**DBSCAN**

**Introduction**

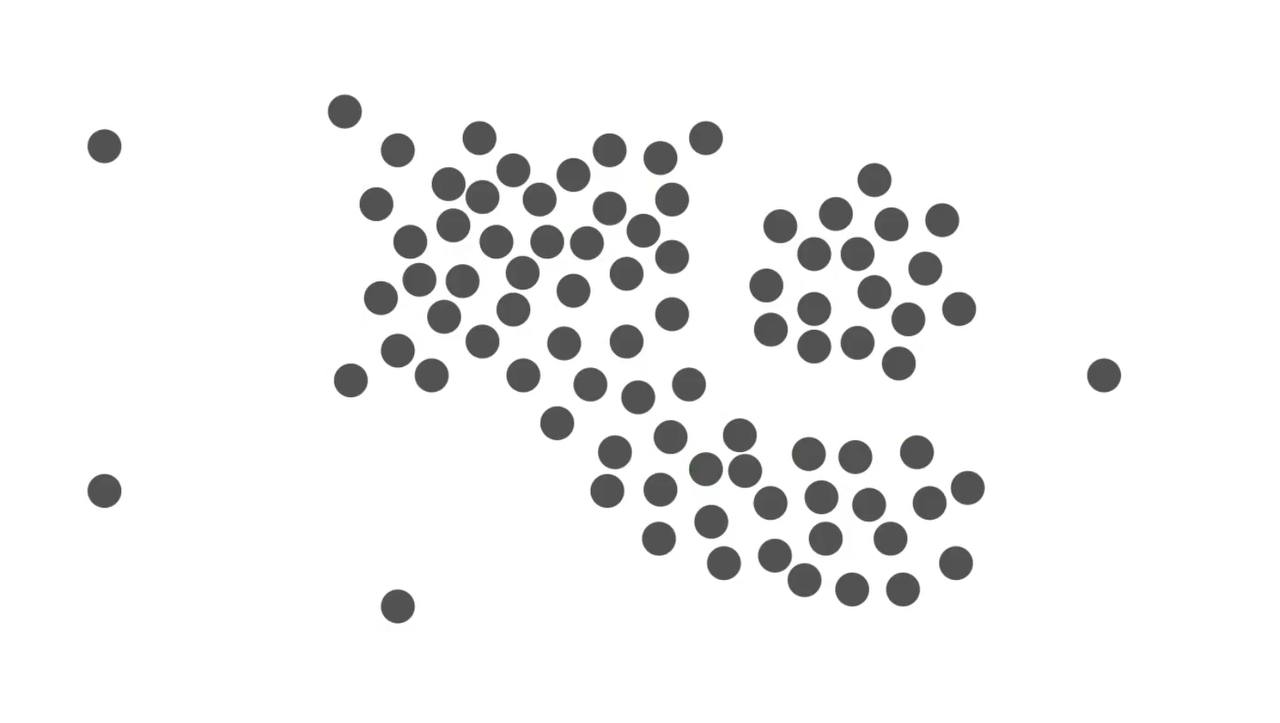
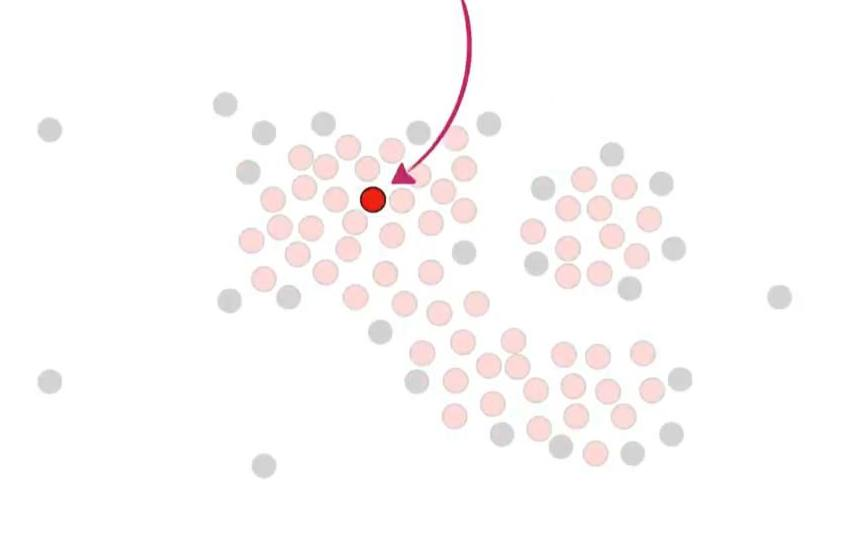
Density-based spatial clustering of applications with noise (DBSCAN) is one of the methods for unsupervised learning used for clustering proposed by [Martin Ester](https://en.wikipedia.org/wiki/Martin_Ester), [Hans-Peter Kriegel](https://en.wikipedia.org/wiki/Hans-Peter_Kriegel), [Jörg Sander](https://en.wikipedia.org/w/index.php?title=J%C3%B6rg_Sander&action=edit&redlink=1), and [Xiaowei Xu](https://en.wikipedia.org/w/index.php?title=Xiaowei_Xu&action=edit&redlink=1) in 1996. while other clustering algorithms like K-means clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are suitable only for compact and well-separated clusters. However, they are severely affected by the presence of noise and outliers in the data fig below. For such datasets DBSCAN is best suitable, which allows for arbitrary cluster shapes, automatically determining the number of clusters, and effectively handling noise and outliers.  


Figure 1: datasets with outliers

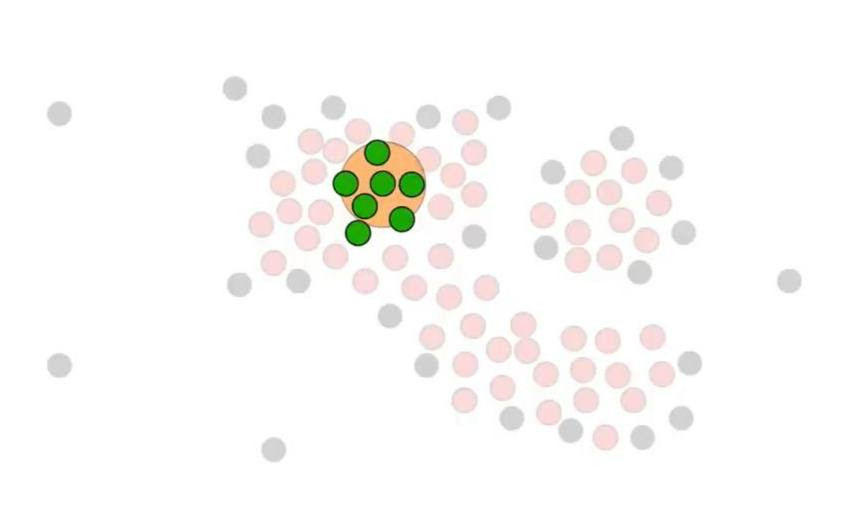
DBSCAN identifies clusters of varying shapes and sizes by grouping points that are densely packed while marking points in low-density regions as noise. It operates using two main parameters: Epsilon (ε), which defines the radius for neighborhood points, and MinPts, the minimum number of points required to form a dense region (core point

**Steps for the algorithm**

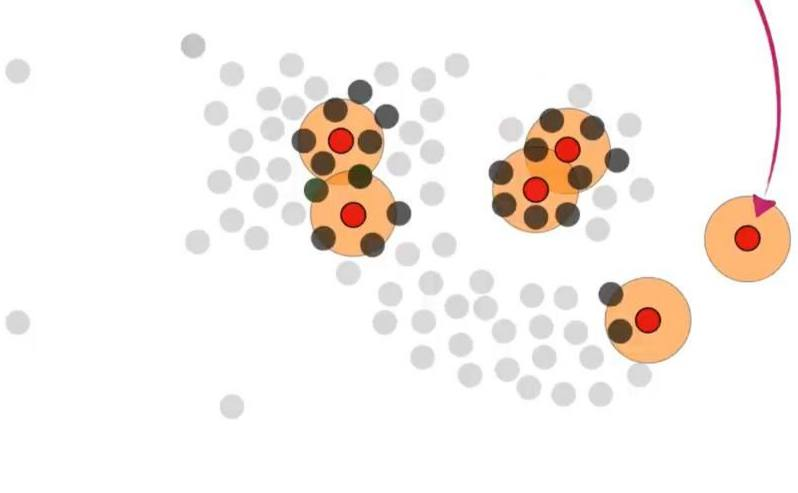
Using the above dataset in figure one let’s see how this algorithm works.

1. Pick randomly a core point 
2. And assign it to a new cluster
3. Identify all points within the ε radius of the current point (using euclidean distance).

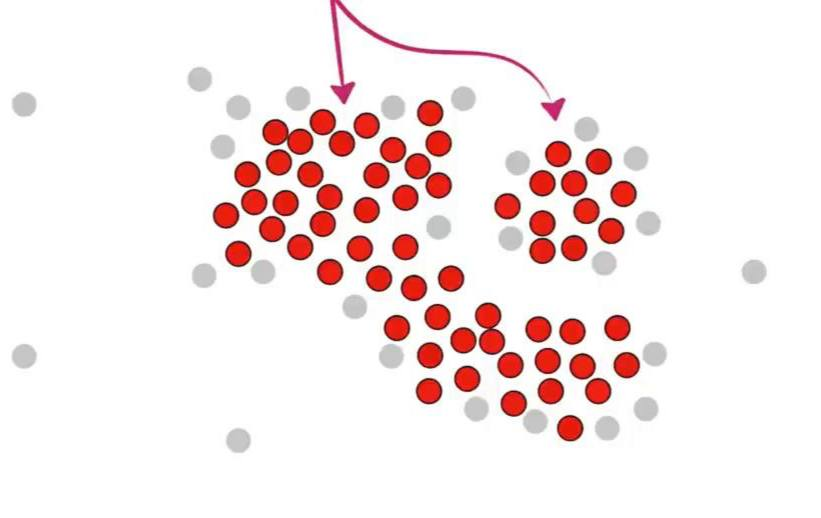
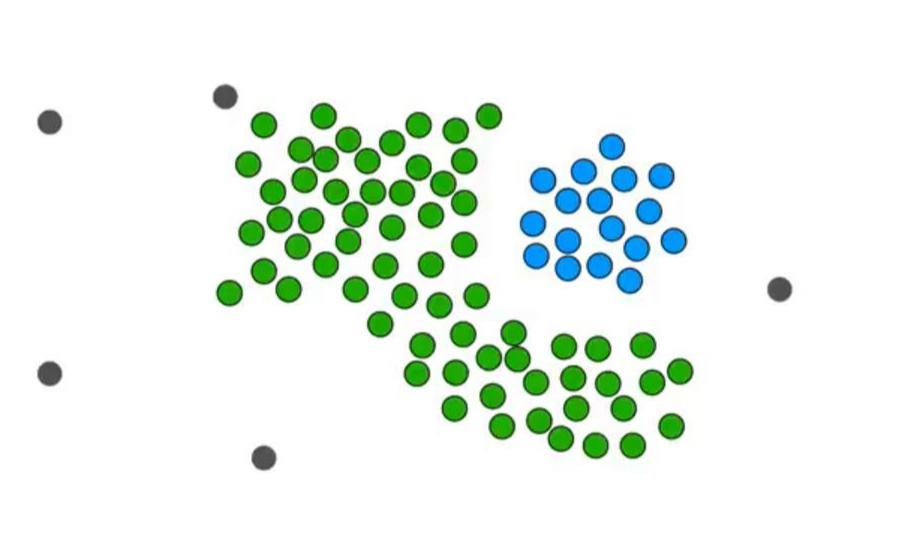
If the current point is a core point (i.e., it has at least MinPts neighbors here in this example is 4), a new cluster is created.

If it’s not a core point, label it as noise (temporarily, as it might be a border point).

1. For each core point in the cluster, repeat the neighborhood search.
2. If new core points are found, add them to the cluster and continue the process until no new points can be added.



1. Move to the next unvisited point in the dataset and repeat the process until all points have been visited.

 finally, the result looks like 

Note that if the non-core point is closer to the non-core point, it can’t be part of the cluster because only points close to the core point can be part of the cluster.

**Conclusion**

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) represents a significant advancement in clustering techniques, particularly for datasets characterized by arbitrary shapes and varying densities. Unlike traditional methods such as k-means, which struggle with noise and outliers, DBSCAN identifies clusters without requiring prior knowledge of their number. By focusing on the density of data points, the algorithm effectively groups closely packed points while marking sparse regions as noise, thereby enhancing the robustness of clustering results. Although DBSCAN's performance can be sensitive to the choice of parameters like Epsilon (ε) and MinPts, its ability to adapt to the inherent structure of complex data makes it a valuable tool in various applications, including geographic data analysis and anomaly detection. As data continues to grow in complexity, DBSCAN remains a powerful option for researchers and practitioners seeking effective clustering solutions.

REFERNCES

<https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/>

https://www.youtube.com/watch?v=RDZUdRSDOok