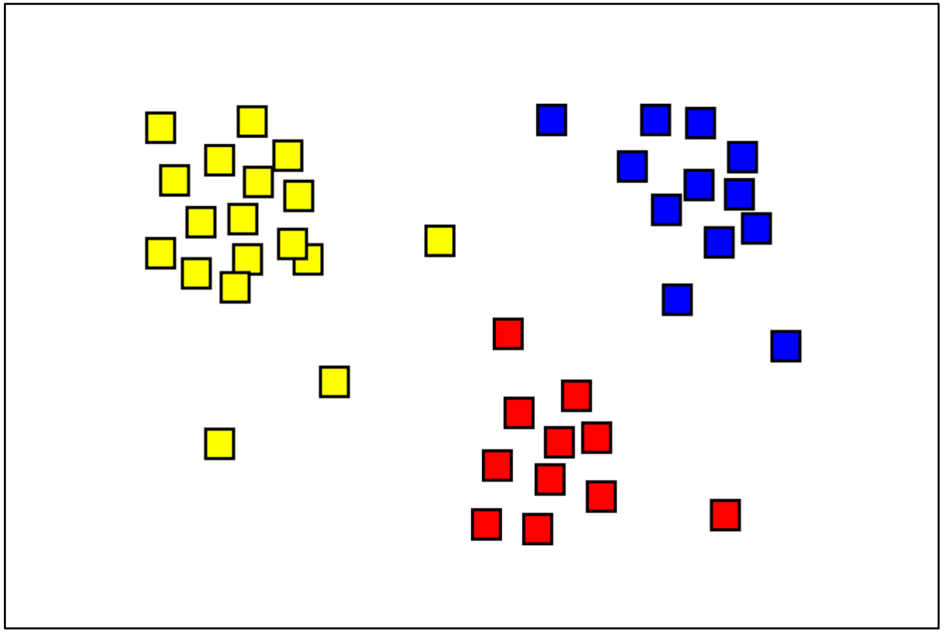
Clustering in Machine Learning

Krishnendu Jana

Contents

What is cluster?

- A cluster is a group of data points that are similar to each other based on their relation to surrounding data points.



What is clustering algorithms ?

- Clustering is an unsupervised machine learning task.
- The aim of the clustering process is to segregate groups with similar traits and assign them into clusters.

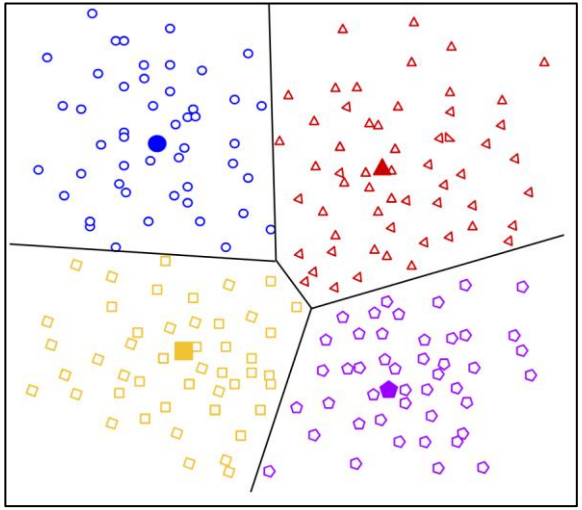
Types of Clustering Algorithms

Several approaches to clustering exist. But the most commonly used clustering algorithms in machine learning are-

1. Centroid-based Clustering
2. Density-based Clustering
3. Distribution-based Clustering

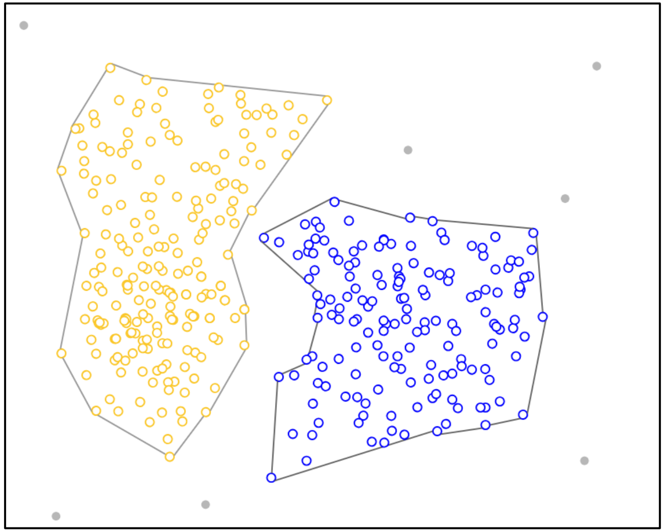
Centroid-based Clustering

- Centroid-based clustering is the easiest of all the clustering algorithms.
- It works on the closeness of the data points to the chosen central value.
- It is a vastly used clustering approach for surfacing and optimizing large datasets.
- Example : K-Means clustering, K-Medoids clustering.



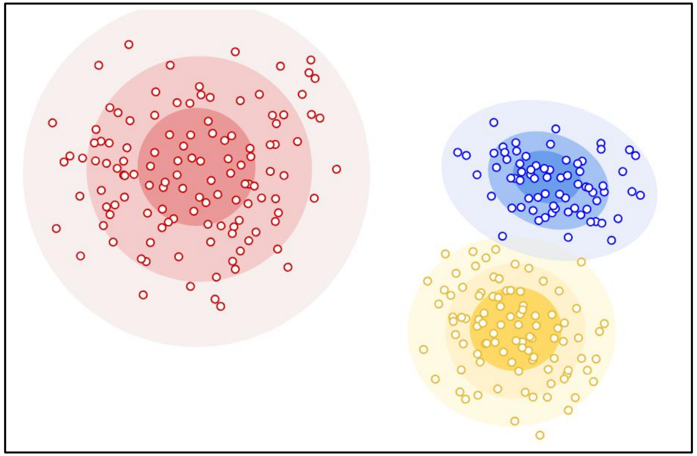
Density-based Clustering

- Density-based clustering connects areas of high example density into clusters.
- This algorithm can create arbitrary-shaped distributions as long as dense areas can be connected.
- Example : DBSCAN



Distribution-based Clustering

- This clustering approach assumes data is composed of distributions, such as Gaussian distributions.
- As distance from the distribution's center increases, the probability that a point belongs to the distribution decreases.
- Example : EM Clustering.



What is DBSCAN?

- DBSCAN – Density Based Spatial Clustering of Application with Noise.
- It is density based clustering algorithm.
- Overcome the problem of concentric circle clustered data in k-Means clustering.

Requirements

- DBSCAN requires only two parameters: *epsilon* and *minPoints*.

Epsilon- The radius of the circle to be created around each data point to check the density

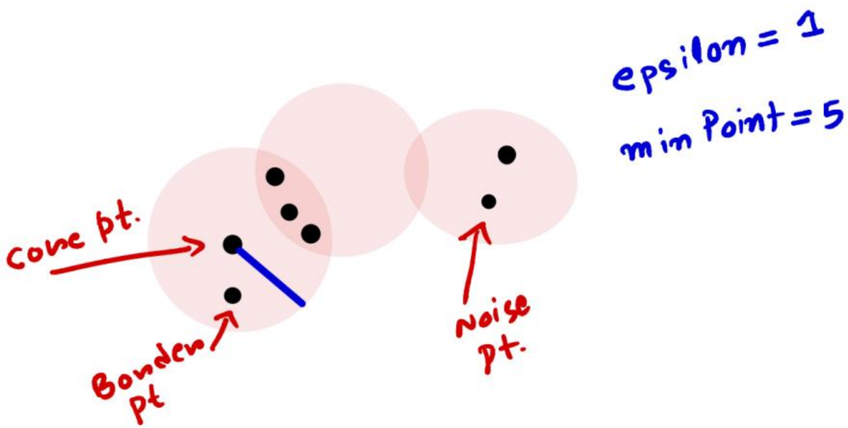
minPoints- the minimum number of data points required inside that circle for that data point to be classified as a **Core** point.

DBSCAN Requirements

Core Points – Those points whose number of neighborhood points are greater than or equal to minPoints.

Border Points - Those points whose number of neighborhood points are less than minPoints and have at least one core point in its neighborhood.

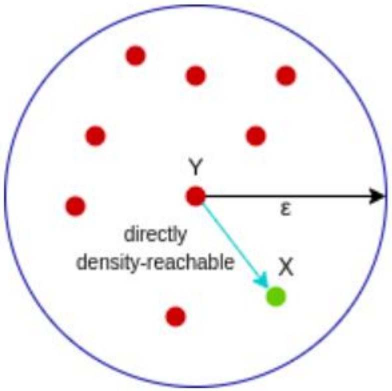
Noise Points - Those points whose number of neighborhood points are less.



DBSCAN Requirements

Directly Density Reachable : A point X is directly density-reachable from point Y w.r.t epsilon, minPoints if,

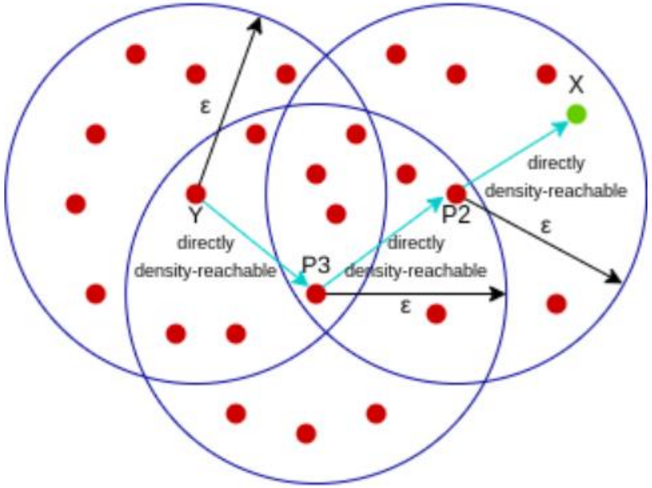
1. X belongs to the neighborhood of Y, i.e, dist(X, Y) <= epsilon
2. Y is a core point



Here, **X** is directly density-reachable from Y**,** but vice versa is not valid.

DBSCAN Requirements

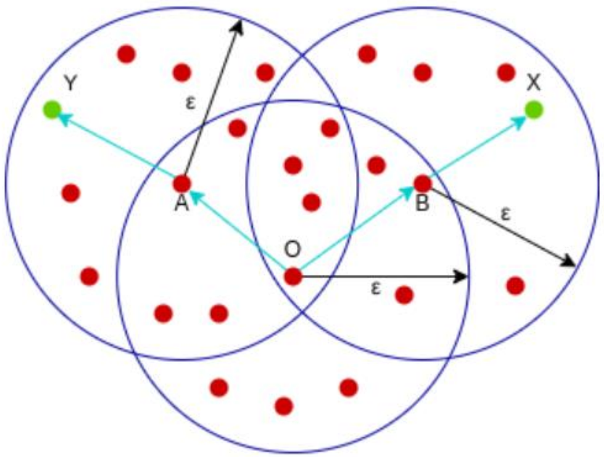
Directly Density Reachable : A point **X** is **density-reachable** from point **Y** w.r.t *epsilon, minPoints* if there is a chain of points $p_1, p_2, \dots, p_n$ and $p_1 = X \; and \; p_n = Y$ such that p_{i+1} is directly density-reachable from p_i.



Here, **X** is density-reachable from **Y** with **X** being directly density-reachable from p_2, p_2 from p_3**,** and p_3 from **Y.** But, the inverse of this is not valid.

DBSCAN Requirements

Density Connected: 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.*



Here, both **X** and **Y** are density-reachable from **O**, therefore, we can say that **X** is density-connected from **Y.**

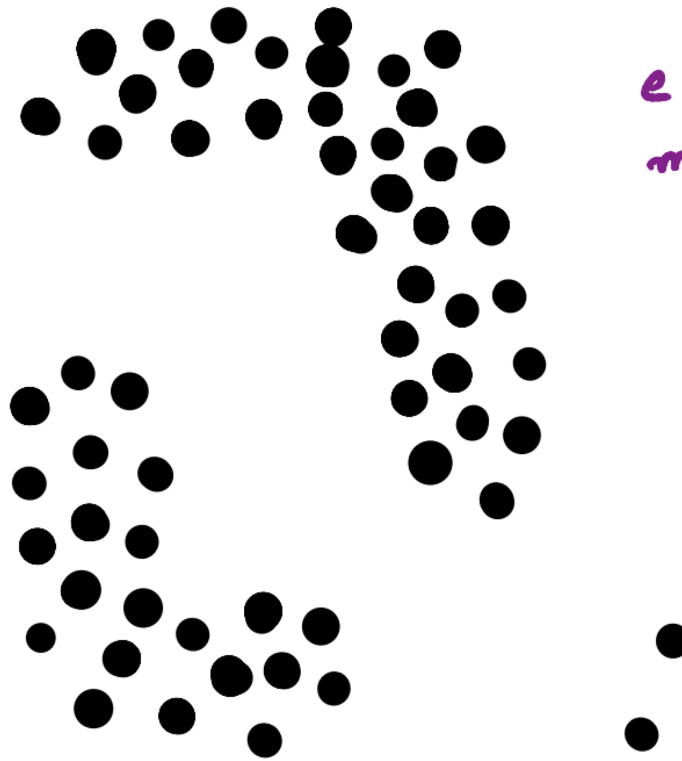
DBSCAN Algorithm

Input : epsilon and minPoints

1. Identify core points, boundary points and noise points.
2. For each un-clustered core points:
    1. Create a new cluster
    2. Add all points that are un-clustered and density connected point to current point and put into same cluster.
3. For each un-clustered boundary point/border point assign to its cluster according to its nearest core points.
4. Leave all noise points.

DBSCAN Dry Run

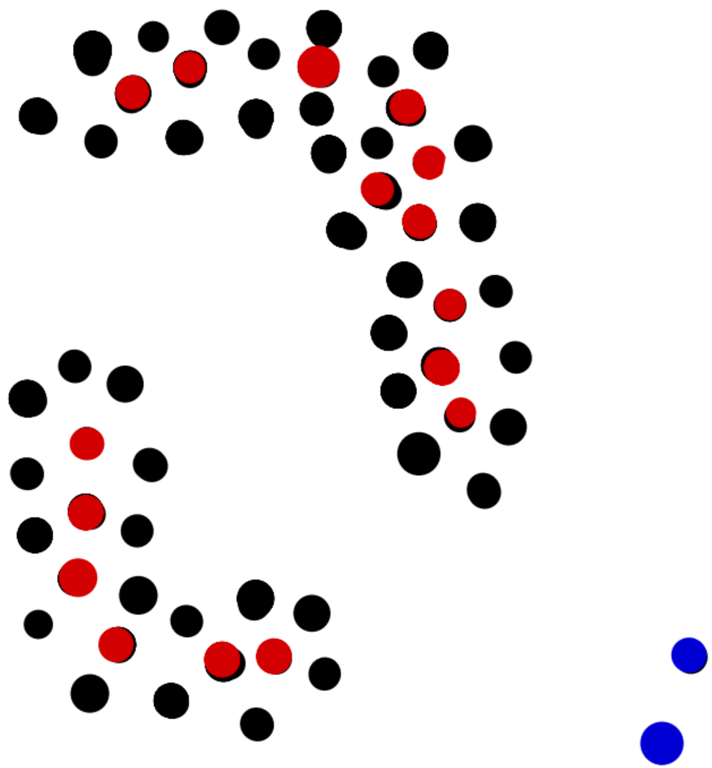
Initial :



epsilon = 1 unit
min Points = 5

DBSCAN Dry Run

step 1:



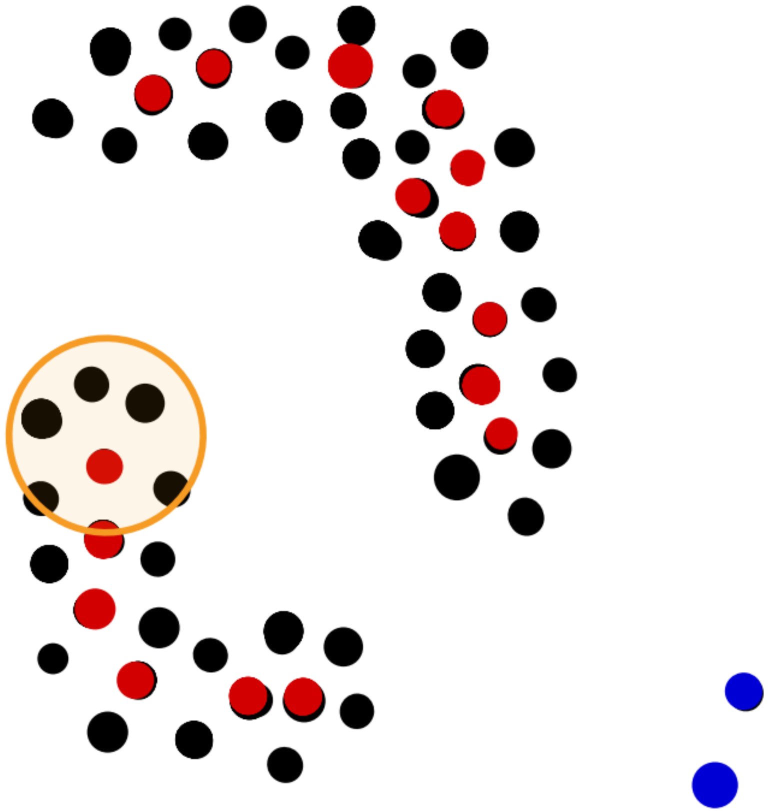
Red → Core pt.

Blue → Noise pt.

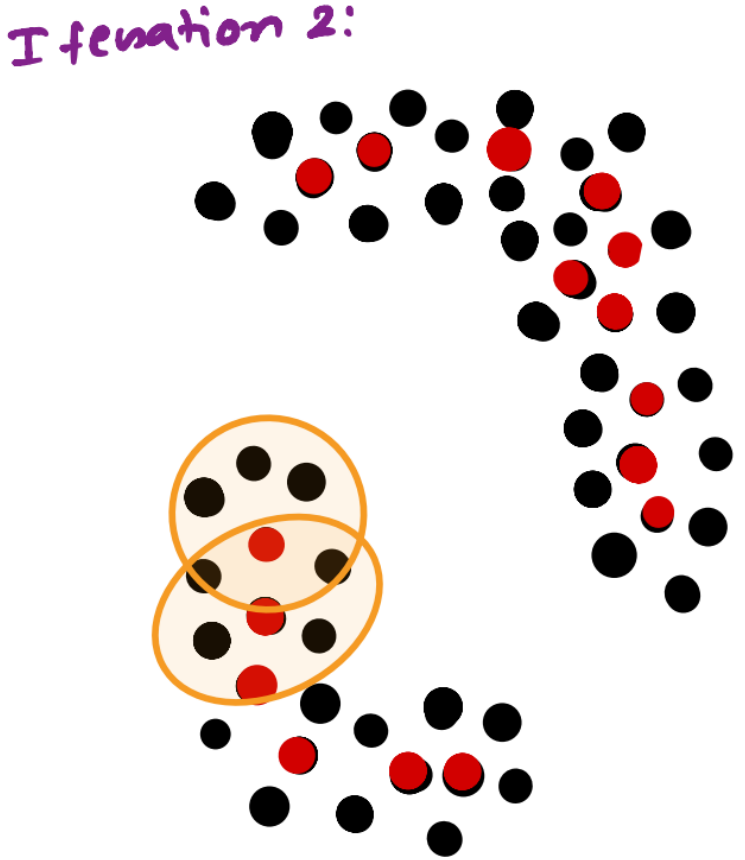
Black → Border pt.

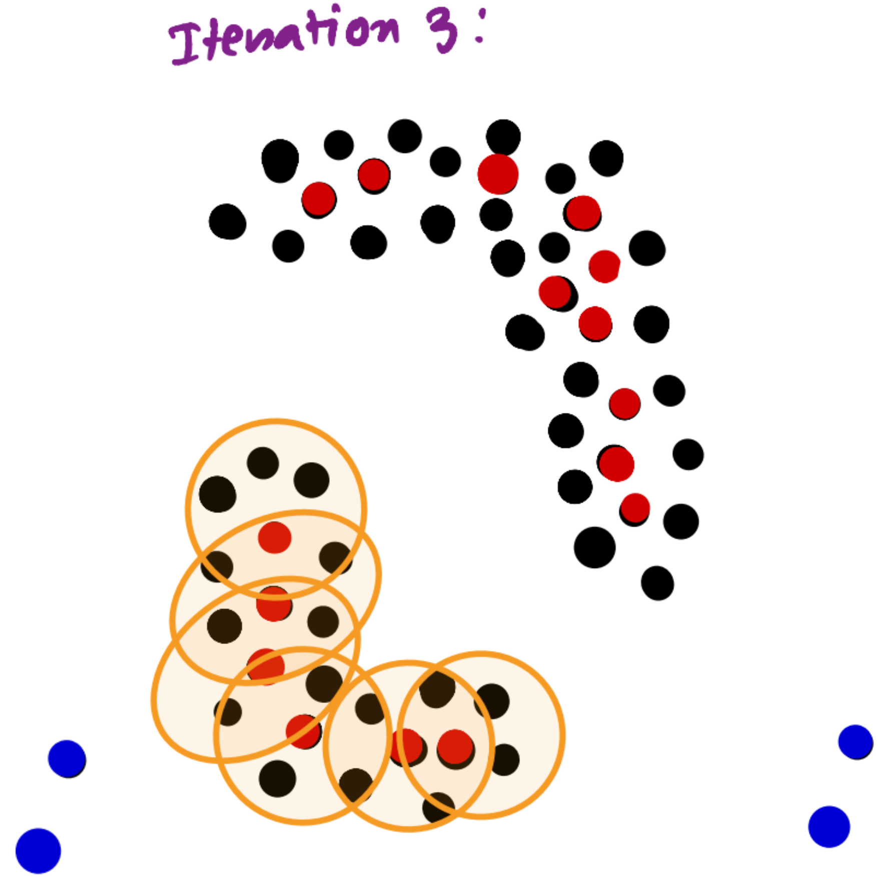
DBSCAN Dry Run

Iteration 1 :



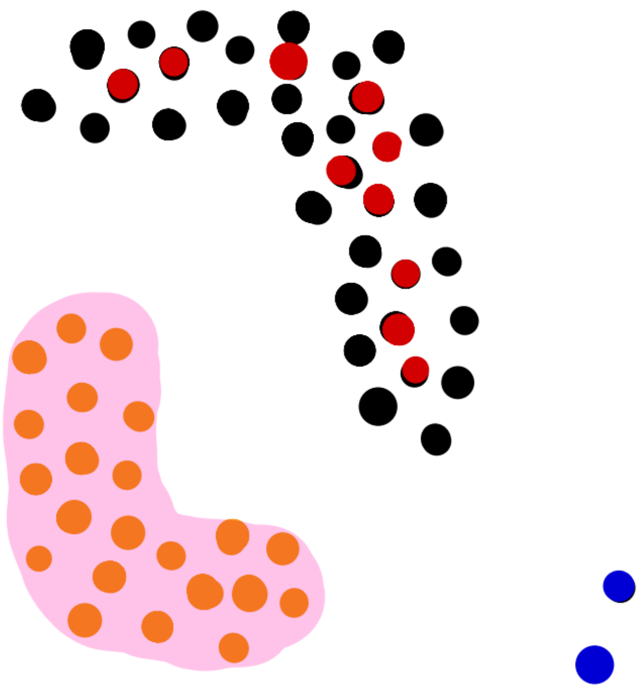
DBSCAN Dry Run



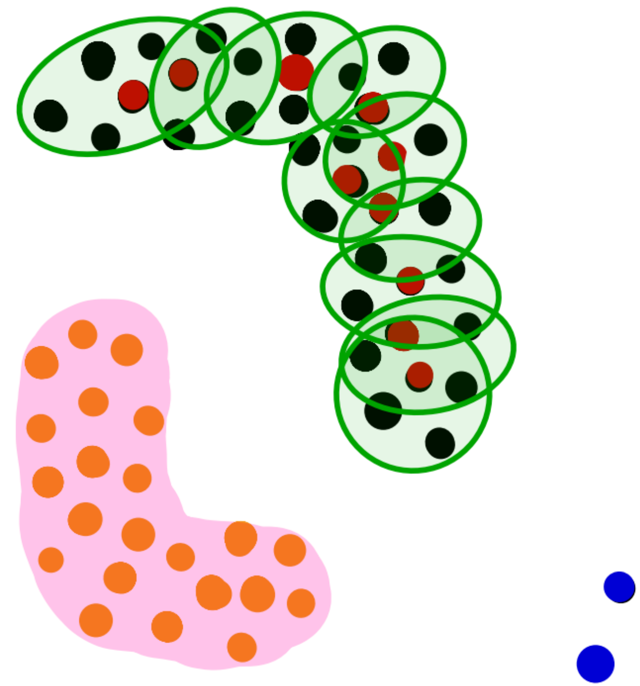


DBSCAN Dry Run

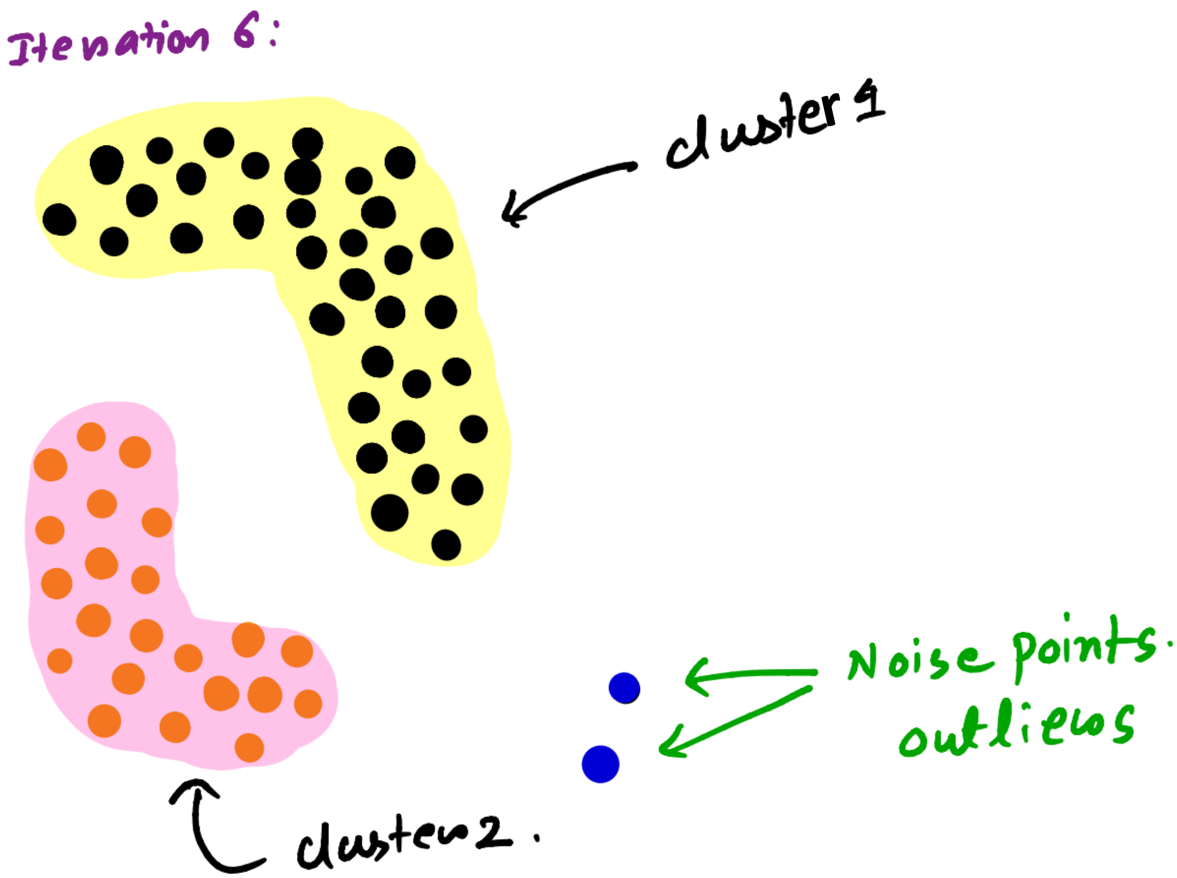
Iteration 4:



Iteration 5:



DBSCAN Dry Run



Advantages of DBSCAN

- DBSCAN can discover clusters of arbitrary shape, unlike k-means.
- It is robust to noise, as it can identify outliers.
- It does not require the number of clusters to be specified in advance.

Dis-advantages of DBSCAN

- It is sensitive to the choice of the epsilon and minPoints parameters.
- It has a high computational cost when the number of data points is large.
- It is not guaranteed to find all clusters in the data.

Referances

[1] Pattern Recognition and Machine Learning , Christopher M. Bishop.

[2] Stanford Handouts written by Chris Piech. Based on a handout by Andrew Ng, https://stanford.edu/~cpiech/cs221/handouts/kmeans.html

[3] Analytics Vidya , https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/