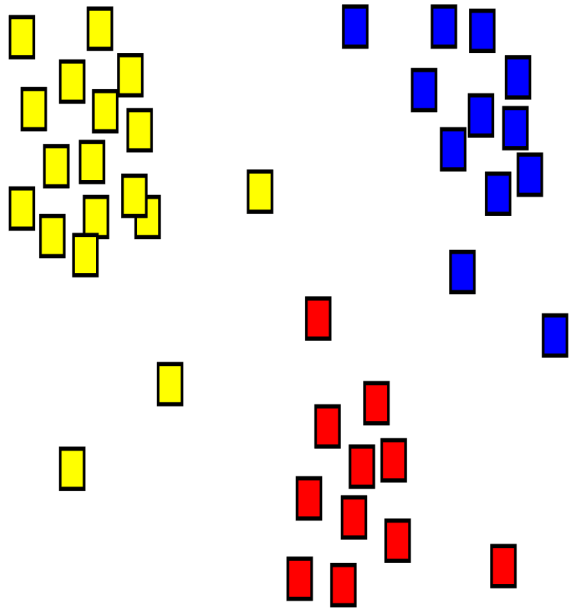
DATA SCIENCE MEI/1

University of Beira Interior
Department of Informatics

Hugo Pedro Proença,
hugomcp@di.ubi.pt, **2021/22**

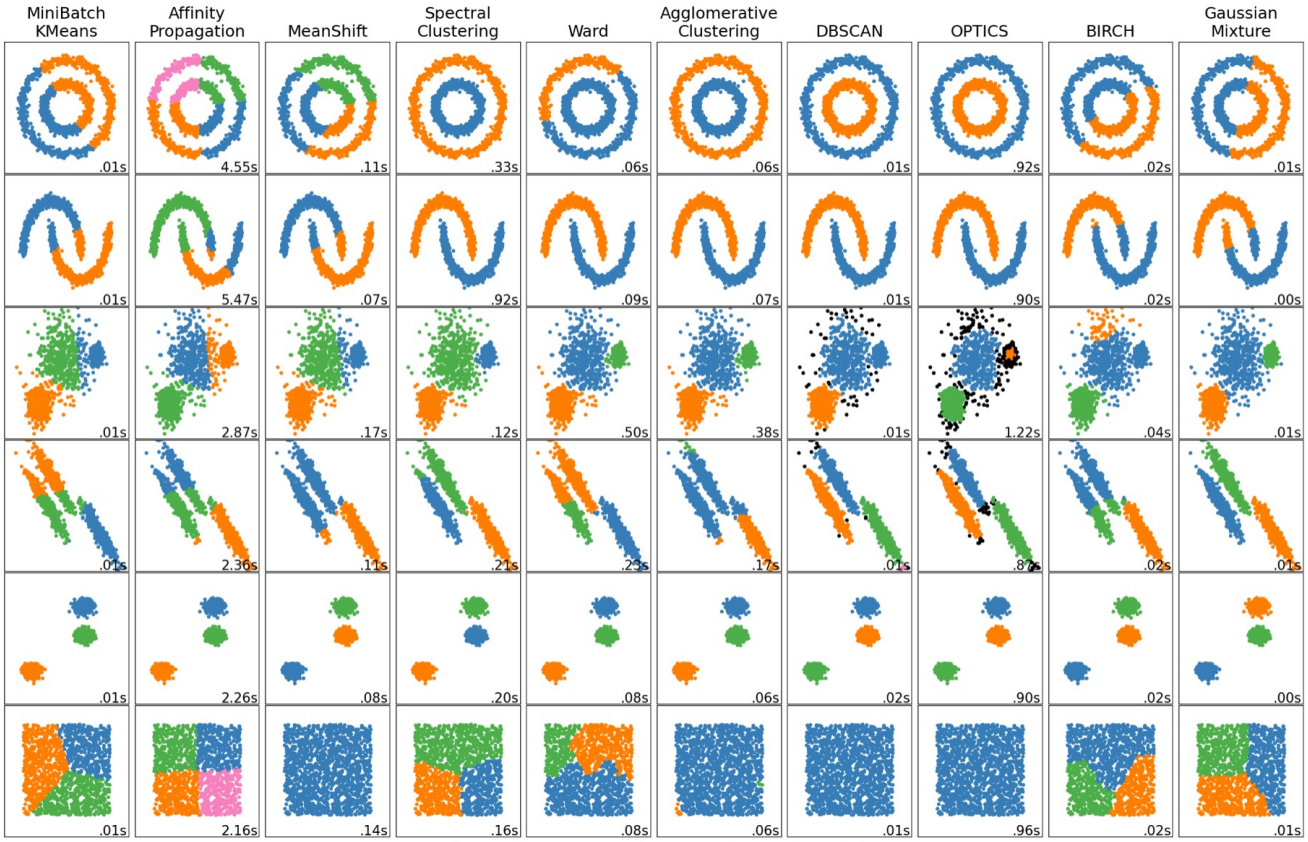
Data Clustering

- **Clustering** can be defined as "the task of dividing the population into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups".
  - In practice, the aim is to segregate groups with similar traits and assign them into clusters.
- It is a form of **non-supervised learning**, in the sense that no label information/supervision in needed.



Data Clustering

- There is a variety of clustering algorithms available for multiple languages.
- As an example, the "*sklearn.cluster*" library offers:



Data Clustering

- The existing clustering algorithms can be broadly divided into two families:
  - **Hard Clustering**: Here, each data point either belongs to a cluster completely or not.
  - **Soft Clustering**: Instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned.

- In terms of the techniques used in clustering algorithms, four main families can be identified:
  - **Connectivity models**: Models are based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away. These models are easy to interpret but lack scalability for handling big datasets. Examples are hierarchical clustering algorithm and its variants.
  - **Centroid models**: These are iterative algorithms, in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters. K-Means clustering algorithm is a popular algorithm that falls into this category. In these models, the no. of clusters is required at the end have to be mentioned beforehand, which makes it important to have prior knowledge of the dataset.
  - **Distribution models**: Based on the notion of how probable is it that all data points in the cluster belong to the same distribution (e.g., Gaussian). A popular example of these models is Expectation-maximization algorithm which uses multivariate normal distributions.
  - **Density Models**: These models search the data space for areas of varied density of data points in the data space. They isolate different density regions and assign the data points within these regions in the same cluster. Popular examples of density models are DBSCAN and OPTICS.

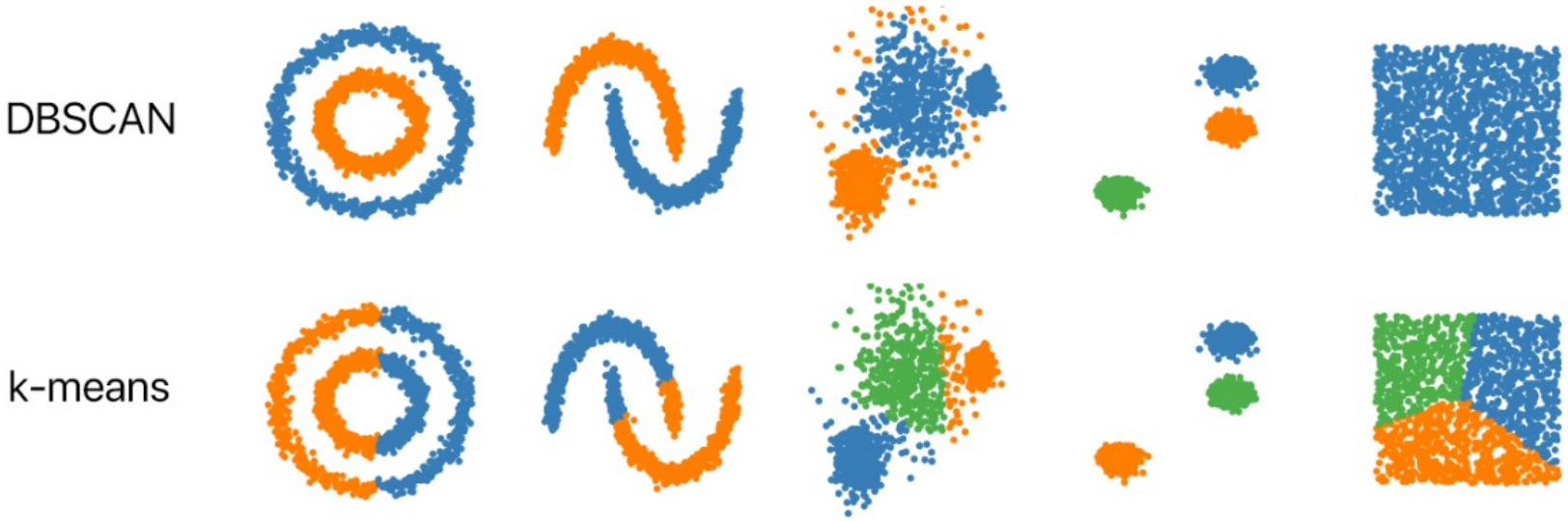
DBSCAN

- The DBSCAN algorithm is a particularly interesting example.
- It was proposed by Martin Ester et al. in 1996. DBSCAN is a density-based clustering algorithm that works on the assumption that clusters are **dense regions** in space **separated** by regions of **lower density**.
- It groups 'densely grouped' data points into a single cluster.
- It can identify clusters in large spatial datasets by looking at the local density of the data points.
- The most exciting feature of DBSCAN clustering is that it is robust to outliers. It also does not require the number of clusters to be told beforehand, unlike K-Means, where we must specify the number of centroids..

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). AAAI Press, 226–231.

DBSCAN

- DBSCAN requires only two parameters: “epsilon” and “minPoints”.
- “Epsilon” is the radius of the circle to be created around each data point to check the density
- “minPoints” is the minimum number of data points required inside that circle for that data point to be classified as a **Core point**.
- In high dimensions the circle most be understood as an hypersphere, where “epsilon” is the radius of that hypersphere, and minPoints is the minimum number of data points required inside that hypersphere.

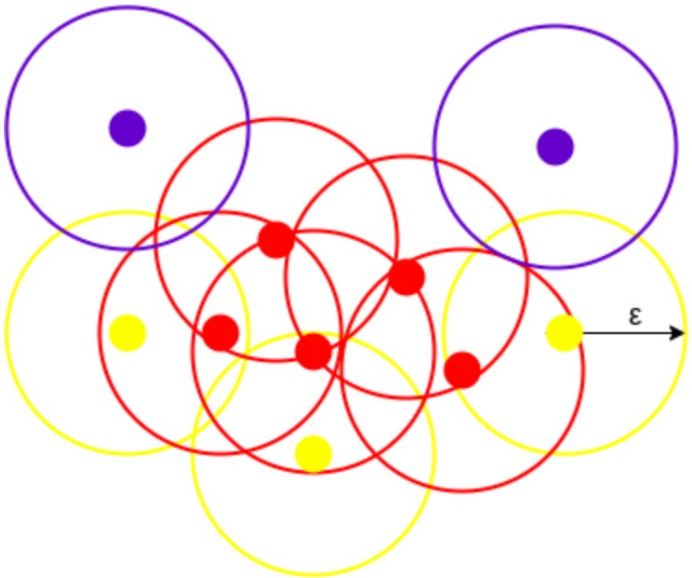


DBSCAN

- Let us consider a 2D example with 10 points. In this case, it will correspond to 10 instances, each one represented with two feature values.

DBSCAN

- The algorithm starts by defining a neighborhood of radius ε around each point, and classifies them into one of three families:
  - **Core**: A point is considered a "Core" if it has at least "minPoints" in its neighborhood.
  - **Border**: A point is considered a "Border" if it has some neighbors, but less than "minPoints"
  - **Noise**: A point is considered an outlier (noise) when there are no neighbors in its neighborhood.
- In the example below, "minPoints=3"



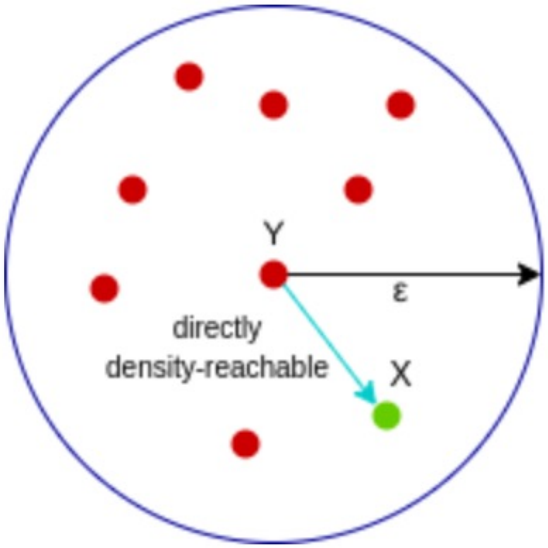
DBSCAN

- The key concepts in DBSCAN are “Reachability” and “Connectivity”.
  - Reachability states if a data point can be accessed from another data point directly or indirectly;
  - Connectivity states whether two data points belong to the same cluster or not.

- Generally, two points can be referred in DBSCAN as:
  - Directly Density-Reachable
  - Density-Reachable
  - Density-Connected

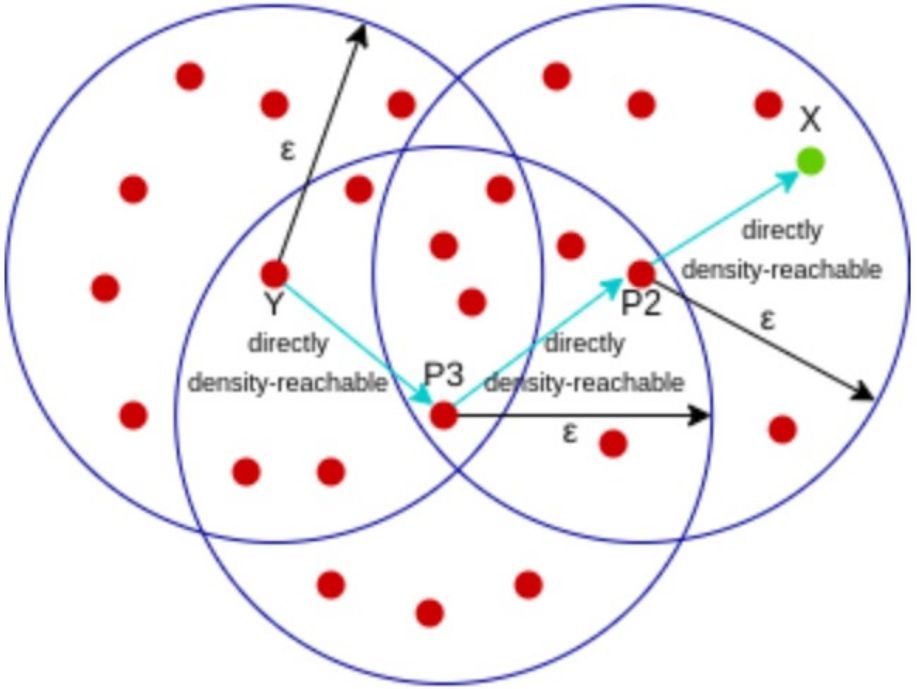
DBSCAN: Directly Density-Reachable

- A point **X is density reachable from Y** when Y is a "**Core**" point and dist(X,Y) < ε.
  - Here dist(,) is typically the Euclidean distance, but other distance metrics can be used.



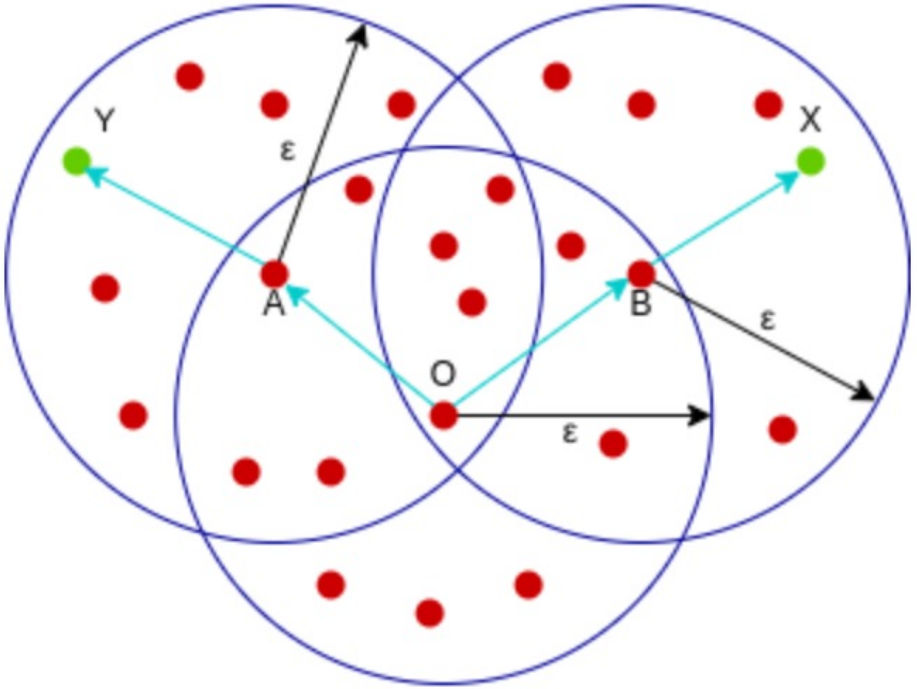
DBSCAN: Density Reachable

- A point X is density-reachable from point Y w.r.t epsilon, minPoints if there is a chain of points p_1, p_2, p_3, …, p_n and p_1=X and p_n=Y such that p_{i+1} is directly density-reachable from p_i.
  - In practice terms, there is the concept of “transitivity” here. If we can move “step-by-step” between points p_1 and p_n , then the latter çpoint is said to be density reachable from the former.



DBSCAN: Density-Connected

- Finally, a point X is density-connected from point Y w.r.t epsilon and minPoints if there exists a point O such that both X and Y are density-reachable from O w.r.t to epsilon and minPoints.
  - In the example below, X and Y are both density reachable from O. Hence, we say that X is density-connected to Y



DBSCAN

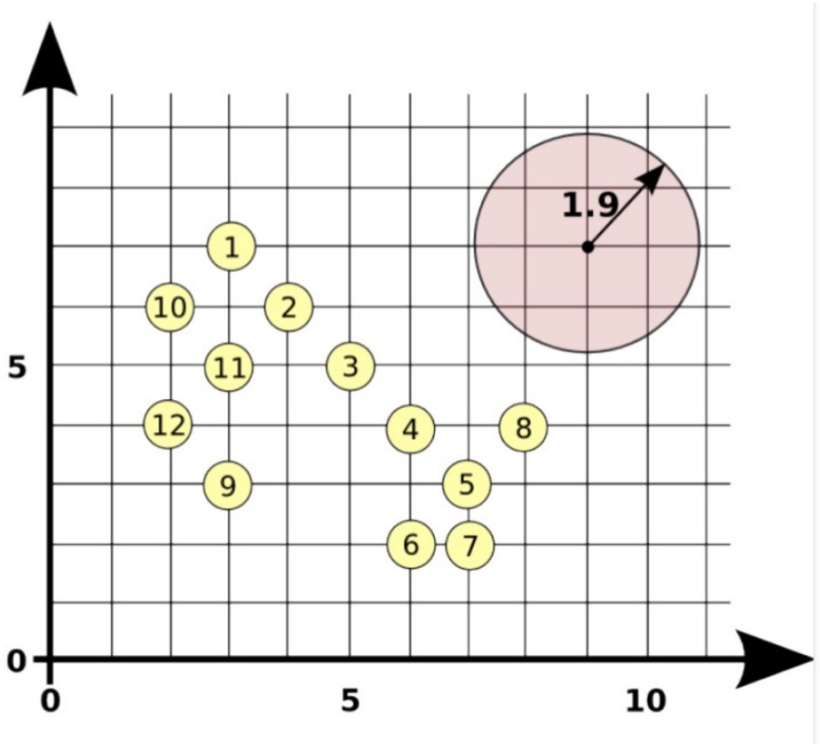
- DBSCAN is very sensitive to the values of the parameters.
- It is very important to understand how to select the values of ε and minPoints.
- A slight variation in these values significantly changes the results produced by the DBSCAN algorithm.
- In practice, the value of minPoints should be at least one greater than the number of dimensions of the dataset, i.e.,
  - minPoints>=Dimensions+1.
  - It does not make sense to take minPoints as 1 because it will result in each point being a separate cluster.
  - Therefore, it must be at least 3. Generally, it is twice the dimensions. But domain knowledge also decides its value.
- The value of epsilon can be decided from the K-distance graph.
  - The point of maximum curvature (elbow) in this graph tells us about the value of epsilon. If the value of epsilon chosen is too small then a higher number of clusters will be created, and more data points will be taken as noise. Whereas, if chosen too big then various small clusters will merge into a big cluster, and we will lose details.

DBSCAN

- Finally, if p is a core point, then it forms a cluster together with all points (core or non-core) that are reachable from it.
  - Each cluster contains at least one core point; non-core points can be part of a cluster, but they form its "edge", since they cannot be used to reach more points.
- According to this definition, clusters satisfy two properties:
  - All points within the cluster are mutually density-connected.
  - If a point is density-reachable from some point of the cluster, it is part of the cluster as well.

DBSCAN Exercise

- Apply the DBSCAN algorithm with $\varepsilon = 1.9$ and minPoints=3
  - Start by discriminate between “Core”, “Border” and “Noise” points.
  - Finally, indicate the clusters obtained.



DBSCAN Exercise: Solution