Outlier Detection

[Unsupervised Learning]

-> Types

-> Eliptical Envelop

-> Isolation Forest

-> Local Outlier factor.

Outliers

Q: Why do outliers exist?

→ Human / Sensor error

→ Real "unusual" data

→ Anomaly ⇒ Not "Normal"

example

→ Novelty ⇒ Never happened before

example

Eg:

→ Fraudulent transactions

→ Sales during covid

→ Electric cars → mileage data

Q: What are some ways to detect outliers?

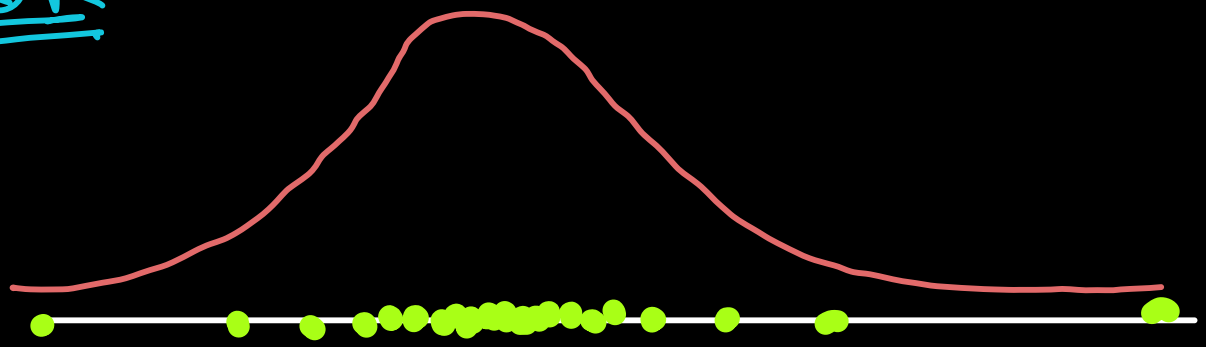
→ IQR → KNN → DBSCAN, etc.

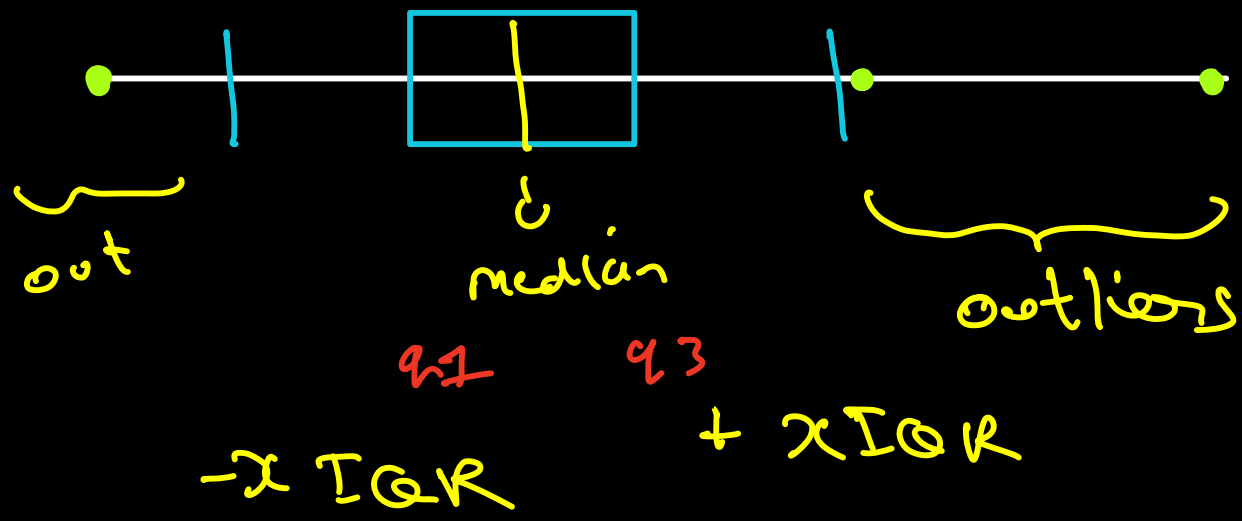
Review

Elliptical Envelope

Let's build better techniques using above ideas.

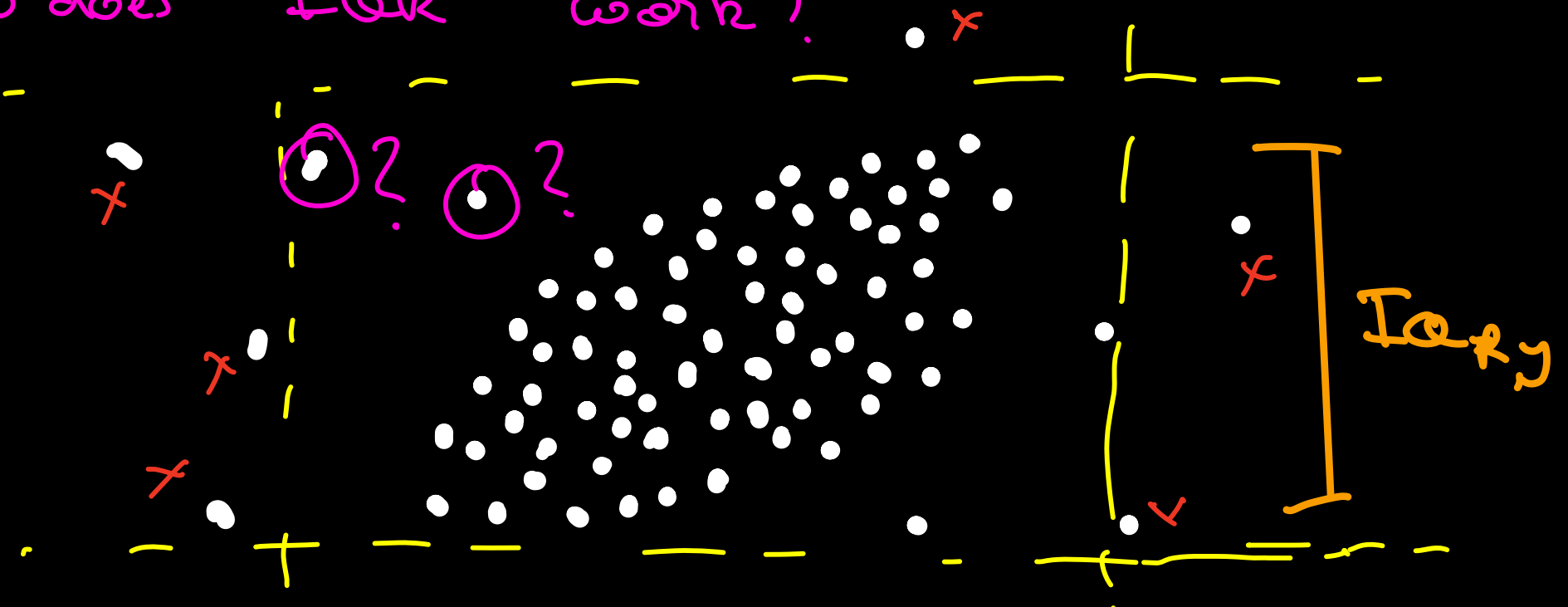
IQR





But what abt 2/n dimensions?

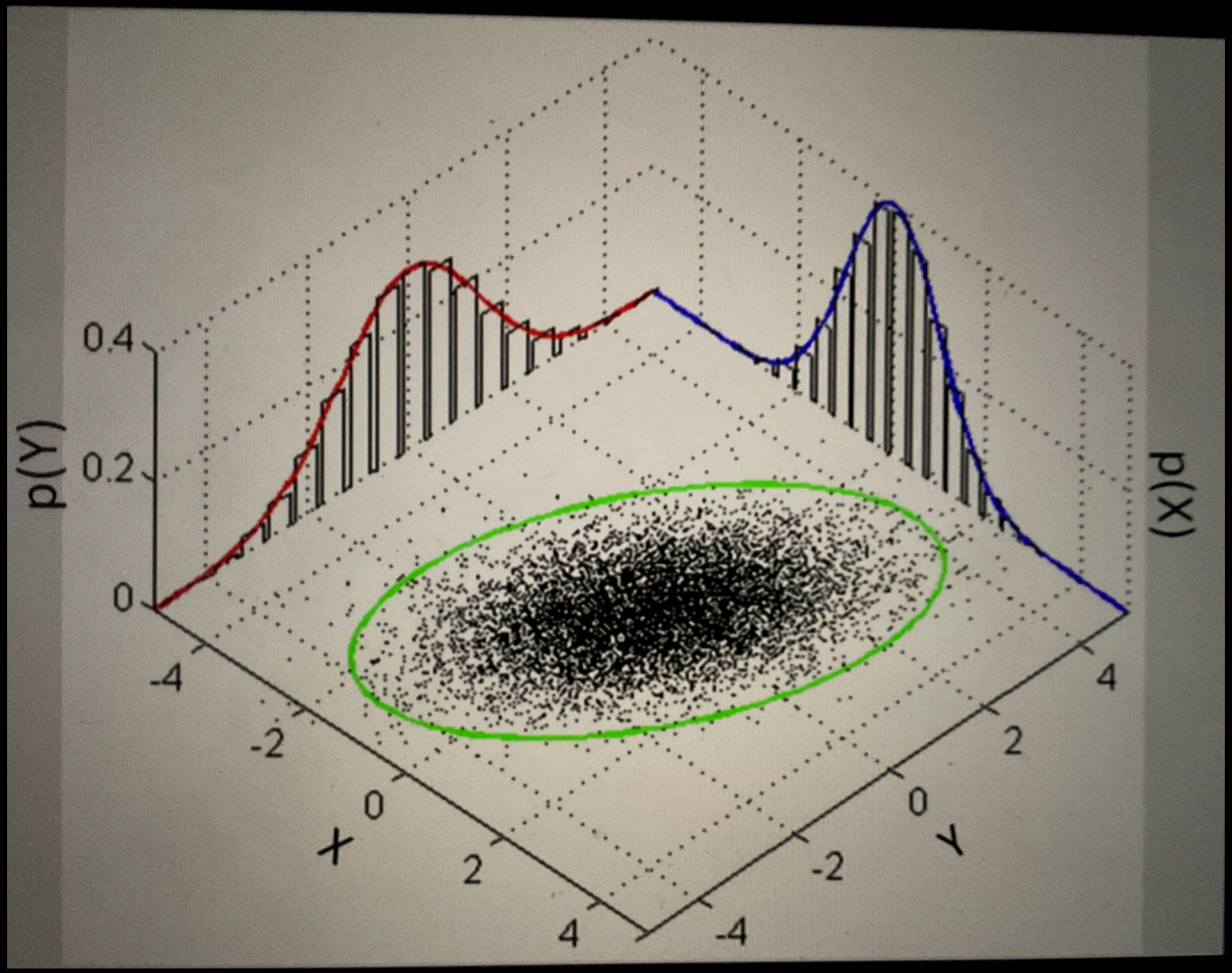
-> does IQR work?



→ N- dimensional IQR will form cuboidal boudaries. Whereas most high dimesional data may fit "globular" boundaries.

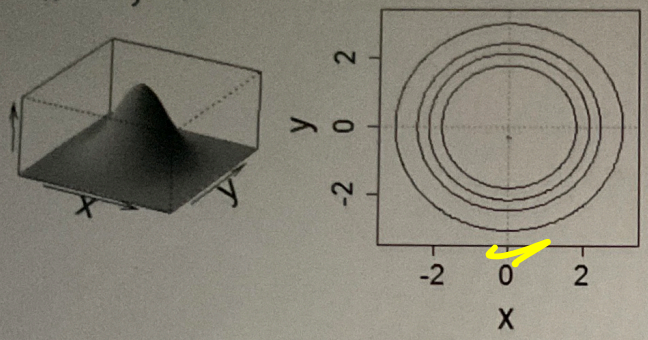
→ Any solutions ?

↳ Multi variate gaussians !

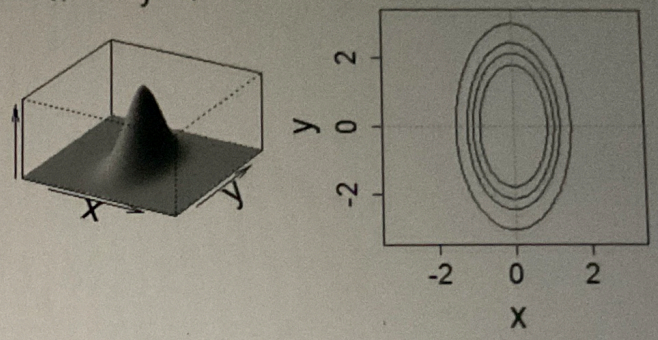


→ wiki: pdf bi variate

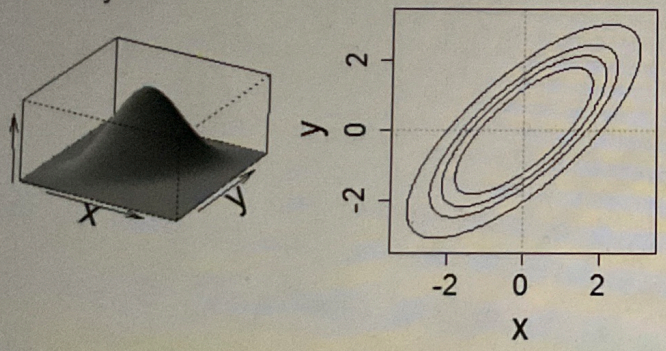
$\sigma_x = \sigma_y,\ \rho = 0$



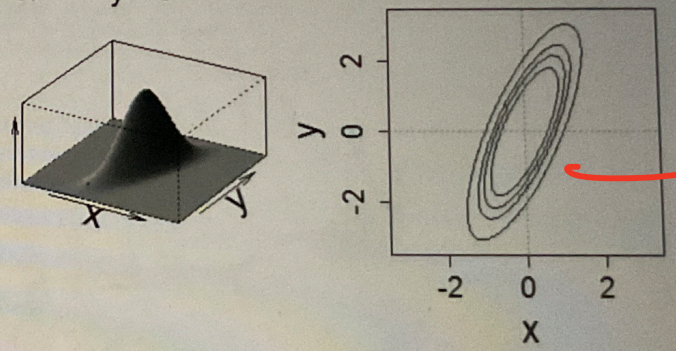
$2\sigma_x = \sigma_y,\ \rho = 0$



$\sigma_x = \sigma_y,\ \rho = 0.75$

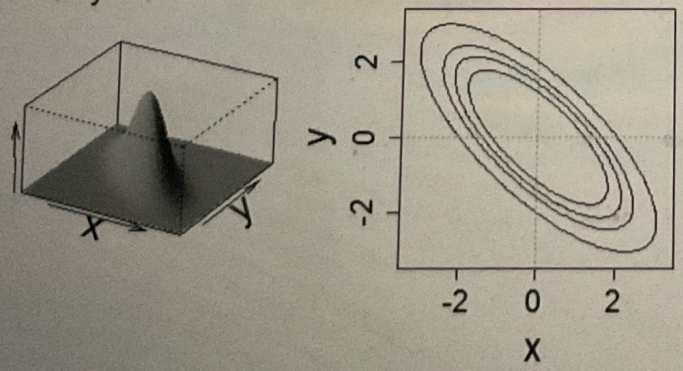


$2\sigma_x = \sigma_y,\ \rho = 0.75$

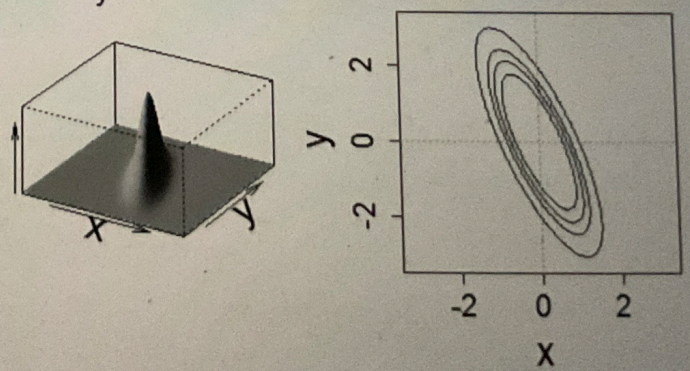


→ ellipse

$\sigma_x = \sigma_y,\ \rho = -0.75$



$2\sigma_x = \sigma_y,\ \rho = -0.75$



So we can fit multivariate gaussian to the data and points with low prob can be called outliers.

→ Assumptions : points follow gaussian distribution

How do we learn the parameters?

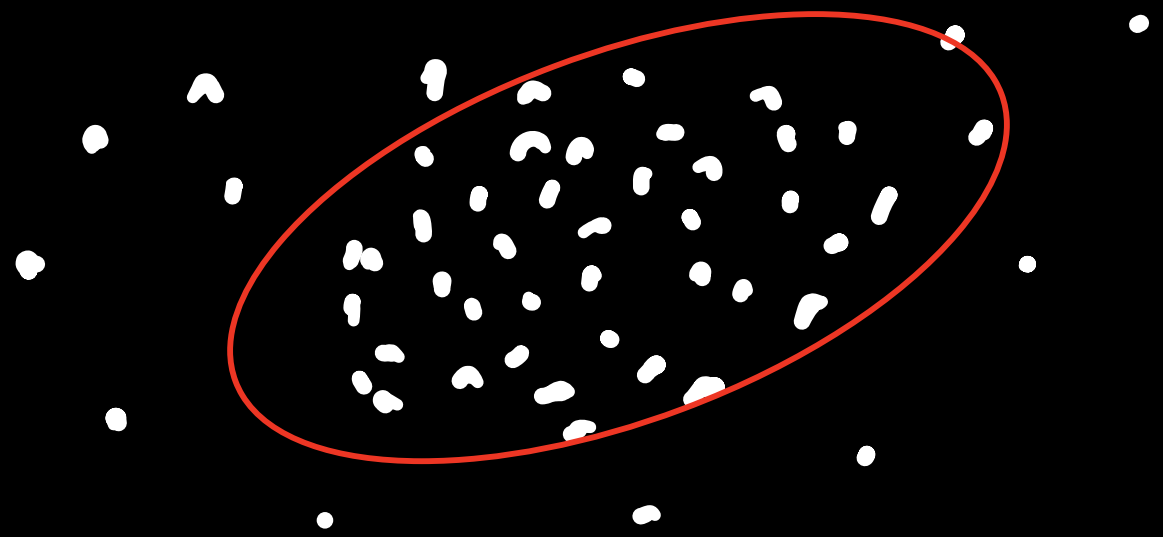
↳ μ, σ, β will be affected by outliers.

↓

For this we can use robust algos ↓

RANSAC

Random sample consensus



→ Randomly sample some points.

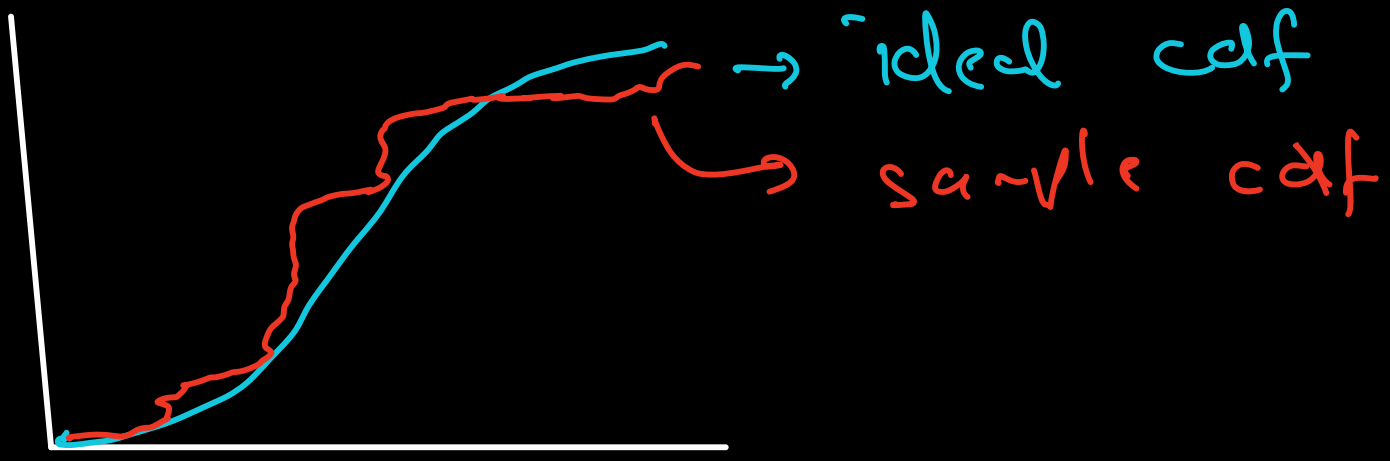
Find μ, σ, ρ of this sample.

Now compute ideal distribution

blw μ, σ, ρ

→ Compare ideal dist with actual sample.

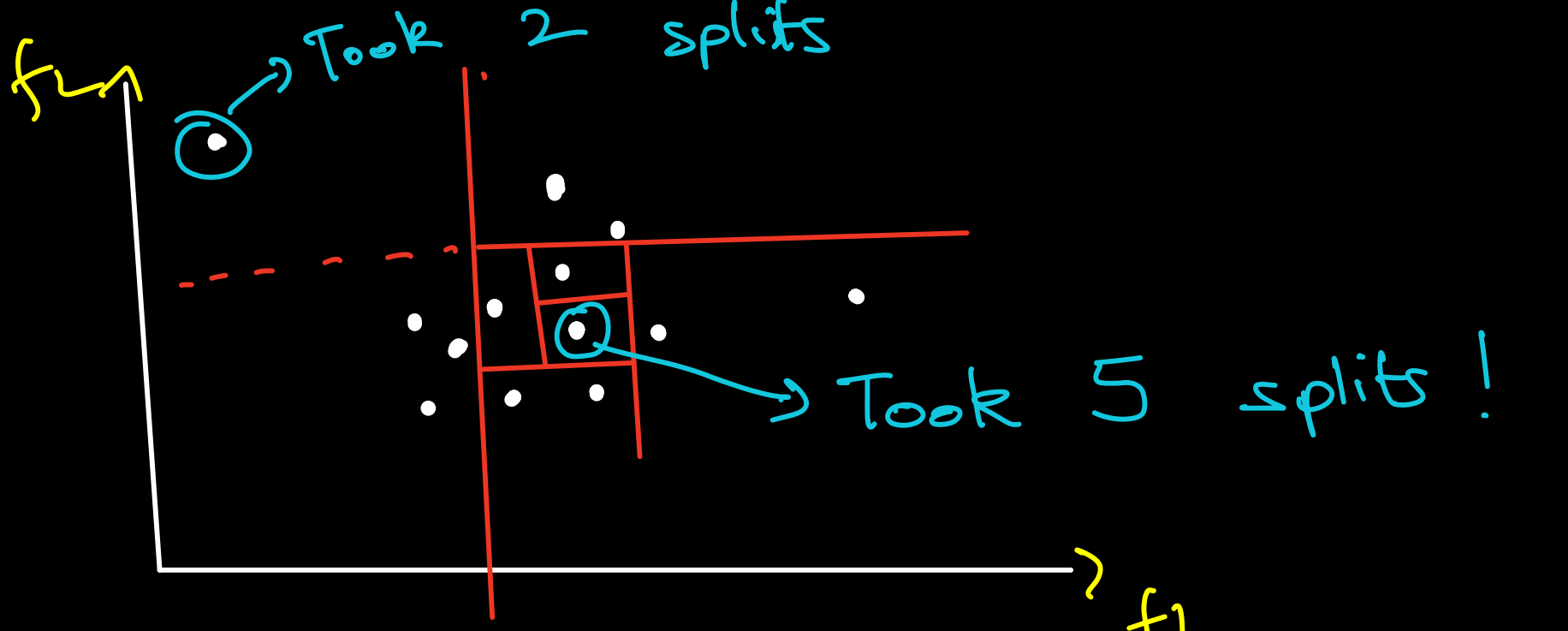
→ Choose the parameters that fit most of the samples / best fits one of the sample / avg etc.



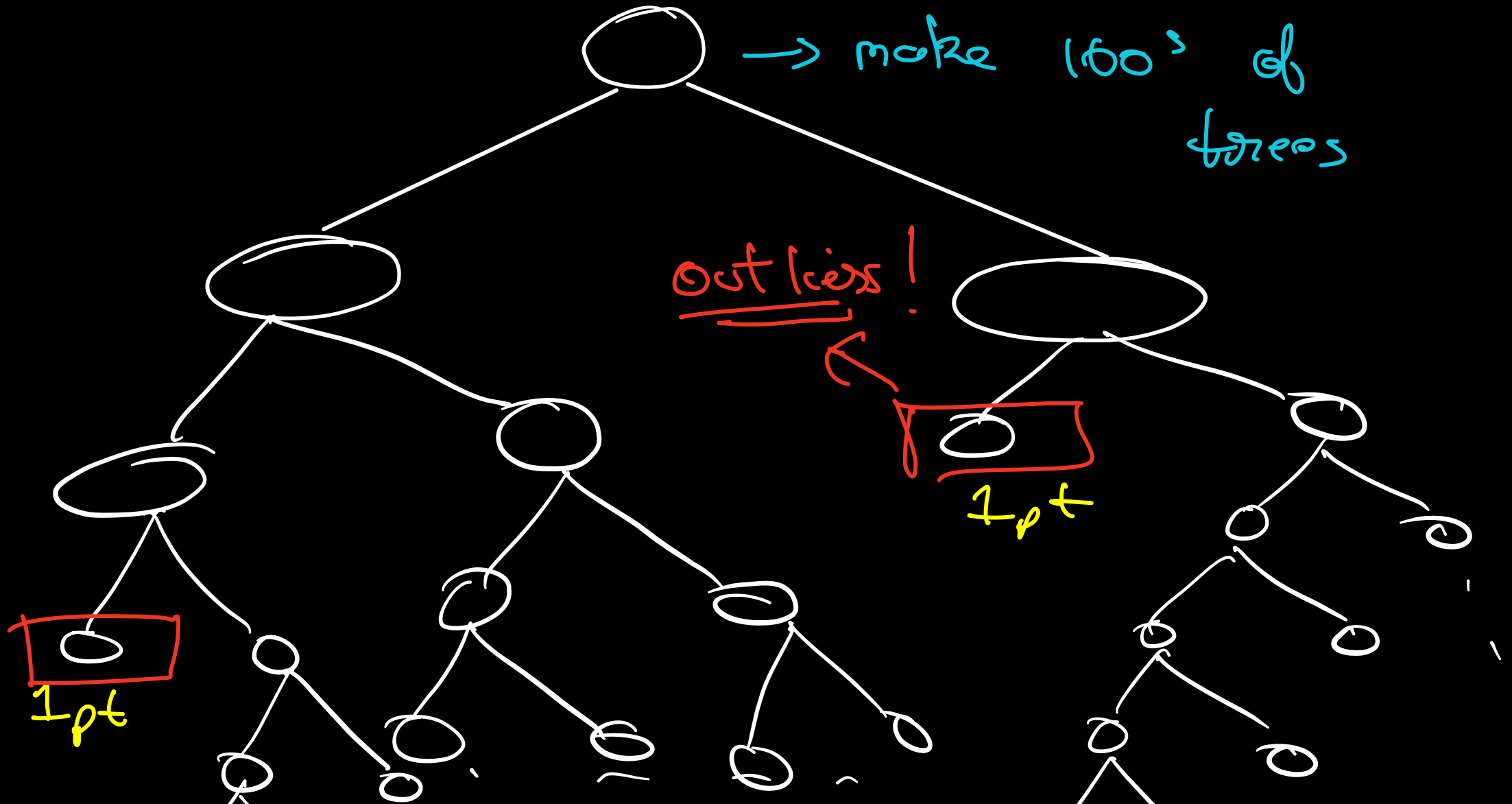
Isolation Forests

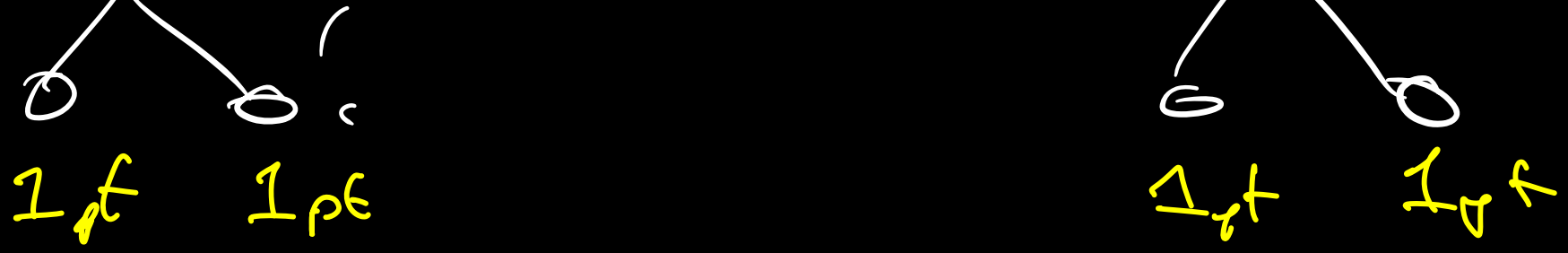
Idea !

↳ Build a lot of trees (random) until each point is a seperate node.



⇒ On avg outliers will take less # splits to be seperated.



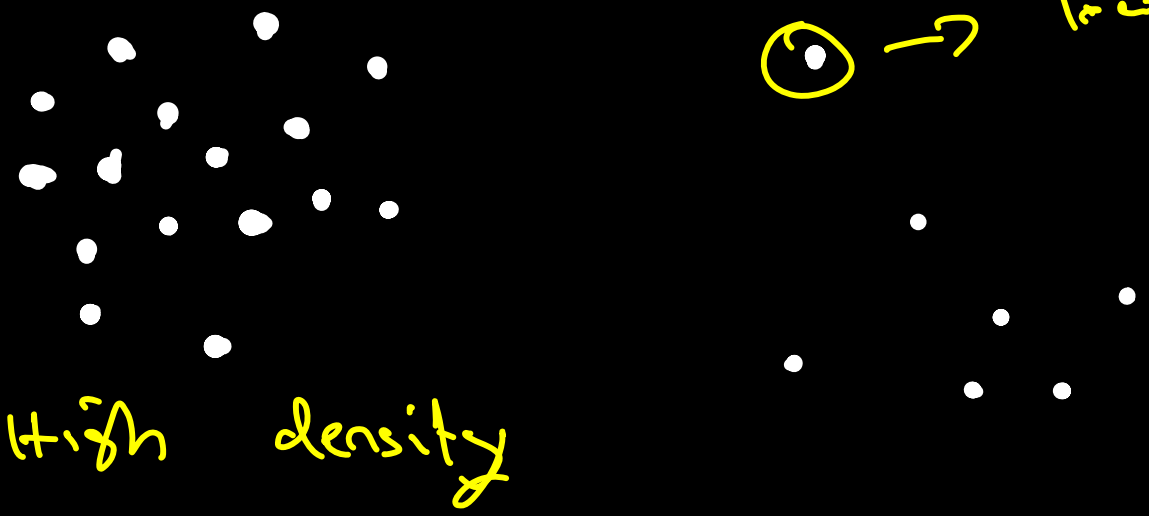


On an avg → the depth of outlier nodes / pts will be lower.

Local Outlier Factor

KNN + DBSCAN

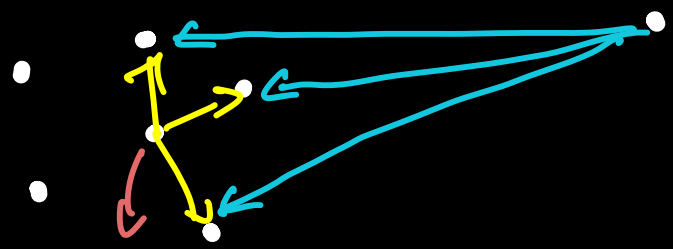
Idea: Outliers exist in low density regions



Density:

Density can be represented as avg distance between points (inverse)

K=3



→ high avg dist
= low density

low avg dist
= high density

Steps:

↳ Find K nearest neighbours of point 'p'

→ calc avg distance of K-nn.

$$\text{density}(p) = \frac{1}{\text{avg k-dist}}$$

Variation:

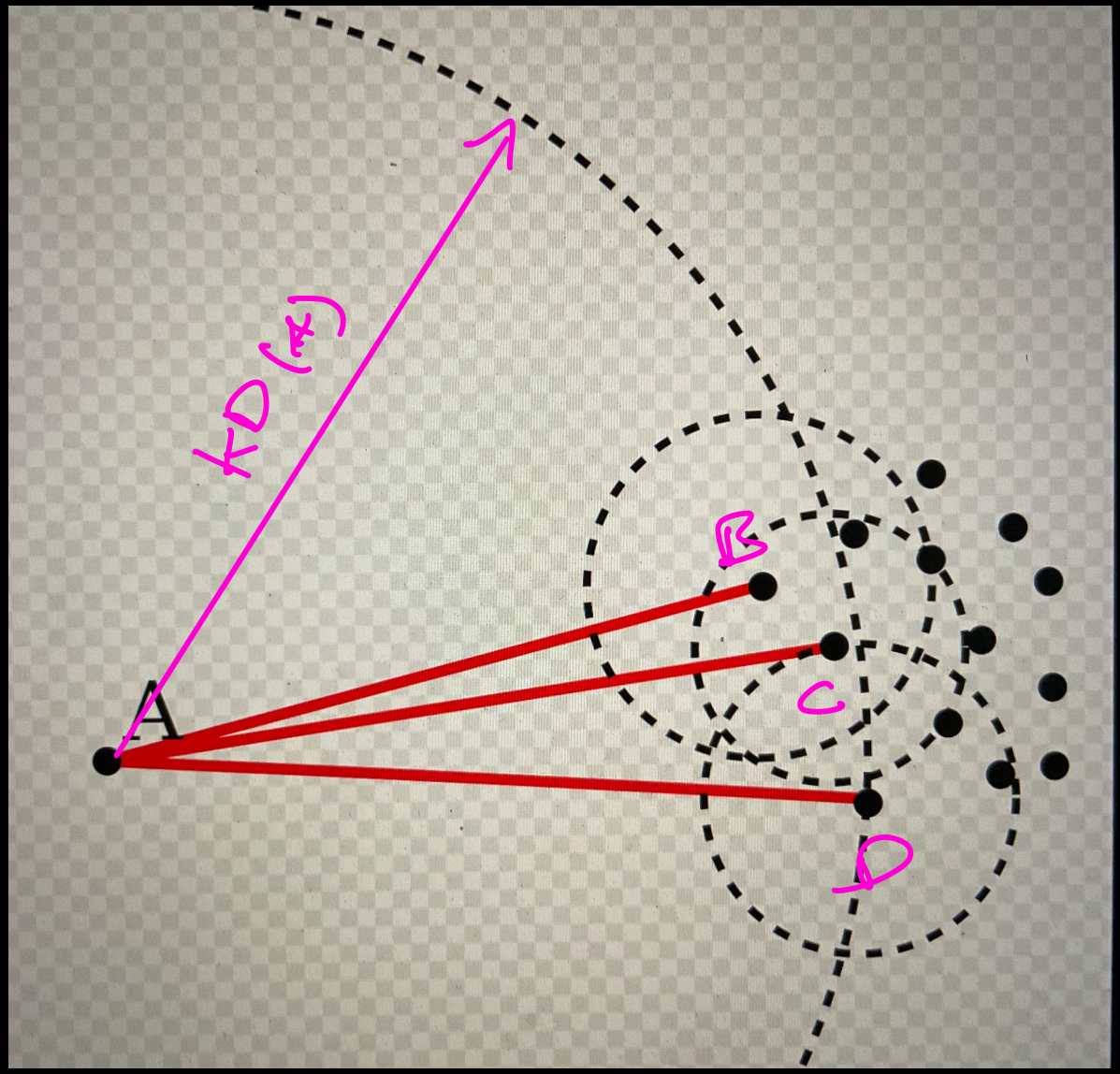
↳ it is recommended, by researchers that max K-dist works better than avg. (more stable boundaries)

max(K neighbours) = distance from K^{th} neighbour

K-dist

Hence,

$$\text{density}(p) = \frac{1}{KD(p)}$$



A has large radius, i.e low density

Now we can compute

Local outlier factor =

$$\frac{\text{Avg density of } k \text{ neighbours}}{\text{Density of } P}$$

Q: Outlier when ?

a) LOF >> 1

b) LOF << 1

c) LOF ~ 0

d) LOF < 0

This algo that we discussed is called "simplified LOF" because it uses

→ K-distance

However, there is a middle ground between avg and max that avoids some edge cases:

Reachability Distance:

$$RD(A, B) = \max(KD_B, dist(A, B))$$

$$\text{Density}(A) = \frac{1}{\frac{1}{K}\sum \text{RD}(A, \text{Neighbours})}$$

This means that we will use distance between A, B but it has to be min equal to K-dist(B)

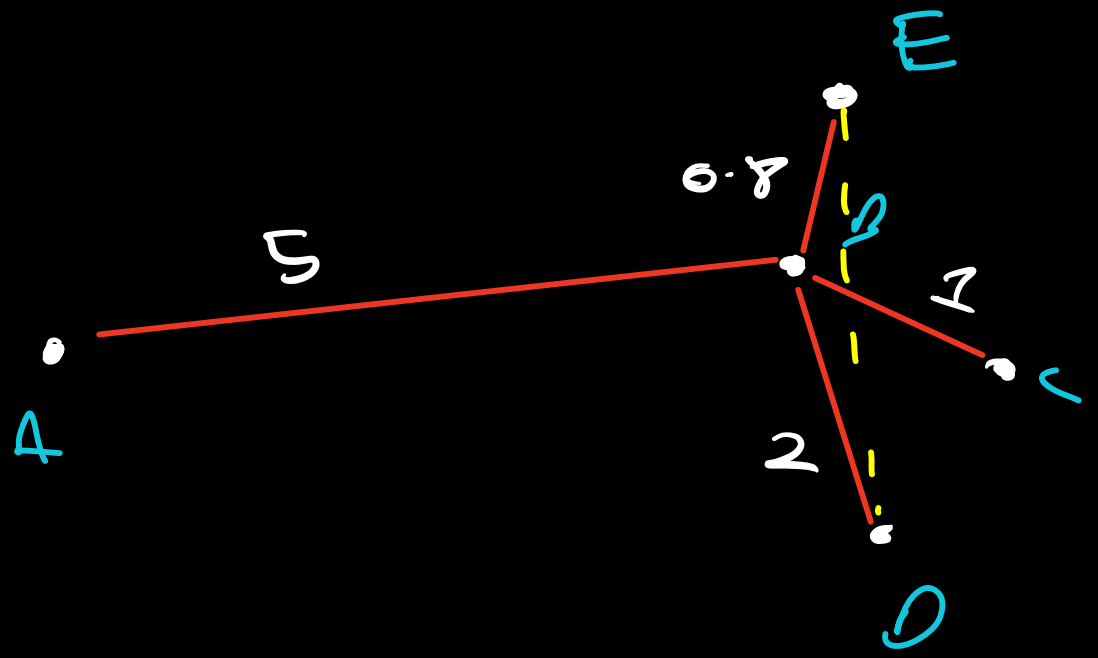
in other words!

RD(me, you):

if you are in my top 'K' but I am not in your top 'K' then let's use actual distance but if

we both are in each other's top 'K' then let's use your K-dist

Note: RD is not Symmetric.



$RD(A, B) = 5$

$RD(C, B) = 2$

$\neq$ not 1

$RD(C, E) = 2$

$RD(C, D) = 2$

$RD(B, E) = \underline{2.5}$

This is less intuitive but has some benefits when 1st neighbour and k^{th} neighbour distance may be very different.

Comparision