Computational Biology master´s: Machine Learning

Applied Unsupervised Machine Learning

Jaime Castro Cernadas
jaime.castro.cernadas@alumnos.upm.es

Cayetana Martin Garcia
cayetana.martin@alumnos.upm.es

16/11/2024

INDEX

1. Introduction

Unsupervised clustering is a fundamental technique in machine learning that focuses on identifying patterns and grouping data based on similarities without any predefined labels or guidance. Unlike supervised learning, where the model is trained with known outcomes, clustering operates purely on the inherent structure of the data. The goal is to partition data points into clusters such that objects in the same cluster are more similar to each other than to those in other clusters.

This work explores 4 different methods of unsupervised clustering/learning: K-Means, Hierarchical Clustering, DBSCAN and Gaussian Mixture Models; applied to a DNA microarrays gene expression database in .csv format.

2. Preprocessing data

To ensure optimal model performance, data preprocessing is essential. This involves cleaning and transforming raw data to eliminate noise, handle missing values, and standardize the format, thereby improving the quality and reliability of the data.

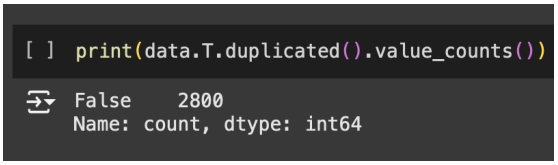
2.1 Data Cleaning

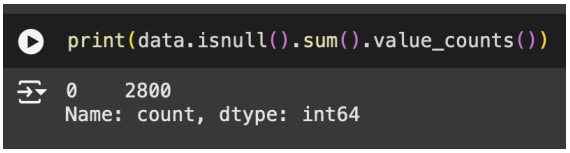
Real-world datasets often contain imperfections such as missing values and outliers. Missing data can arise from various reasons, including data entry errors or incomplete records. Outliers, on the other hand, are data points that deviate significantly from the norm and can skew the analysis. To address these issues, techniques like imputation (filling in missing values) and outlier detection/removal are employed.

Additionally, duplicate data points can also impact data quality. These redundant entries can lead to biased results and inaccurate analysis. Identifying and removing duplicates is crucial for ensuring data integrity.

2.1.1Missing Values and Duplicates

This dataset doesn´t have any missing values or duplicates, so this preprocessing steps are not needed.





2.1.2 Categorical Values

Out of the 2800 features, 2 are object datatype, which are not accepted for our machine learning models.

To deal with this problem, we considered 2 options: OneHotEncoder and LabelEncoder.

Both of this methods give a label to each unique value in the categorical features, but in different ways. LabelEncoder gives each unique categorical value a unique numerical value starting from 0. OneHotEncoder, in the other hand, turns a categorical feature with K possible values into K binary features.

OneHotEncoder method was chosen because only having binary values for this features allows us to control the effect that the scaling might have in them.

2.2 Data Scaling

Data transformation is a crucial step in data preprocessing to ensure that machine learning models can effectively learn from the data. Two common techniques for transforming numerical data are normalization and standardization.

- Normalization: Scales data to a fixed range, typically between 0 and 1. This is beneficial for algorithms that are sensitive to the scale of input features, such as neural networks.
- Standardization: Scales data to have a mean of 0 and a standard deviation of 1. This is useful for algorithms that rely on distance metrics, such as Support Vector Machines and K-Means clustering.

Seen that standardization is a better fit for our clustering algorithms, two methods were discussed: StandardScaler and RobustScaler.

StandardScaler scales features by subtracting the mean and dividing by the standard deviation, centers the data to have a mean of 0 and scales it to have a standard deviation of 1.Works well for normally distributed data with few outliers, but it´s extremely sensitive to the presence of outliers.

RobustScaler scales features using statistics that are robust to outliers. Specifically, it removes the median and scales the data according to the interquartile range (IQR). More

robust to outliers , as it focuses on the median and IQR, which are less affected by extreme values.

Since our data has many different features and scale values, the RobustScaler was chosen for it´s capability for taking on outliers.

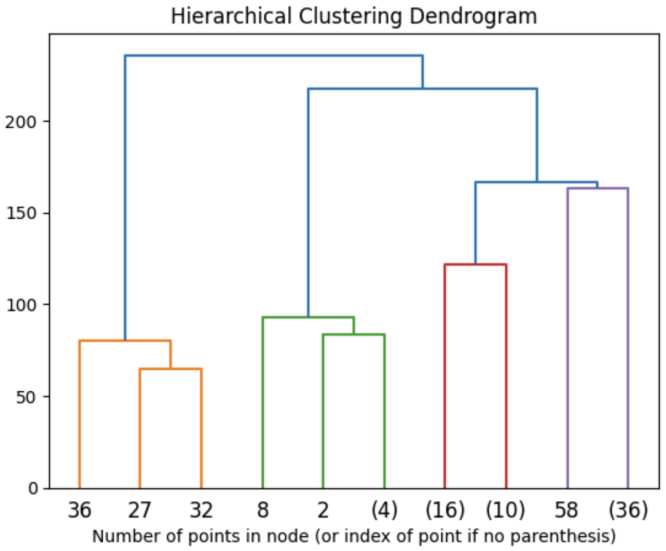
3. Clustering algorithms

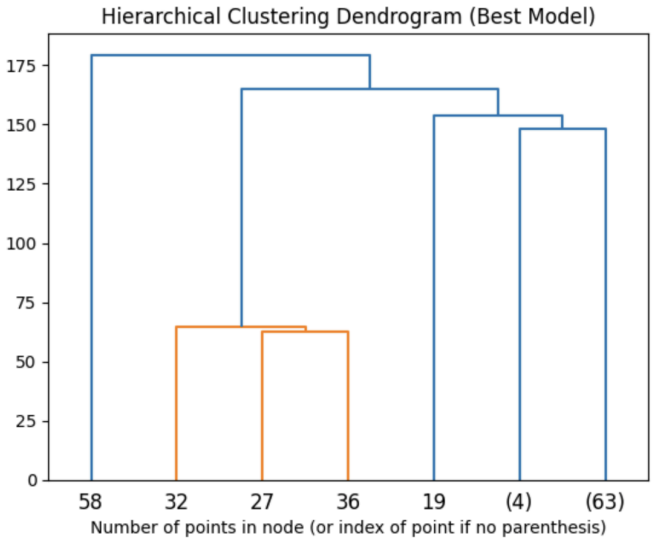
3.1 Hierarchical clustering

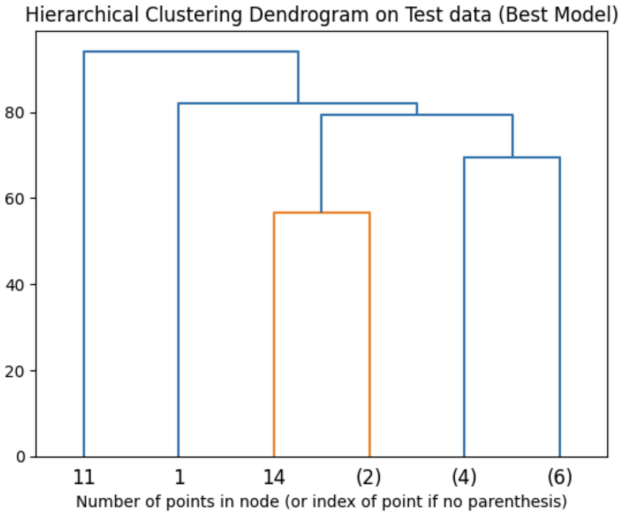
Hierarchical clustering is an unsupervised machine learning method that creates a hierarchical structure out of comparable data points. This structure is frequently shown as a dendrogram, which is a diagram that resembles a tree. At various granularities, the relationships between data points are represented by this hierarchical structure.Every data point in the dendrogram is grouped together at the root. The groups become increasingly smaller and more specialized as we proceed down the tree. At the tree's leaves, each data point eventually forms a distinct cluster.This approach offers a graphic depiction of the connections among samples at various granularities.

It has an object named AgglomerativeClustering that uses a bottom-up method to achieve hierarchical clustering. Each observation begins in its own cluster, and clusters are then gradually combined. The measure for the merging technique is determined by the linkage criteria:

- Ward: reduces all clusters' sum of squared differences.
- maximal or full linkage: the greatest distance between cluster pair observations is minimized.
- Average linkage: the average of the distances between each pair of cluster observations is minimized.





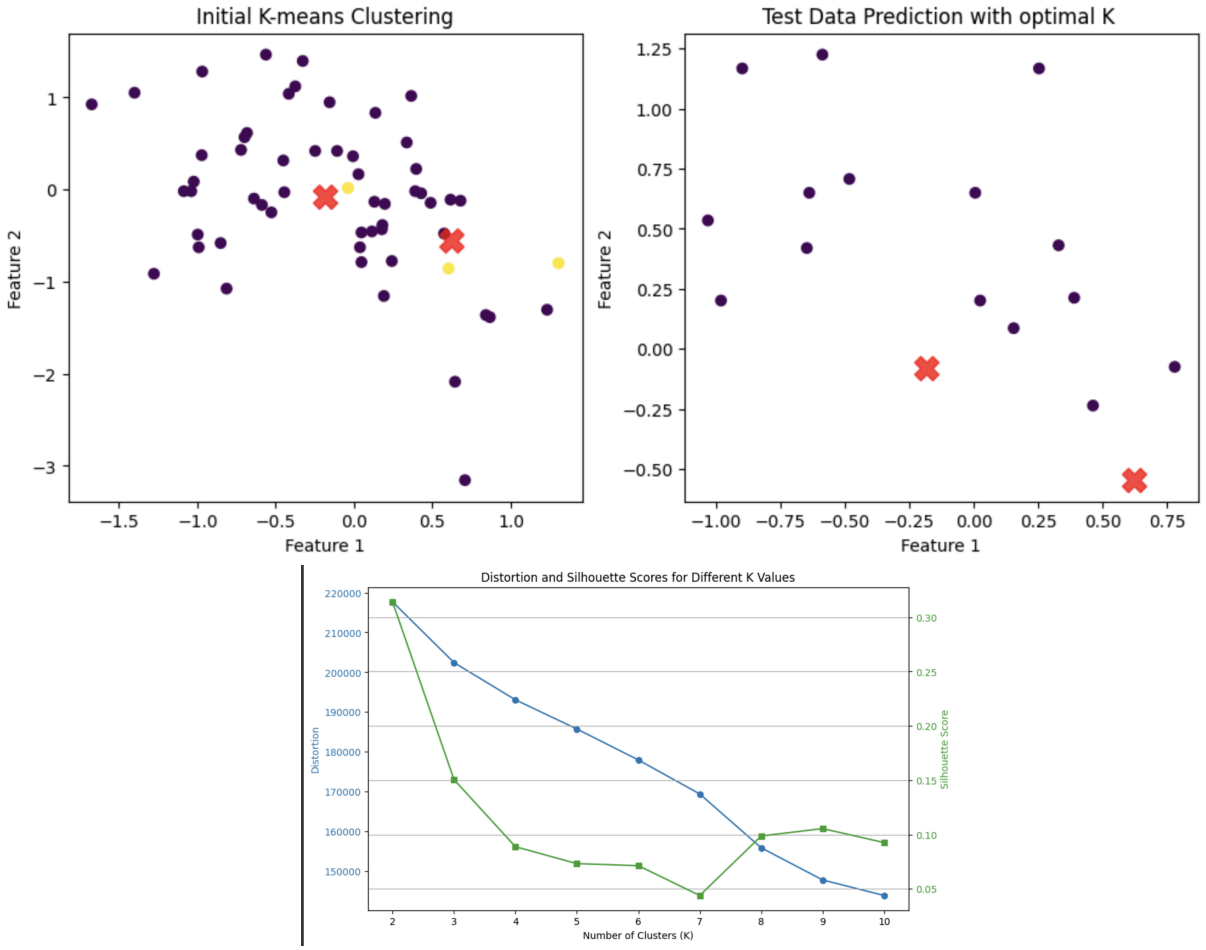


	BASIC MODEL	OPTIMIZED MODEL	VALIDATED MODEL
SILHOUETTE SCORE	0.2579029487506173 7	0.33692195792246304	0.2153102295199417
PARAMETERS		'linkage': 'average', 'n_clusters': 2	
CLUSTERS	5 CLUSTERS: 1 BIG CLUSTER THAT CONTAINS 4 SMALL CLUSTERS (2 WITH MORE SIMILARITY THAN THE OTHERS)	2 CLUSTERS: 1 DISPERSED CLUSTER THAT CONTAINS ANOTHER WITH GREATER SIMILARITY	2 CLUSTERS: 1 DISPERSED CLUSTER THAT CONTAINS ANOTHER WITH GREATER SIMILARITY

3.2 Partitional clustering: K-Means

A sort of clustering technique called partitional clustering divides the data into a collection of non-overlapping groups so that every data point is a member of precisely one cluster. The goal of this method is to divide the dataset into a certain number of clusters, each of which will include comparable data points.

K-means is a popular partitional clustering technique that tries to minimize the within-cluster variation in order to split a dataset into a certain number of groups (K). The algorithm updates the cluster centers (centroids) and assigns data points to clusters iteratively until convergence.



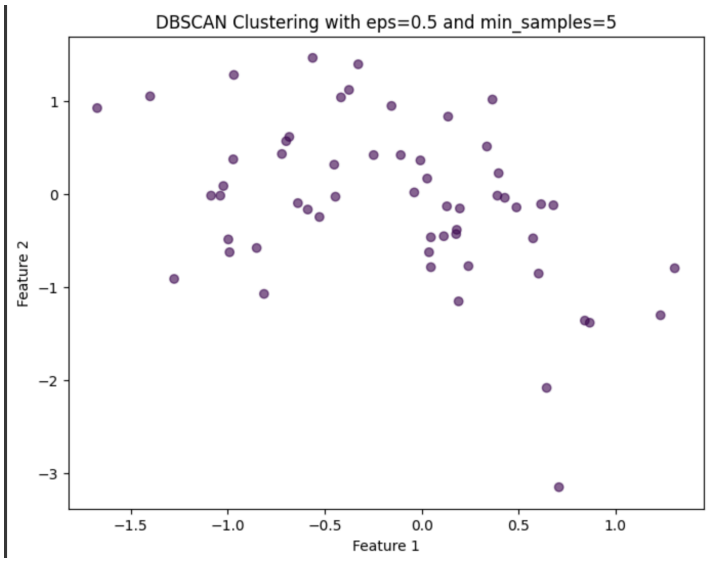
In the first image, we observe 2 clusters of different sizes and locations, although they are somewhat dispersed. Both groups are highly dispersed around their centroids, indicating a possible separation of groups based on the observed characteristics, as well as specific differentiation within each cluster. The location of the centroids suggests that the algorithm attempted to group the data so that each point is close to its respective centroid. However, it was not entirely successful, which may indicate less coherent clustering in this section of the space.

In the third, the Silhouette score measures how similar points within a cluster are compared to points in other clusters. A higher score suggests better separation and cohesion of the clusters. In this case, the Silhouette score is highest when K=2, indicating that, according to

this criterion, dividing the data into 2 clusters is the most suitable option. Although K=2 was used in the graph on the right for the test data, only one cluster was formed.

3.4 DBSCAN

Unlike k-means, which assumes convex clusters, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) method finds clusters as high-density regions divided by low-density areas. Core samples, which are points encircled by a particular number of neighbors (min_samples) within a set distance (eps), are the fundamental idea of DBSCAN. Closely spaced core samples and the non-core samples that go with them,which are core samples' neighbors but not cores themselves,form clusters.

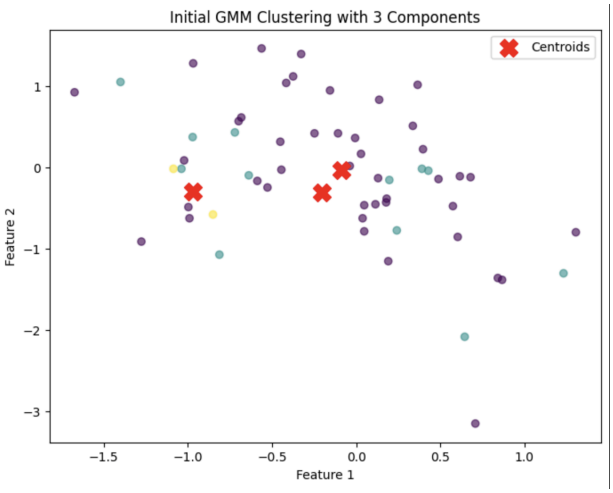


The failure of DBSCAN to form clusters with any combination of parameters suggests that the data does not have a dense or well-defined structure that the algorithm can identify. This could be due to a sparse distribution of the data, poorly matched parameters, or the presence of noise that prevents the detection of meaningful clusters.

3.5 Gaussian Mixture Model

A GMM is a probabilistic model that makes the assumption that the data points come from a combination of a finite number of Gaussian distributions, each of which has unidentified parameters. By taking into account both the covariance structure of the data and the centers of the latent Gaussian distributions, mixture models may be thought of as an extension of k-means clustering.

To fit GMM to data, the GaussianMixture class applies the Expectation-Maximization (EM) method. Additionally, this class provides tools for calculating the BIC (Bayesian Information Criterion) to find the ideal number of clusters in the dataset and displaying confidence ellipsoids in multivariate models. A Gaussian Mixture Model may be learned from training data using the GaussianMixture.fit method, and each test sample can be assigned to the most likely Gaussian component using the GaussianMixture.predict method.




```
n_components: 2, covariance_type: full, BIC: 29927097.941515323
n_components: 3, covariance_type: full, BIC: 45822468.168186456
n_components: 4, covariance_type: full, BIC: 61719936.166944385
n_components: 5, covariance_type: full, BIC: 77612265.03266968
n_components: 6, covariance_type: full, BIC: 93513307.97341433
n_components: 2, covariance_type: tied, BIC: 14055898.98939329
n_components: 3, covariance_type: tied, BIC: 14066144.830334531
n_components: 4, covariance_type: tied, BIC: 14076347.3134544
n_components: 5, covariance_type: tied, BIC: 14086590.971822038
n_components: 6, covariance_type: tied, BIC: 14096797.959691368
n_components: 2, covariance_type: diag, BIC: 452706.5275081146
n_components: 3, covariance_type: diag, BIC: 455240.12708234356
n_components: 4, covariance_type: diag, BIC: 464333.01351821405
n_components: 5, covariance_type: diag, BIC: 474975.5435186548
n_components: 6, covariance_type: diag, BIC: 483130.2921958134
n_components: 2, covariance_type: spherical, BIC: 533939.7039268106
n_components: 3, covariance_type: spherical, BIC: 535058.941089438
n_components: 4, covariance_type: spherical, BIC: 512845.86003598577
n_components: 5, covariance_type: spherical, BIC: 515760.41144614737
n_components: 6, covariance_type: spherical, BIC: 518755.5672917124

Best Parameters: {'covariance_type': 'diag', 'n_components': 2}
Best BIC: 452706.5275081146
```

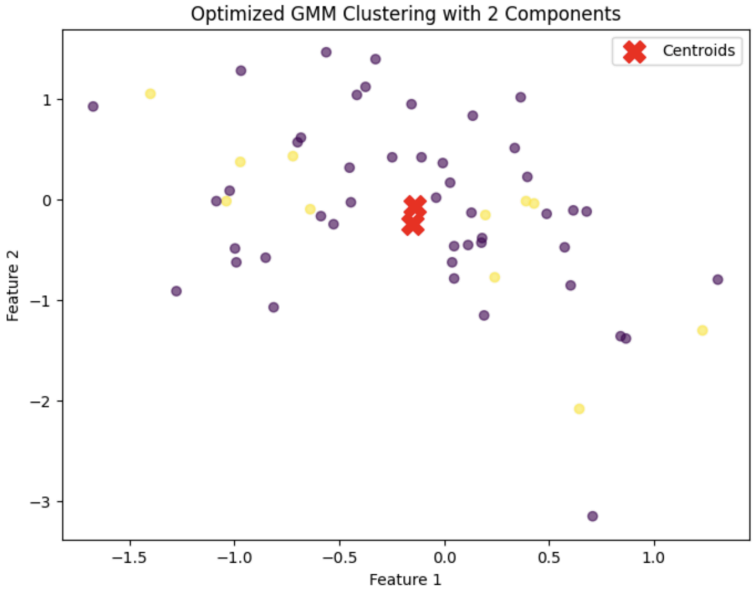
The image number "..." represents the clusters identified by the GMM model. In total, three clusters are formed, each with data points distributed across the cluster area and scattered around their centroids. These points show some degree of dispersion both within each individual cluster and relative to the centroids, indicating that the clusters are not perfectly compact but still exhibit a general grouping around central points.

On the right side of the image, the covariance and BIC (Bayesian Information Criterion) for each number of clusters are displayed. The plot indicates that the optimal number of clusters is 2, as it corresponds to the point where the BIC is lowest, suggesting that this configuration provides the best balance between model fit and complexity.

In this image, we see a Gaussian Mixture Model (GMM) clustering result with 2 components.

The color coding represents the 2 clusters formed by the GMM model. We can observe a notable separation among some points, but there are also a few data points within the clusters that appear dispersed and somewhat isolated from others. This dispersal could be a sign that these data points do not fit perfectly within the clusters and may have an impact on the overall cluster quality.

Comparing this to the Silhouette and BIC graphs from the previous analysis, it appears that a GMM with 2 components achieves the optimal BIC (which was lowest for 8 components), it provides a balance in terms of simplicity and interpretability. The choice of 2 components could be an attempt to capture distinct groups in the data, although the resulting clusters may lack the compactness or cohesion we would desire for a more definitive classification.



4. Selected Cluster

The Partitional Clustering model is particularly suitable in this context because of its focus on grouping data in a way that maximises internal cohesion and minimises overlap between clusters. This type of model, which includes methods such as K-Means, operates by dividing the dataset into a specific number of groups (K clusters), so that each point is assigned to the cluster whose centre (or centroid) is closest. In this case, analysis of the data shows that a 2-cluster partition offers the most optimal configuration, supported by a high Silhouette Score.

The Silhouette Score is a key indicator in clustering, as it assesses how similar points within a cluster are compared to points in other clusters. In this case, the score reached its highest level when K=2 was set. This indicates that dividing the data into two clusters achieves the optimal balance between cohesion and separation, avoiding both excessive fragmentation (which could create clusters that are too specific or fragmented) and the clustering of disparate points in the same group.

The Partitional Clustering model places a centroid for each cluster, representing the 'middle' centre of each group of points. Although the clusters show some dispersion around their centroids, the placement of these central points reveals that the model has grouped the data in a way that maximises the proximity of each point to its cluster, reflecting the underlying patterns in the data. This dispersion may indicate inherent variability in the data, but the model is still able to organise the points into meaningful clusters, suggesting a suitable data structure for this method.

The model was tested with different values of K, but K=2 proved to be the optimal number of clusters according to the Silhouette Score. This confirms that a 2-cluster partition is the most representative for the analysed data, which optimises the balance between internal variability and cluster separation.

5. Conclusions

We have implemented four unsupervised clustering models (UML) to try to identify patterns in DNA gene expression data. After evaluating the results of each method, we found that both Hierarchical Clustering and Partitional Clustering provided the most promising results. However, neither of these models achieved a sufficiently clear or consistent clustering of the data.

During the data pre-processing process, we identified a high level of outliers and a large disparity in data characteristics. These conditions significantly affect the ability of clustering models to form well-defined groups, as outliers and extreme variations distort the internal cohesion of clusters and generate a dispersion that makes it difficult to identify consistent patterns.

To improve clustering, we propose a dimensionality reduction using Principal Component Analysis (PCA). This approach would reduce the complexity of the data by retaining only the most relevant features, minimising the impact of outliers and improving homogeneity within each cluster. By restructuring the data space with PCA, clustering models could identify more meaningful patterns and improve the quality of the results.

In summary, while the current models provide a useful basis, incorporating PCA as a step in the clustering process could help to improve the quality of the results.

6. References

https://scikit-learn.org/stable/modules/clustering.html