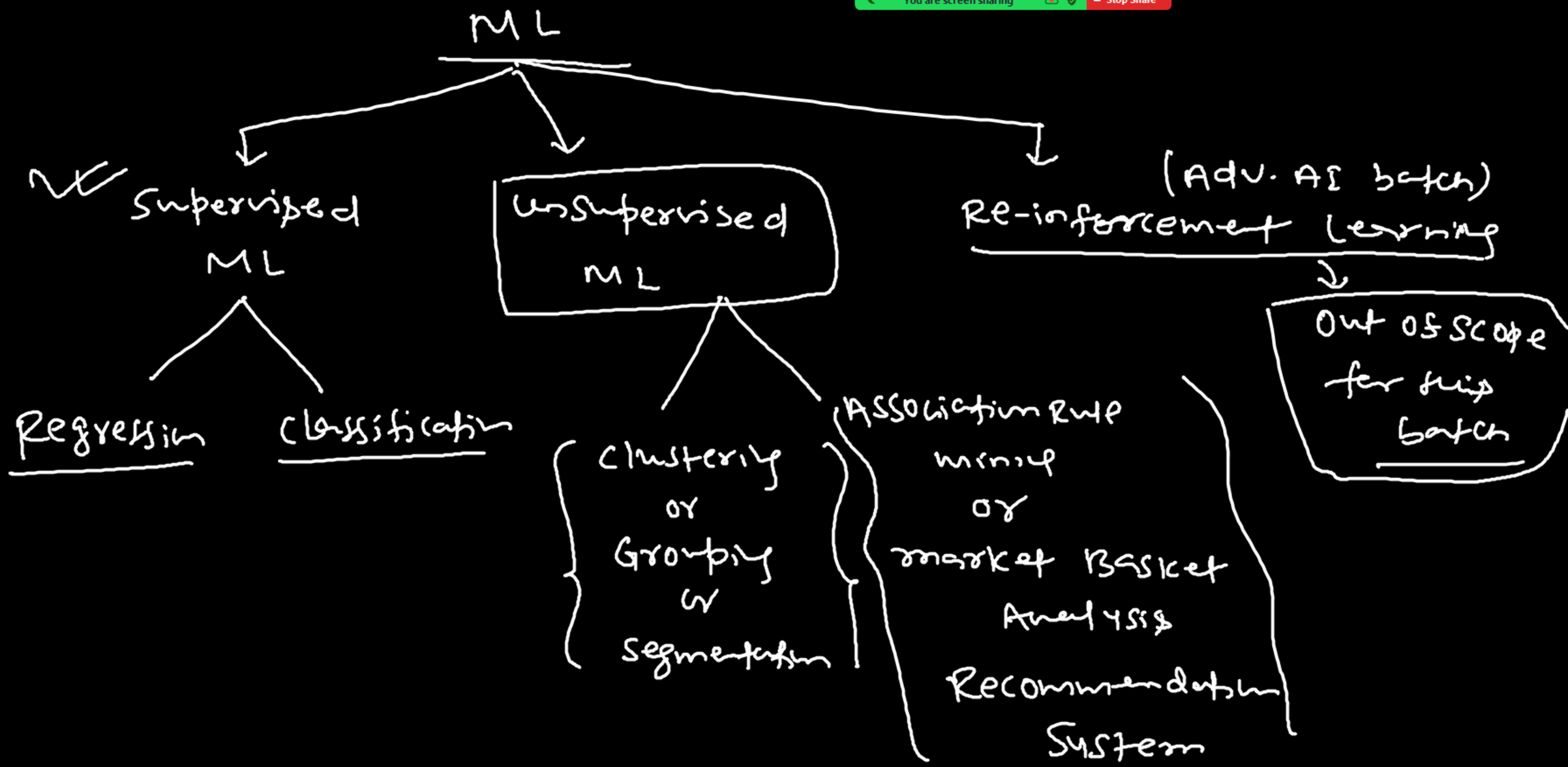


Q: What is Clustering ?

Regression & Classification

$D = \{x_i, y_i\}$, $y = f(x_i)$ (Dep. variable)

$y \in 0,1$ or $0,1,2,3\ldots$

$y \in \mathbb{R}^q$

x_i = Indv variable

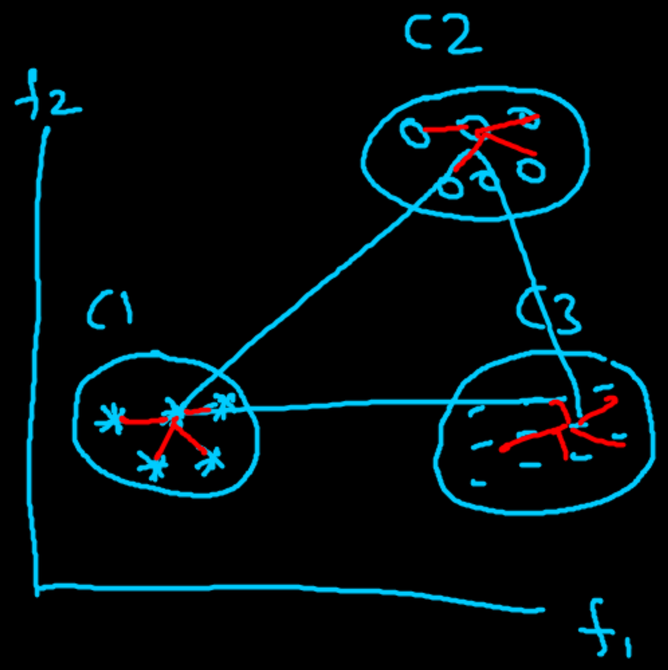
"Clustering"

$D = \{x_i\}$, No y_i | No dep. variable / Output / Result variable

x_i: Given

y_i: Not given

Task :- Group / Cluster "Similar" data point



Task of clustering

-> Group | cluster | segment "Similar" Point

-> points are close together by the help of Euclidean dist

Point 1 :- points in a cluster are closer together.

Point 2 :- points in different cluster are far away.

Supervised ML — Classification_report, Accuracy_Score, Adj_R-Sqr, MAE, MAPE, MSE, RMSE

how to measure

Unsupervised ML — K-means / Hierarchical, DBSCAN

↓ ↓

General purpose

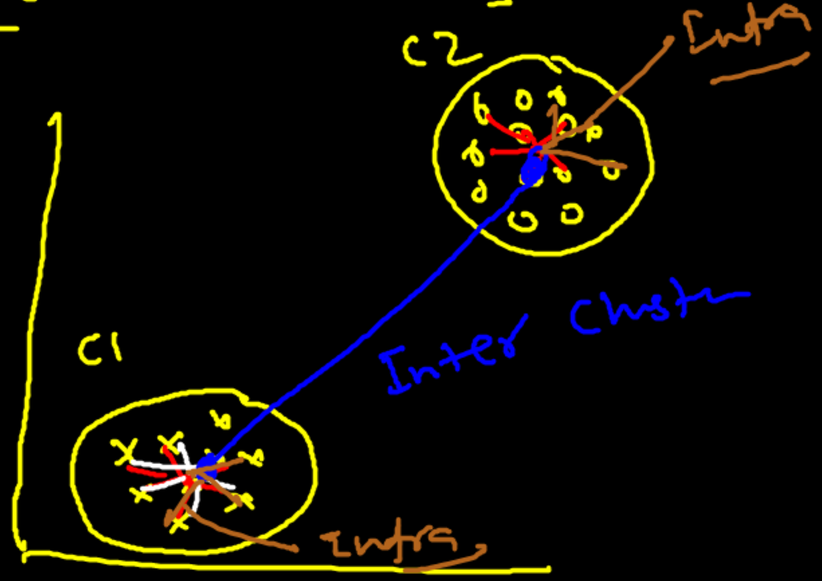
Clustering → Data mining topic

→ DEP/output/label/result

$D = \{x_i\}$: There is no y_i variable

↑ Given

Measure — Learning



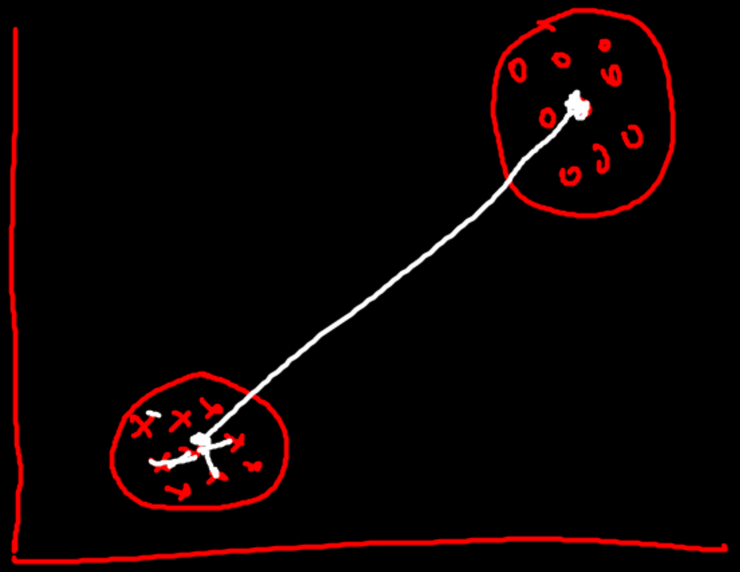
① Intra-cluster :- within a cluster → dist - very low

↳ smaller

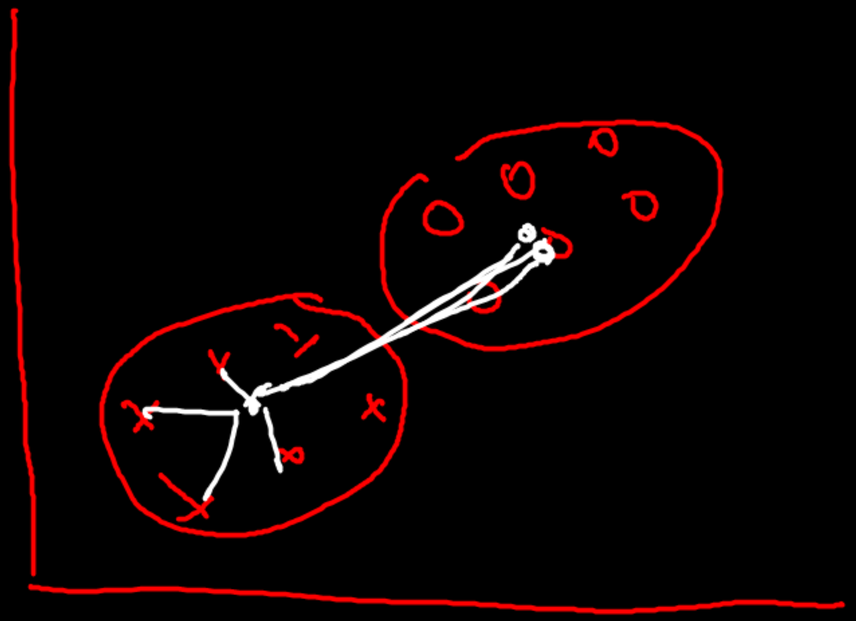
② Inter-cluster :- across or between cluster → very dist - high - Good

Core idea

- low intra clust
- high inter clust



A



B

Q:- which cluster is better?

Ans :— Inter:- dist = very high = A

intra = dist = " small = A

K-Means Cluster → popular, stable, elegant approach

	Height	Weight	
1.	168	60	K1
2	185	72	K2
3	170	56	K1
4	190	78	K2
5	185	77	? K2
6	183	89	?
7	165	66	? K2

K1 (168, 60)

K2 (185, 72)

K = 2 → Elbow method – how many "K" is required

↳ Randomly/Selected

New cluster (K1)

(168, 60) (170, 56)

$\frac{168+170}{2}$, $\frac{60+56}{2}$

What is (170, 56) = K1 or K2 ?

$ED1 = \sqrt{2^2 + 4^2}$

$ED2 = \sqrt{15^2 + 16^2}$ =

K1 = (169, 58)

(169, 58)

K2

190, 78 : K1 or K2 = ?

K1 = 169, 58

K2 = 185, 72 K

$FD1 = \sqrt{21^2 + 20^2}$

$ED2 = \sqrt{5^2 + 6^2}$ K

ED2 < ED1

$\frac{190+185}{2}, \frac{78+72}{2}$ ∴ K2 = (187.5, 75)

18.5, 77 = ? < K1 K2

$K1 = \sqrt{16^2 + 19^2}$

$K2 = \sqrt{(2.5)^2 + 2^2}$ K

$\left(\frac{187.5+185}{2}, \frac{77+73}{2}\right)$

K2

K-Means :- very very hard to solve mathematically

in computer Science → "complexity Theory"

exponentially Time complexity

"Approximation algorithm" - hard prob

hacks & maths

Lloyd's algorithm → V. V. Simple

K-means : Lloyd's Algorithm

D : { x

C1 C2

Rule ① :- random initialize

→ randomly pick K points from D

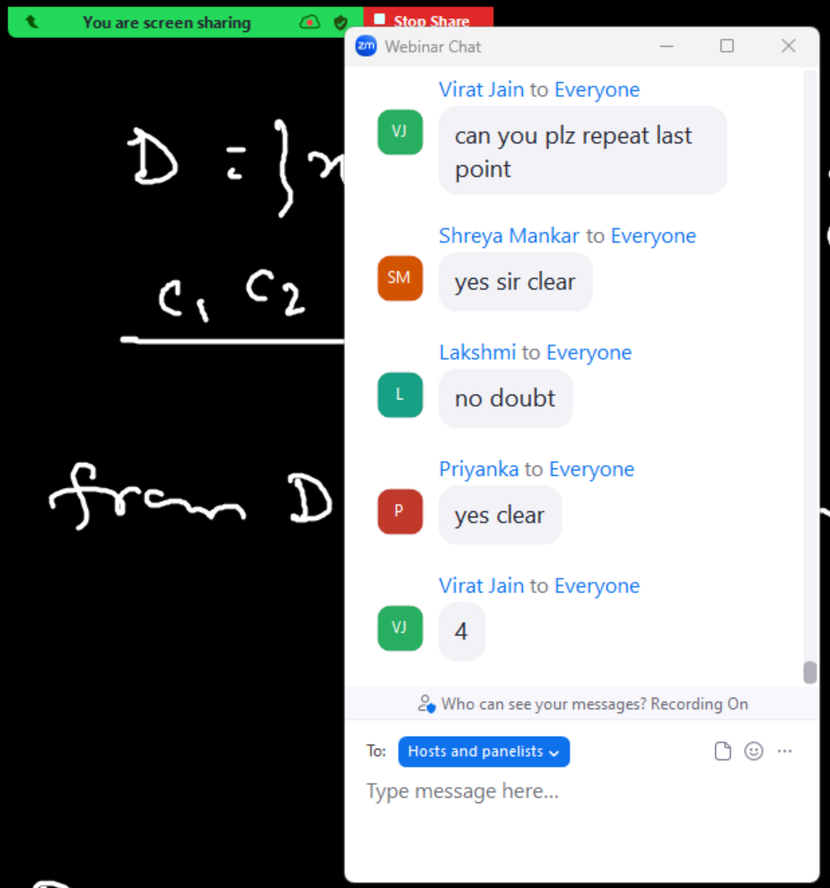
C1 C2 C3 - —

Rule 2 :- Assignment

iterate-loop {

for each points x_i in D

→ select the nearest one Eucliy-dist



Rule 3 :- Recompute centroid of update the

Rule 4 :- Repeat step 2 & 3 until

Computer Science — Complexity Theory → Approximation algorithm

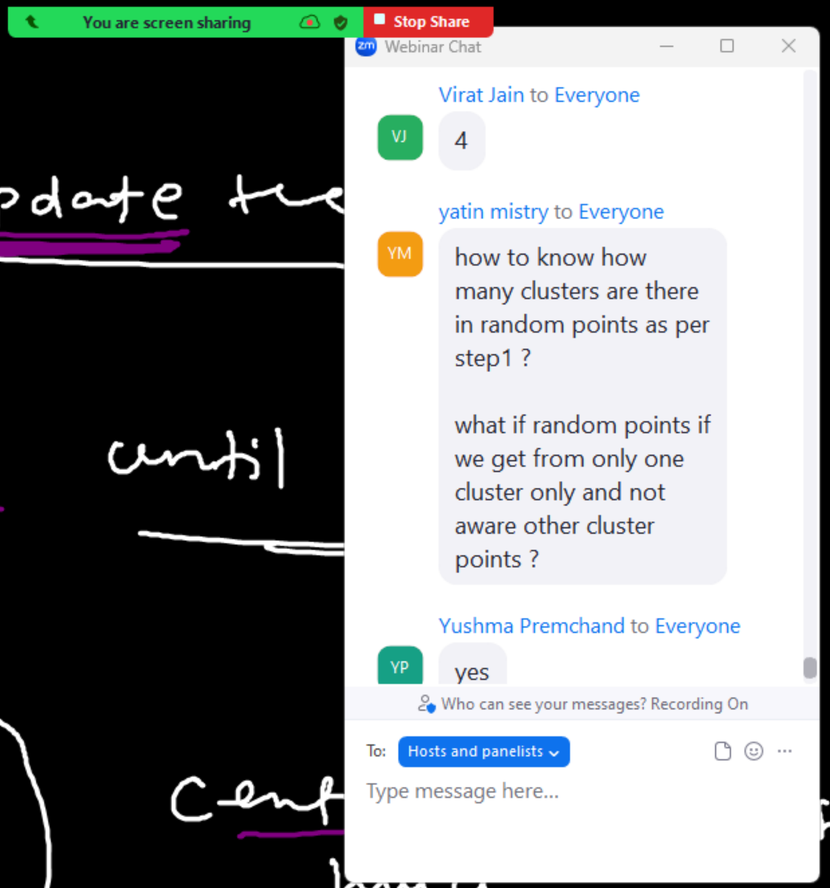
very hard → very easy | simple

Lloyd's method

Cent... change

↓

dist between new centroid & old centroid is very small.



Silhouette Score