Unsupervised learning

Marta Fernández González, Alberto González Delgado

We have applied k-means and Hierarchical Unsupervised Learning Algorithms in order to classificate neurons based on eight morphological features of their axons. These Algorithms have permitted us to analyse unlabelled data and find the internal architecture of the data. We have selected the parameters that lead into the best classification of our data.

1. Introduction

Machine learning (ML) is considered a component of artificial intelligence although it endeavours to solve problems based on historical examples [1]. ML techniques can be classified into supervised and unsupervised techniques. Supervised ML uses labelled data which allows to guide the training process. On the other hand, unsupervised ML works on unlabelled data so that the labels must be discovered by the learning algorithm [2]. Unsupervised ML algorithms are suitable for creating labels in the data that are subsequently used to implement supervised learning tasks [1].

Clustering is an unsupervised approach to ML. It is a method for finding cluster structure in a given data set. This cluster structure is characterised by the greatest similarity within the data in the same cluster and the greatest dissimilarity between different clusters. Clustering methods can be divided as probability model-based and nonparametric approaches. The probability model-based approaches use a mixture likelihood approach to clustering. For nonparametric approaches, clustering methods are based on a function of similarity or dissimilarity measures. Clustering methods can be divided into partitional and hierarchical methods [3].

Partitional methods represent the data using finite clusters. The most popular partitional method is the k-means clustering [3].

The k-means algorithms find k clusters following these steps: First, k points are selected as initial centroids and all the points are assigned to the closest centroid. Then, the centroid of each cluster is recomputed, and the rest of the points are assigned again to the closest centroid. This last step is repeated until the centroids don’t change [4]. The drawback of this method is that k-means need to be given a number of clusters and the cluster number is, generally, unknown. In this case, it can be used as validity indices (Bayesian information criterion [BIC], Akaike information criterion [AIC], etc) for defining the number of clusters in k-means. These indices are supposed to be independent of clustering algorithms [3].

BIC has been used for model selection among a given group of models based on a likelihood function of the model [5]. It determines the predictive ability of a model based on a measure. It identifies the most stable model from a set of models. It is chosen as the model that has lower BIC [6]. BIC is defined as:

$$BIC = -2 \times ln(\hat{L}) + ln(n) \times k$$

L: log-likelihood
N: number of observations
K: number of parameters estimated

SSE (Error Sum of squared) is defined as the sum of squared differences between each observation and its group's mean. It can be used as a measure of variation within a cluster. An SSE equal to 0 means that all the data in a cluster is identical. Its formula can be written as: [7]

$$SSE = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

Silhouette Coefficient or Score is a metric used to calculate the goodness of a clustering technique. Its values go from 1 (means clusters are well apart from each other and clearly distinguished), 0 (the distance between clusters is not significant) to -1 (clusters are assigned in the wrong way). [8]

$$s = \frac{b - a}{\max(a, b)}$$

a: average intra-cluster distance
b: average inter-cluster distance

Hierarchical clustering does not require any knowledge about the appropriate number of clusters beforehand. It creates a tree-structured clustering where sibling clusters partition the observation covered by their parent. Hierarchical structuring provides a view of data at several levels of partitions [9]. Hierarchical clustering can be performed in an agglomerate or divisive fashion. Agglomerative Hierarchical clustering Technique initially considers each data point as an individual cluster. At each iteration, the similar clusters merge with other clusters until only a single cluster remains. This technique can be visualized using a Dendrogram, which is a tree-like diagram that records the sequences of merges or splits.

There are certain approaches which are used to calculate the similarity between two clusters. In this work we used four different approaches:

1. MIN: Also known as single-linkage algorithm, the distance between clusters is defined by the distance between their closest members.
2. MAX: Also known as the complete linkage algorithm, the distance between clusters is defined by the distance between their furthest members. This method is computationally expensive.

3. Group Average: The percentage of the number of points of each cluster is calculated with respect to the number of points of the two clusters if they were merged.
4. Ward's Method: It specifies the distance between two clusters, computes the sum of squares error (SSE), and successively chooses the next clusters based on the smaller ESS. Ward's Method seeks to minimize the increase of SSE at each step. Therefore, minimizing error.

2. Methods

Watch supplementary material scripts.

Dataset

We have used a dataset of neurons' features saved in a csv document. The dataset is obtained from
https://computationalintelligencegroup.github.io/neurostr/doc/measures/prebuilt.html.

The dataset contains information about branch length averaged over the branches of the axon ("length.avg"), the total number of bifurcations in an axon (N_bifurcations), a measure of branching asymmetry, averaged over the branches of the axon (partition_assymetry.avg), axonal width (width), axonal height (height), branch bifurcation amplitude, averaged over the branches of the axon (remote_bifurcation_angle.avg), ratio of the branch length and Euclidean distance between its endpoints, averaged over the branches of the axon (tortuosity.avg), and the number of bifurcations from the soma to a branch, averaged over the branches of the axon (centrifugal_order.avg).

Visualising and preprocessing

We have used pandas library to read and realize exploratory data analysis. In order to visualise the data, we have used matplotlib.pyplot library.

The analysis of the data has been realised using sklearn library. In order to get the data in the same scale, we have standardised the scale of the data using sklearn.preprocessing.StandarScaler.

Principal Component Analysis - PCA

When the data has a lot of features, it makes performing the analysis very challenging. We have used skleanr.decomposition.pca to reduce our 8-dimensional data into 2 dimensions so that we can plot and understand the data better. After the dimensionality reduction, we obtain two new components that don't have meaning and are just the two main dimensions of variation.

K-means Clustering

First, we have applied sklearn.cluster.KMeans to a NumPy's array created with the dataset and using a different number of clusters (2, 3 and 4). In order to visualize the different clustered dataset, we have plotted each cluster with a different colour.

In order to determine the number of clusters that fits the better with the data, we have applied the SSE method, and BIC (sklearn.mixture.GaussianMixture) and Silhouette score (sklearn.metrics.silhouette_score). With the aim to make the analysis more interpretable, we have plotted the results of each method in graphs of score versus the number of clusters.

Hierarchical Clustering

We have applied scipy.cluster.hierarchy functions. Linkage function performs hierarchical/agglomerative clustering. We performed linkage analysis with four different approaches:

- Single: Perform single/min/nearest linkage on the condensed distance matrix.
- Complete: Perform complete/max/farthest point linkage on a condensed distance matrix.
- Average: Perform average/UPGMA linkage on a condensed distance matrix.
- Ward: Perform Ward's linkage on a condensed distance matrix.

In order to visualise the different clustered dataset, we used the scipy dendrogram function that plots the hierarchical clustering as a dendrogram.

An interesting number of clusters is found graphically in the dendrogram finding the largest horizontal space that doesn't have any vertical lines. It represents that the clusters have a significant distance between them. To corroborate the appropriate number of clusters, we also performed a validity score analysis for each linkage method. We calculated the silhouette score for the number of clusters between 2 and 10. Then, we performed Agglomerative clustering with the number of clusters with the highest score using Scikit-Learn AgglomerativeClustering.

3. Results and discussion

Watch supplementary material scripts.

Visualising and preprocessing

Figure 1 corresponds to two cells from the first part in the exploratory data analysis that we have realised using the pandas library.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 219 entries, 0 to 218
Data columns (total 8 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   length.avg                      219 non-null    float64
 1   N_bifurcations                  219 non-null    int64  
 2   partition_asymmetry.avg         219 non-null    float64
 3   width                           219 non-null    float64
 4   height                          219 non-null    float64
 5   remote_bifurcation_angle.avg    219 non-null    float64
 6   tortuosity.avg                  219 non-null    float64
 7   centrifugal_order.avg           219 non-null    float64
dtypes: float64(7), int64(1)
memory usage: 13.8 KB
```

```
df.duplicated().sum()
```

```
0
```

Figure 1. Recording of the script. It shows the results of printing information about the data and the searching for duplicated data.

The first cell contains information about the non-nulls values count and type of data in each column. In RangeIndex we were told that there are 219 rows in each column, so as the non-null count is also 219, we determined that there is no missing data. It is an important aspect due to several methods that do not work with null values. In addition, all the data is numeric (float64 or int64). In the second cell, we have searched for duplicate values. There are no duplicated values and neither columns repeated so it is no need to clean the data from duplicate values.

With the aim of visualising the distribution of the data and the relation between the features, we have plotted both the distribution histogram of all the data and the scatter matrix (Figures 2 and 3).

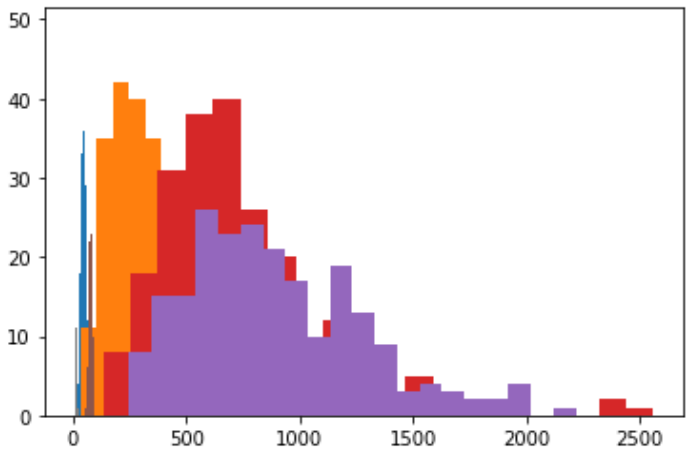


Figure 2. Histogram of the data distribution plot.

In Figure 2, we can see that the features, each of them represented by a colour, are distributed in different scales. This is also observed in the Figure 3.

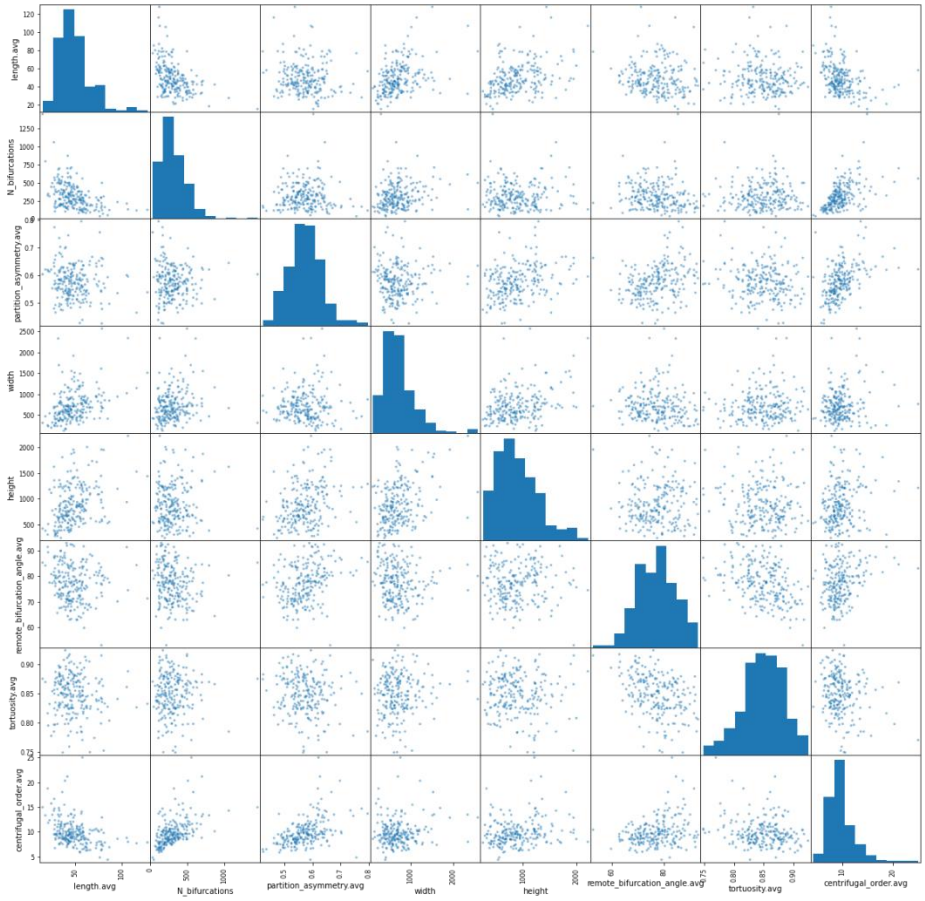


Figure 3. Scatter matrix plot of all the features.

In Figure 3, we can visualise how the features are related to each other. In addition, in the diagonal, we can see the data distribution of each feature. Notice that there is no clear group separation between variables in the scatterplots. Due to this, and in order to ensure that all variables have the same weight when comparing data points, we have standardised the data, using sklearn.preprocessing.StandadScaler.

The distribution of the data after the standardisation process is shown in Figure 4. It can be seen that the data now has a similar scale. As a conclusion, all the variables have the same weight, and it can be used to apply them to a clustering algorithm.

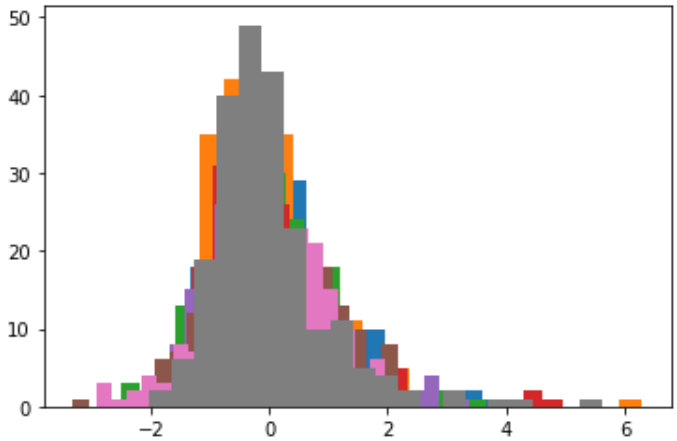


Figure 4. Histogram of the data distribution plot, after standardising the data.

As there is no clear separation between variable scatter plots (Figure 3) and we have 8 dimensions of data, we decided to perform a decomposition of the data (pca) into two components to simplify the plotting of the labels and try to clarify the interpretation.

K-means Clustering

As a preliminary visualisation, we have applied the k-means algorithm to our standardized data defining different numbers of clusters (2 to 5). Then, we have plotted (Figure 5) the result using colours to differentiate the groups created by the algorithm. We have used the decomposed data for the graphs.

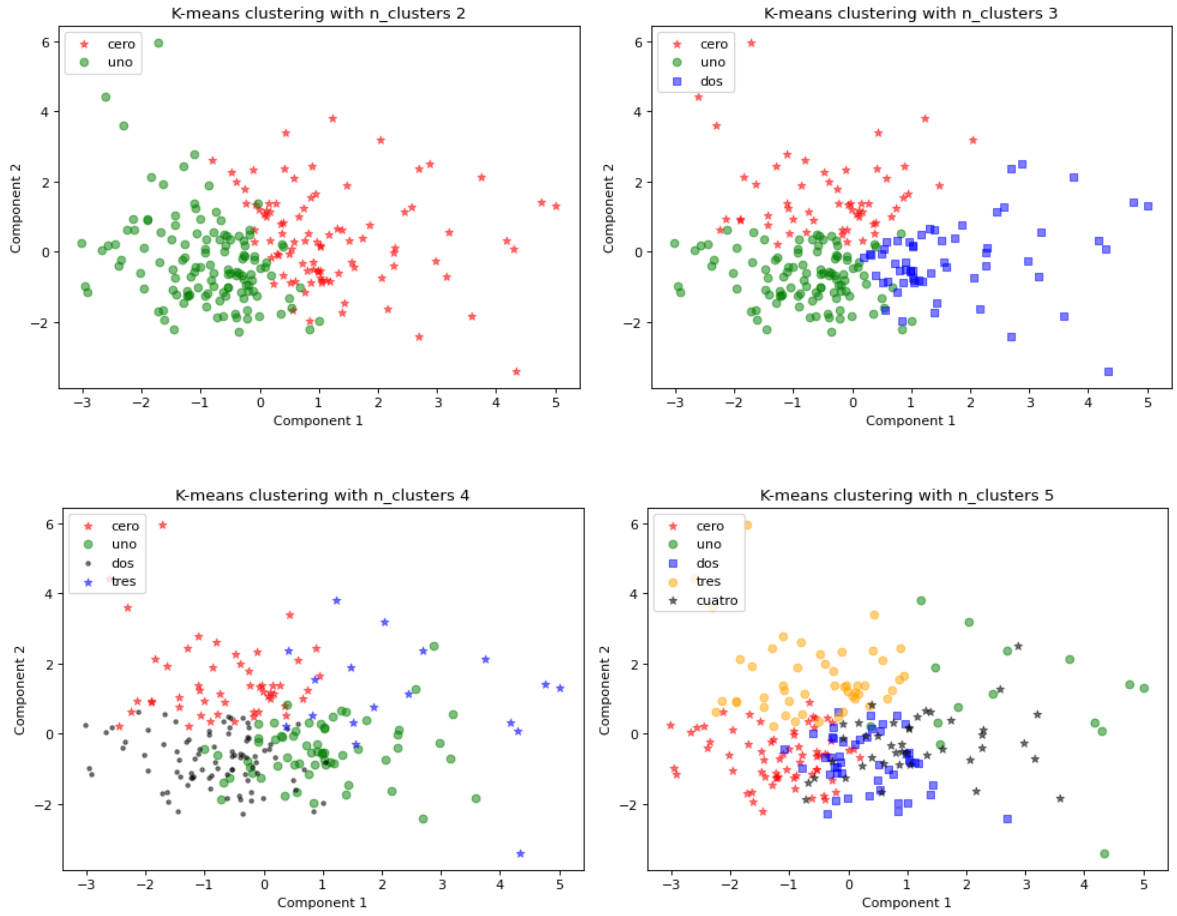


Figure 5. Clusters obtained for 2 features (2 clusters upper-left; 3 clusters upper-right; 4 clusters down-left, 5 clusters down-right)

As it was said above, the drawback of k-means clustering is that it needs to be given a number of clusters [3]. In Figure 5 it is shown that we do not obtain very clear and differentiated groups, and that this differentiation is worse as the number of clusters is higher. We have applied three validity indices with the aim to find the number of clusters that gets the best partitional model of the data. In Figure 6 we have shown the graph of SSE versus the number of clusters (from 1 to 10).

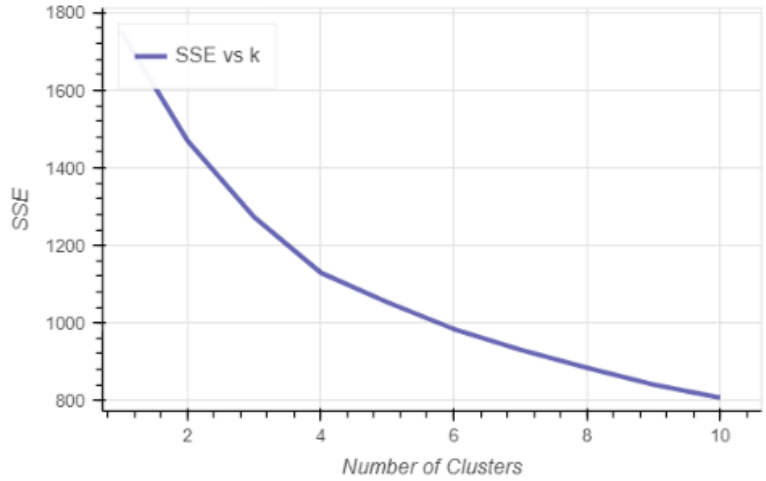


Figure 6. Values of SSE obtained for each number of clusters

As it is expected, the SSE decreases for a higher number of clusters (Figure 6). The best improvement of SSE value occurs when the number of clusters is 2, 3 and 4. It means that 2 is the number of clusters that lead into the best improvement of the error. For more than 4 clusters, the improvement is again lower although it is lower from 3 than 2 and from 4 than 3.

The following figure (Figure 7) is a diagram of BIC vs number of clusters (from 0 to 50).

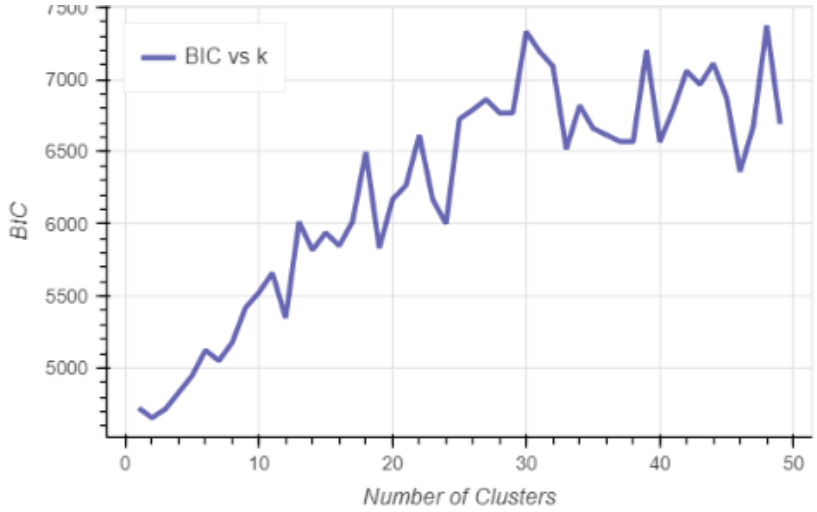


Figure 7. Values of BIC obtained for each number of clusters

BIC values grow while the number of clusters is higher (Figure 7). So, it means that the data fits better with a lower number of clusters. The best (lowest) BIC value obtained in this analysis is while the data is divided into 2 clusters. Regarding the other candidate (4 clusters), BIC is higher, indicating a worse model. Nevertheless, for 4 clusters, BIC is still one of the lowest, compared to the rest of the data. Moreover, the BIC values seem to be high, indicating that the partition of the data is not as well as it could be expected.

The last validity index used is Silhouette score (Figure 8). Here we can see Silhouette score versus the number of clusters (from 2 to 10).

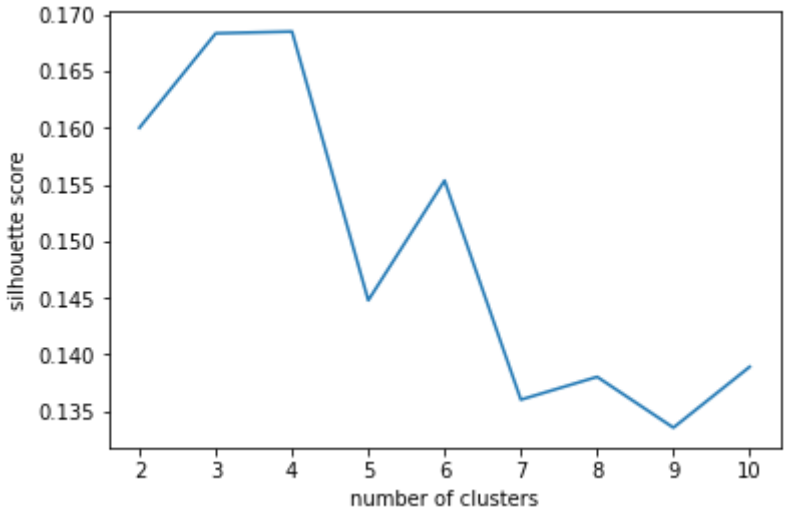


Figure 8. Silhouette scores obtained for each number of clusters

The Silhouette score is positive, indicating that the clusters have been assigned in the correct way. Nevertheless, the Silhouette score is near to 0, indicating that the distance between clusters may not be significant. According to the Silhouette score, 3 and 4 clusters fit the best with our data (Silhouette score ~0.167). In addition, 2 clusters have a Silhouette score near to the maximum values obtained (Silhouette score = 0.160). So, based on Silhouette score, the best partition of the data is using 3 and 4 clusters. Nevertheless, using 2 clusters we also obtain a good partition of the data. For a higher number of clusters, the partition is worse. This result is similar to the ones obtained for BIC.

The three validity indices used (BIC, SSE and Silhouette score) are consistent in the number of possible best number of clusters (between 2 and 4). However, the best number of clusters is not clear. So, as a conclusion we can provide a range of possible numbers of clusters that permit the best partition of the data. The data should be partitioned between 2 and 4 clusters. Comparing the partitional plots in Figure 5, we have selected 3 clusters as the best way to partition the data (Figure 5 upper-right).

Despite being the best clustering obtained, we can see that well differentiated clusters are not observed. It is indicated by the general low Silhouette scores obtained, that the distance between clusters is not significant for any of 2-10 clusters. On the other hand, as a pca analysis was necessary to observe the data, we cannot interpret the characteristics of the neuron clusters obtained, as the axes represent a combination of all the features decomposed into two dimensions and the components are meaningless.

Figure 9 shows the Silhouette analysis for k-means clustering using 3 clusters. Each colour corresponds to one cluster.

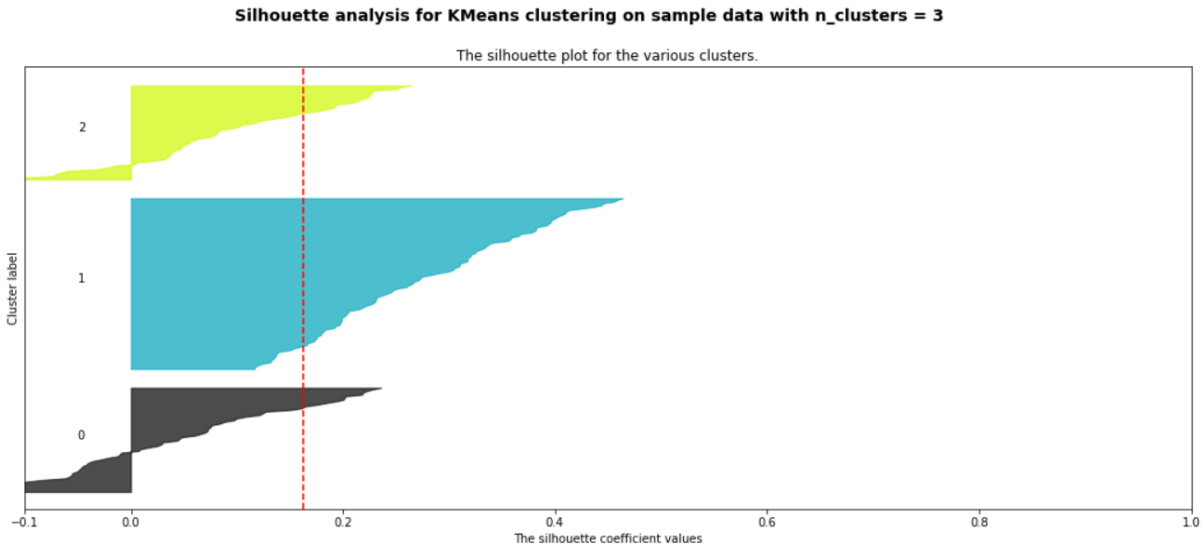


Figure 9. Silhouette analysis for k-means clustering using 3 clusters

We can see that, in general, all the data has a positive Silhouette score (Figure 9). The data assigned to the second cluster (blue) is totally well-assigned. The other two clusters (grey and yellow) contain some data assigned in the wrong way (negative Silhouette score). The mean Silhouette score is ~0.17 corresponding to the best value obtained in the previous analysis (Figure 8).

However, it is important to note that looking at the values observed in the validity scores plots (high in both SSE and BIC and low in Silhouette score), it indicates that the clusters assigned are not significant and k-means may not be the best option to assign labels to the data.

Hierarchical Clustering

We performed linkage analysis to our scaled data and plotted a dendrogram with four different approaches. For each linkage method, we analysed the dendrogram and decided an interesting number of clusters for an Agglomerative cluster.

Ward linkage method:

Figure 10 shows the dendrogram obtained when we apply the Ward linkage method for hierarchical clustering on the data. We combined silhouette score and graphical analysis to get a good number of clusters for this linkage method. Figure 10 contains the agglomerative clustering using Ward method.

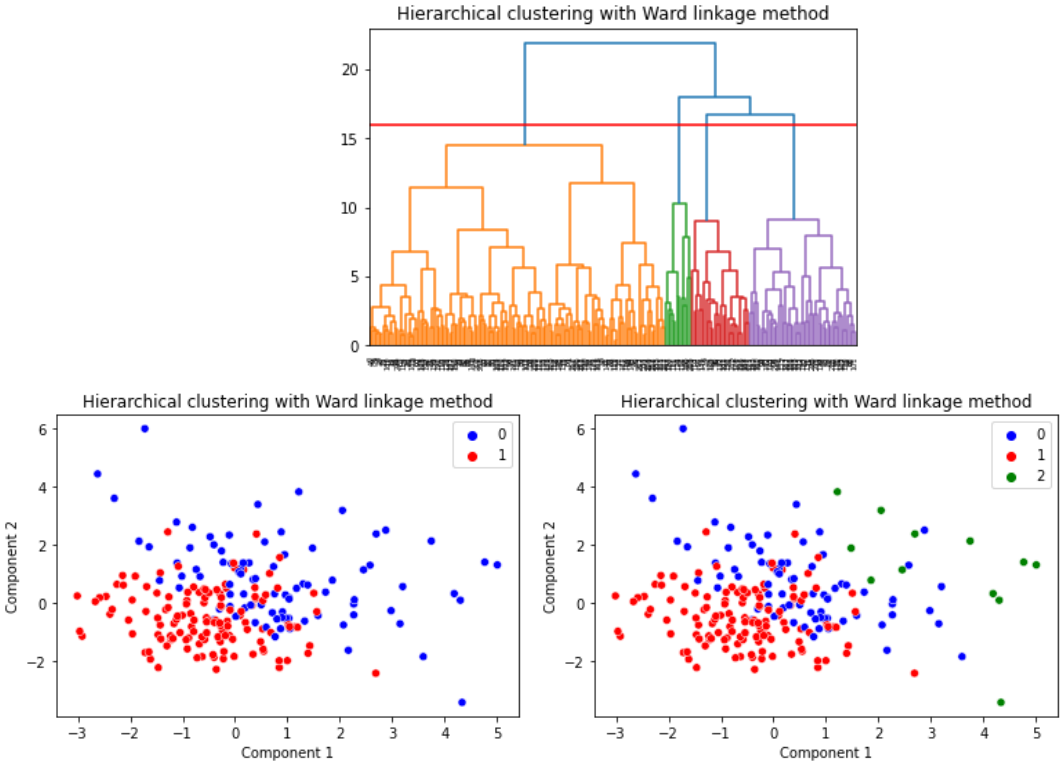


Figure 10. Dendrogram for Hierarchical clustering performing ward linkage method. Agglomerative clustering for 2 and 3 clusters using ward linkage method.

For the ward linkage method, based on the best silhouette score, we decided to create 2 clusters, obtaining a value of 0.164. We also partitioned the data for 3 clusters as the silhouette score is 0.1637. As we can see in the data plot the analysis is better for 2 clusters and we reasonable differentiate 2 groups even though they are not really separated.

Single linkage method:

Figure 11 shows the dendrogram obtained when we apply the Single linkage method for hierarchical clustering on the data. At the right we can see the partition of the data performing agglomerative clustering using this method.

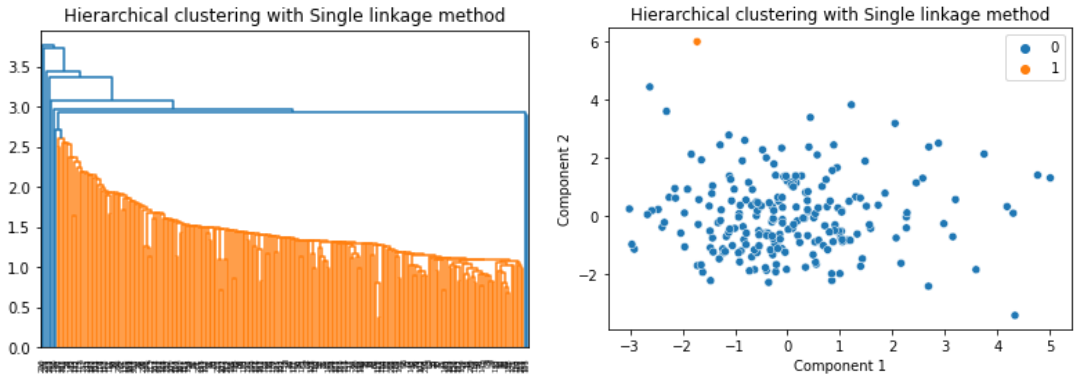


Figure 11. Dendrogram for Hierarchical clustering performing single linkage method. Agglomerative clustering for 2 clusters using ward linkage method.

For the single linkage method, we decided to create 2 clusters and we obtained a silhouette score of 0.46. At first, we think this linkage method is good as the silhouette score is high, but this is because the method is not able to identify groups and it is generating a unique big cluster and classifying one or two outliers as a different group. The single linkage method is not good to clusterize our data.

Complete linkage method:

Figure 12 shows the dendrogram obtained when we apply Complete linkage method for hierarchical clustering on the data. At the right we can see the partition of the data performing an agglomerative clustering using this linkage method.

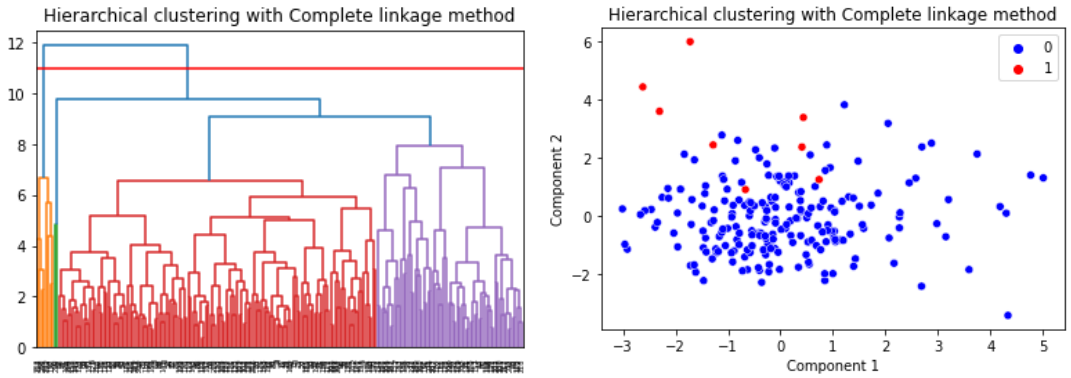
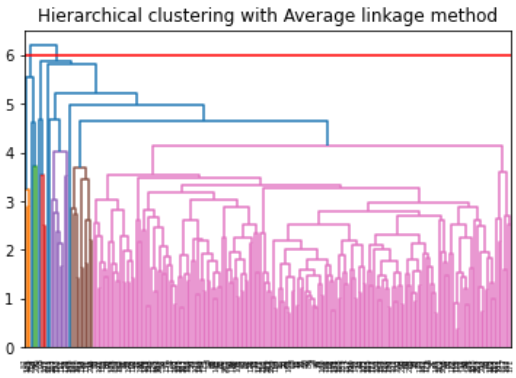


Figure 12. Dendrogram for Hierarchical clustering performing complete linkage method. Agglomerative clustering for 4 clusters using ward linkage method.

For the complete linkage method, we decided to create 2 clusters obtaining a silhouette score of 0.29. The method shows a bad partition of the data as we cannot properly differentiate the two groups.

Average linkage method:

Figure 13 shows the dendrogram obtained when we apply the Average linkage method for hierarchical clustering of the data. Down, we can see the partition of the data performing an agglomerative clustering using this linkage method.



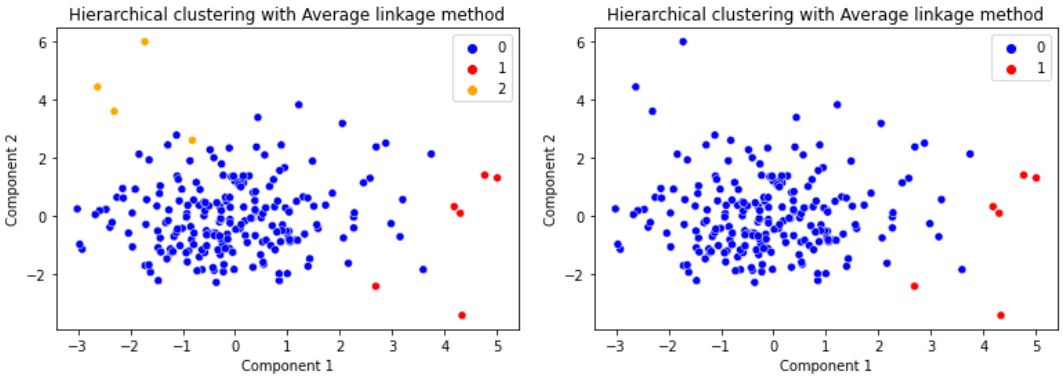


Figure 13. Dendrogram for Hierarchical clustering performing average linkage method. Agglomerative clustering for 6 clusters using ward linkage method.

For the average linkage method, we created 3 and 2 clusters obtaining silhouette scores of 0.335 and 0.40 respectively. As seen in the partitioned data graphs, this method also creates a big group and other small groups that do not differentiate well.

Comparing the hierarchical clusterization of our data, it seems that Ward linkage method performs a reasonable clustering. Although, it is the one with the lowest silhouette score, we think the best Hierarchical approach is using ward linkage method. This is because using other linkages methods we obtain a big cluster with almost all data points and another group with very few data points, which is not good.

Figure 14 shows the Silhouette analysis for Hierarchical Agglomerative clustering using Ward linkage method and 3 clusters. Each colour corresponds to one cluster.

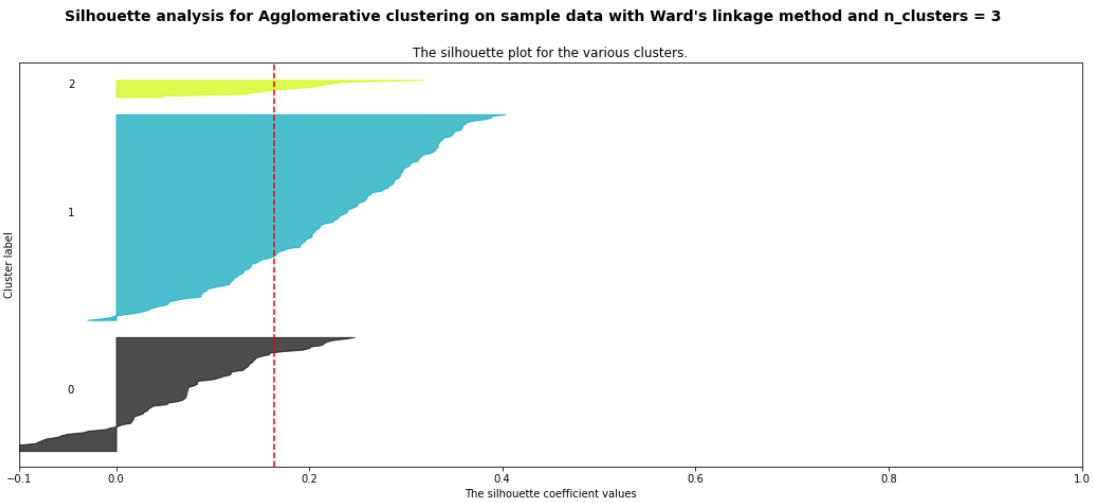


Figure 14. Silhouette analysis for Hierarchical Agglomerative clustering using Ward linkage method and 3 clusters. Each colour corresponds to one cluster.

We can see that, in general, all the data has a positive Silhouette score (Figure 14). The data assigned to the second cluster (blue) is almost totally well-assigned and the first cluster (yellow) is totally well-assigned. The other cluster (grey) contain some data assigned in the wrong way (negative Silhouette score). The mean Silhouette score is ~0.16.

4. Conclusion

Comparing the best k-means and hierarchical models we see that we obtained a similar differentiation performing both k-means clustering and Hierarchical Agglomerative clustering with Ward linkage method using 3 clusters.

Although, we concluded that the best clustering method to apply to this data is partitioning the data (k-means) because, despite having almost the same silhouette score, it is slightly higher, and the partition plot is better for k-means.

Because we used pca, the clusters are represented over two dimensional components that are meaningless, which combined with the fact that there are no well differentiated partitions makes interpretation impossible.

As we have explained before, all cluster methods may not differentiate groups well enough. It would be necessary to collect more feature information about the neurons that correlates better, which will help to perform a better clustering analysis without pca and to be able to interpret the data clusters.

Supplementary material:

For the code of the analysis read the collab notebook:
https://colab.research.google.com/drive/1lDJAfeCZzALuVoFn6tXuamKf5WX9e6qC?usp=sharing

5. References

1. Berry W., Wah-Yap,B., Mohamed, A. (2020), Supervised and Unsupervised Learning for Data Science, *Springer*, ISBN 978-3-030-22474-5. https://doi.org/10.1007/978-3-030-22475-2
2. Palacio-Niño, J., Berzal, F. (2019), Evaluation Metrics for Unsupervised Learning Algorithms. Available on: https://arxiv.org/pdf/1905.05667.pdf [Visited on: 10/11/2022]
3. Sinafa,K., Miin-Sheen, Y. (2020), Unsupervised k-Means Clustering Algorithm. DOI: 10.1109/ACCESS.2020.2988796
4. Steinbach, M., Karypis, G., Kumar, V., (2000), A Comparison of Document Clustering Techniques. Available on: https://conservancy.umn.edu/bitstream/handle/11299/215421/00-034.pdf?sequence=1&isAllowed=y [Visited on : 12/11/2022]
5. Lee, Eun Ryung; Noh, Hohsuk; Park, Byeong U. (2014). Model Selection via Bayesian Information Criterion for Quantile Regression Models. Journal of the American Statistical Association, 109(505), 216–229. doi:10.1080/01621459.2013.836975
6. López Sánchez, N-. (2019) Aplicación y comparativa de cuatro modelos de clustering para datos GTEx (TFM Universitat Oberta de Catalunya for Master en Bioinformática y Bioestadística) . Available on: https://openaccess.uoc.edu/bitstream/10609/90626/6/vlopezsanchTFM0119memoria.pdf [Visited on: 12/11/2022]
7. Error Sum of Squared. Available on: https://hlab.stanford.edu/brian/error_sum_of_squares.html [Visited on: 12/11/2022]
8. Bhardwaj, A., (2020), Silhouette Coefficient Available on: https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c [Visited on: 12/11/2022]

   Talavera, L., (1999), Feature Selection as a Preprocessing Step fot Hierarchical Clustering, *Proceedings of the Sixteenth International Conference on Machine Learning,* Available on: https://www.researchgate.net/profile/Luis-Talavera-3/publication/2272714_Feature_Selection_as_a_Preprocessing_Step_for_Hierarchical_Clustering/links/0deec53aec6ca5dbf1000000/Feature-Selection-as-a-Preprocessing-Step-for-Hierarchical-Clustering.pdf [Visited on: 12/11/2022]
9. Galarnyk, M. (2017). PCA using Python (scikit-learn). Available on: https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60 [Visited on 14/11/2022]
10. Reddy-Patlolla, C., (2018). Understanding the concept of Hierarchical clustering Technique. Available on: https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec [Visited on 14/11/2022]
11. SciPy documentation(Hierarchical scipy). Available on: https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html
12. Sampaio, C., (n.d),Definitive Guide to Hierarchical Clustering with Python and Scikit-Learn. Available on: https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/ [Visited on 14/11/2022]