clustering algorithms' input only contains features/components that explain the most variance in the original dataset. A method used to accomplish this is called **principle component analysis [5]**.The resulting transformation of this analysis contains components/synthetic features that are in numbers but a fraction of the original number of features, however they contain most of the variance from the original dataset.

2.2 Clustering

The clustering part takes off when the data is preprocessed and the principle components are created. Most of the findings from the clustering analysis were derived using the K-means algorithm, which uses the **Euclidean distance [6]** as a similarity metric. Using this approach is most efficient as it doesn't need a lot of computing power with large datasets and its interpretation is quite straightforward. The original dataset is indexed mostly by retailer(Rubriek, Rubriek_Taal combo) occurance, as shown below:

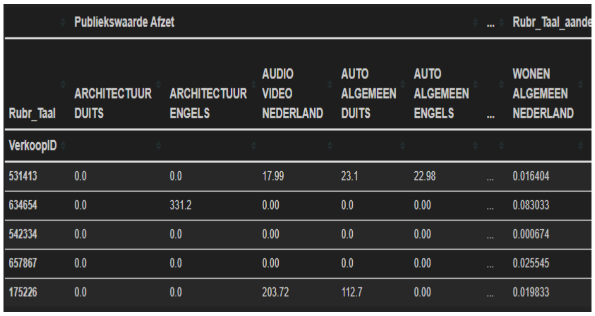


Klant	Omschrijving_Rubriek	Assortimentsbreedte	Aantal_Talen	Verkocht	Retour_Percentage	Gemiddelde_Frequentie	Omzet
507	AUTO_ALGEMEEN	30	1	23	0,825757576	13	131,75
507	AUTO_MOTOR_OVERIG	30	1	40	0,731543624	7	170,1
507	AUTO_OLDTIMER_CLASSIC	30	1	5	0,791666667	6	25,35
507	AUTO_SPORT	30	1	3	0,951612903	6	15,85

There are mainly two ways considered to deal with this in the preprocessing step. Even though traces in the **jupyter notebooks** created for this project, can be found of method (1) mostly for the sake of robust and correct clustering method (2) was the main approach taken in all cluster trials. Brief descriptions of both methods :

1. Encode the categorical variables (e.g. 'Omschrijving_Rubriek') as binary dummy variables for each retailer, then perform **(MCA) multiple correspondence analysis [7]** on the dummy variables in order to find reduce the dimensionality. When the components are extracted, they are merged with the mean values of the other variables per retailer. The number of dimensions with this method is low, however the informative distribution of the continuous variables per categorical feature (e.g. 'Omschrijving_Rubriek') is lost, only the occurrence of the categorical variable is taken into account by the MCA components.

2. Make a pivot table where the categorical variable is linked directly to another numerical variable, which results in a multi-indexed long set of columns with a large number of numerical variables. Needless to say if a combination is not possible for a certain retailer (e.g. Assortimentsbreedte: Auto_Sport does not exist) then the missing value is simply replaced with a 0. This leads to a large number of dimensions (dim > 2000) , however no information is lost and this in turn serves as a better input for the PCA dimension reduction technique.



	Publiekswaarde Afzet					...	Rubr_Taal_aande
Rubr_Taal	ARCHITECTUUR DUITS	ARCHITECTUUR ENGELS	AUDIO VIDEO NEDERLAND	AUTO ALGEMEEN DUITS	AUTO ALGEMEEN ENGELS	...	WONEN ALGEMEEN NEDERLAND
VerkoopID							
531413	0.0	0.0	17.99	23.1	22.98	...	0.016404
634654	0.0	331.2	0.00	0.0	0.00	...	0.083033
542334	0.0	0.0	0.00	0.0	0.00	...	0.000674
657867	0.0	0.0	0.00	0.0	0.00	...	0.025545
175226	0.0	0.0	203.72	112.7	0.00	...	0.019833

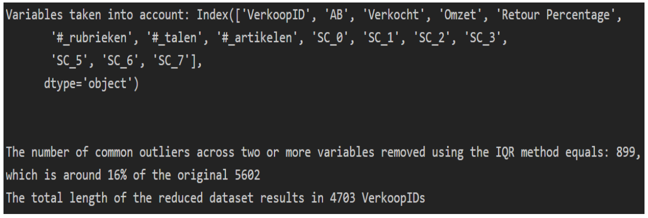
The clustering algorithm is fed these so called principle components that represent a certain percentage of the variance from the original dataset and the algorithm tries to find structure given certain parameters. In the case of non-hierarchical algorithms like K-means, GMM and Agglomerative Clustering the number of clusters need to be specified , hierarchical and density based algorithms like HDBSCAN or MeanShift do not require the specification of the number of desired clusters, however other parameters need to be specified e.g. the minimum number of points in a cluster, the minimum number of points that need to be away from a certain cluster for a group of points to be considered a cluster etc. All these parameters and randomization always lead to different results, which is why there is no exact solution to the problem but a best possible approximation of the truth. A way to decide on the best result is for example to look at tests like the Silhouette test and choose the highest possible score with a reasonable number of clusters, while taking into account how much variance(number of principle components) the clustering algorithm has considered from which these results are derived.

3 Main Findings

Below you may find the most effective trials in segmenting 5602 retailers with different dimensions/variables and the corresponding business intelligence (practical differences) behind the clustered retailers. Every cluster trial is supplemented with a graph of the different clusters, a graph of the methods used to determine the number of clusters and the differences between the given clusters.

3.1 Clustering retailers (zoomed out)

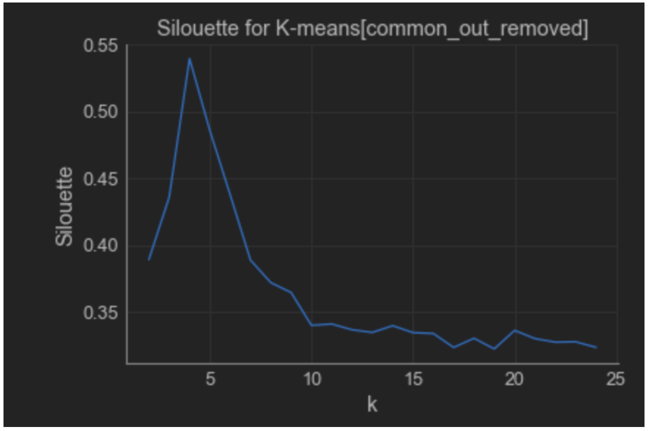
The composition of the reduced dataset (outliers removed)for this trial is as follows :



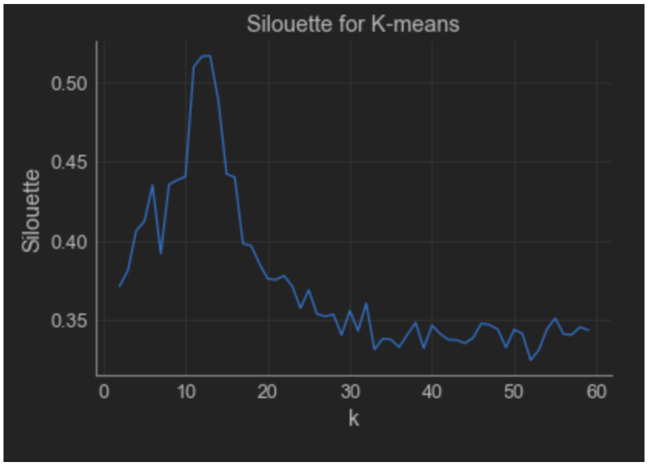
```
Variables taken into account: Index(['VerkoopID', 'AB', 'Verkocht', 'Omzet', 'Retour Percentage',
       '#_rubrieken', '#_talen', '#_artikelen', 'SC_0', 'SC_1', 'SC_2', 'SC_3',
       'SC_5', 'SC_6', 'SC_7'],
      dtype='object')



The number of common outliers across two or more variables removed using the IQR method equals: 899,
which is around 16% of the original 5602
The total length of the reduced dataset results in 4703 VerkoopIDs
```

Note: 'Verkocht' is the number of articles sold in a certain retailer, 'AB' = 'Assortimentsbreedte', SC' = binary variables of 'Seizoenscode' and '#_' means: number of.

The non-reduced form also takes the 899 retailers into account in the clustering analysis. Below are the Silhouette score test findings for different entries in K-means. The point of doing this is to see for what number of clusters most of the variance is explained, so we can prevent bias. Conventionally we choose the highest silhouette score:



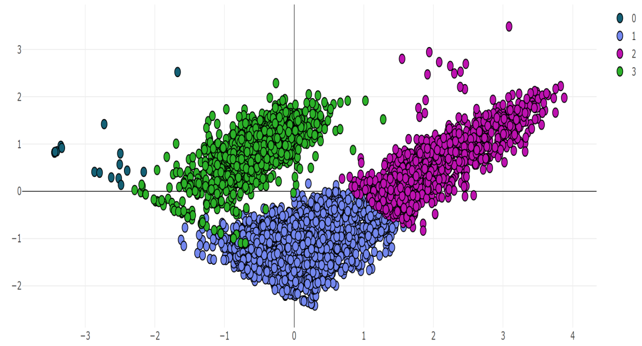
Note: The optimal number of clusters equals 4 for the reduced form(robust analysis without the 899 entries) with score = 0.54



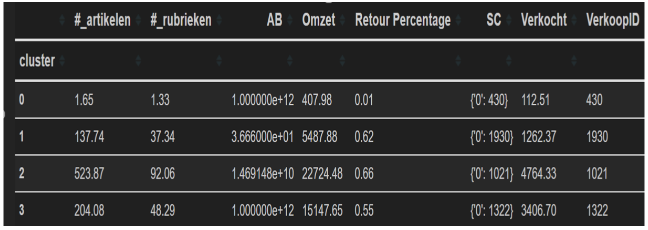
Note: The optimal number of clusters equals 13 when we take all retailers into account, with score = 0.52

3.1.1 Reduced and robust clustering analysis

Clustering on the dataset with removed outliers, thus taking only the 4703 entries into account gives us an unbiased and very intuitive result:



Note: The numbers displayed don't represent a meaningful interpretation, these are the values of the first two principle components of the standardized dataset, which explain 82 percent of the variance in the dataset. The standardized distance between the points is what's intuitive.

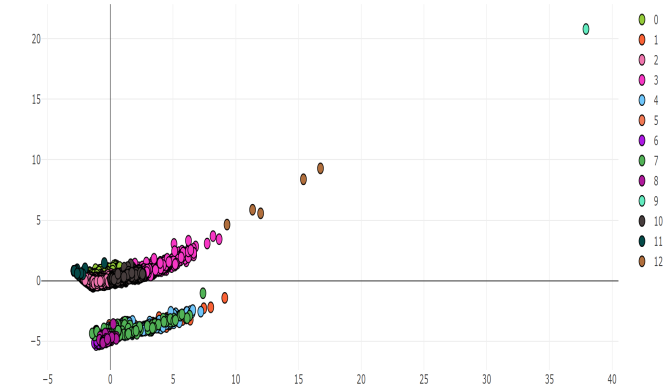


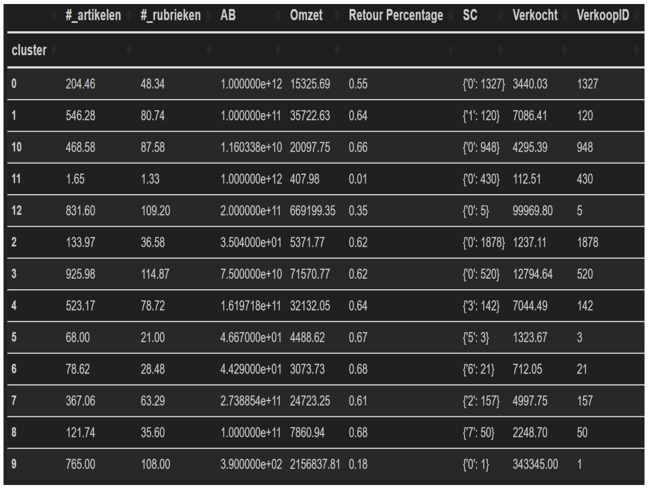
	#_artikelen	#_rubrieken	AB	Omzet	Retour Percentage	SC	Verkocht	VerkoopID
cluster								
0	1.65	1.33	1.000000e+12	407.98	0.01	{'0': 430}	112.51	430
1	137.74	37.34	3.666000e+01	5487.88	0.62	{'0': 1930}	1262.37	1930
2	523.87	92.06	1.469148e+10	22724.48	0.66	{'0': 1021}	4764.33	1021
3	204.08	48.29	1.000000e+12	15147.65	0.55	{'0': 1322}	3406.70	1322

Above you can see the average values of every variable of interest with the corresponding cluster. The 'VerkoopID' column displays the number of retail points in that cluster.The major separation of the retailers in the different brackets are decided on how large their 'Omzet' is and the ' aantal artikelnummers(#_artikelen)' . Afterwards the model separates the retailers with restricted 'AB' from the unlimited ones. Moreover it is also fair to notice that the retailers with 'SC' values other than 0 have been excluded from the dataset as outliers. "Cluster 0" is a special kind of retailers, so it was decided that they will be removed from further analysis.

3.1.2 Full variant

Here are the findings of the clustering analysis performed with K-means on the full dataset, 899 outliers included:



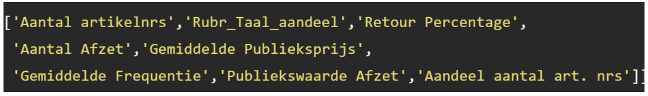


	#_artikelen	#_rubrieken	AB	Omzet	Retour Percentage	SC	Verkocht	VerkoopID
cluster								
0	204.46	48.34	1.000000e+12	15325.69	0.55	{'0': 1327}	3440.03	1327
1	546.28	80.74	1.000000e+11	35722.63	0.64	{'1': 120}	7086.41	120
10	468.58	87.58	1.160338e+10	20097.75	0.66	{'0': 948}	4295.39	948
11	1.65	1.33	1.000000e+12	407.98	0.01	{'0': 430}	112.51	430
12	831.60	109.20	2.000000e+11	669199.35	0.35	{'0': 5}	99969.80	5
2	133.97	36.58	3.504000e+01	5371.77	0.62	{'0': 1878}	1237.11	1878
3	925.98	114.87	7.500000e+10	71570.77	0.62	{'0': 520}	12794.64	520
4	523.17	78.72	1.619718e+11	32132.05	0.64	{'3': 142}	7044.49	142
5	68.00	21.00	4.667000e+01	4488.62	0.67	{'5': 3}	1323.67	3
6	78.62	28.48	4.429000e+01	3073.73	0.68	{'6': 21}	712.05	21
7	367.06	63.29	2.738854e+11	24723.25	0.61	{'2': 157}	4997.75	157
8	121.74	35.60	1.000000e+11	7860.94	0.68	{'7': 50}	2248.70	50
9	765.00	108.00	3.900000e+02	2156837.81	0.18	{'0': 1}	343345.00	1

Note:The picture illustrates how outliers can affect the clustering algorithm. Notice that the distribution is more or less still the same for the big 3, just the outliers have skewed the number of clusters towards the outliers.

3.2 Clustering retailers (zoomed in)

Next the focus lies in the distribution of features in a more aggregated level ('Rubriek- Taal combinatie per verkooppunt') multi-indexed on the following variables:



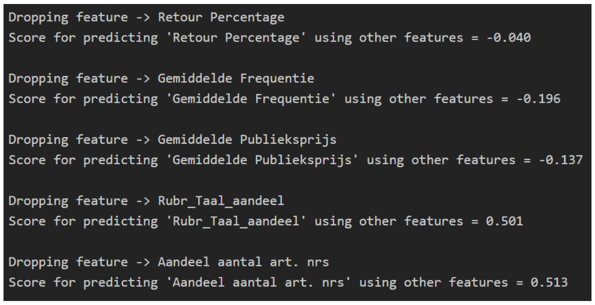
```
['Aantal artikelnrs','Rubr_Taal_aandeel','Retour Percentage',
 'Aantal Afzet','Gemiddelde Publieksprijs',
 'Gemiddelde Frequentie','Publiekswaarde Afzet','Aandeel aantal art. nrs']]
```

Here the way the data is structured and the number of variables taken into account has a lot of influence on the clustering analysis. Because of this 'Rubriek-taal' aggregation level the number of variables per retail point grew to around 5000, which made outliers detection that more difficult. A way of dealing with this is to perform IQR-analysis before pivoting on retail point, so when the data is still structured on a 'Rubriek-taal' aggregation level, from which the most extreme combination could be removed while keeping the most if not all retail points. The other way of course is to look at a retailer-level and the 5000 features, which is very computing expensive and at the end the result was biased, because of high dimensionality all retailers can be outliers in two or more features. There were many trials performed here and all outliers detection methods were tried of which the IQR of course was most robust and some gave promising results of which only the best one is described in more detail :

1. Trial variant that resulted in 19 clusters(at this point the 430 + 9 (from table 1) special cases were removed with a couple of most common outliers on retailer level, total equal to 766) based on only the first four principle components which explained 33% of the variance in the dataset. The total number of dimensions considered equaled 2163 and the cluster separation score (Silhouette Score) was equal to 0.4. From the 19 clusters the big three were further sub-clustered. In this way we use K-means as anomaly/ more robust outliers detection method first and then we look for further structure in the points that lay close to one another. The sub-clustering process did not gave more insight, it just resulted in a more or less same distribution with 3 big clusters. This finding from the sub-clustering was deemed to be biased as it didn't take into account all the available information (e.g. info from the 16 other clusters).

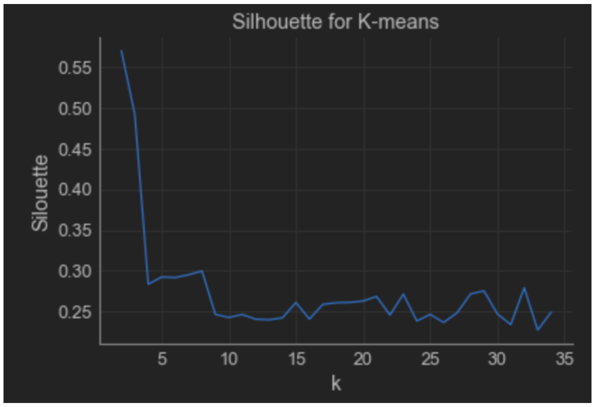
2. Trial variant that resulted in 17 clusters excludes variable 'Retour Percentage' as it wasn't deemed very interesting as there weren't many differences across retailers except for the already removed 430 instances with very low 'Retour Percentage' values. The structure of the input for the analysis is more or less the same as the K-19 variant : explained ratio equals 38%, which incorporates 9 principle components and the separation score equaled 0.38. However the distribution of the retail points for the big three was more even. *(For more info please consult the jupyter notebooks or python scripts saved in the Betapress database.)*

3. Trial variant that showed most promise resulted in 32 clusters. The way going about this was first removing all features that were highly correlated with one another leading to a more robust and objective analysis. This resulted in the following five main features extracted from the DecisionTreeRegressor-method with train-test split of (70:30) :



```
Dropping feature -> Retour Percentage
Score for predicting 'Retour Percentage' using other features = -0.040

Dropping feature -> Gemiddelde Frequentie
Score for predicting 'Gemiddelde Frequentie' using other features = -0.196

Dropping feature -> Gemiddelde Publieksprijs
Score for predicting 'Gemiddelde Publieksprijs' using other features = -0.137

Dropping feature -> Rubr_Taal_aandeel
Score for predicting 'Rubr_Taal_aandeel' using other features = 0.501

Dropping feature -> Aandeel aantal art. nrs
Score for predicting 'Aandeel aantal art. nrs' using other features = 0.513
```

Then IQR- analysis was performed on the low -dimensional dataset with 'Rubr-taal' combinations as index per retail point to detect outliers. This resulted in 1222 combinations of 'Rubr-taal' per retail point that were seen as extreme cases for more than 2 features (out of the 5), which represented around 0.4% of the whole dataset (300 000 + entries). Afterwards the dataset was restructured to indexing 5117 individual retail points and 1540 features/columns. Using this the principle components were computed, representing 49% of the variance in the dataset, which then served as input for different clustering algorithms of which K-means yet again gave the best results in VerkoopID distribution per cluster and Silhouette score of 0.28. Of course this is not the optimal case, choosing less clusters led always to a higher separation score, however the interest lay in getting the highest amount of clusters, while maintaining a high score. The next closest number of chosen clusters to this score ranged between k = 4 and k=9, so it was safe to choose more clusters for the same average Silhouette score.



This trial took more variation of the dataset into account with the most important features and gave the best retail-distribution across many clusters, which are the main reasons why this is the most desired clustering result. For an overview on the clustering analysis please visit table 2 of the Appendix section.

According to literature [1] the clustering algorithm HDBSCAN is most robust to high dimensions so it was also used as an alternative. Even though HDBSCAN doesn't need the upfront specification of the number of desired clusters like K-means, the algorithm needs other parameters to be tweaked for the best possible results. In most, if not all cases using this algorithm resulted in finding a few clusters of a few distinct retail points, while the majority of the retailers were classified as extreme cases and could not be assigned a cluster. For an overview on this please consider table 3 in the Appendix section.

4 Conclusion

In short depending on how many dimensions we take into account and which algorithm we use the number of clusters and the distribution across segments can differ. There is no optimal solution to this problem, only a best possible representation of reality. There is clear trade-off between the number of dimensions we put in the clustering algorithms and the number of clusters that come out as output.The differences between retailers are best seen when we only take a few variables with the highest variance into account, which is why the trial resulting in 32 clusters is most robust to bias and thus will most closely showcase the most important differences of the retail points.

References

[1] Github, Clustering algorithms
`https://github.com/scikit-learn-contrib/hdbscan/blob/master/docs/comparing_clustering_algorithms.rst`

[2] IQR mthod for defining outliers
`https://www.purplemath.com/modules/boxwhisk3.htm`

[3] Explanation of DecisionTreeRegressor
`https://cambridgespark.com/content/tutorials/getting-started-with-regression-and-decision-trees/index.html`

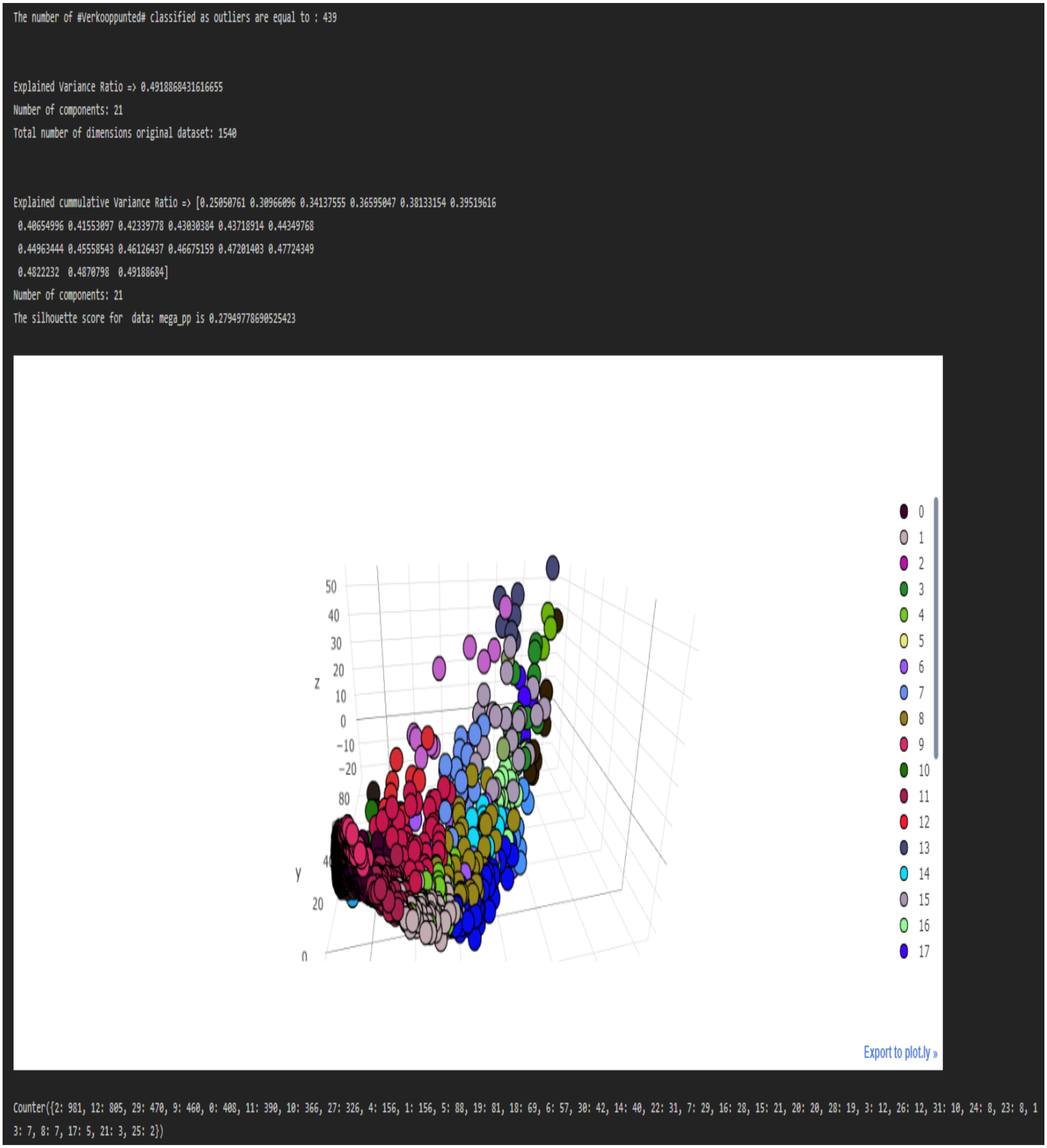
[4] Curse of dimensionality explained, FreeCodeCamp
https://medium.freecodecamp.org/the-curse-of-dimensionality-how-we-can-save-big-data-from-itself-d9fa0f872335

[5] Visual demo of how PCA works
`http://setosa.io/ev/principal-component-analysis/`

[6] Euclidean distance ,measure for similarity
`http://www.pbarrett.net/techpapers/euclid.pdf`

[7] Brief description of MCA(difference to PCA), use in excel files included
`https://www.xlstat.com/en/solutions/features/multiple-correspondence-analysis-mca`

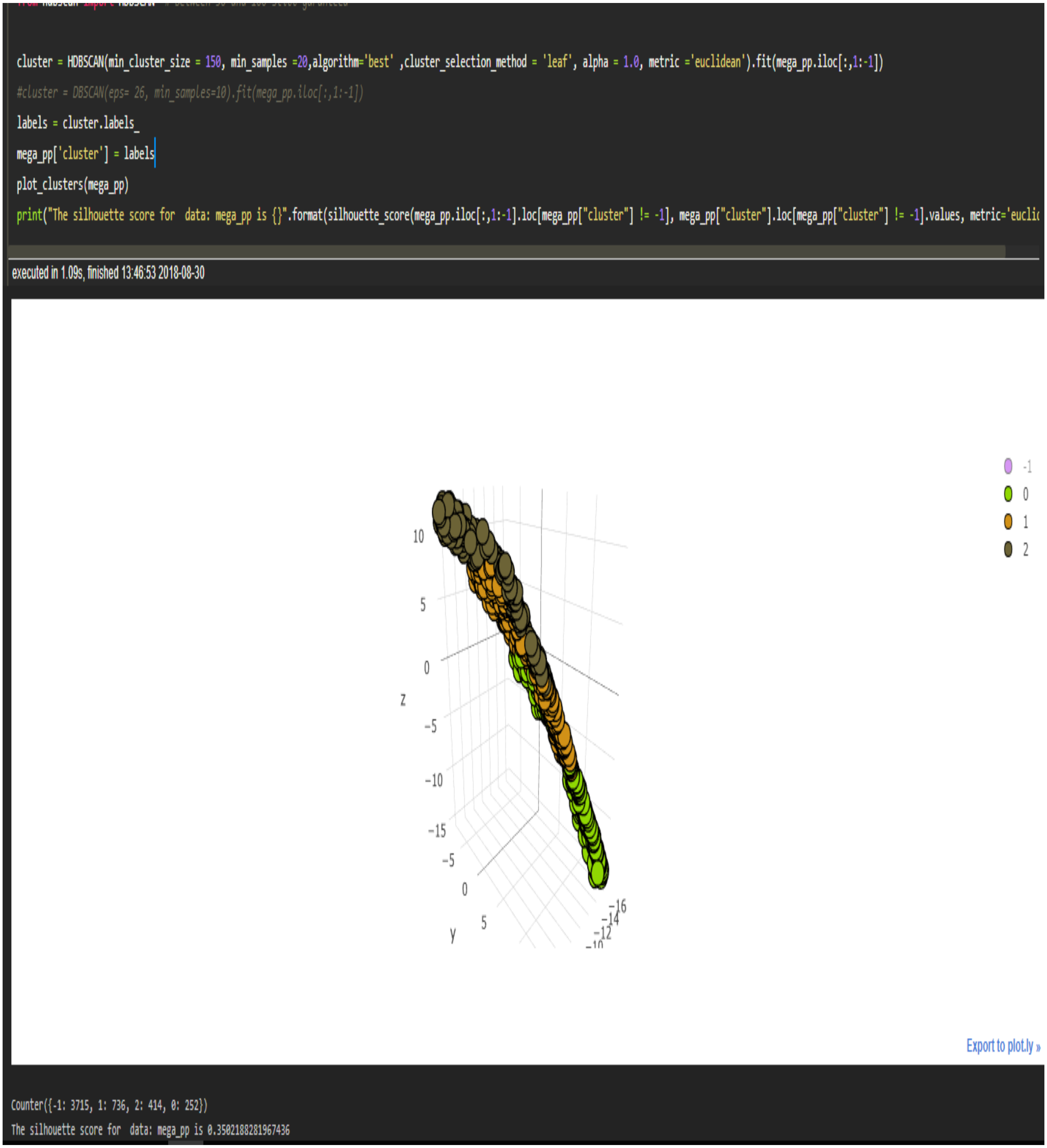
APPENDIX



```
The number of #Verkooppunted# classified as outliers are equal to : 439



Explained Variance Ratio => 0.4918868431616655
Number of components: 21
Total number of dimensions original dataset: 1540



Explained cummulative Variance Ratio => [0.25050761 0.30966096 0.34137555 0.36595047 0.38133154 0.39519616
 0.40654996 0.41553097 0.42339778 0.43030384 0.43718914 0.44349768
 0.44963444 0.45558543 0.46126437 0.46675159 0.47201403 0.47724349
 0.4822232  0.4870798  0.49188684]
Number of components: 21
The silhouette score for  data: mega_pp is 0.27949778690525423
```

```
Counter({2: 981, 12: 805, 29: 470, 9: 460, 0: 408, 11: 390, 10: 366, 27: 326, 4: 156, 1: 156, 5: 88, 19: 81, 18: 69, 6: 57, 30: 42, 14: 40, 22: 31, 7: 29, 16: 28, 15: 21, 20: 20, 28: 19, 3: 12, 26: 12, 31: 10, 24: 8, 23: 8, 1
3: 7, 8: 7, 17: 5, 21: 3, 25: 2})
```

Table 2: Results of the 32k clustering, K-means



```
from hdbscan import HDBSCAN # between 50 and 100 still garanteed


cluster = HDBSCAN(min_cluster_size = 150, min_samples =20,algorithm='best' ,cluster_selection_method = 'leaf', alpha = 1.0, metric ='euclidean').fit(mega_pp.iloc[:,1:-1])
#cluster = DBSCAN(eps= 26, min_samples=10).fit(mega_pp.iloc[:,1:-1])
labels = cluster.labels_
mega_pp['cluster'] = labels
plot_clusters(mega_pp)
print("The silhouette score for  data: mega_pp is {}".format(silhouette_score(mega_pp.iloc[:,1:-1].loc[mega_pp["cluster"] != -1], mega_pp["cluster"].loc[mega_pp["cluster"] != -1].values, metric='euclic
```

executed in 1.09s, finished 13:46:53 2018-08-30

```
Counter({-1: 3715, 1: 736, 2: 414, 0: 252})
The silhouette score for  data: mega_pp is 0.3502188281967436
```

Table 3: Results of the 32k clustering, HDBSCAN

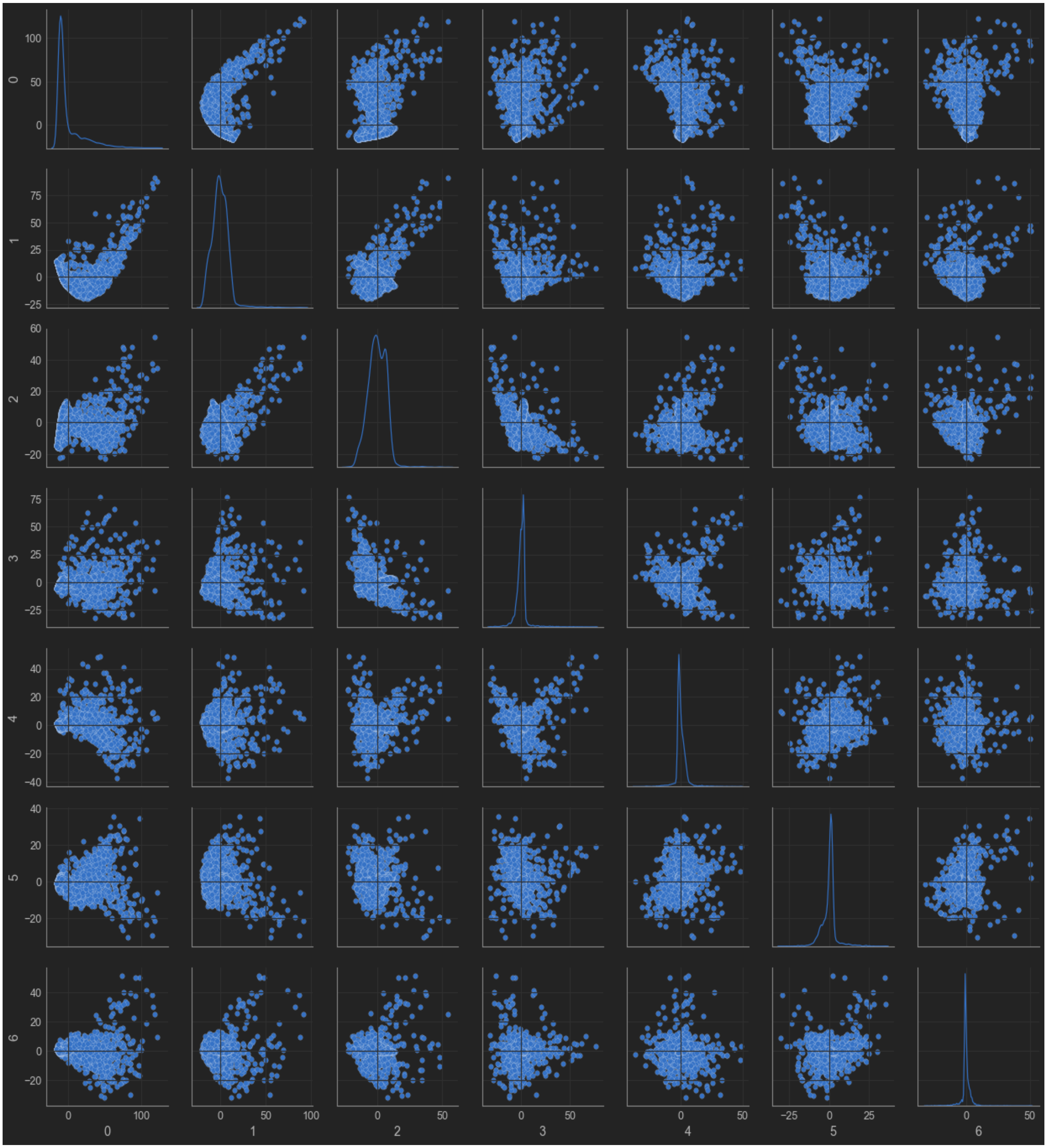


Table 4: Standardized distribution of the first 7 principle components. Note here that most of the retailers for the most cases lie close to one another and in almost all paired dimensions can be split in 4 a 6 clusters.