Jose Valero
Lucas Fayolle

Lab session week 7 -
Density-based clustering (DBSCAN)

Profesor : Elies Fuster i Garcia

Block I. Data loading and exploration

1. **Exploring data suitability for Density-Based Clustering**

   The requirements for a density-based clustering approach, such as DBSCAN, include clusters that are densely packed and separated by sparse or empty regions.

   As can be seen in the scatter plot (Figure 1), the data shows clear, dense clusters of points, with well-defined spaces between them. This separation indicates that DBSCAN could successfully identify these clusters, as the algorithm is designed to detect areas of high point density. In addition, there are some isolated points and small scattered groups that could represent noise, which DBSCAN can handle by excluding them from any cluster.

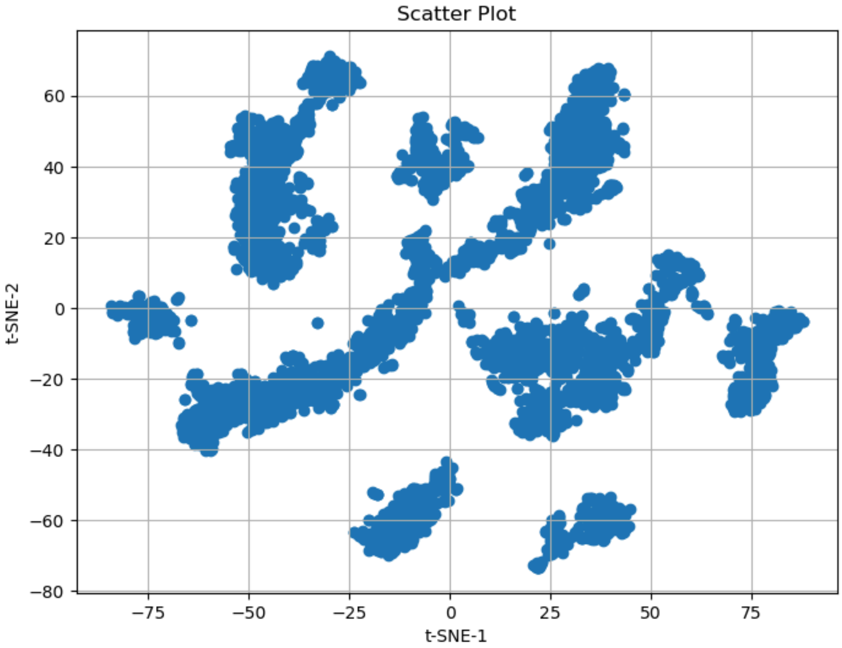

Figure 1: Scatter plot showing data distribution and potential cluster structures

2. **Analyzing pairwise euclidean distances**

   Once we have verified that it is feasible to use DBSCAN, we proceed to visualize the histogram of distances and the graph of ordered distances. These visualizations help us to better understand the structure of the dataset and to validate the choice of parameters for density-based clustering.

   In the distance histogram (Figure 2), we observe that most of the points are located at a distance between 45 and 80 from other points. This peak indicates areas of high density in the dataset, which is crucial for the DBSCAN algorithm, which relies on the existence of dense regions to form clusters. The frequency distribution gradually decreases to the right, suggesting that points further away are in less dense or possibly isolated areas. This "long tail" could represent the separation between clusters or outlier points, which the algorithm would interpret as noise.

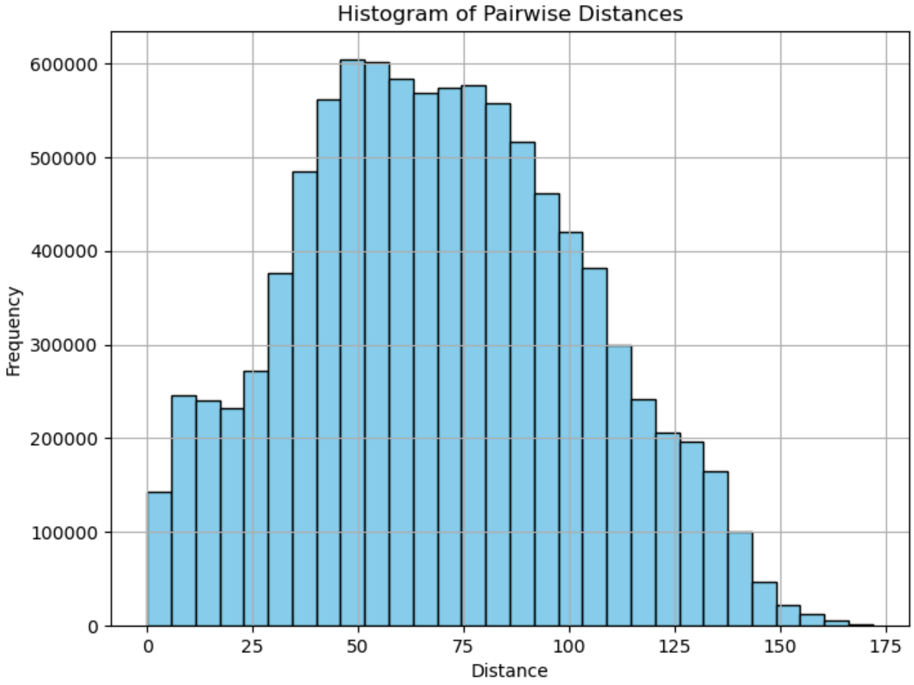

Figure 2: Histogram of pairwise Euclidean distances

The line plot with the ordered distances (Figure 3), on the other hand, provides a cumulative visualization that allows us to observe the transition from the closest to the farthest distances in the data set. In the early portions of the curve, we see a smooth increase in distances, indicating that, for most points, there are other nearby points at a moderate distance. However, as we move to the far right of the plot, the increase in distance becomes more pronounced, suggesting the existence of points farther and farther away from each other.

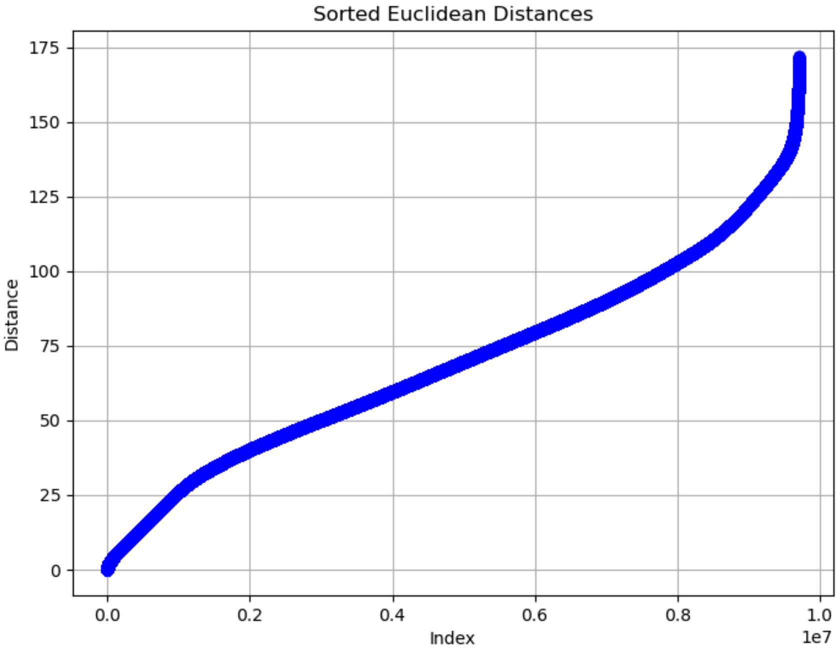

Figure 3: Sorted Euclidean distances plot

3. Exploring k-Graph distances

Having analyzed the dataset and determined its feasibility for a density-based clustering approach, we proceed with the calculation of the optimal parameters for the DBSCAN algorithm, specifically *ε* (which defines the neighborhood radius within which the points must be to be considered part of a cluster) and *minPts* (the minimum number of neighbors needed within this radius for a point to be considered a core point).

To establish a suitable minimum value of *minPts*, we take the dimensionality of the dataset (*D*) as a reference and add one to it, thus ensuring a minimum number of neighbors to consider a point within a cluster.

The value of *ε* is estimated by analyzing the ordered distance plot for each value of *k* (equivalent to *minPts*). The distance to the k-th nearest neighbor is calculated for each point and these distances are sorted in ascending order. Then, a "bend" is sought in the plot, which represents an abrupt change in slope, indicating a transition from points within clusters to edge points or noise. In this case, an angle criterion is used to find this bend by selecting a significant distance index that exceeds an angle threshold. This method allows us to visually identify the optimal cutoff point for ε at each value of *k*.

The following table summarizes the results obtained for different values of *minPts* and their corresponding values of ε, estimated by analysis of the ordered distance plots:

$minPts = tau = k$	Estimated ε value
3	1.903
4	2.357
5	2.679
6	2.733
7	2.813
8	3.161
9	3.395
10	3.721

To better understand how the parameter ε varies as a function of different values of *k*, ordered distance plots are generated. These plots represent the distances to the k-th nearest neighbor for each point, sorted in ascending order, and allow us to visually identify the "elbow" that suggests the optimal value of *ε*. Below, we present an example of one of these graphs to illustrate the *ε* selection process and analyze how points within clusters differ from isolated or edge points.

Looking at the ordered distance plot for $k = 3$ (Figure 4), one notices a pattern where most of the distances stay close to zero at the first few points and then increase abruptly, forming an elbow representing the value of ε. In this case, the bend is at $\varepsilon \approx 1.903$. This inflection point suggests that this is a good choice for ε when $minPts = 3$, as it adequately captures the cluster density without including isolated or noisy points.

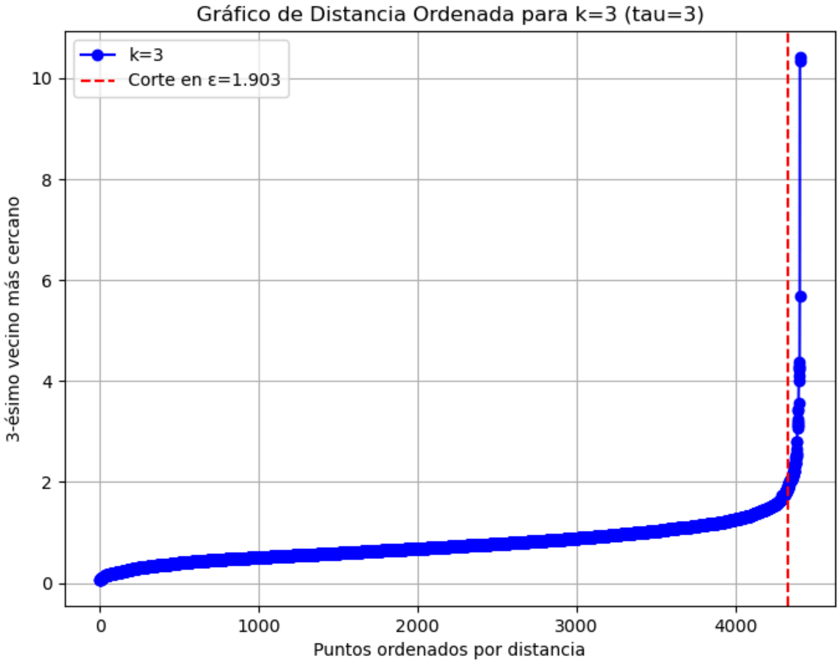

Figure 4: Elbow plot for determining optimal epsilon

Block II. Patients clustering with DBSCAN

1. **Selecting hyperparameter combinations for DBSCAN**

   To determine the optimal values of *minPts* and ε, a search in a hyperparameter space based on the values estimated in the previous block was used. Specifically, combinations of *minPts* and ε that had already been shown to be good candidates in the ordered distance plots were tested. This approach can be considered a form of constrained grid search, since we are testing a restricted set of values based on previous analyses rather than exploring a wider range or performing a completely random search.

   In addition, evaluation using the Silhouette Score provides a measure of the cohesion and separation of the clusters formed. This metric provides an indication of the quality of the clustering, with higher values suggesting well-defined and separated clusters.

   After testing the combinations, the best values were determined to be $minPts = 9$ and $\varepsilon \approx 3.395$, which produce **10 clusters** with a Silhouette Score of 0.1117. Although the Silhouette Score is positive, indicating moderate separation between clusters, the relatively low value suggests that there may be some overlap or that the clusters are not completely cohesive.

2. **Analyzing DBSCAN clustering results**

   Once the optimal parameters for DBSCAN have been determined, we proceed to represent the clusters obtained in a scatter plot. In this visualization (Figure 5), each cluster is identified with a different color, while the black dots represent the outliers, i.e., those points that DBSCAN has classified as noise because they do not meet the minimum density requirements.

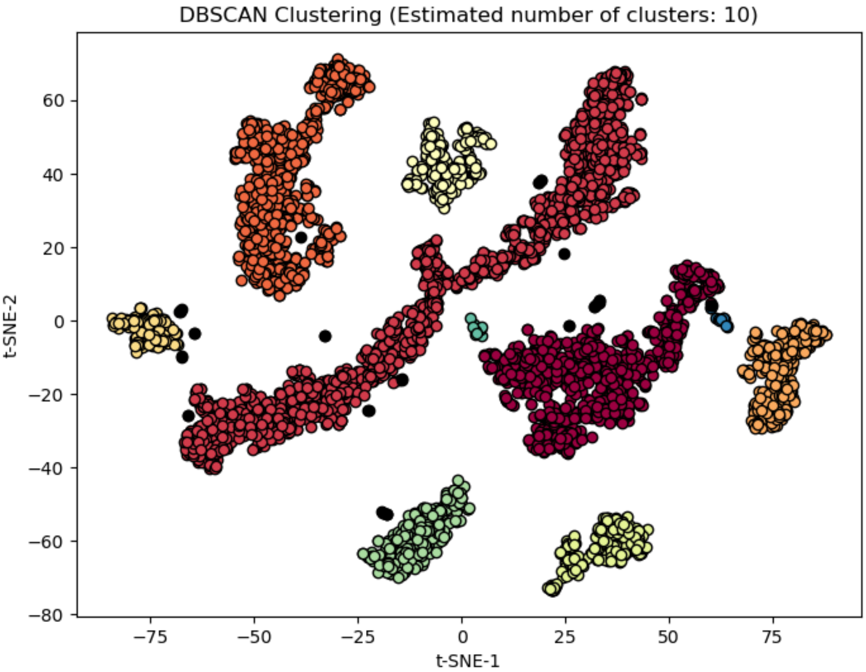

Figure 5: Scatter plot of DBSCAN clusters with identified outliers (black points)

3. Exploring point types in DBSCAN

Once the clusters have been identified with DBSCAN, it is important to analyze the different types of points that make up each cluster: core points, edge points and noise points. DBSCAN classifies points into these categories based on their local density and their relationship to neighbors. **Core points** are those with sufficient neighbors within ε radius, allowing them to form the "heart" of the clusters. **Edge points** are at the boundary of clusters and have fewer neighbors than core points, but still belong to a cluster. Finally, **noise points** do not meet the minimum density requirement and are therefore considered as isolated points or outliers.

In the scatter plot (Figure 6), these types of points are represented with different colors, providing a clear view of the internal structure of the clusters and how DBSCAN handles less dense or isolated points.

Observing the graph, we note that most of the points are classified as **core points** (in blue), which confirms the presence of high density areas within the clusters. These core points form the central body of each cluster, providing cohesion and defining the dense areas of the dataset.

At the boundaries of the clusters, we find the **edge points** (in green), which connect the core points but do not have enough neighbors to be classified as such.

The **noise points** (in red) appear scattered in space, separated from the main clusters. These points represent low density areas and are considered by DBSCAN as isolated points, not belonging to any cluster.

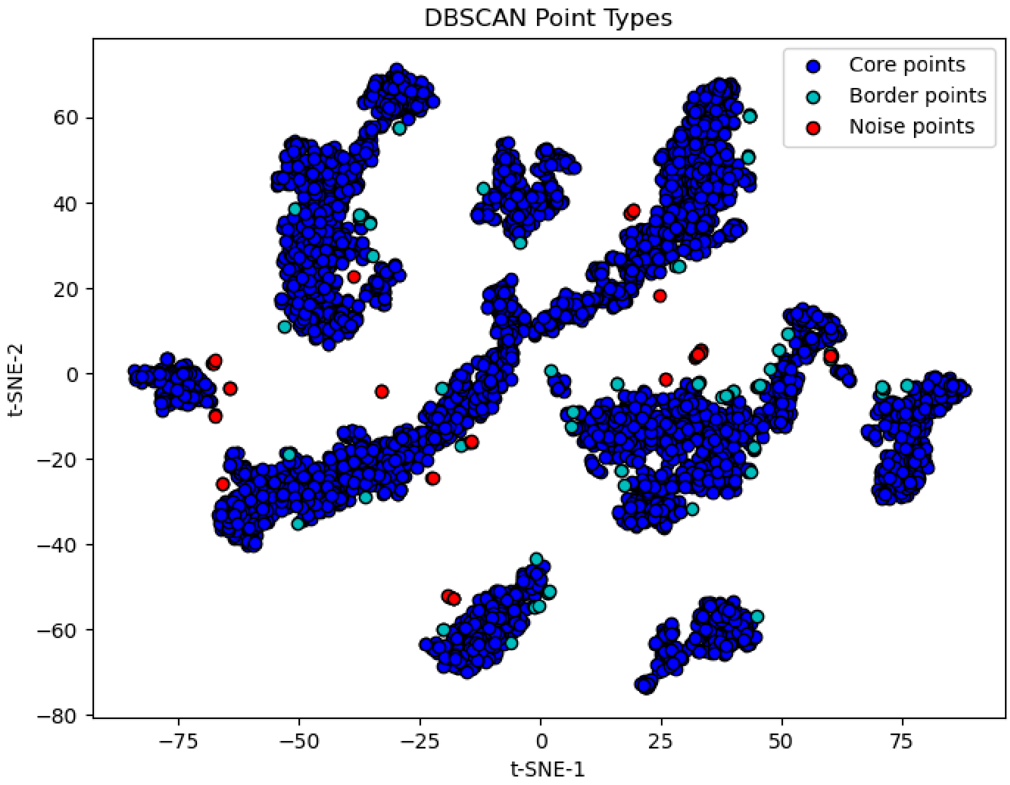

Figure 6: Classification of Points by DBSCAN: Core, Border, and Noise

4. **Specific questions:**

- **Why have I not asked you to split your data into a training and test set, or into a pure training and validation set? Justify your answer.**

  In the context of unsupervised learning with the DBSCAN algorithm, there is no need to divide the data into training and test or validation sets. This is because there are no labels or target values to predict; the goal is to discover patterns and structures inherent in the entire data set.

- **Do you expect your selected/proposed method to set DBSCAN hyperparameters (question 1 of this block) to work properly with other data of the same dimensionality? Do you think it should perform well in higher dimensions beyond 2 and 3? Justify your answer.**

  The proposed method for establishing DBSCAN hyperparameters could work well with other data of the same dimensionality if they share similar characteristics in terms of density and distribution. However, since DBSCAN is sensitive to the scale and density of the data, ε and $minPts$ may need to be adjusted to fit different data sets.

  In dimensions higher than 2 or 3, the method may not be as effective due to the "curse of dimensionality" (Banks & Fienberg, 2003, 249), which makes it difficult to identify dense structures in high-dimensional spaces. In such cases, additional techniques such as dimensionality reduction or the application of alternative algorithms more suitable for high-dimensional data may be necessary.

References

Banks, D. L., & Fienberg, S. E. (2003). *Encyclopedia of Physical Science and Technology* (Third Edition ed.). Academic Press. https://doi.org/10.1016/B0-12-227410-5/00164-2