Practical Application 3

Machine Learning Assignment

David Cabornero Pascual
david.cabornero@alumnos.upm.es

June 18, 2022

1 Introduction

In this third assignment, a clustering problem is going to be handled. Concretely, it is going to be a dataset about *Online Shoppers Purchasing Intention*, in other words, about marketing. The main objective is to implement four main algorithms: *Hierarchical Clustering*, *K-means*, *K-medians* and *Gaussian Mixture*.

Finally, a discussion about the interpretation of results will be carried out, trying to find out metrics of quality and regarding what each cluster classify.

2 Problem Description

The dataset [4] used in this Assignment is from the Bahcesehir University of Istambul and provides useful information that would classify different types of customers navigating on certain web sites. The original one is a prediction dataset which classes were whether a certain customer ended up buying a product or not. Removing this class, this dataset provides us a classic problem: customer clustering. We have more than 12.000 instances and 17 attributes in this dataset:

- *Administrative*, *Informational* and *Product Related* represent the number of pages of each type visited by the potential buyer during their session.
- *Administrative Duration*, *Informational Duration* and *Product Related Duration* are related to the previous variables and describe the total time spent on each category.
- *Bounce Rate* is the rate of visitors that enter on the page and don't do any extra requests.
- *Exit Rate* is the proportion of page views which were the last on the web page.
- *Page Value* is the average value of all the pages visited during the user's session.
- *Special Day* indicates the closeness to a specific special day (such as Mother's Day or Valentine's Day), when it is more likely that the buyer buys a product. It only affects the dates before the special day occurs and it is a number between 0 and 1.

- *Operating System*, *Browser*, *Region*, *Traffic Type* are some nominal values related to how the customer accessed the web page.
- *Visitor Type* is a nominal variable with three values: *New Visitor*, *Returning Visitor*, *Other*.
- *Weekend* is a boolean value indicating whether the session was carried out on a weekend and *Month* indicates the proper month of the session. For some reason, this dataset does not include data of January and April, so there are only ten months.

Some of these variables are strongly correlated or are worthless, even before knowing anything about its behaviour with the algorithms. These problems are going to be handled in the next section.

3 Methodology

For this assignment, *scikit-learn* [3] library of Python[1] will be utilized to define the clusters and the respective plots because of problems with cache memory in *Weka*. On the contrary, *Weka* [5] is going to be used for preprocessing purposes and to represent the final plots that will help us to identify different clusters.

Regarding the attributes, some of them have been removed since they have not much relevance for this problem: *Operating System*, *Browser*, *Region* and *Traffic Type*. Furthermore, the *Duration* attributes were strongly correlated with the number of times that each customer visits each kind of page. The same problem has been found with *Bound Rate*, that had the strongest correlation with *Exit Rate*. Besides, some of the algorithms that have been used cannot handle categorical and numerical attributes at the same time on *scikit-learn*, so *Month*, *Weekend* and *Visitor Type* have been one-hot-encoded with the *NominalToBinary* method in *Weka*. With all this modifications, now the algorithm is going to handle 19 attributes, but the attribute *Month* is going to be merged again before showing the plots in order to make easier its lecture.

Since some of the algorithms can present problems when we work with different scales, a general standardization is going to be carried out. With this modification, the algorithm *K-means* should work better and the variance shown on *Gaussian Mixture* should be more accessible (this part will be discussed on the next sections).

Four classifiers are going to be used. Firstly, we have **Hierarchical Clustering**, in which we are going to use the agglomerative approach. Two important features must be chosen: the linkage function and the number of clusters. Although, there is not an easy way to get a Knee Plot in *scikit-learn* for this algorithm, so we are going to choose these parameters regarding the corresponding dendrograms generated. Four linkages will be tested: *simple*, *complete*, *average*, and *Ward's method*.

In the second place, we have the **K-means** and **K-medians** method, which the unique important attribute now is the number of clusters that is going to be generated. This number was decided with a Knee Plot, based on the *inertia* attribute[2] for the first case and on the

[1] Except the algorithm used for K-medians, which was extracted from *pyclustering* library [2].

[2] The inertia is the sum of the squared distances of samples to their closest cluster center

Index	AdminDur	InfoDur	ProdDur	ExitRates	PageValues	SpecialDay	Weekend
8635	0.0	0.0	784.88	0.0	360.95	0.0	0.0
9238	2657.32	1949.17	29970.47	0.03	0.0	0.0	0.0
5152	2629.25	2050.43	43171.23	0.02	0.76	0.0	1.0
8071	3398.75	2549.37	63973.52	0.03	0.0	0.0	0.0
12108	0.0	0.0	1107.92	0.01	226.68	0.0	0.0
7925	504.25	1665.07	8768.48	0.03	1.83	0.0	0.0
6165	2407.42	434.3	23050.1	0.01	0.0	0.0	0.0

Table 1: Some samples isolated in at least one dendrogram. The bold ones appear in more than one.

sum of squared errors for the second one. The final results were 11 clusters for *K-means* and 10 clusters for *K-medians*.

Finally, we have the **Gaussian Mixture** model with an Expectation-Maximization (EM) approach. The kind of distribution chosen was the *diag* one, where each component has a variance for each attribute. With this, we only have to decide again the number of components, and this decision will be based again on a Knee Plot, but in this case we are going to use the BIC (Bayesian Information Criterion) metric. The number of chosen components is 4.

4 Results

Regard to hierarchical clustering, we can see the four dendrograms based on the four linkage criteria mentioned before on Figure 3 and Figure 4. When the number showed below has a parenthesis, it means the number of samples on this branch. On the contrary, when we have not parenthesis the number is the index of the unique sample on the branch. The interpretation is going to be shown in the next section, but the best model that has been chosen is the Ward's method with 8 clusters. We can also see the isolated samples of the tree on Table 1.

The months are one of the most relevant attributes for almost all the clustering methods, so multiple scatter plots of *Number of Cluster against Month* are found in Figure 1.

Regarding K-means, K-medians and Gaussian Mixture, we can see in the Scree plots of Figure 2 that the Python library *KneeLocator* has chosen 11 clusters as the best result for K-means, 4 clusters for K-medians and 4 components for Gaussian Mixture. Furthermore, the centroids of each cluster of K-means is included in Table 2, as well as the medians of each cluster of K-medians in Table 6.

Finally, the mean and variance of each component of Gaussian mixture is included in Table 3 and Table 4 respectively. Moreover, we have an special table with the variances related to months on Table 5.

Cluster	Admin_Dur	Info_Dur	ProdRelated_Dur	PageValues	SpecialDay
0	71.2	30.7	812.3	4.0	-0.0
1	93.7	45.8	1802.3	7.1	-0.0
2	109.3	35.7	1253.4	7.6	0.0
3	37.9	22.7	986.0	2.0	0.7
4	75.2	35.5	1058.3	6.8	0.0
5	125.9	38.7	1117.0	8.6	0.0
6	81.8	28.2	962.2	6.9	0.0
7	106.7	35.5	1272.7	5.9	0.0
8	59.1	20.5	1213.4	3.4	-0.0
9	78.9	45.5	1217.6	4.1	0.0
10	16.9	2.4	471.0	0.9	0.2

Table 2: Relevant attributes of the Centroids in K-means.

Comp.	Admin_Duration	Info_Duration	ProdRelated_Duration	PageValues	SpecialDay
0	82.5	33.8	1221.4	5.8	0.1
1	78.2	38.8	1113.7	7.5	0.0
2	78.9	45.5	1217.6	4.1	0.0
3	16.9	2.4	471.0	0.9	0.2

Table 3: Means of each component in Gaussian Mixture (regarding only relevant variables).

Comp.	AdminDur	InfoDur	ProdDur	ExitRates	PageValues	SpecialDay	RetClient
0	0.98	0.89	0.95	0.98	0.85	1.16	0.92
1	1.09	1.46	1.31	0.97	2.07	0.0	1.48
2	1.45	2.04	1.22	0.93	0.35	0.0	0.89
3	0.25	0.01	0.19	2.01	0.09	2.46	0.04

Table 4: Variance of each component in Gaussian Mixture before reversing the normalization(regarding only relevant variables).

Comp.	Feb	Mar	May	June	Jul	Aug	Sep	Oct	Nov	Dec
0	0.0	1.18	1.13	1.23	0.0	1.23	1.23	1.22	1.14	0.0
1	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.08	0.13
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 5: Variance of months in Gaussian Mixture before reversing the normalization.

Cluster	AdminDur	InfoDur	ProdDur	ExitRates	PageValues	RetVisitor
0	53.95	16.0	234.26	0.01	18.16	0.0
1	0.0	0.0	493.75	0.05	0.0	0.0
2	16.4	9.23	1076.45	0.01	5.67	0.0
3	46.2	0.0	280.05	0.01	0.0	0.0
4	68.88	0.0	1578.86	0.01	27.16	1.0
5	0.0	0.0	1665.7	0.06	0.0	1.0
6	0.0	0.0	222.2	0.08	0.0	1.0
7	32.92	6.62	745.31	0.03	0.0	1.0
8	0.0	42.4	1140.8	0.0	0.0	1.0
9	0.0	0.0	5764.37	0.06	0.0	1.0

Table 6: Relevant attributes of the Medians in K-medians.

5 Discussion

5.1 Hierarchical Clustering

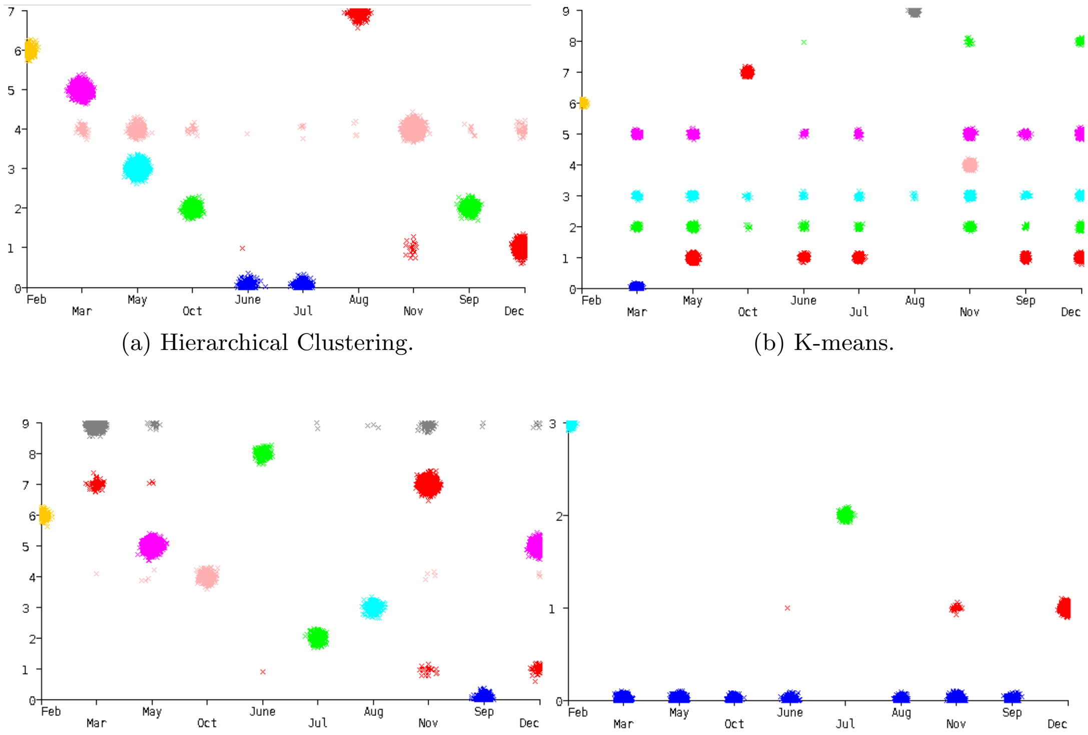
The reason to select the Ward's method is simple: the rest agglomerate more than 12.000 elements in one cluster after developing more than 10 clusters. Which is the reason behind this behaviour? Fortunately, this dendrogram returns the index of these samples (repeated over different linkages), so we can try to figure out which is the problem.

As we can see in Table 1, the results shown on the table are generally outliers. Firstly, the 99th percentile of *PageValues* is 85.5. Regarding the duration, the 95th percentile of *ProductRelated_Duration* is 43.000, the respective one in the *Administrative_Duration* is 348 and in *Informational_Duration* is 195. With this information, it's reasonable to think that average, complete and single linkage are more sensible to outliers than the Ward's method, that seems more robust (the tree is more extended and we are unable to see a single node).

Is there any reasonable explanation to justify this behaviour? On *Hierarchical Clustering*, this means that there is a huge dissimilarity between the outliers and the rest of the elements. It makes sense with simple-linkage, it is sensitive to outliers since it understands the dissimilarity as the distance between this point and the nearest point in the cluster. The outlier is a greater issue if the problem remains with complete linkage, though. It means that the distance between the outlier and the furthest point of the cluster is a huge value too. Indeed, if the maximum and minimum distance between the cluster and the outlier are great, it is reasonable to expect the same behaviour with the average-linkage. Regarding the Ward's method, this kind of dissimilarity has been probably strongly penalized because it takes into account how different are the points inside the cluster.

Once the Ward's method has been chosen, we can see a point where there is not more new branches after a long distance difference: between 125 and 150. That is because the selected Hierarchical Clustering has had 8 clusters, since *scikit-learn* does not provide an easy tool to make a more elaborated method (unlike the rest of algorithms).

Regarding the scatter plot of Figure 1, we can see that most of the clusters separates the months. The months that are more similar and consequently remain together are, firstly June



(a) Hierarchical Clustering.

(b) K-means.

(c) K-medians

(d) Gaussian Mixture

Figure 1: *Number of Cluster vs Month* scatter plots of different algorithms.

and July, secondly September and October and finally December and November are slightly mixed. It's interesting how nearer months are grouped on the same cluster; it stipulates a certain similarity between them, since *Month* is a categorical variable and we are not giving information about proximity between months.

Furthermore, we can observe too that the cluster 4, apart from November and a bit of December, has a small part on May. It's possible that November and May has similarities regarding the patterns of the customers.

5.2 K-means

This algorithm is based on finding points associated to each cluster, called centroids. Each centroid represent the most centric point of each cluster based on the average between all the points in the cluster. That's because each median can be interpreted as a representative of each class, so we can expect that the majority of the points in the cluster would have this structure.

In this case, the corresponding scatter plot of Figure 1 reflects that in general the clusters do not separate the months, except February, October and August, that are clearly separated by Clusters 6, 7 and 9 respectively. Cluster 4 has only samples from November, but there are more clusters that have results about November.

What can we induce from the centroids? There is some information that we can extract regarding the Table 2. For instance, the cluster 1 has the largest *ProductRelated_Duration*, which means that the average user of this cluster is the one who spend more time looking for products. The cluster 10 is one of the most representatives, with the smallest durations and the worst *PageValues*, showing that these users come from some low-quality pages and do not spend much time on the current page. Another interesting case is the cluster 3, where we have again low quality pages, but now a special day is near and the duration on the page is better. Finally, we can see what is the particularity of Cluster 4, that only has a subset of November: it seems like we have a good *PageValues* while the duration on the web page is more or less on the average.

5.3 K-medians

As we can see with the scree plot of Figure 2, *ScreeLocator* chooses in this case 4 clusters, but we can see that this plot is not as smooth as the rest. This algorithm computes the error as the Sum of Squared Errors, and even taking the average of 10 executions the scree plot is too irregular. Looking carefully the results, one can see the case of 4 clusters as a local optimum, but the 10 cluster case could be a better optimum (looking the previous clustering methods it makes sense).

Again, we can observe that the corresponding plot *Cluster-Month* in Figure 1 reveals that the clusters are organized basically by months. Some interesting conclusions can be extracted: May and December are on the same cluster, a small part of December shares cluster with November and November and March clusters are very mixed. It can give us some information about the months that are more similar taking into account the rest of the variables.

Regarding the medians extracted from Table 6, we have more or less the same information than in the previous algorithm: each median is a representative of each cluster. Now, the medians are not based on the average of the points of the cluster, but on the median of the points of the cluster. It makes the result less sensitive to outliers, which could be a big issue (as we saw with the Hierarchical Clustering).

With this information, we could know for instance that the main difference between Cluster 7 and 9 (that are mixed with March and November) is the time spent in pages related to the products (eight times larger on Cluster 9). We can also extract on which months we have more new visitors (generally, the second half of the year) or the high *PageValues* (average value of the pages visited during the session) on the clusters that correspond to September and October.

5.4 Gaussian Mixture

We obtained some majoritarian clusters with all the algorithms, but in this case the number of samples classified as the first type is more than 80%, and we have to take this into account.

With this model, each cluster has a Gaussian distribution associated. Since the *diag* type has been selected, this Gaussian distribution has a diagonal matrix as the covariance matrix. With this information, the Table 3 is the mean vector of each distribution (simplified with the most interesting attributes) and the Table 4 is the same case, but with the diagonal of the covariance matrix.

In this case, the diagonal shown is the one that we obtain before reverting the standardization. This is not the real diagonal, but the real diagonal takes into account two different issues: the variance of the density function itself and the variability of the attribute. With the standardization, we can forget the second issue and focus on the first one: the largest the variance is, the flatter our distribution will be (regarding a certain attribute).

Regard to the Component 3, we can observe that it has the same characteristics as the Cluster 10 in K-means: people who come from bad web pages and spend a small amount of time on the page. Observing Table 3 and Table 4, we can see that the means are more or less similar (except *PageValues*) and they have more or less the same variances, except *SpecialDay*, but the majority of the samples has a 0 value on this attribute (which means that this variable is not relevant or that we are regarding a discrete variable). The most reasonable question is why the Component 0 has associated the 80% of the samples.

The answer is on the Table 5, where we can see that on the majority of months the Component 0 has a large value and regarding the Components 1 and 2 the variance is almost zero. With this information and the scatter plot extracted from 1, we can see that again the clusters are mostly separated by months. Concretely, the most different months that deserve a cluster are July, December and February.

6 Conclusion

In this assignment, four clustering algorithms have been applied to a customer dataset in order to obtain distinct results: *Hierarchical Clustering*, *K-means*, *K-medians* and *Gaussian Mixture*. For these tasks, *scikit-learn*, *pyclustering* and *Weka* have been used.

The most important variable on this dataset is the *Month*, which separated the cluster in three out of the four cases (*K-means* is the exception). It gives us two important characteristics of the dataset: each month has a certain distribution of attributes and some of the months are more similar than the rest. For instance, we observed similarities between November and May, November and March, September and October, June and July and so on.

Some of the dendrograms generated by the Hierarchical Tree had leaves of one sample after a few number of branches. This happened with simple, complete and average linkage. Finally, these nodes were the outliers of the dataset, causing that some of the clusters would be composed by only one sample. Therefore, one possible solution is to remove a certain number of outliers.

Another algorithms are sensitive to outliers too. Concretely, *K-means* algorithm can present issues when there are outliers or the scale. The second one was solved with standardization, but the outliers are still a problem. In this case, an alternative could be *K-medians*, which is more or less similar but more robust to the outliers.

K-means is the only algorithm that divided the dataset more or less independently of the *Month* variable. Nonetheless, the another three algorithms have not given the same results, neither in terms of number of clusters nor in terms of the features of the clusters. It has given us a more richer information about our dataset and the different ways that we can employ to divide it.

Finally, we have to regret that the library *scikit-learn* have not provided more properties that inform the user about the quality of the algorithms in each case. However, finding metrics to evaluate our results with clustering is harder than with supervised learning.

References

[1] C. Bielza Lozoya and P. Larrañaga Mújica. *Data-driven computational neuroscience : machine learning and statistical models.* Cambridge University Press, Cambridge, 2020.

[2] A. Novikov. PyClustering: Data mining library. *Journal of Open Source Software*, 4(36):1230, apr 2019.

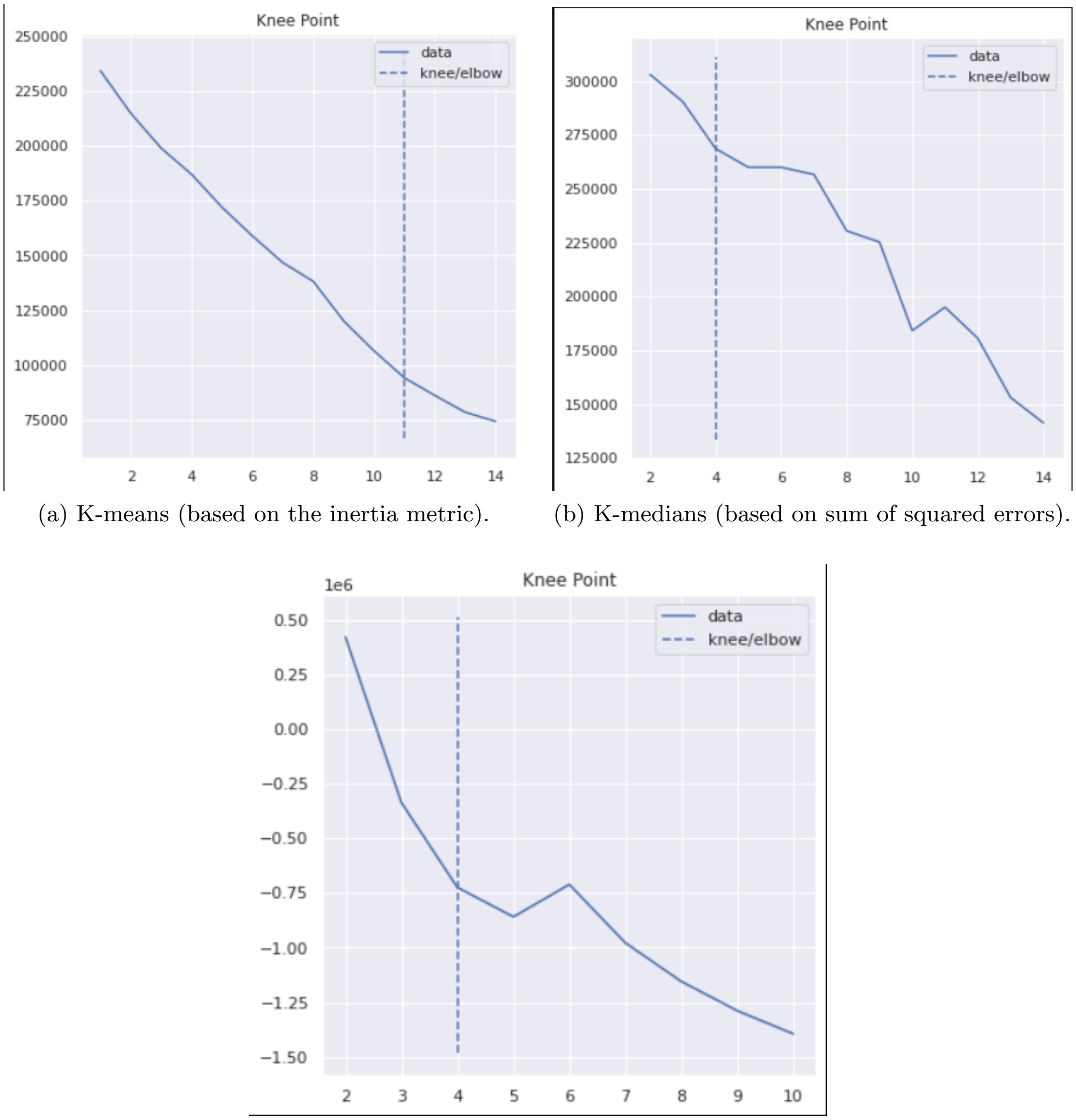
[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Kastro. Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks. *Neural Computing and Applications*, 31(10):6893–6908, 2018.

[5] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, San Francisco, 2nd edition, 2005.

A Some extra images

Here we can find some extra images with higher size, allowing the reader a better lecture.

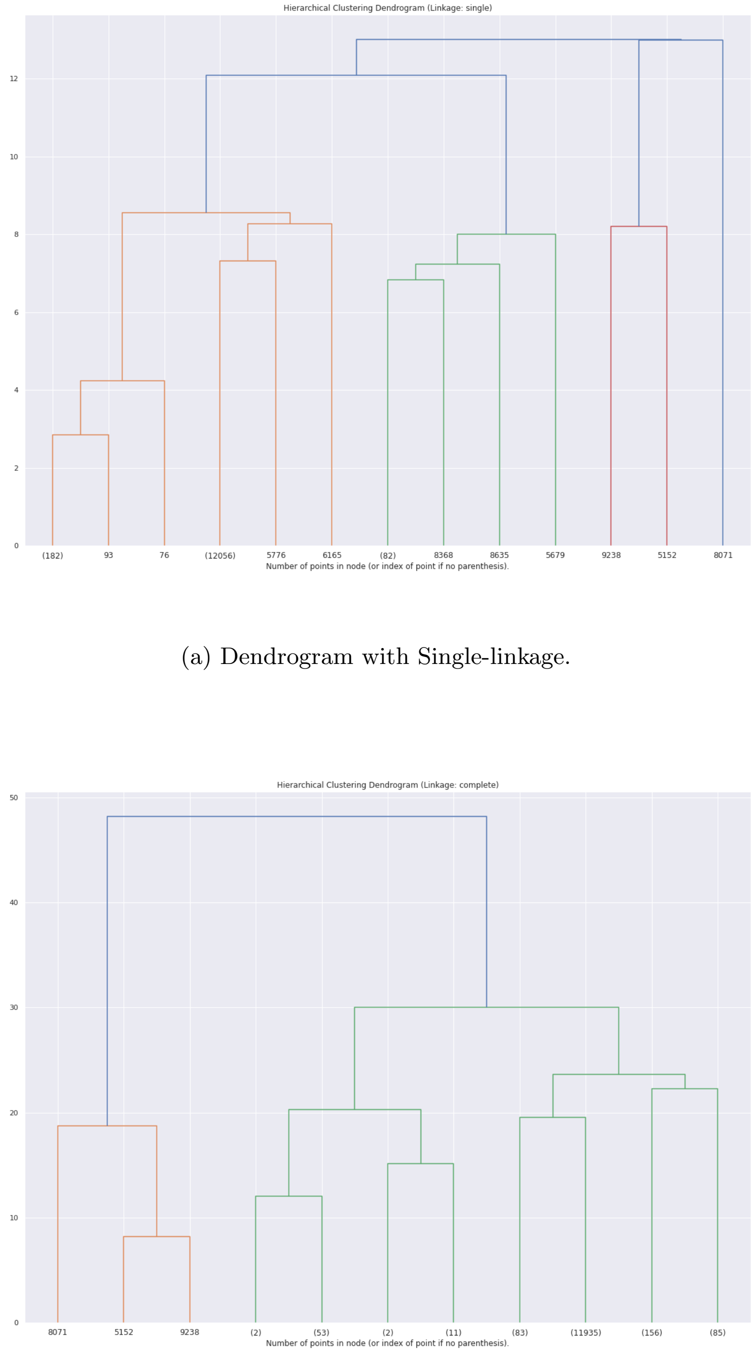


(a) K-means (based on the inertia metric).

(b) K-medians (based on sum of squared errors).

(c) Gaussian Mixture (based on BIC).

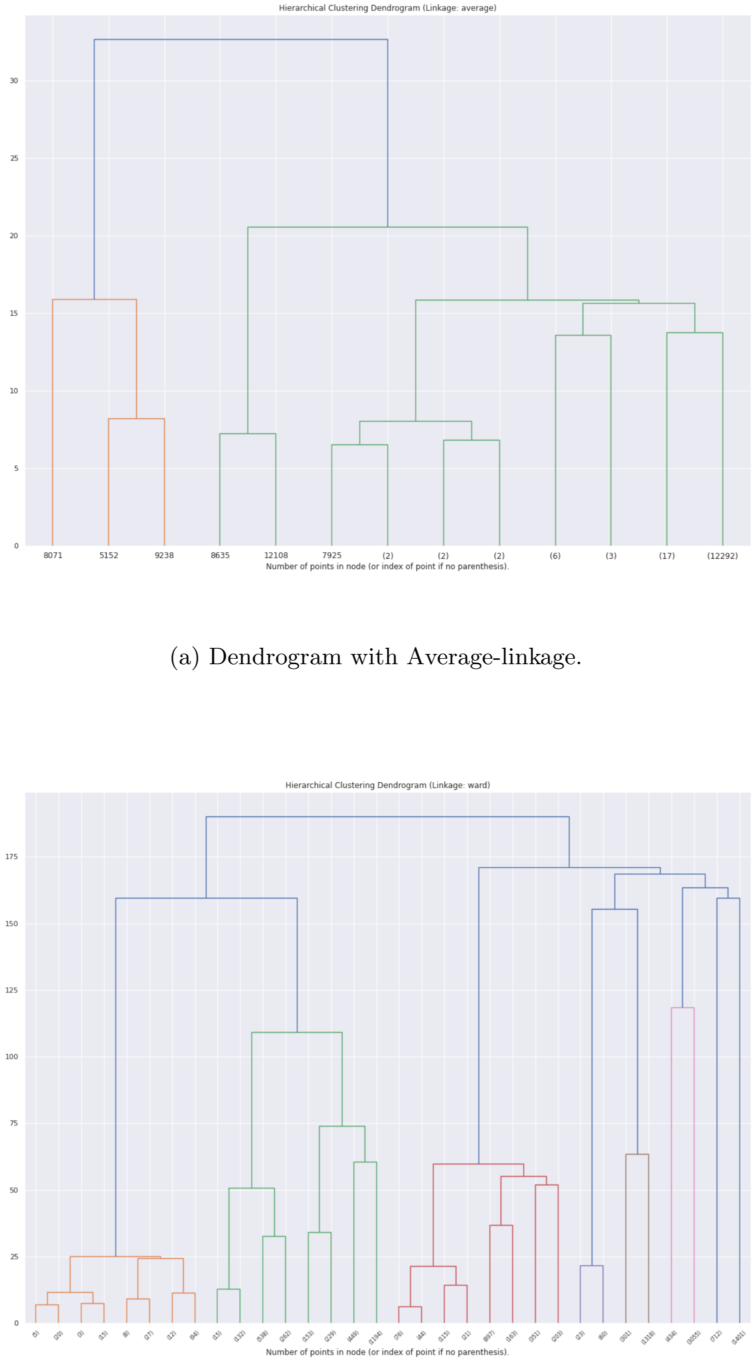
Figure 2: Results of the corresponding Scree Plots.



(a) Dendrogram with Single-linkage.

(b) Dendrogram with Complete-linkage.

Figure 3: Dendrogram of the Hierarchical Clustering with some linking methods (1).



(a) Dendrogram with Average-linkage.

(b) Dendrogram with Ward's method.

Figure 4: Dendrogram of the Hierarchical Clustering with some linking methods (2).