11) DBSCAN

- Stands for Density Based Spacial Clustering of Applications with Noise
- This is one of the few algos that actually cluster the data rather than partitioning it, meaning it will help us find outliers rather than putting them into different clusters
- Is a "true clustering" algorithm
- It can have points that dont belong to any clusters (meaning they're outliers)
- **Strengths:**
  - No need to specify the number of clusters
  - Allows for noise
  - Can handle arbitrary-shaped clusters
- **Weaknesses:**
  - Requires two parameters (epsilon and n_clu)
  - Finding appropriate value for these hyperparameters (aka fine tuning them) can be difficult, especially in higher dimensions
  - Does not do well with clusters of different densities

How it works

We are making the assumption that points in a cluster should be a certain distance from one and another, and within a certain neighbourhood. So we would randomly select points from these higher density regions and slowly expand our clusters, and as we expand, we only include points that are at a certain distance from the points already included in that cluster. The algorithm ends when all points have been classified as belong to a cluster, or belonging to noise. We need the following required inputs:

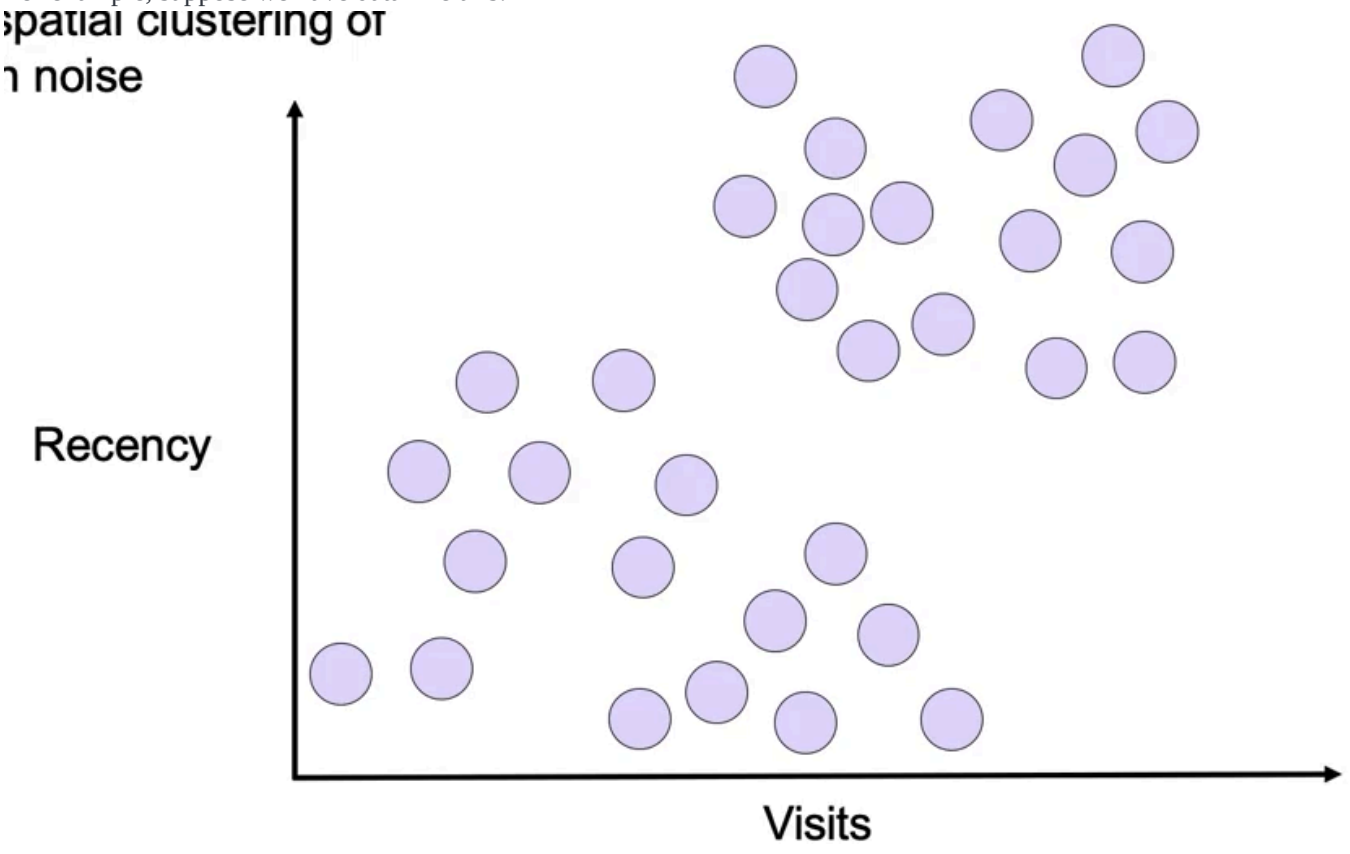
- **Metric:** Function to calculate distance
- **Epsilon:** Radius of the local neighbourhood
- **n_clu / min_samples:** Determines density threshold, meaning min amount of points for a particular point to be considered as a core point of the cluster

There are three different possible labels for any given point:

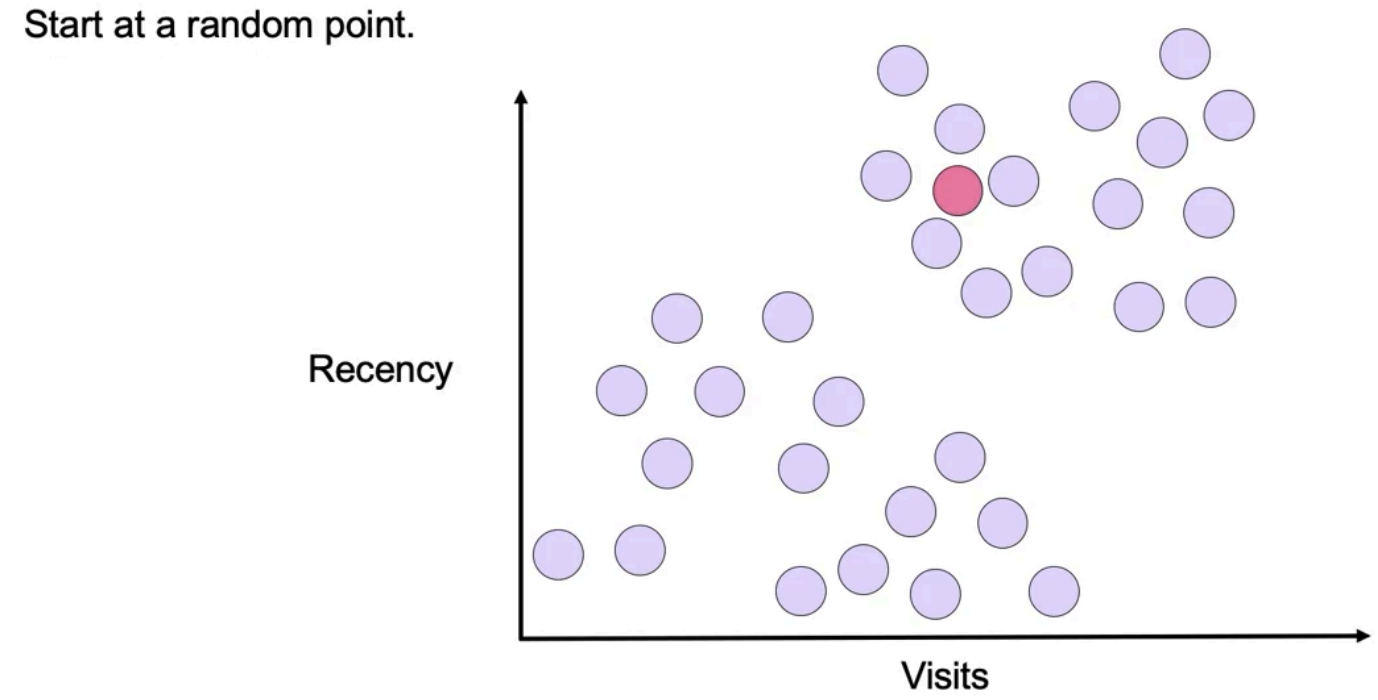
- **Core points:** Points that have atleast n_clu-1 neighbours (ie atleast n_clu-1 points in epsilon-radius)
- **Density reachable aka border points:** Points which are neighbours of a core point but doesnt have enough neighbours to be a core point

- **Noise:** Points which are not part of any cluster, meaning a point which has no core point as its neighbour (meaning a point which is atleast epsilon distance away from every core point)

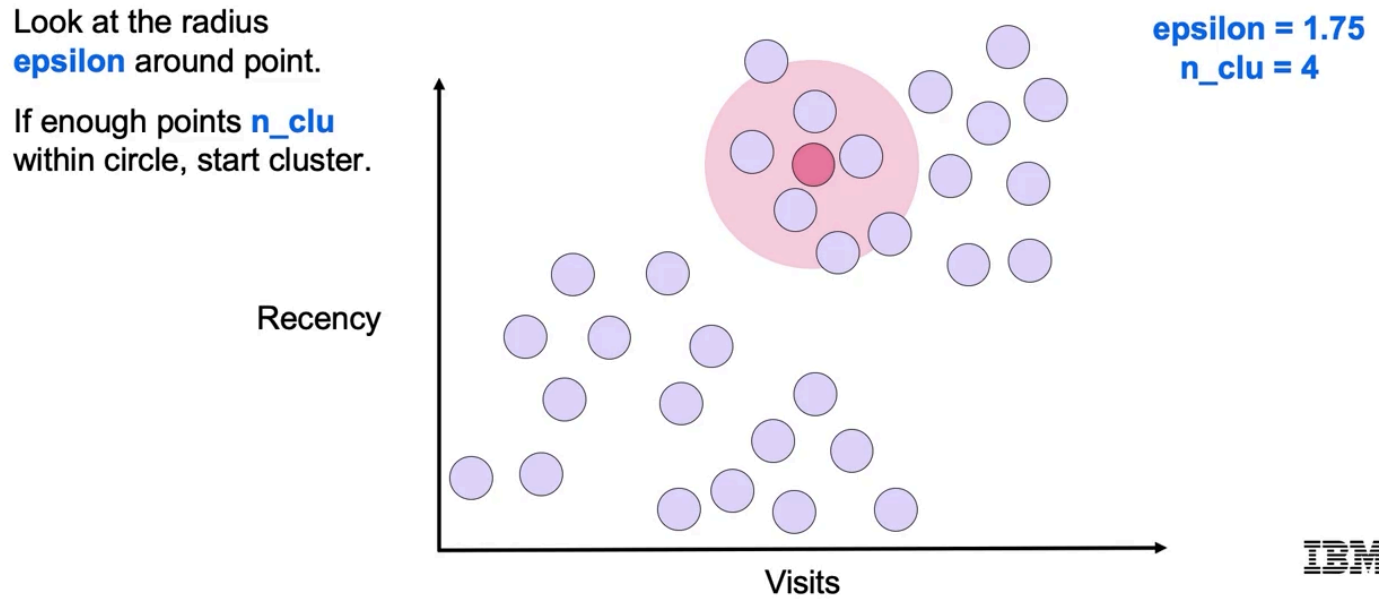
For example, suppose we have data like this:



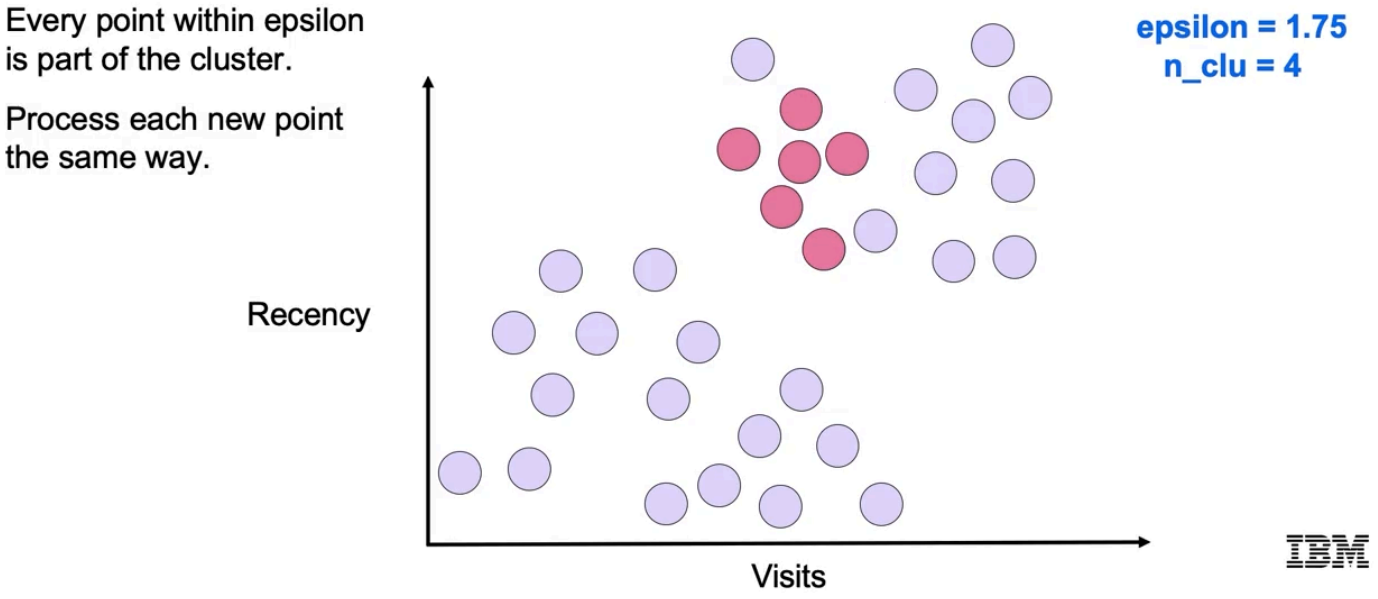
First, we will choose a random point like so:



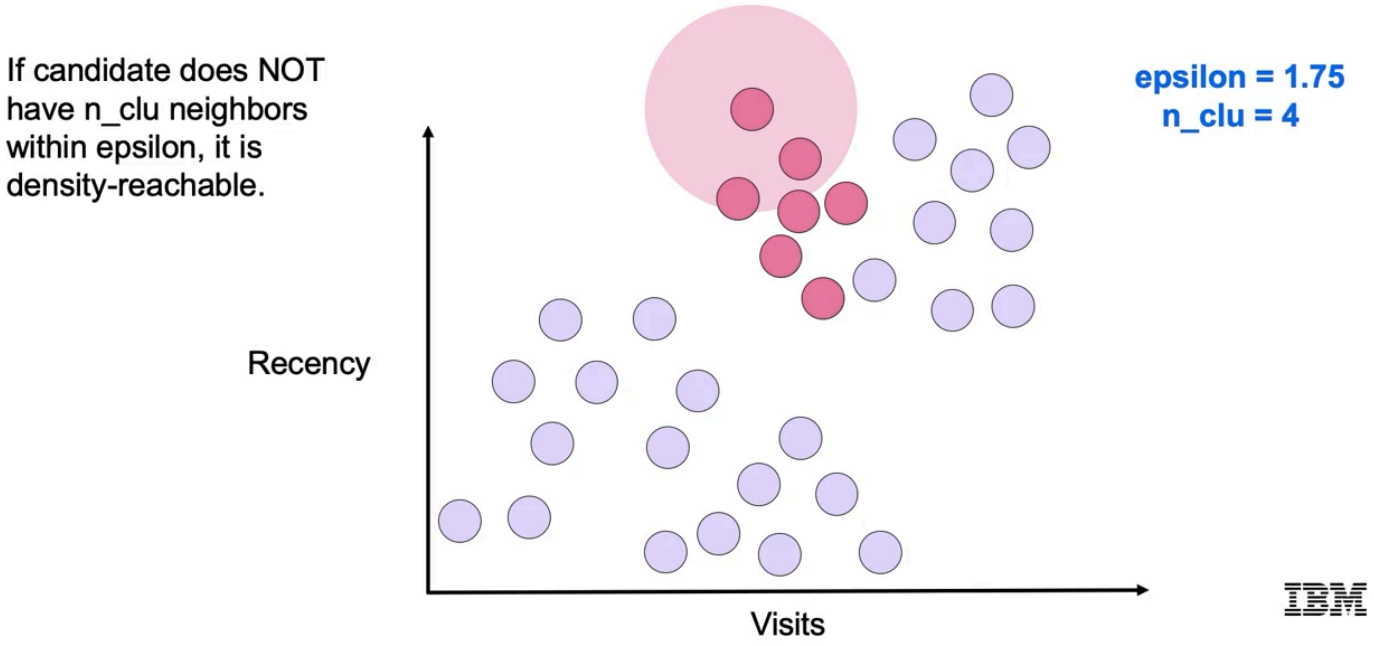
Then we look at the radius epsilon around that point and if enough points are there, we start a cluster



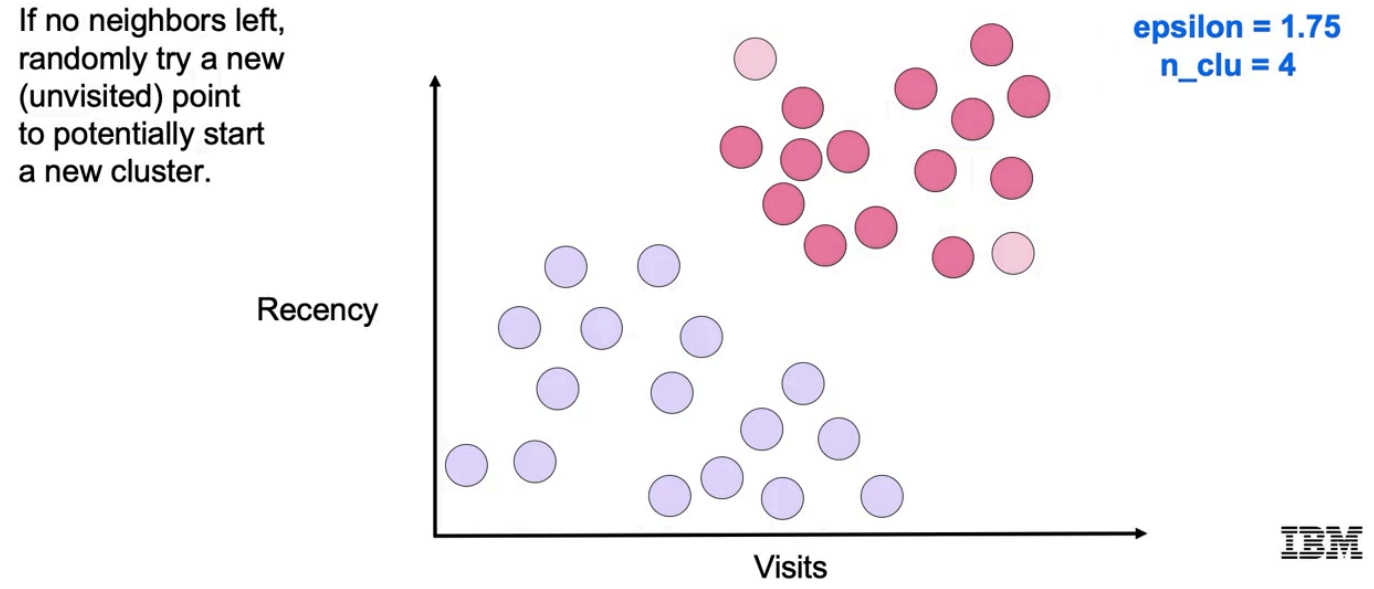
Since enough points are there, we include those points in our cluster like so (remember, all these are core points):



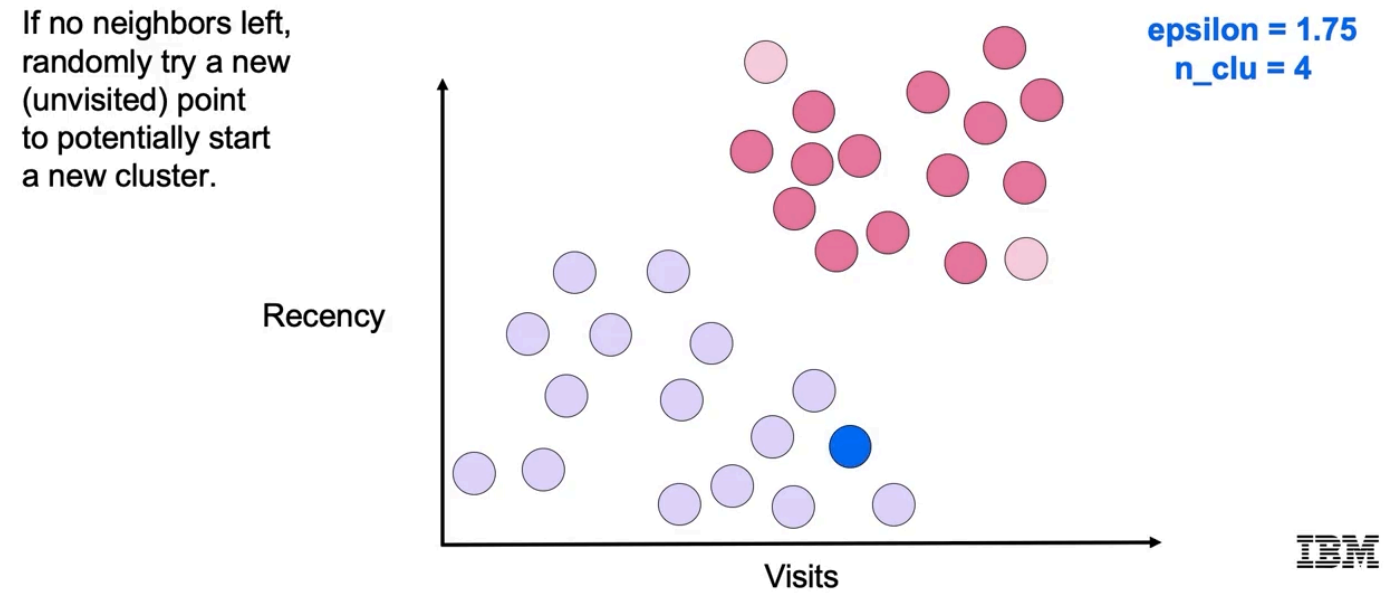
Some points in our cluster dont have less than n_clu-1 points near them, so they become density-reachable points, like this point:



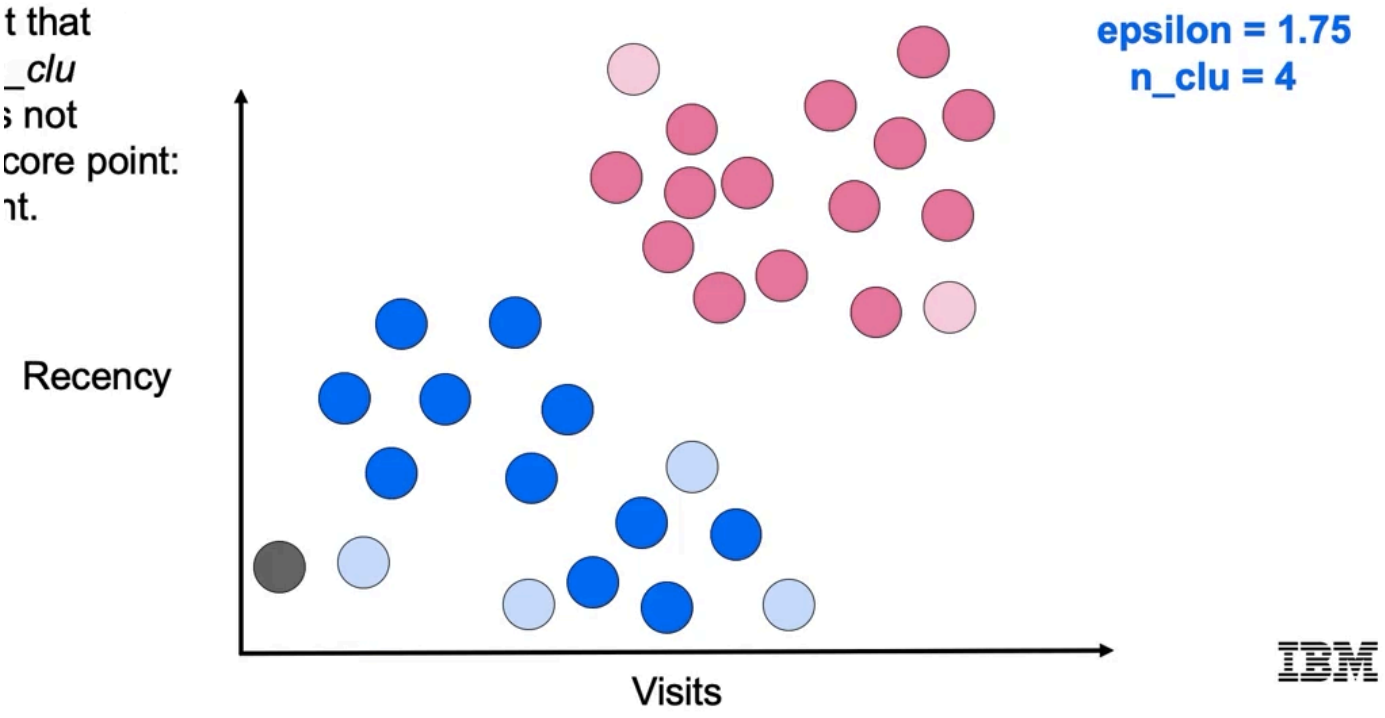
We keep going through all the points in the cluster until no point is left visited by this "chain reaction", so we end up with something like this:



Once this is done (ie there are no neighbours left), we will randomly try an unvisited point to potentially start a new cluster, like so:



And we will repeat the same process with this point/cluster, so at the end we are left with the following clusters:



Notice how the point in the bottom left is black. That is because it is classified as a "noise point" because it doesnt have a core point or n_clu points in its epsilon-radius-neighbourhood

Code

```
from sklearn.cluster import DBSCAN

model = DBSCAN(eps=0.5, min_samples=5, metric='euclidean')
model.fit(X)
clusters = model.labels_
```