Third Assignment

Sergi Carol and Balbina Virgili

May 15, 2018

Introduction

The aim of this third assignment is to successfully develop an analytical study of different methods that can be applied to obtain the clustering analysis of a desired dataset. To be able to do this aim, we have decided to first, briefly theoretically study several methods that can be used to develop a cluster analysis and then, using our already acquired knowledge of the first assignment, implement them using *KNIME* tool. Our objective by the end of this project is being able to obtain, show and objectively compare the results performed for each of the possible implemented clustering method. This way, we expect to acquire theoretical and practical knowledge of *Hierarchical Clustering* and *Clustering using K-means*.

This document begins with an small explanation of our chosen dataset, follows with a short introduction of basics about clustering and, then, it continues with our developed workflow and the explanation of the different methods that we have used to develop the cluster analysis of the data. Finally, this document ends with some conclusions of our experience and the results obtained during the development of the assignment.

1 Dataset

The dataset that we have chosen for this assignment is titled *wine* [1]. This dataset contains the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars, by determining the quantities of 13 constituents found in each of the three types of wines. More concretely, it has the following attributes for each wine listed:

- **Wine**. Class identifier (1,2,3).
- **Alcohol**. Range values [11.03 - 14.83]
- **Malic acid**. Range values [0.74 - 5.8]
- **Ash**. Range values [1.36 - 3.23]
- **Acl** (Alcalinity of ash). Range values [10.6 - 30]
- **Mg** (Magnesium). Range values [70 - 162]
- **Phenols**. Range values [0.98 - 3.88]
- **Flavanoids**. Range values [0.34 - 5.08]
- **Nonflavanoid phenols**. Range values [0.13 - 0.66]
- **Proanth** (Proanthocyanins). Range values [0.41 - 3.58]
- **Color intensity**. Range values [1.28 - 13]
- **Hue**. Range values [0.48 - 1.71]
- **OD** (OD280/OD315 of diluted wines). Range values [1.27 - 4]
- **Proline**. Range values [278 - 1680]

2 Introduction to Clustering

Clustering [2] is considered the classification of patterns, such as observations, data items, or feature vectors, into groups, also known as clusters. It has been used in many contexts so it is widely considered as a very useful step in explanatory data analysis because it tries to find the similarity of the data by trying to group them by similarity groups through analysis. Then, intuitively, items within a valid and defined cluster are more similar to each other than a pattern belonging to a different cluster. However, clustering is likely to conclude with differences in results and contexts as many different acceptable groups classification could be obtained from a given data.

A small study of different clustering techniques that can be applied on our chosen data has been developed. To be able to demonstrate that our acquired knowledge on this techniques is the right one, we have used *KNIME* to implement the following algorithms with our desired data. The combination of both methods could usually be used to obtain more accurate results, but this is out of out assignment scope.

- Hierarchical Clustering [3]
- K-means Clustering [4]

As we already have a classifier identifier of each wine, which divides them in three different groups, we desire to reproduce this already given classification with the rest of the data proportionated. Note that some error can be obtained because, as stated in the source web page [1], the initial data had 30 attributes but some of them have been lost. In spite of this, our initial hypothesis is to obtain three different clusters to classify all wines.

3 KNIME

After theoretically analyze the different algorithms mentioned to perform clustering, our objective is to implement them with *KNIME* in order to test which one gives a better clustering solution for our aims which are the following:

- Be able to predict from which region a wine is from by looking at their different attributes.
- Group the different wines by the kind of wine they produce.

In order to achieve this aim we will use two different clustering approaches, *Hierarchical and K means*.

To do so we will use the tool *KNIME*, which we have already used in a previous deliverable, to read the dataset, normalize it, and finally visualize the different clusters. The *workflow* that we implemented for being able to resolve and analyze our stated purpose is showed below.

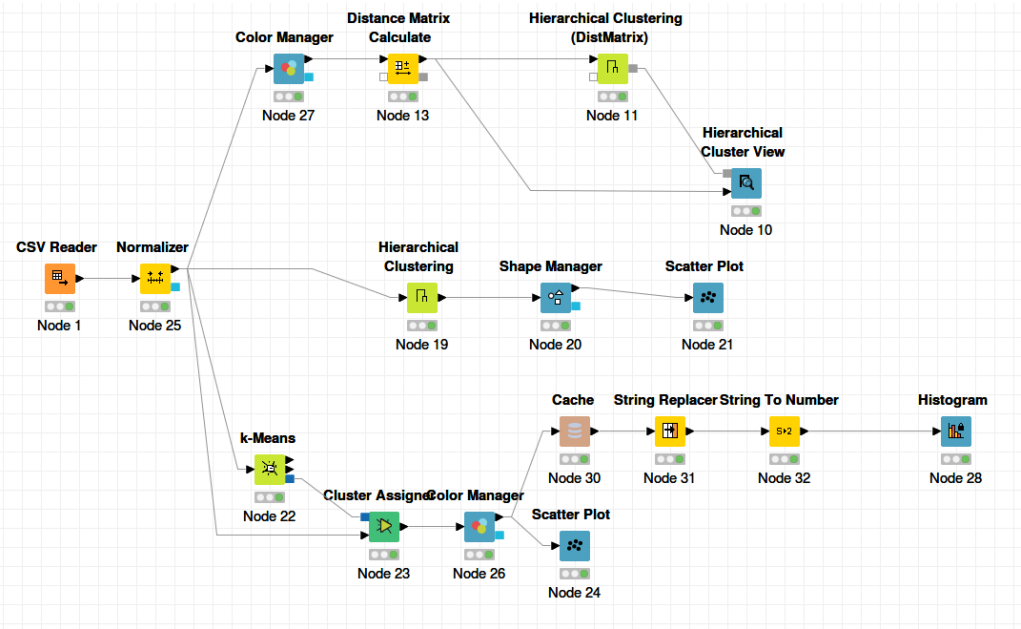


Figure 1: Workflow implemented with KNIME

As it can be seen, we first have needed to normalize the data of our dataset in order to develop the most accurate analysis, due to the different scale that the attributes have in our sample. That is why the normalize node for all the attributes of the dataset has been used. Then, we have implemented our data with the selected clustering algorithms, hierarchical and k-means. It can be noticed that we performed the hierarchical clustering twice, because we want to check if there are any differences between using the *Distance Matrix* version or not.

4 Hierarchical Clustering

Our first approach is to use the *Hierarchical* algorithm in order to obtain a cluster classification for our chosen dataset. As we have already explained, our ideal result would be to obtain three different clusters, that each of them represents one of the three different cultivars that a wine is extracted from. For being able to explore it, the following two methods for calculating hierarchical clustering are being implemented. The color manager node, which has assigned a color to each wine cultivar, has been added for both methods in order to show the results in more clearly way.

The Hierarchical clustering algorithm is useful to obtain different cluster classifications from a

given data, without any predefined number of clusters previously assigned. The initial state of this algorithm is that each point or individual of the set given is a cluster. Then, it computes until all different alternatives are calculated by merging two closest clusters. So we can note that is an iterative algorithm. The basic process, given a set of N items to be clustered and an N*N distance matrix, can be defined as follow [5]:

1. Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
3. Compute distances (similarities) between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

There are several ways to calculate the distance between items and clusters, which must be defined according to the needs of each problem. For our both implementations we have used *euclidean distance* [6] to compute items distances and *Complete Linkage* to compare clusters.

The several ways that the distances between clusters can be computed are breefly explained below:

- *Single Linkage*: the distance between one cluster and another one is considered to be equal to the shortest distance from any member of one cluster to any member of the other cluster.
- *Complete Linkage*: the distance between one cluster and another cluster is considered to be equal to the greatest distance from any member of one cluster to any member of the other cluster.
- *Average Linkage*: the distance between one cluster and another one is considered to be equal to the average distance from any member of one cluster to any member of the other cluster.

The following two implementations of the hierarchical algorithm have been needed to be able to obtain both, a dendogram extracted from a distance matrix and with no initial number of clusters to be defined, and a scatter plot extracted from the solution with three clusters initial stated.

4.1 Using Distance matrix

The workflow implemented for our first approach of the hierarchical algorithm is shown on the image below.

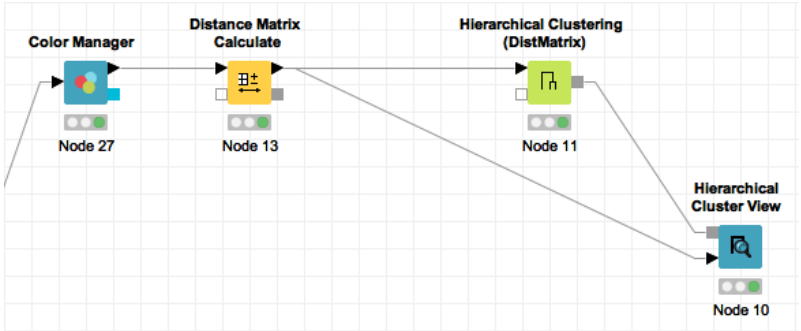


Figure 2: Workflow of Hierarchical Clustering using Distance matrix

As we can see, first the distance matrix for the items of our dataset has been calculated using *Distance Matrix Calculate*, by assigning the distance selection parameter to *Euclidean*, extracting the column *Wine* and appending a new column with name *Distance* with the output calculated on this node. Then, the already calculated matrix has been passed to the *Hierarchical Clustering (Distance Matrix)* node which input the algorithm using a distance matrix input column. The

linkage type for the algorithm has been assigned to *Complete Linkage*, as it has been initially defined. Finally, the *Hierarchical Cluster View* node has been added in order to show the dendogram with all combinations of closest clusters calculated. The results computed using this part of the implementation are showed below.

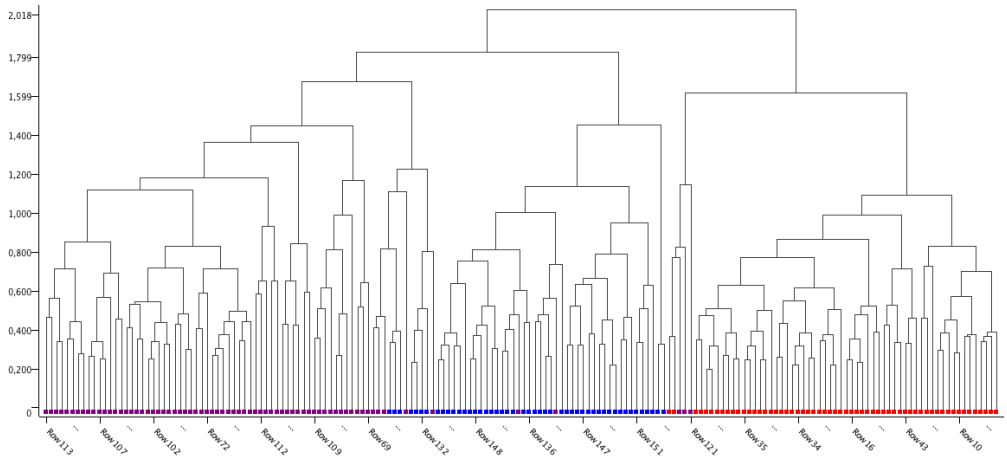


Figure 3: Dendogram

After deeply analyze the dendogram retrieved, we can realize that there are several different classifications of the wines that can be considered as good ones, such as 3, 5, 7 or even 24. But as our initial hypothesis was obtaining wines from each 3 cultivars, we can see that our solution represents quite well this classification because just few individuals (3-5 individuals) are not good classified on each cluster. We can easily compare this thanks to the color managed used for each range of *Wine* column of our dataset. From this solution obtained, we suspect that we can also extract some information about similar wines in terms of similar composition that is the reason why the classification of 5 cluster will be further inspected.

4.2 Without using Distance matrix

The workflow implemented for our second approach of the hierarchical algorithm is shown on the image below.

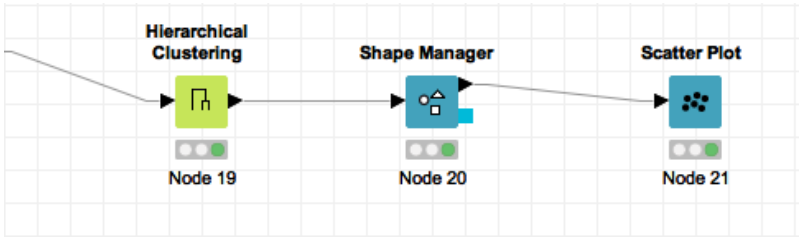


Figure 4: Workflow of Hierarchical Clustering without using Distance matrix

As we can see, no previous calculation of the distance matrix is needed for this implementation, as the *Hierarchical Clustering* node will compute the whole process to hierarchically cluster the input data. To be able to compare the new results obtained with the previous ones, we have set the same parameters as before. So we have assigned the distance function to *Euclidean*, the linkage type to *Complete* and we have extract the column *Wine* from the analysis. Furthermore, this *Hierarchical Clustering* node allows to set the number of output clusters to an specific number.

As we already decided that the most interesting number of clusters for our analysis is 3 and 5, we have performed the analysis twice to be able the scatter plot for both results.

Below this lines, we can observe that the dendogram obtained with this implemented approach is the same as the one obtained with the first one. Then, there is no difference on the results retrieved between the two implemented approaches of hierarchical clustering, and with this second approach we have been able to obtain our desired results by correctly configuring just one node, instead of three.

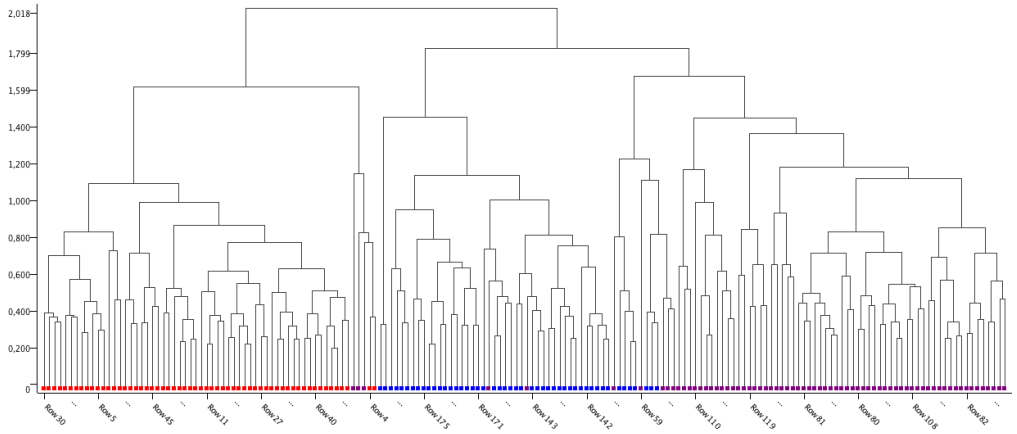
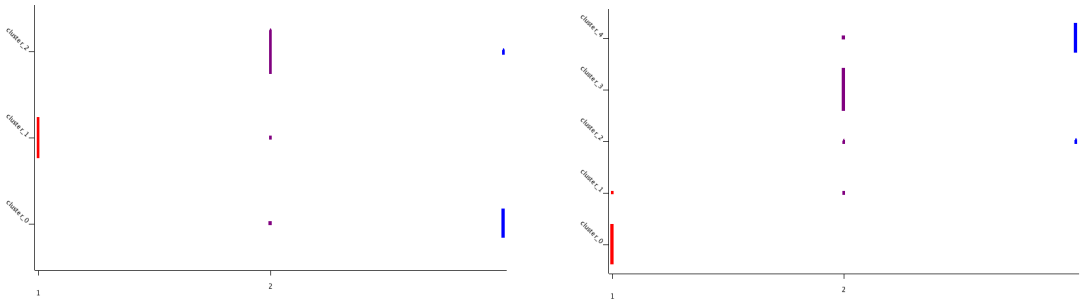


Figure 5: Dendogram

Below we can see the scatter plot output when the initial value of three and five clusters is defined.



(a) Scatter plot for 3 clusters

(b) Scatter plot for 5 clusters

On the one hand, the aim of the results retrieved with 3 clusters is to separate the different wines into the 3 different regions. As we can see, the clusters generated seem to be quite accurate with our chosen wine data as all wines of type *Wine 1* are well classified on *Cluster 1* and there are only few wines of *Wine 2* misclassified on *Cluster 0* and *Cluster 1* instead of *Cluster 2*, and few of *Wine 3* that are misclassified on *Cluster 2* instead of *Cluster 0*.

On the other hand, the aim of the results retrieved with 5 clusters is to extract a possible classification of the wines in terms of similar composition. With the results obtained, we can see that the implemented node has now retrieved 5 different clusters. Now, almost all wines of type *Wine 1* are assigned to *Cluster 0* and *Cluster 3* is all formed by wines of type *Wine 2*. Furthermore, on *Cluster 1* several wines of type *Wine 1* and *Wine 2* are joined and on *Cluster 2* some wines of *Wine 2* and *Wine 4* are classified. Finally, *Cluster 4* is composed by almost all wines of *Wine 3* and a few of *Wine 2*.

All in all, we can say that almost all wines that grown in the same cultivar have a similar composition, but there are also some wines of type *Wine 2* that have much the same composition with wines of other cultivars but no similarity is found on wines of type *Wine 1* and *Wine 3*.

5 Clustering using K-means

Our next approach is to use the *K-means* algorithm in order to check how it performs against the *Hierarchical* clustering, as with the previous clustering we will use the same dataset, and will use 3 and 5 clusters.

The K-means algorithm is used to try and minimize the square error. Although the goal of a cluster is to create an unknown amount of cluster by itself with the given data, this becomes impossible to do with the *K-means* approach, but instead the algorithm requires to give either the final number of clusters or the centroids of the cluster.

The algorithm works in two steps, the first one is to define **K** centroids, one for each cluster, and the second one is two assign the data to the nearest centroid. It is important to note that K-means is also an iterative algorithm, meaning that once the data has been assigned a new centroid is calculated for each cluster and the procedure begins again. This is due to the fact that the algorithm is significantly sensitive of the initially random selected centroids, and iterating through the algorithm a number of times reduces the effect [4]. Two sum up the steps taken are:

1. Choose K centroids at random in the data plane.
2. Assign each point to the closest centroid.
3. Recalculate the centroids.
4. Repeat steps 2 and 3 until convergence or *max_ iter*.

Below we can see an image of our implementation in *Knime*.

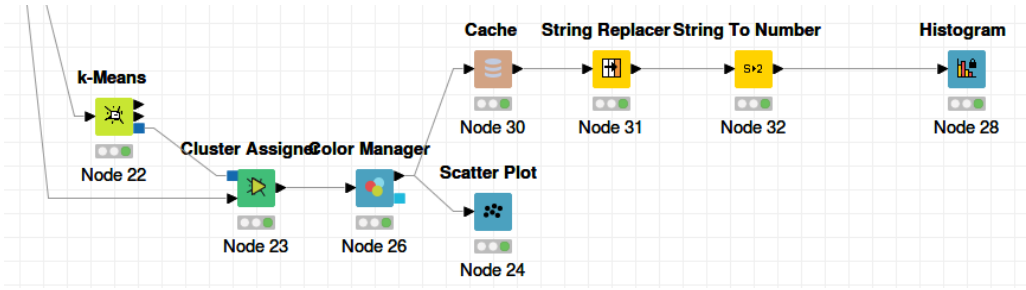
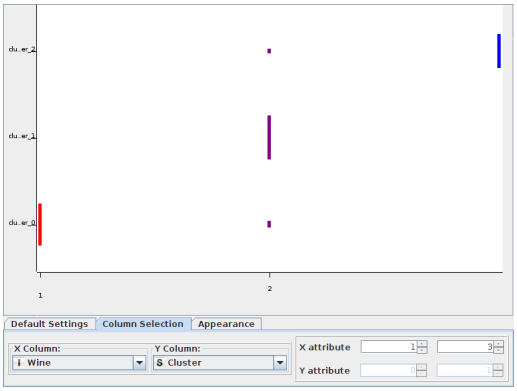


Figure 7: K means workflow

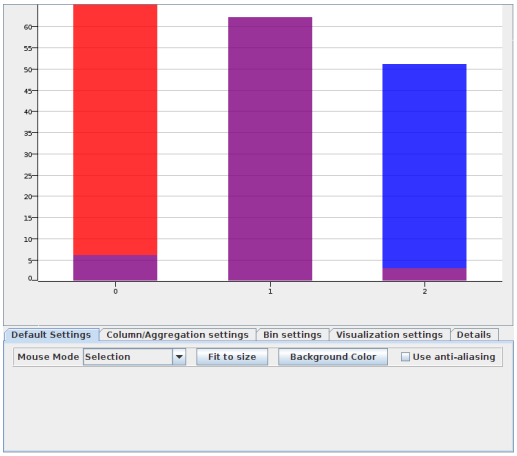
5.1 3 Clusters

With the 3 clusters our aim is to separate the different wines into the 3 different regions, to do so we have removed the *Wine* attribute from the cluster so to not lead the cluster algorithm into making decision based on that attribute. In this case we also set the number of maximum iterations to 99 in order to ensure that convergence is achieved. Then we assign the output to a color manager in order to color the 3 different wines for a better visualization of the output. finally we visualize the data through an scatter plot, and an Histogram (hence why we use the *String Replacer* and *String to Number*).

Below we can see the scatter plot and the histogram output.



(a) Scatter plot for 3 centroids



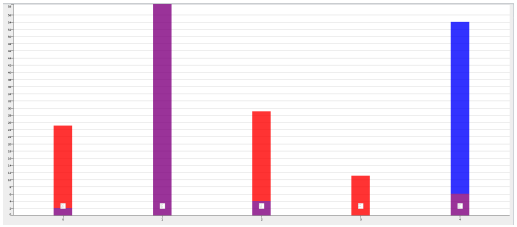
(b) Histogram for 3 centroids

As we can see the clusters appear to be quite accurate with the wine data. Only **3** wines misclassified from *Wine 2* to *Cluster 2* and **6** wines from *Wine 2* to cluster 0, while cluster 1 has no misclassified elements. All in all we can consider this cluster to be quite a success.

5.2 5 Clusters

The next approach is to use 5 clusters, since with the Hierarchical cluster we saw the possibility of there actually being 5 clusters, we do this is order to find likeliness between the different wines.

(a) Scatter plot for 3 centroids



(b) Histogram for 3 centroids

It is quite clear that in cluster 1 we have indeed an unique wine which is only produced in the region for *Wine 2*, meanwhile it seems that the region of *Wine 1* produces a variety of different types of wines which also does *Wine 2*, last but not least, it seems that *Wine 3* only produces one kind of wine which also appears to be produced by *Wine 2*.

6 Conclusions

With this paper we hope to have expressed our learning procedure through the clustering techniques, as well as showing how our two initial objectives, separating wines by region, and separating them by type, have been accomplished by both algorithms. Both algorithms have advantages and disadvantages, some of them are:

- Hierarchical clustering performance becomes quite bad with a large number of individuals since it has a cubic complexity computation.
- On the other hand it can create as many clusters as necessary, and its on the user to interpret how many clusters are needed.
- K-means performance is good regardless of the number of data that we have.
- But it required previous knowledge of the number of clusters which are going to be used.

As such we do believe that using the two techniques might be the better approach, hierarchical in order to detect the number of clusters and k-means to check the grouping of the data.

It is also worth noting that we could also use some methods such as the silhouette method [7] to estimate the number of clusters needed for our data, in our case we have not used such methods since we already know our target cluster numbers.

Both approaches seem to provide similar results in terms of accuracy when grouping the data, as well as we have the same accuracy on both approaches for the hierarchical clustering. Yet, as we have explained before, the hierarchical works well on datasets with low amount of data, while k-means performs better on large datasets.

When explaining the results it is quite clear how we can definitely separate the wines by region using clustering, in both approaches, hierarchical and k-means clustering, the wines are nearly perfectly separated by region. If we try to separate them by type, by using 5 clusters, we can see how the wines from the second region are nearly all of them of one kind, but they also have wines from three other types, while wines from the first region are split in three different kinds, in which the second region also makes two wines of the same type. The third region only makes one kind of wine which is also partially made in the second region.

References

[1] Dua Dheeru and Efi Karra Taniskidou. *UCI Machine Learning Repository*. 2017. URL: https://archive.ics.uci.edu/ml/datasets/wine.

[2] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. "Data clustering: a review". In: *ACM computing surveys (CSUR)* 31.3 (1999), pp. 264–323.

[3] *Hierarchical clustering*. URL: https://nlp.stanford.edu/IR-book/html/htmledition/hierarchical-clustering-1.html.

[4] Matteo Matteucci. *Clustering - K-means*. URL: https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html.

[5] Matteo Matteucci. *Clustering - Hierarchical*. URL: https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html.

[6] *Euclidean distance*. 2018. URL: https://en.wikipedia.org/wiki/Euclidean_distance.

[7] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53 –65. ISSN: 0377-0427. DOI: https://doi.org/10.1016/0377-0427(87)90125-7. URL: http://www.sciencedirect.com/science/article/pii/0377042787901257.