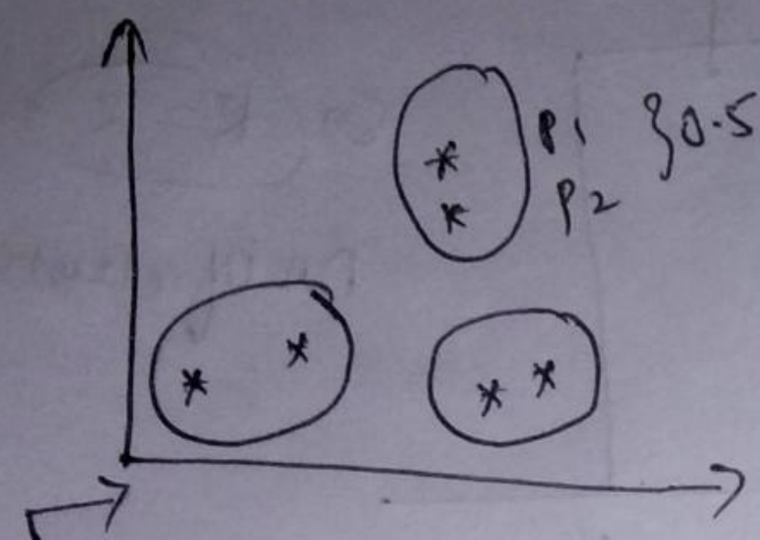
Hierarchial Clustering

⇒ unsupervised

⇒ like K-means but technicaly different



In hierarchial cluster firstly considery all points an different cluster.

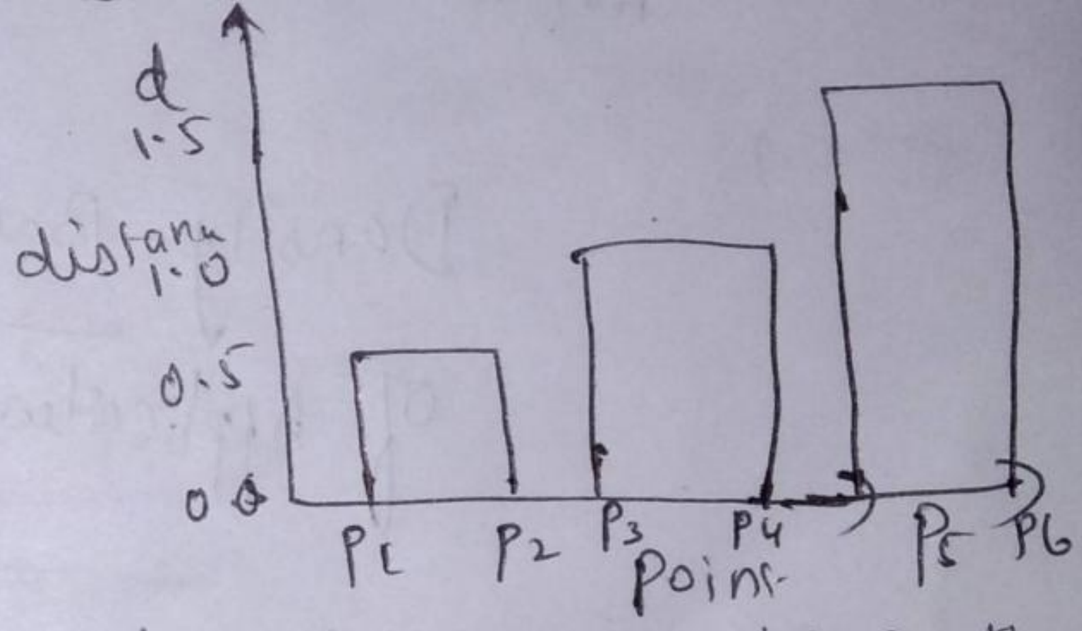
Now: Two nearest point are considered as same cluster.

this points are specify dendogram

$P_1 \to P_2$ (dist : 0.5)

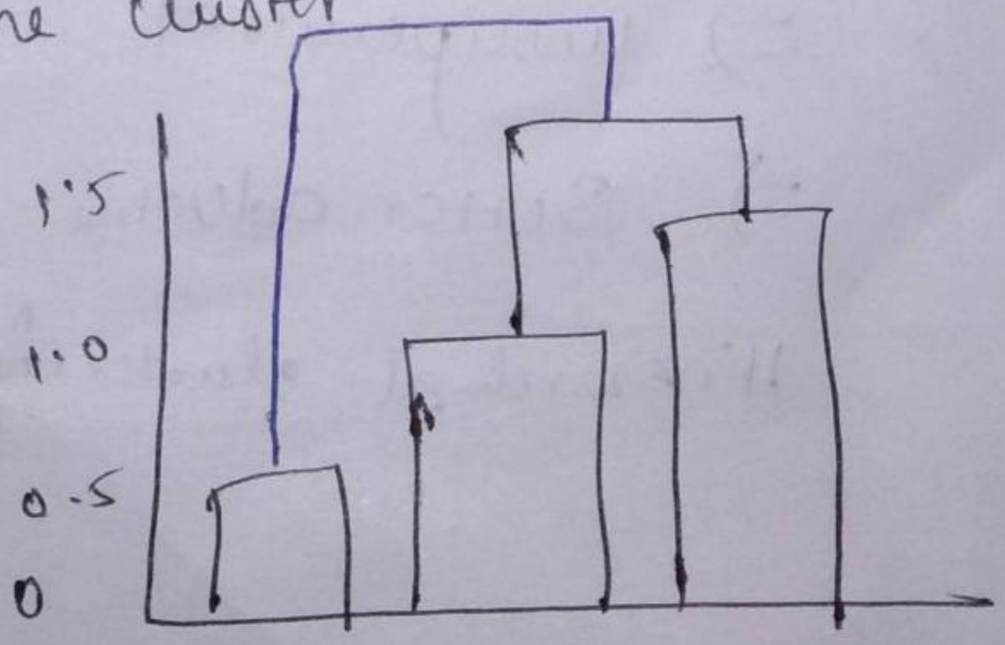
$P_3 \to P_4$ (1.0)

$P_5 \to P_6$ (1.5)



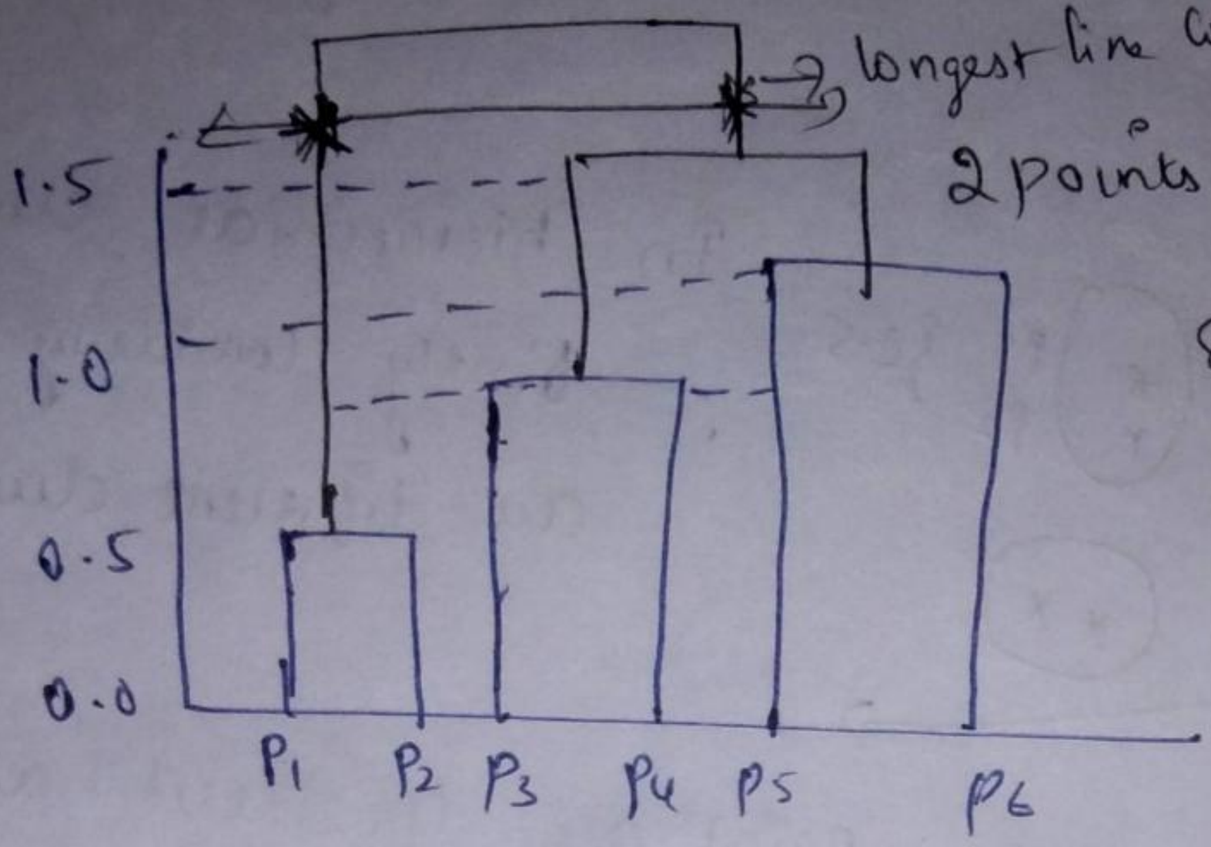
After that which one is next neighbour nearest in ~~order~~ that new cluster (P3, P4, P5, P6) is nearest

to that is same cluster



Then $P_1 P_2 P_3 P_4 P_5 P_6 \to$ Same cluster

Okay now how to select K-values (no of cluster) that we want



So K=2

No of cluster : (2)

we need to find longest vertical line of the horizontal

Density Based Spatial Clustering of Application with Noise

DBSCAN:

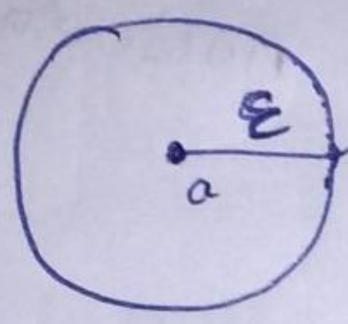
⇒ unsupervised

⇒ Better advance of K-means, Hierarchical clustering.

4 important Concepts:

- Epsilon
- Core point
- Border points
- Noise

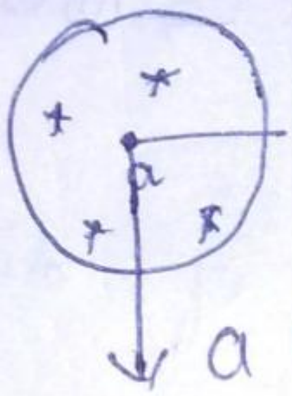
} create cluster.



radius

create decision boundary (Epsilon)

Now assume minpts = 4.



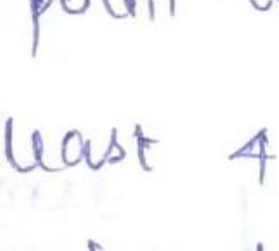
midpoints present in boundary

a (core point)

when core point achieve

at least 4 (or) more points are in decision boundary then decision boundary create.

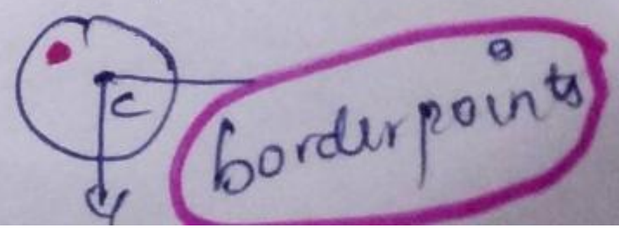
I have point c (mp = 4)



create radius

① create boundaries help of Epsilon

② suppose doesn't satisfy midpoints = 4 but we have atleast one core points (boundary) in boundaries



One point ∅ I creat boundary help of Epsilon:

None of these satisfy

minpoints &
any core points
doesn't meet mid points → noise (outliers)

DBScan → works very well noise data

Prons:

work very well → noise & outliers

drawback:

dimensionality of difficult to grouping data together.

too many dimension dbscan suffer.

doesn't work well when dealing with cluster of varying densities.

Hyperparameter:

eps = epsilon (radius)

min_samples = mid points

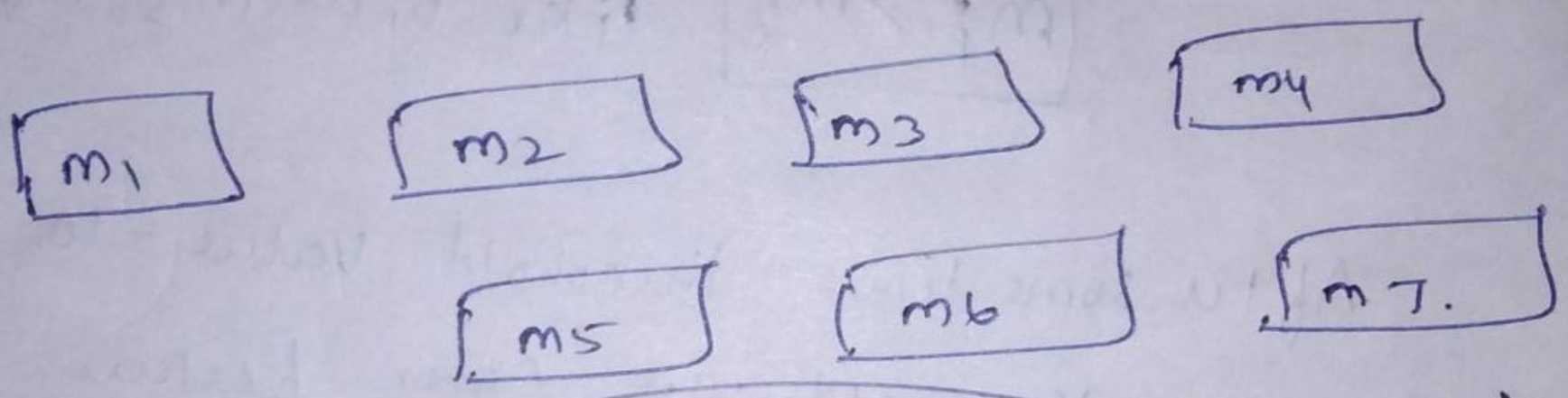
metric = 'Euclidean', 'Manhattan'.

Principal Component Analysis:

Curse of dimensionality:

↳ dimension (features) (i/p, dep)
↳ attributes

I am going to take dataset that dataset
$m_1 \rightarrow$ 2 feature, $m_2 \rightarrow$ 5 features, $m_3 \rightarrow$ 100 features
$m_4 \rightarrow$ 500f, $m_5 \rightarrow$ 200f, $m_6 \rightarrow$ 1000, m_7 - 2000



Now I taken (House sales price) prediction

bisty I take m_1, dataset In m_1 dataset

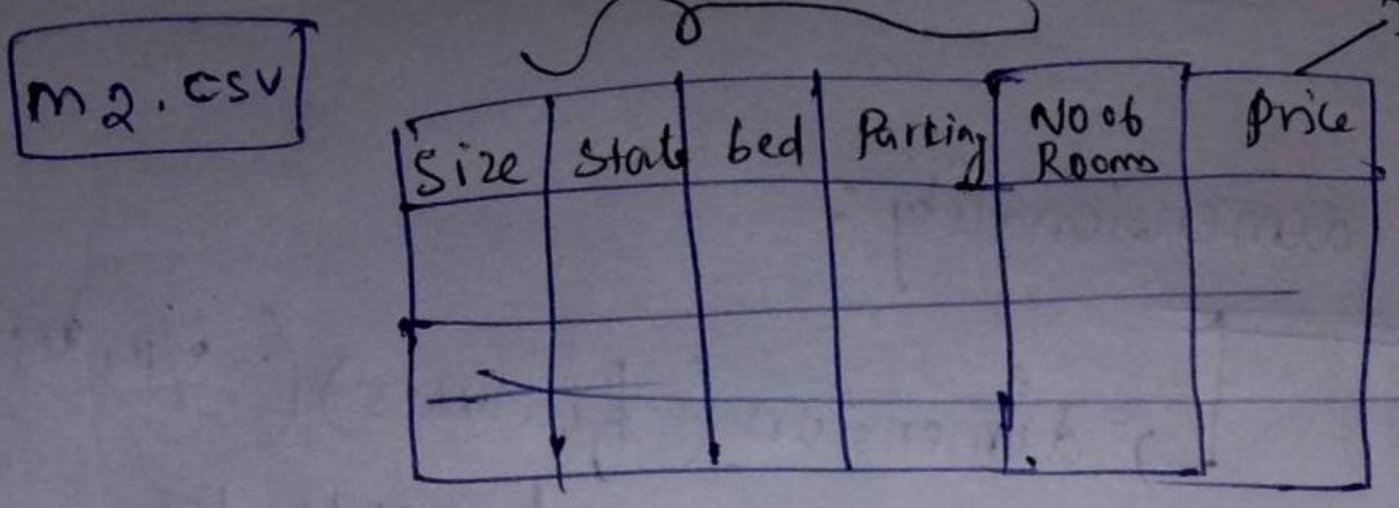
have (2 features) one is bedroom, another is size. } i/p

price → dependent.

Now m_1 accuray acc1 = some value
(80% - 85%)

Then take m2 dataset In m2 dataset

have 5 features



Now predict Rate use independent (5 features)
So get learn lot of details through 5 features
Compare than 2 features.

So m2 model accuracy is high

$m_1 > m_2$ like we in m_3, m_4.

After some time threshold value = 10.

Assume threshold = 10 ~~After~~ feature Reach the threshold what happens.

m_4, m_5 → more than 10 features.

more than 10 features your feeding in lot of information in particular so that model get confused.

machine has confuse model if it is increasing exponentially

After threshold accuracy ↓.

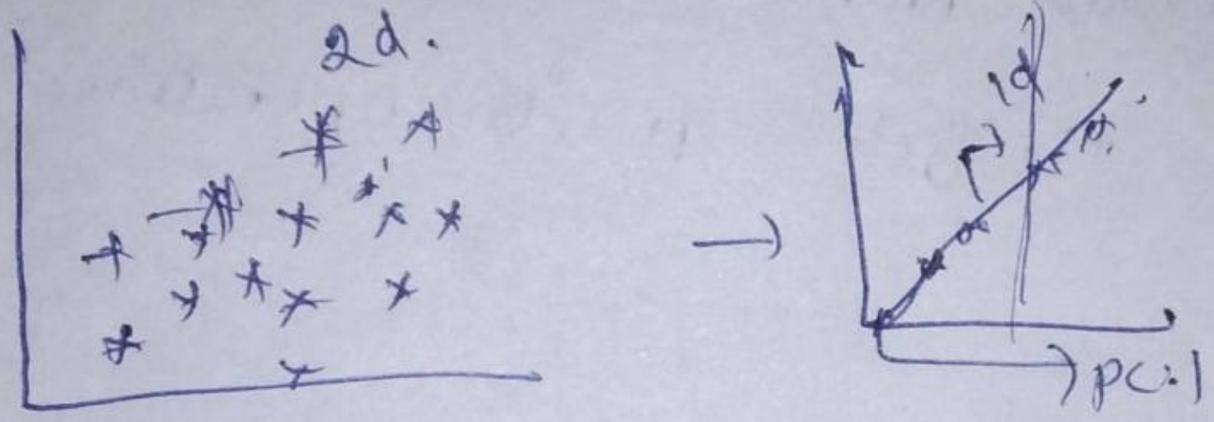
Principal Component Analysis:

Reduce number of dimension

As the number of dimension increase it is a curse.

Because accuracy impact with no of dimension increase.

2d → 1d



① best fit line calculate

② ~~Square~~ error calculate

③ Perpendicular (orthogonal) line.