- DBSCAN Intro
- Key Concepts in DBSCAN
  1. Min-points
  2. Eps (Epsilon)
  3. Core-points
  4. Border point
  5. Noise Point
- DBSCAN Algo
- Key Hyperparameters
- Code implementation
- Introduction to Anomaly/Novelty/Outlier Detection
- Distribution Based Anomaly Detection
  - RANSAC (Random Sample Consesus)
  - Eliptic Envelope
- Sklearn Implementation

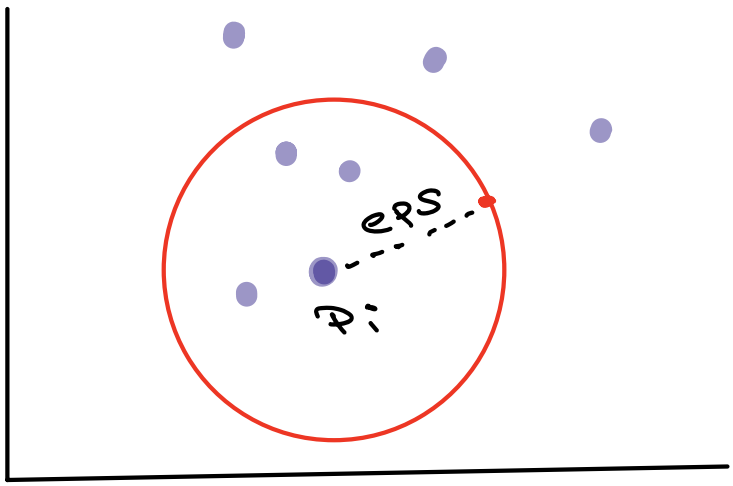
DBSCAN

- Density-Based Spatial Clustering of Application with Noise
- DBSCAN handles outlier effectively
- Scales well with Big Datasets
- No need to define n_clusters

Core Concepts

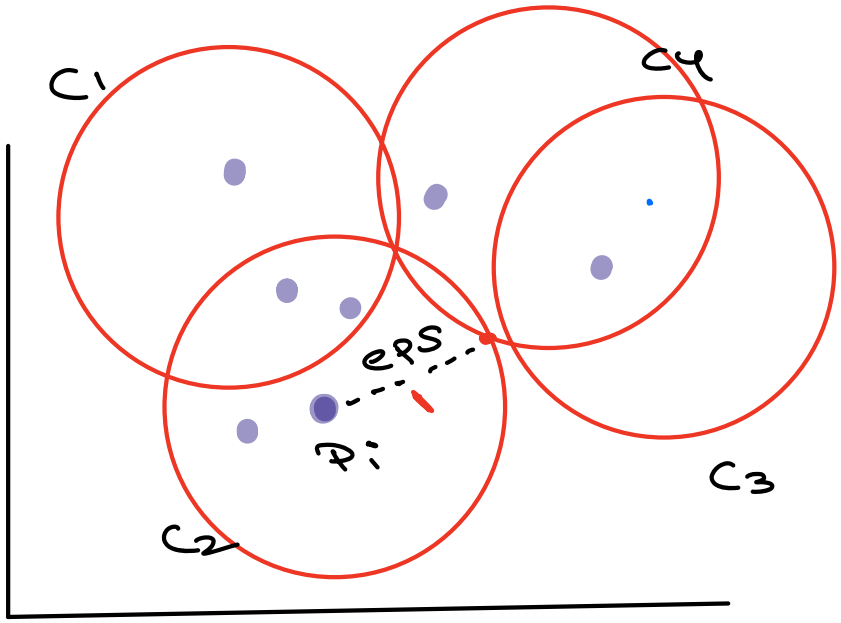
1. Min points
2. Eps
3. Core-point
4. Border-point
5. Noise-point

density @ P_i = # of points that are within some radius eps



- 4 (including P_i)
  3 (Excluding self)

⇒ Min-pts → No of points allowed within eps radius for to be considered Dense regions



min-pts ⇒ 3

eps ⇒ 1

C1 ⇒ 3

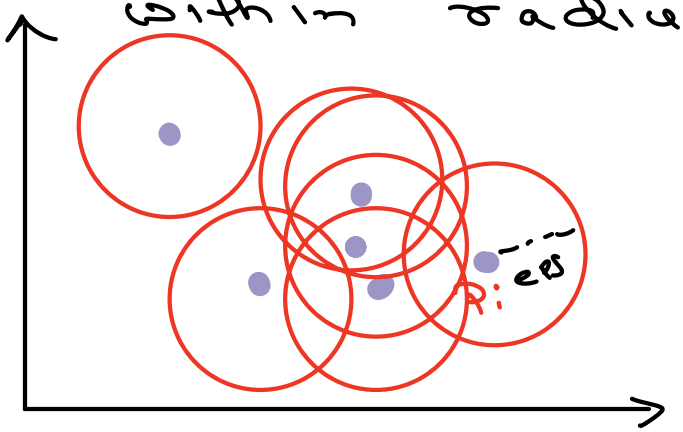
C2 ⇒ 4

C3 ⇒ 1

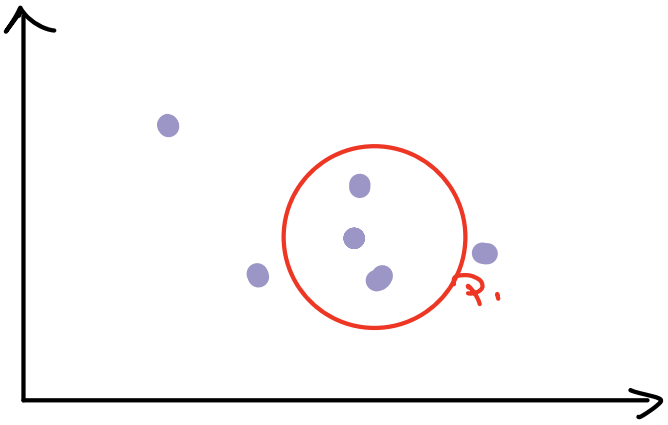
C4 ⇒ 2

Core - points

A point which has #p ≥ min-point within radius of eps



min-pts ⇒ 2



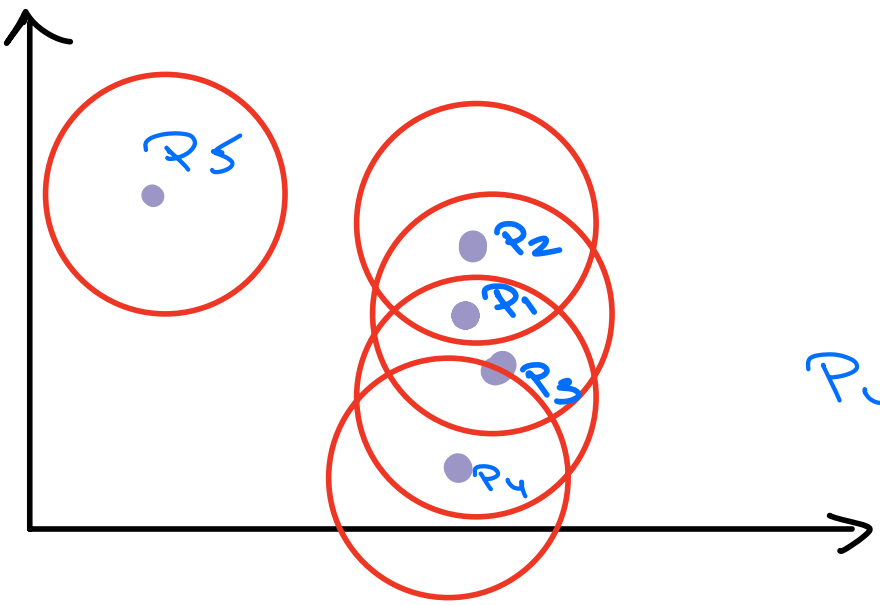
min-pts = 2

D.R of min-pts is

C.P of n-1

Border-point

- A point which is not a Core point
- It lies in neighborhood of point Q which is a Core-point



min-pts = 2

$P_4, P_2 \Rightarrow B.P$

$P_1, P_3 \Rightarrow C.P$

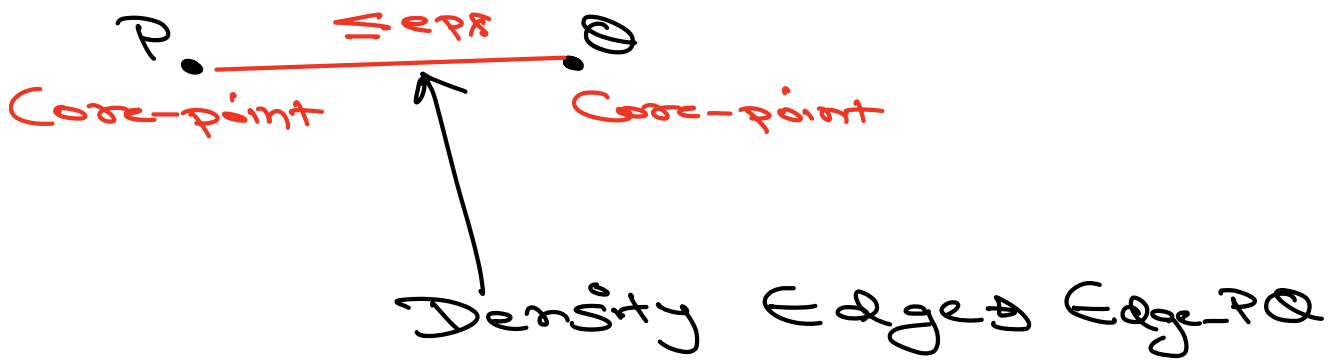
Noise-Points

- A point which is neither Core nor Border

Ex: P_5 is Noise point

Density Edge / Connection

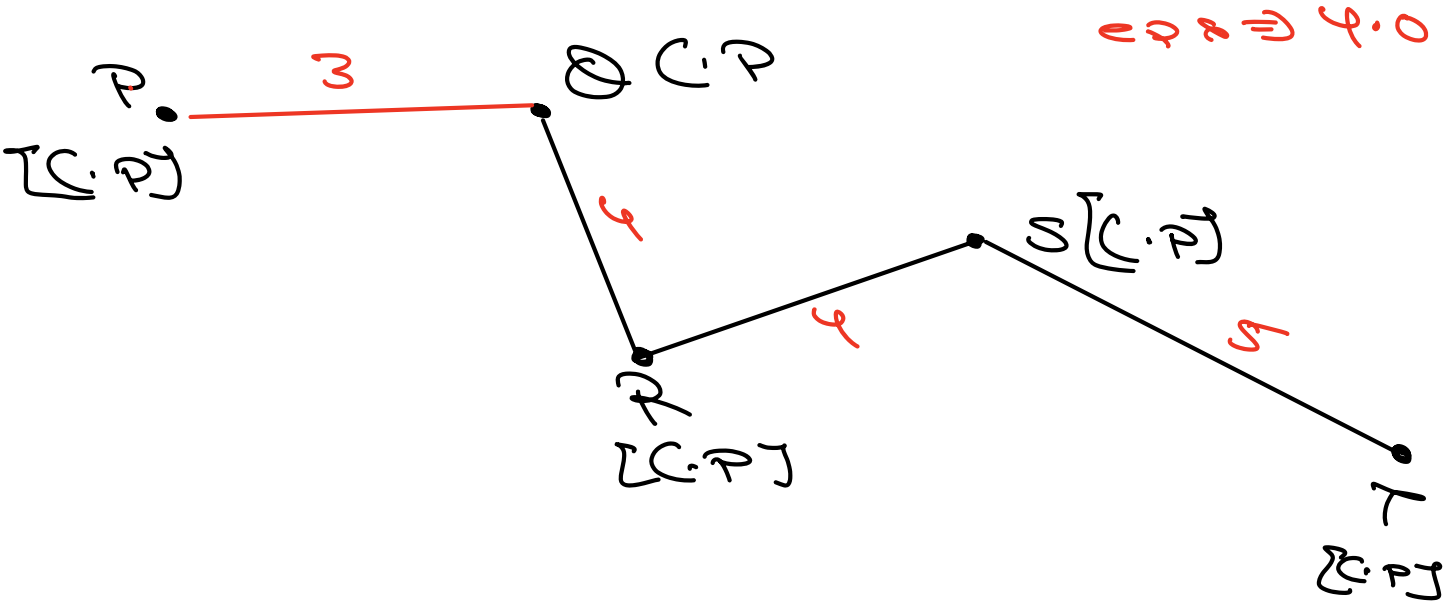
- If Edge connecting two Core points such that $len(edge) \leq Eps$



Density Connected points

Two points P and Q are called D.C.P,

1. if Both P and Q are Core points
2. if there exist Density Edges Connecting point P and Q



$P \to S$: DCP

$P \to T$: No D.C.P

Q For two points to be D.C.P can the Dist. between them be greater than $\in$?

Yes

DBSCAN ALGO

Step 1: Annotate Every point as C.P, B.P or N.P

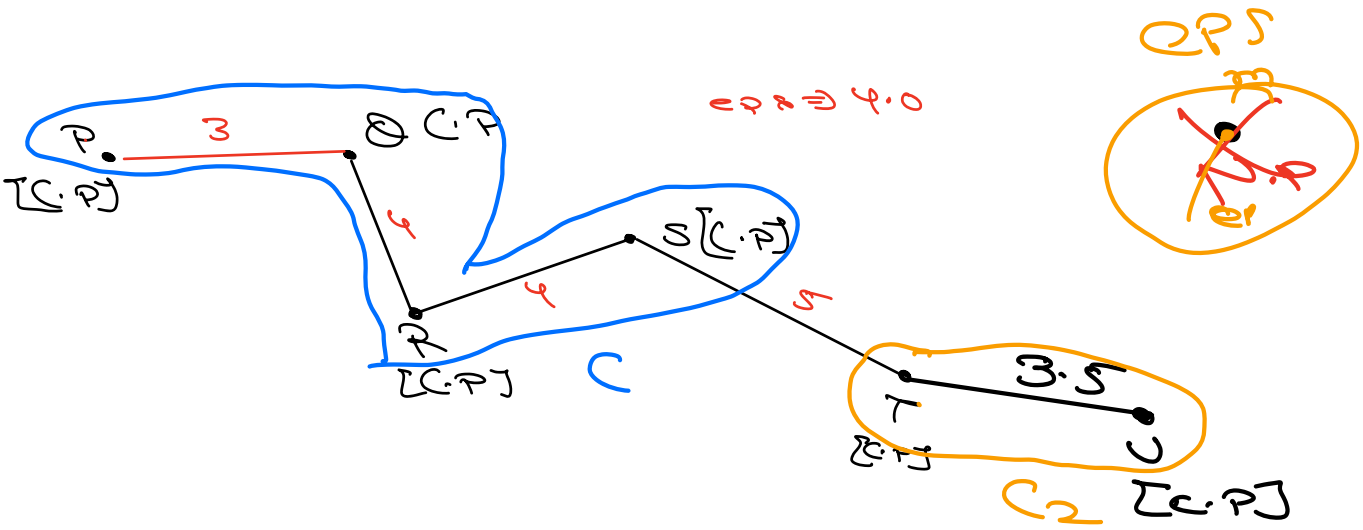
B.F $\Theta(N^2)$

- RangeQuery(D, x_i, Eps) T.C. N^2 / $N \log N$

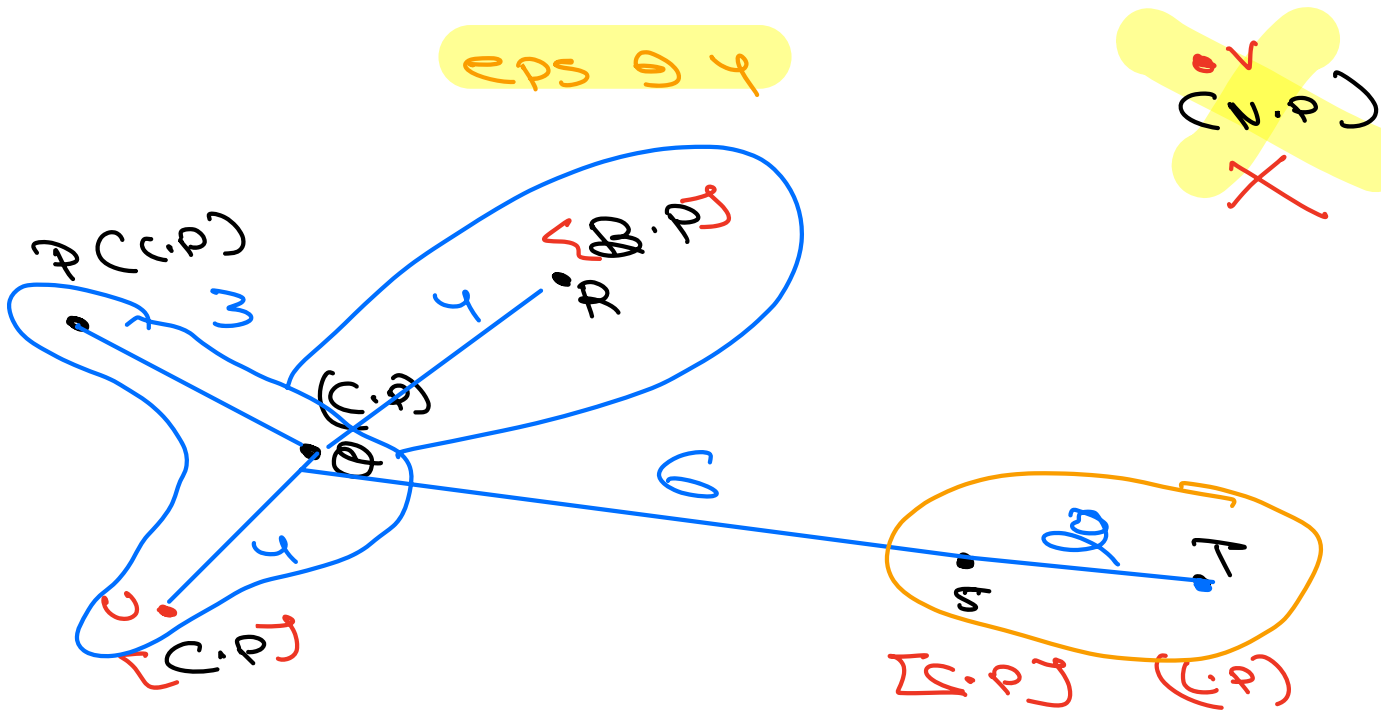
Step 2: Remove All Noise-points

Step 3: For each CorePoint i.e. not part of any Cluster

- Create a newCluster with Point P
- Add all the density connected points to this Cluster



Step 4: Assign each Border-point to its nearest Cluster



C_1 = P Q R U

C_2 = S T

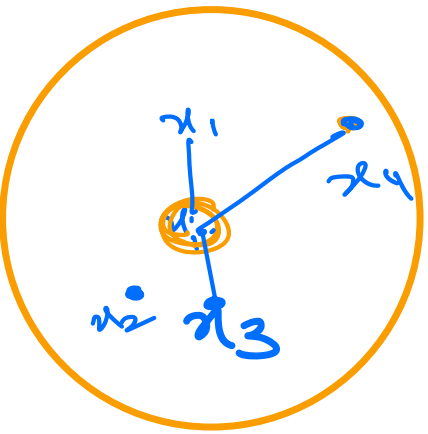
100.5
6.10

Hyperparameter

- min_pts ⇒ Rule of thumbs → dimensions
  - ↳ min_pts ≥ D+1
  - ↳ Typically D×2

If D.S is very Noisy ⇒ higher Value of min-pts

② Eps :



With a fix value of min-pts

$x_1 \rightarrow d_1$

$x_2 \rightarrow d_2$

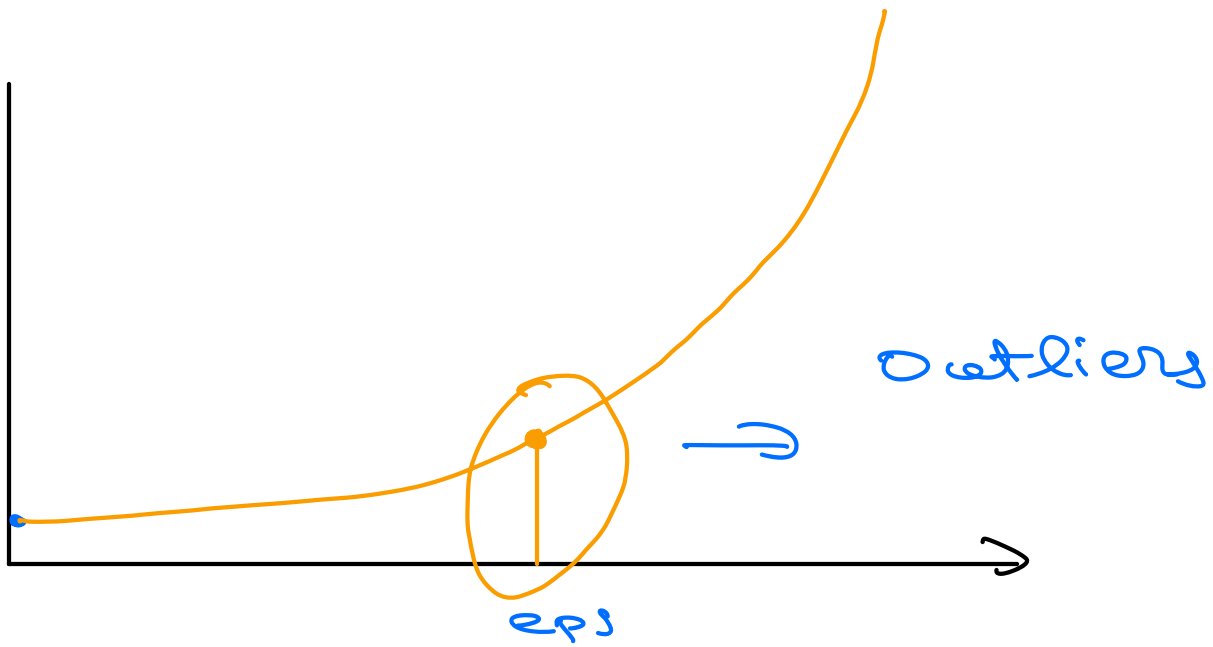
$\vdots$

$x_n \rightarrow d_n$

distance of farthest B.P as C.P

Ci min-pts = 4 then we take d_i as 4th high

③ Sort all distance and plot them



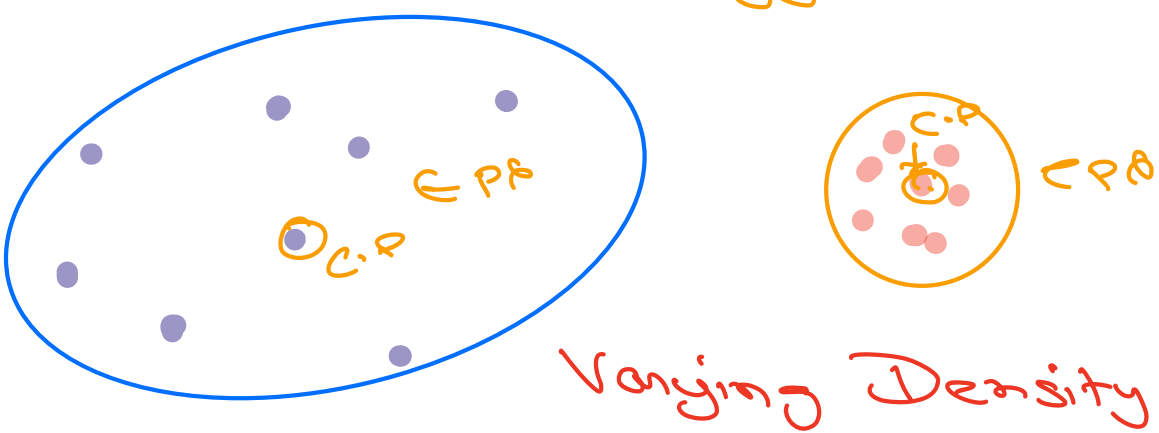
Euldiean Distance

Advantages

1. Resistant to Noise
2. Can handle any shape and size of Cluster
3. Doesn't require n_clusters a priori
4. Only two parameters ⇒ Eps, minPts
5. Speed : Thanks to DB Community NLogN

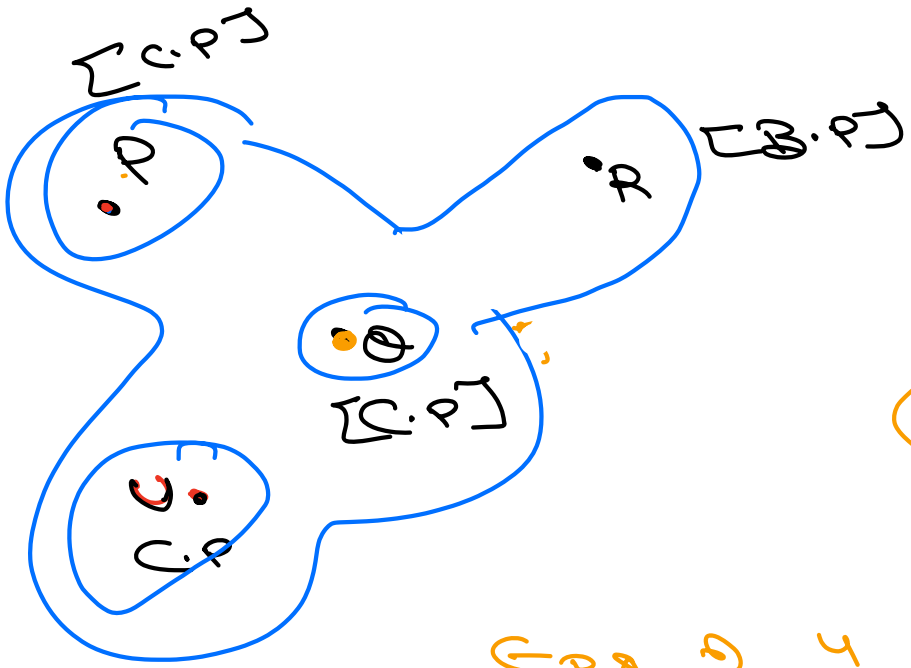
Limitation

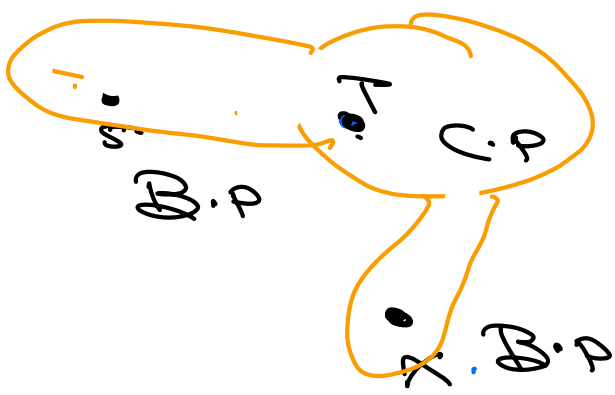
1. Very sensitive to choice of Hyperparameter
2. Can't Handle different Densities



Varying Density

- Topics Remaining
  1. DBSCAN Implementation
  2. Anomaly Detection
  3. RANSAC and Elliptic Envelope



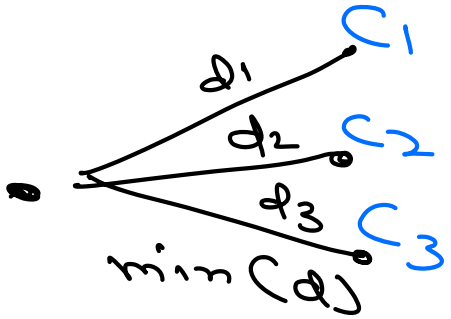


Eps = 4
min_pts = 2

Online Clustering Model

↳ Can predict without retraining

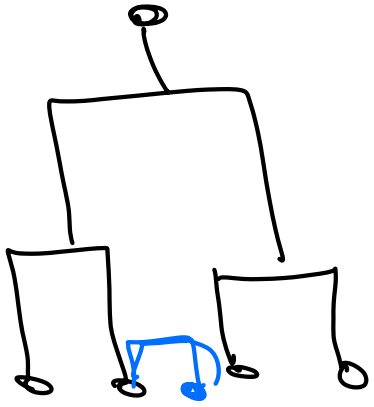
Kmeans → Centroid



- Predict

Offline Clustering Model

- Refit the full dataset to perform prediction
- Process full DS + new-point
- Predict Method Not Avail



Build Dendrogram