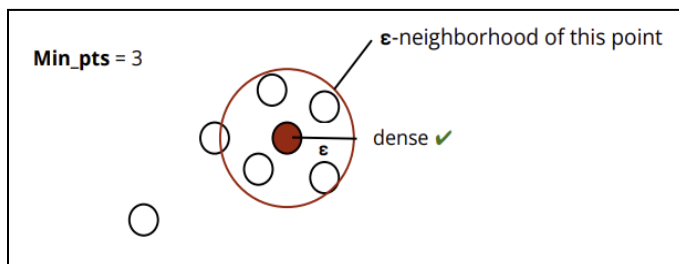


I. Density Based Clustering

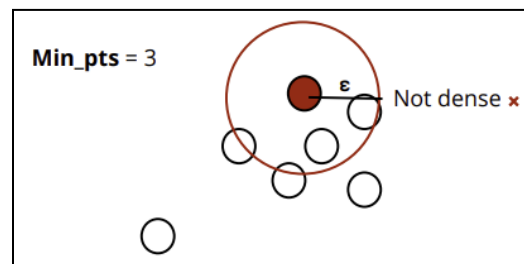
A clustering method that groups together data points that are closely packed (high-density regions) and separates them from areas of low density (sparse regions or noise).

It's different from K-means because it doesn't assume a specific cluster shape or number of clusters — it finds clusters based on the density of points in space.

How should we define density: density is defined by counting how many points fall within a fixed radius (ϵ) around a point. If there are at least `min_pts` points in that area, **it's considered a dense region**.



A. This is a dense region

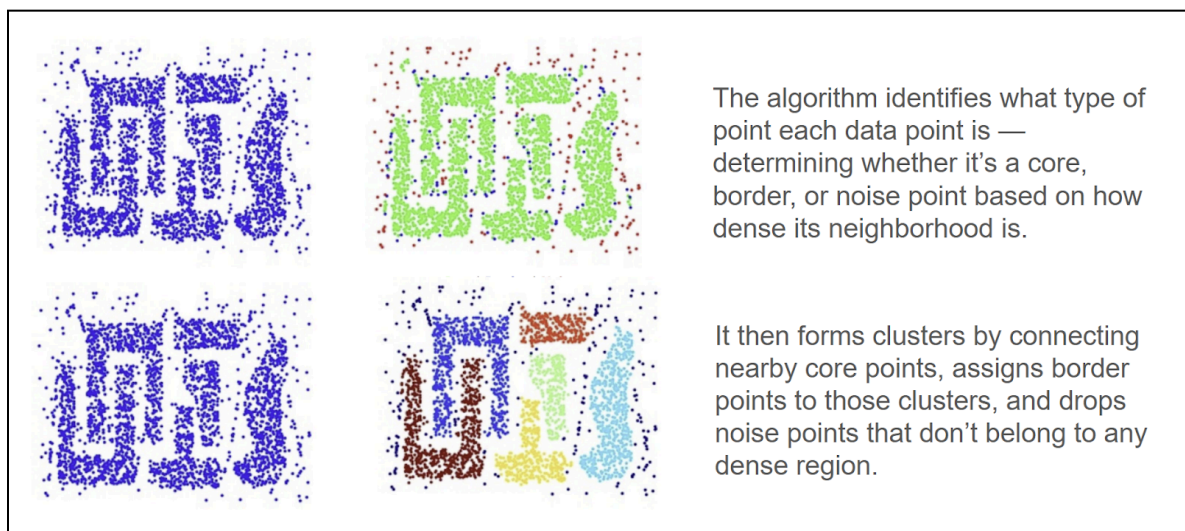


B. This is not a dense region

But the datapoint in visual B was part of a dense section in visual A this shows that a point can appear “not dense” on its own (fewer than `min_pts` points within its ϵ -radius) but still belong to a dense cluster if it's near other dense (core) points.

So, we distinguish between three types of datapoints:

- Core point: has at least `min_pts` points within its ϵ -radius (the center of a dense region).
- Border point: doesn't meet the density requirement itself, but lies within the ϵ -radius of a core point.
- Noise point: is too far from any dense region — not a core or border point, so it's treated as an outlier.



II. DBScan Algorithm

In DBScan, the parameters ϵ (the neighborhood radius) and min_pts (the minimum number of points needed to form a dense region) are given as inputs.

The algorithm steps are:

First take a datapoint:

1. Find the datapoint's ϵ -neighborhood (identify points within the chosen radius)
2. Check if it's a core point (those with at least min_pts neighbors inside their ϵ -radius)

Then if it's a core point:

3. Iterate through neighborhood points to find any other core points within the neighborhood assign them to our cluster.
4. Then go to the next datapoint in the dataset (and restart steps)

But if it's not a core point:

5. Check if it's a border point, label it as such if it is
6. Otherwise, it's a noise point, so label it as such
7. Then go to the next datapoint in the dataset (and restart steps)

Lastly once we iterate through all points and perform the above steps on them:

8. Assign border points to their nearest cluster.

In short, DBSCAN builds clusters by chaining together dense areas and ignores sparse or isolated points as noise.

