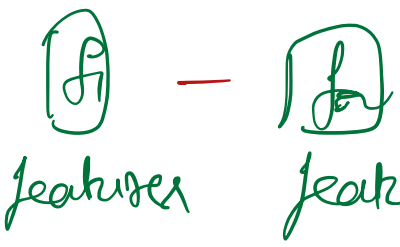
Unsupervised:

1. K-Means clustering
2. Hierarchial clustering
3. Silhoutte score
4. DBScan clustering

⇒ we dont have any specific o/p in unsupervised ML



features features → based on these features we divide them into clusters similar kind of data

→ Can be used in custom Ensemble modelling.

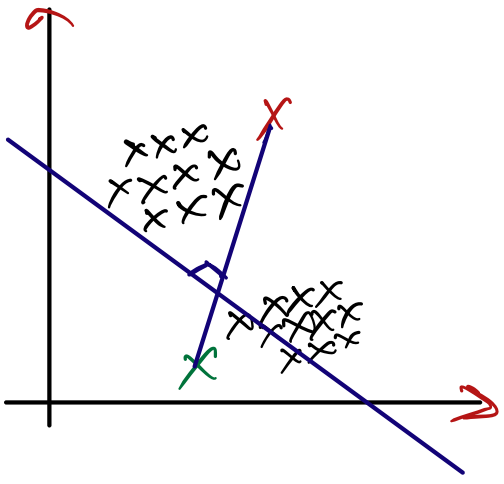
supervised algo.

Centroids

Steps:

1. we try diff 'k' values → select suitable
2. we initialize 'k' number of centroids.
3. Compute the (avg) to update the centroid

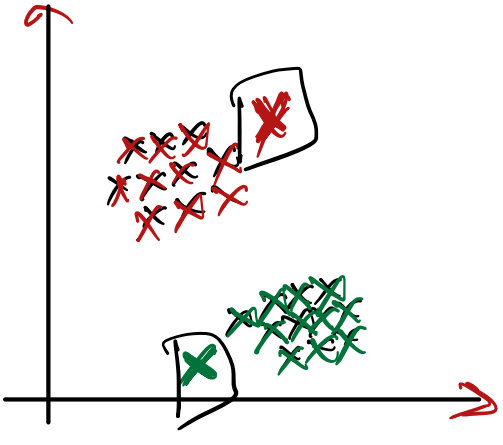
Sit back, and enjoy the show :D



(1) first we consider plotting the whole data which we have.

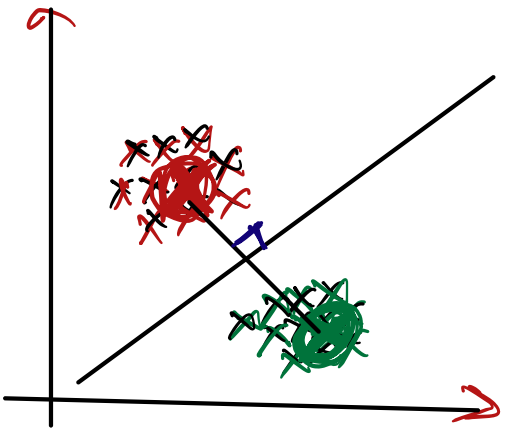
(2) then we take '2' random points and try to find a perpendicular line.

(3) After that we use Euclidean dist to find the centroid.



⇒ So, from the above results, we consider the points near to red as red category and green as green category.

⇒ after this, we find avg of every category and then we will get a new centroid which is inside that particular category.



→ now, we can see that, all the points are categorised properly

→ we call them as clusters, and those centroids are represented by the 'K' values

note:
→ if there are '2' clusters then value of K=2

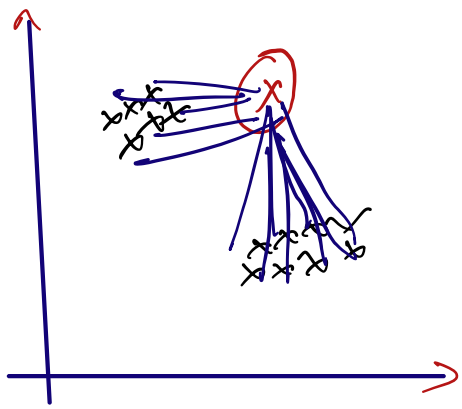
how do we decide the k-value?
⇒ we use Elbow method → optimize 'K' value

→ we take iterations, and for every iteration we will construct a graph with respect to k-value and WCSS

WCSS →

within cluster sum of square- ---

⇒ before plotting the graph, let's understand diff graphs w.r.t to 'K' value.

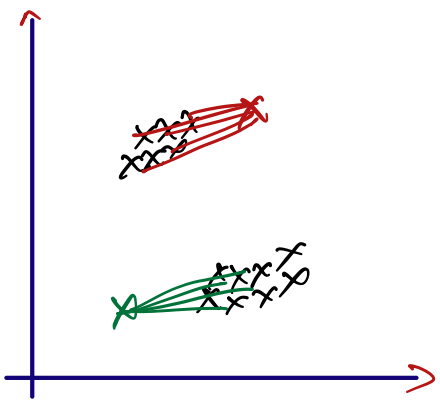


K=1

⇒ we took one centroid, and we try to find dist from all the points.

So, after considering distances we sum up and find square.

⇒ So, this is going to give us a very high WCSS

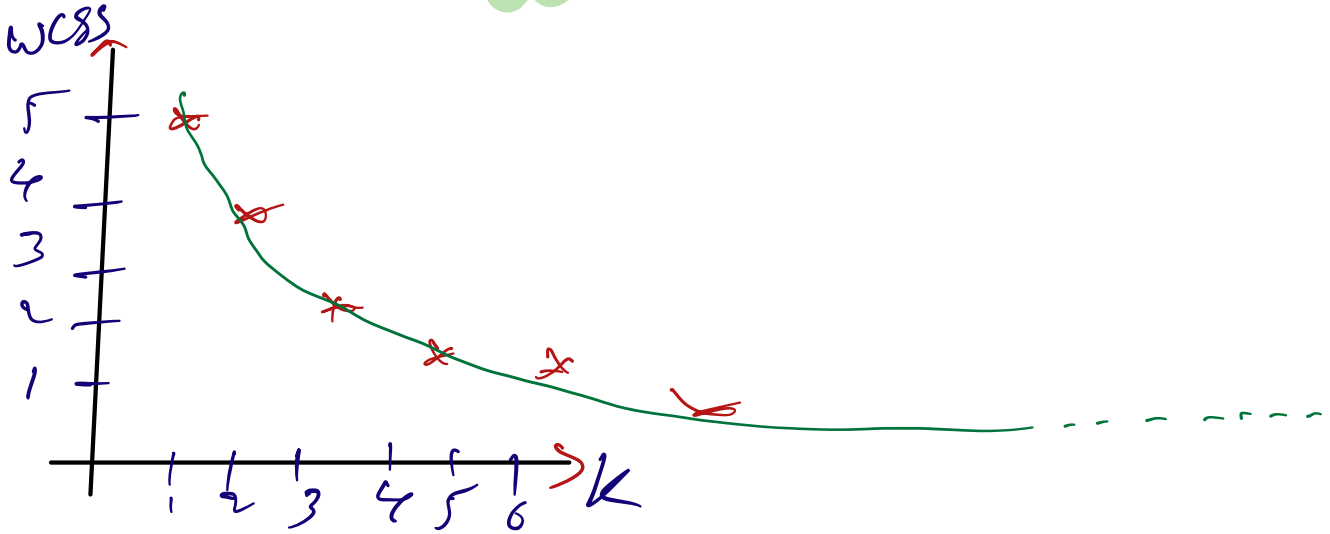


K=2

→ in this case, we have 2 centroids and their individual distances have been WCSS value compared to K=1

⇒ So, the conclusion is, as we take more 'K' value WCSS value will decrease

⇒ and we plot the graph for 'K' and WCSS and we call it Elbow method



So, for what value of 'K' WCSS is giving an elbow style of graph (the drop in the graph)

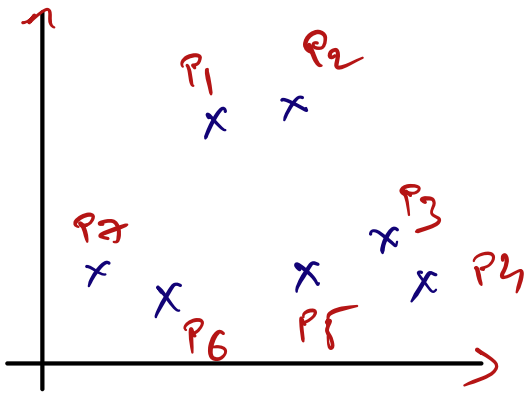
we prefer that particular value of K for dividing values into clusters.

note: we can also say, where there is abrupt change, we consider that 'K' value....

$$WCSS = \sum_{i=1}^{n} (C_i + X_i)^2$$

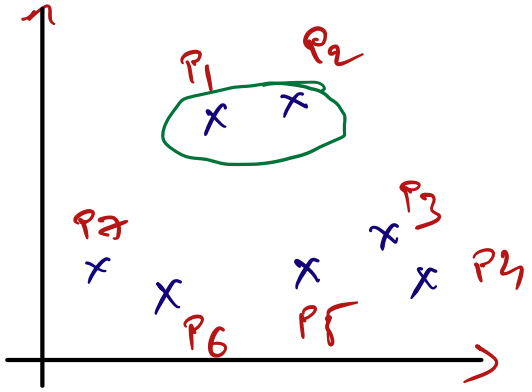
Hierarchial clustering:

→ let, we have various points plotted as shown below.

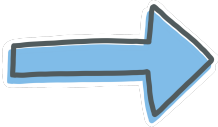


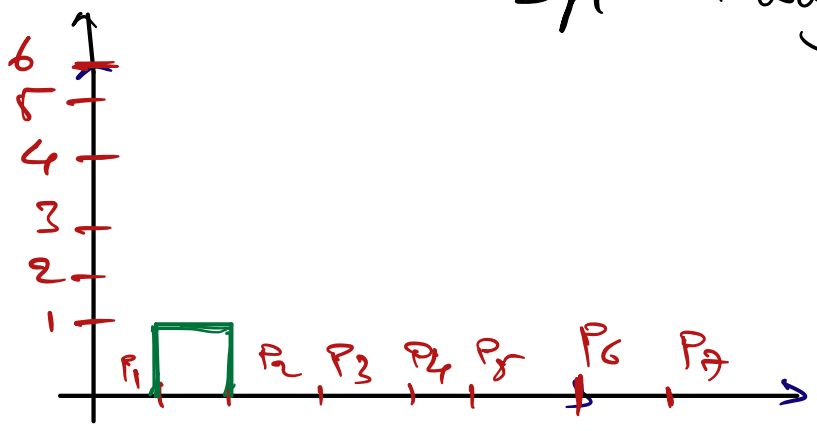
→ now, according to the distance b/w points, we group points / values

→ firstly, lets consider P_1 and P_2, as the distance b/w them is less, we group both of them as one entity

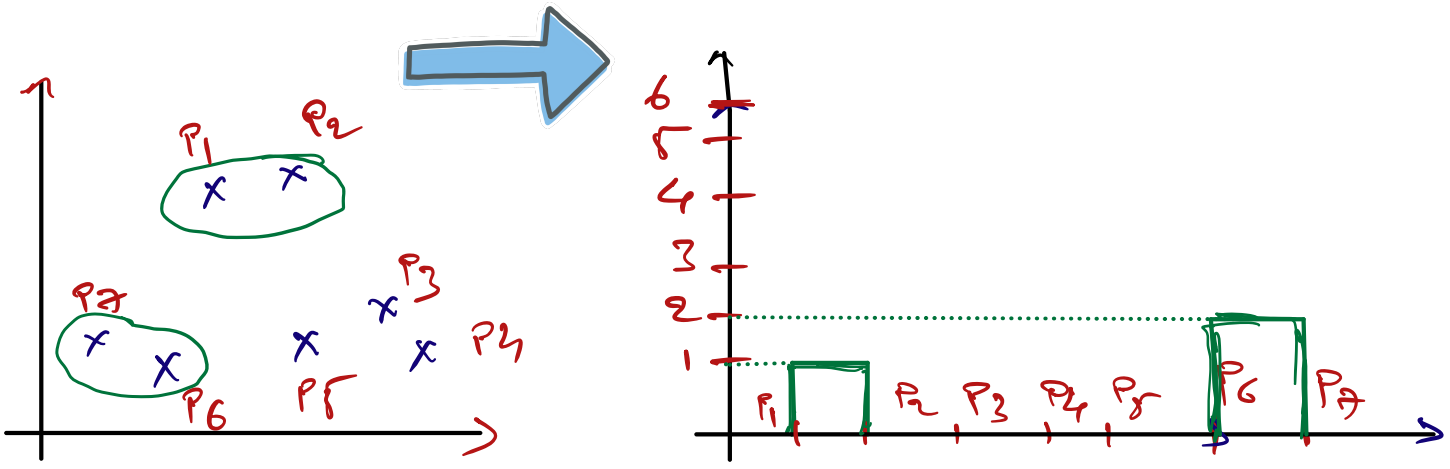


→ now, depending on the grouping, in the backend we create another graph in the below shown way



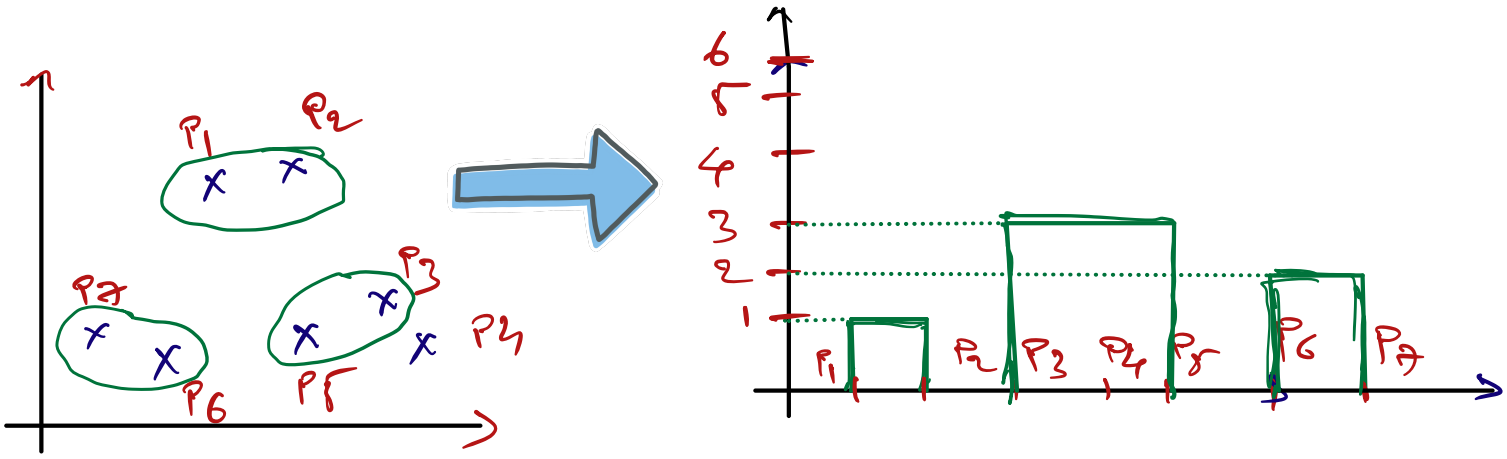


→ in the same way, we will combine P_6 and P_7 as well.

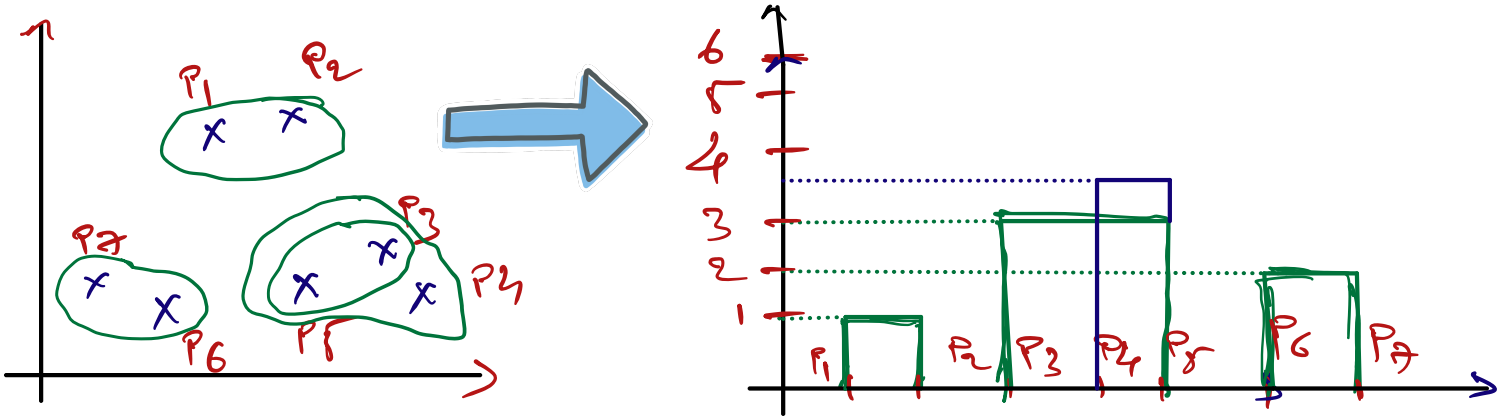


→ Lets assume that, the distance b/w P_6 and P_7 is greater than P_1 and P_2, for eg:- 9

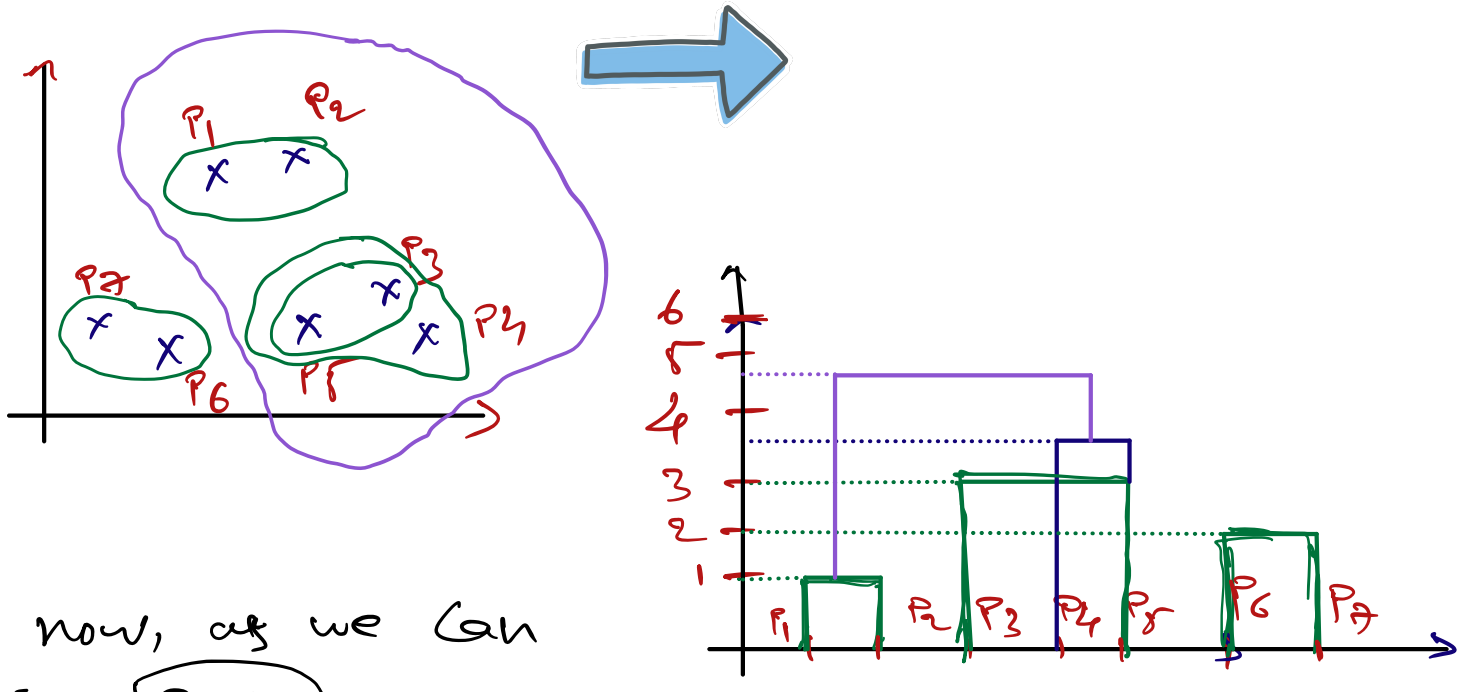
→ next, we will group P_3 and P_5 and also lets assume its dist is even more greater than the previous two



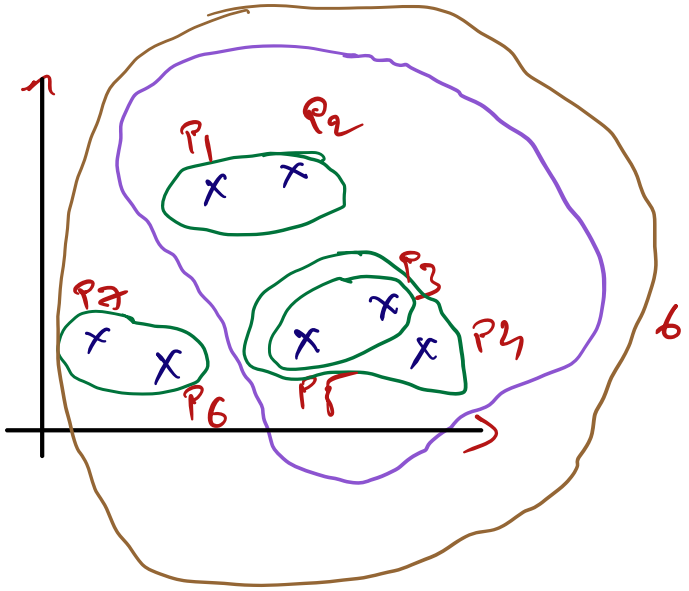
→ now, we will combine P_4 with P_3 and P_5

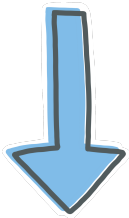


→ as we can see, P_3, P_4, P_5 are nearer to $P_1\ P_2$ group, we combine them.

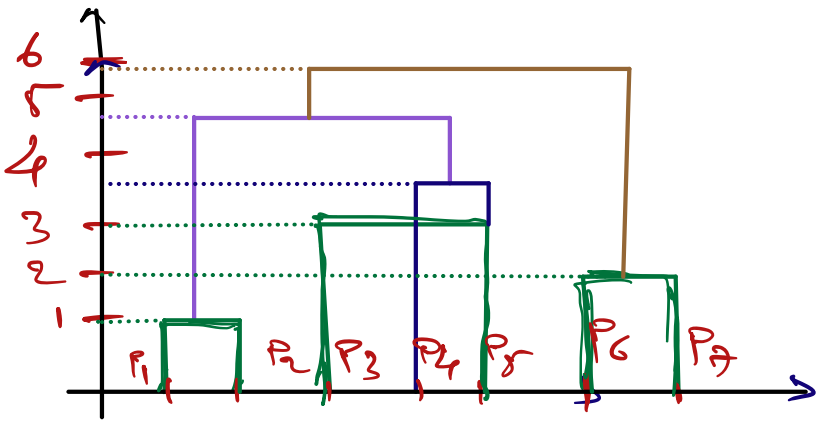


→ now, as we can see $P_6\ P_7$ as near to the purple grp, we join them.





This is !! ⟸ called as



Dendogram

How do we find out, how many groups should we consider for the given number of points???

Ans! You need to find the longest vertical line that has no horizontal line passed through it in a Dendogram

DON'T FORGET

note: there should not be any horizontal line even if we extend any

TEST:

Maximum time is taken by which clustering?

Ans:

Hierarchical clustering.

Dataset is small ⇒ Dendogram
Dataset is large ⇒ K-means

note:

⇒ we should take centroids very far from the data points, cuz it might create confusion with no. of clusters.

⇒ we solve this error with the help of

K means ++

↳ it will make sure, centroids are far.

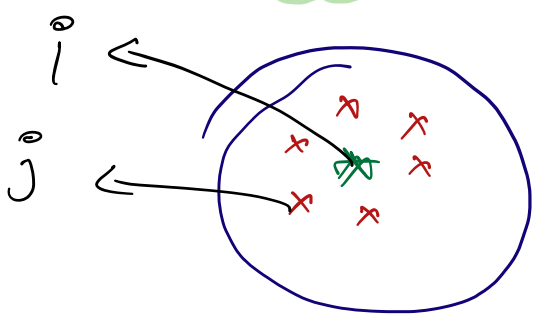
Validating clustering models:

→ we use **Silhouette** score.

Steps:

(1.) In a cluster, we compute the distance b/w centroids and the points. $a(i)$

⇒ summation
⇒ Avg

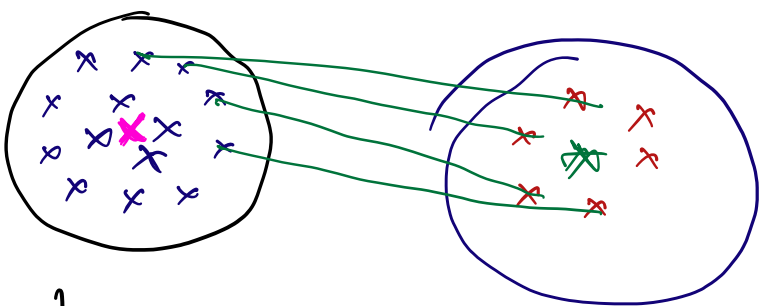


formula:-

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I,\ i \neq j} d(i,j)$$

(2.) we will calculate distance from each and every point from one cluster to the another cluster.

⇒ avg



TEST

If we have a good cluster, wether a(i) will be greater or b(i)

Ans: $b(i) \gg a(i)$

⇒ Value of silhouette clustering will be in range [-1 to +1]

formula:

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$

formula:- !!

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad \text{if } |C_I| > 1$$

and

$s(i) = 0$ if $|C_I| = 1$

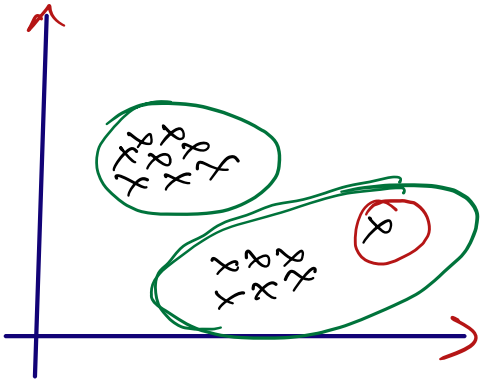
This page is for future notes :—

DBScan Clustering :-

Density Based Spatial Clustering of Applications with Noise

(0) Epsilon
(1) Min Points
(2) Core points
(3) Border points
(4) Noise points

why we use DBScan?

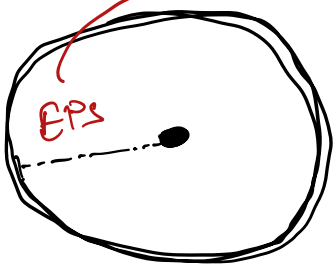


→ In K-Means, we consider this case as 2 clusters

→ But this extra point is called as an outlier (or) noisy point, as it is seperate from the similar data.

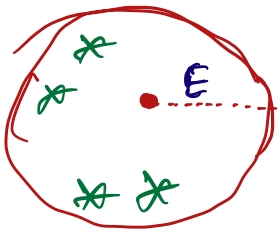
→ to overcome this error we use DBScan, which leaves out the outliers when clustering.

Epsilon: → radius of a circle



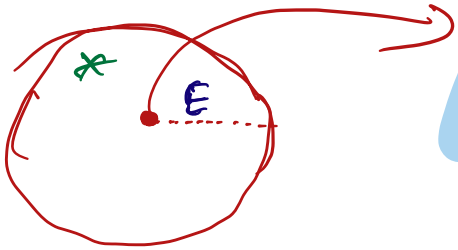
Min point: It is a **Hyperparameter**

if **min pts = 4**



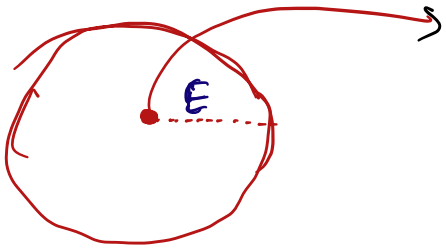
→ If in this Epsilon distance, we have the same min no. of min points, then the **red point** is called as **Core point**

⇒ If we dont have points equal to our min point, we call it



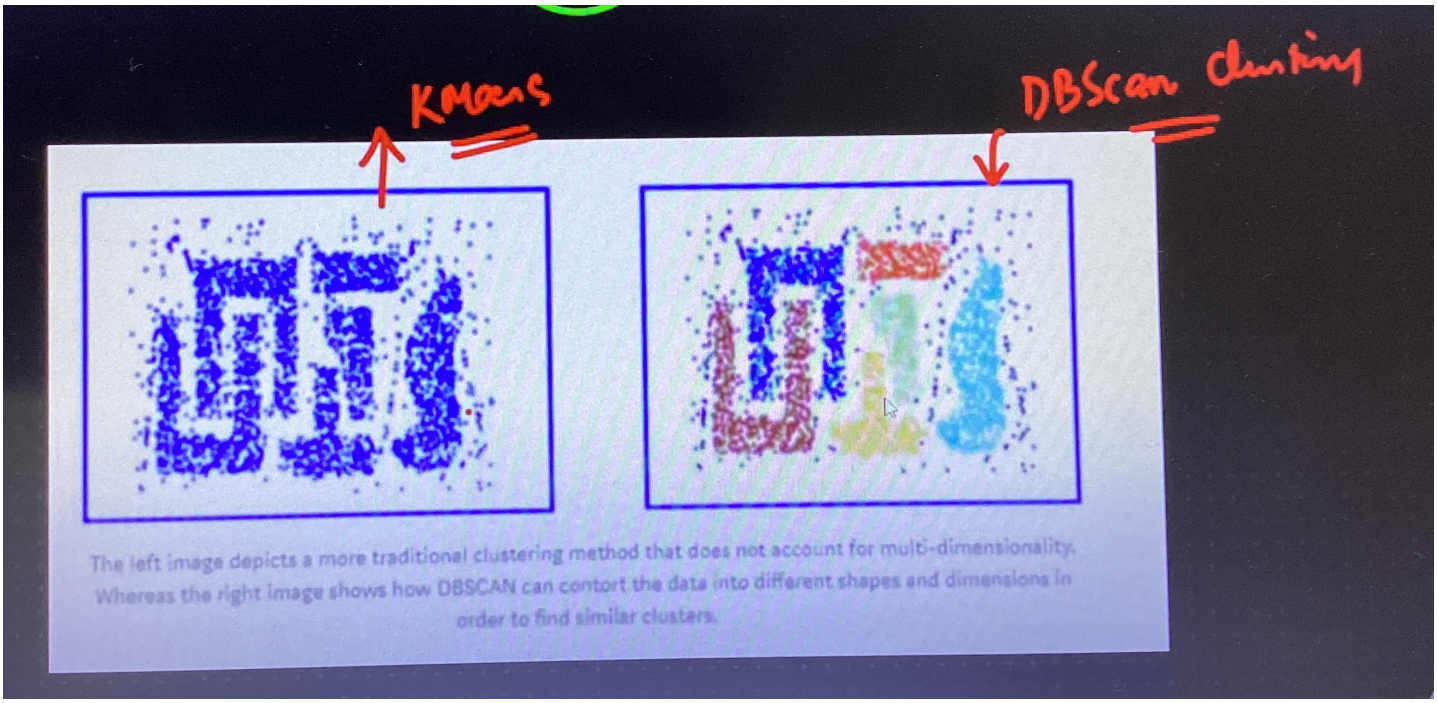
Border point

⇒ If we dont have any point inside our epsilon distance, we call



noise point

⇒ So, when we have a noise point it will get neglected.



Agglomerative Clustering :-

→ it is a part of hierarchial clustering and we create dendograms as well for this

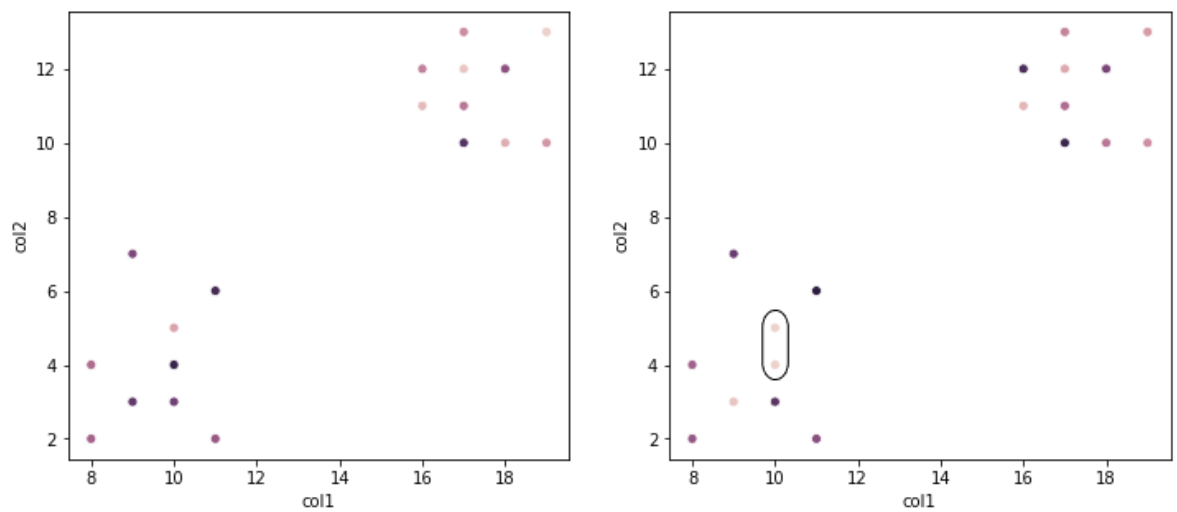
→ More points are considered as one cluster and far points are divided into diff clusters

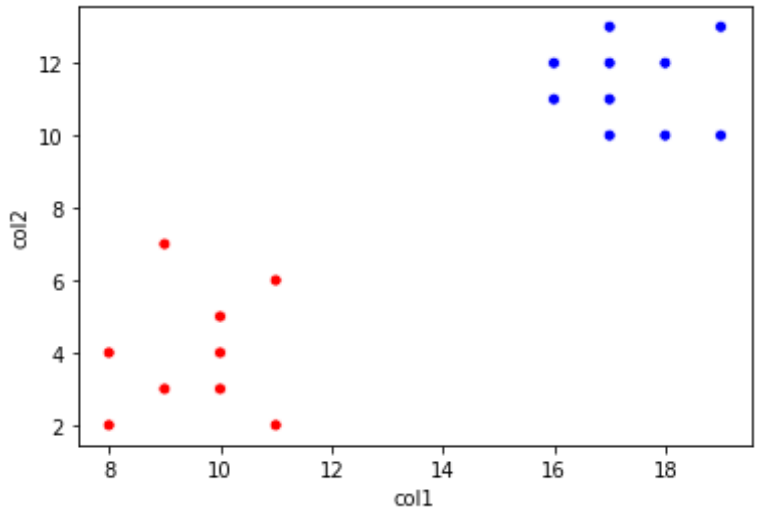
intuition:

→ It is a bottom-up approach, initially, each data point is a cluster of its own, further pair of clusters are merged as one moves up the hierarchy

Steps:

1. Initially, all the data-points are a cluster of its own
2. Take two nearest clusters and join them to form one single cluster.
3. proceed recursively step 2 until you obtain the desired number of clusters.





How?

→ To obtain the desired number of clusters, the no. of clusters need to be reduced from initially being (n) clusters (n equals the total number of data-points)

→ Two clusters are combined by computing the similarity b/w them.

There are some methods which are used to Calculate the similarity b/w two clusters.

1. Distance b/w two closest points in two clusters
2. Distance b/w two farthest points " " "
3. The avg distance b/w all points " " "
4. Distance b/w centroids of two clusters.

⇒ There are several pros and cons of choosing any of the above similarity metrics.

Inverse:

→ The inverse of agglomerative clustering is divisive clustering which is also known as DIANA (Divise Analysis) and it works in "top-down" manner.

→ It begins with the root, in which all objects are included in a single cluster.
→ At each step of iteration, the most heterogenous cluster is divided in two.

Association Rule Mining:

→ it is a method for identifying frequent patterns, correlations, associations (or) casual structures in data sets found in numerous databases such as relational databases, transactional databases, and other types of data repositories....

→ it can also deal with categorical/object data....

→ Given a set of transactions the goal of association rule mining is to find the rules that allow us to predict the occurrence of a specific item based on the occurrences of other items in the transaction.

It consists of two parts:

1. an antecedent (if) and
2. a consequent (then)

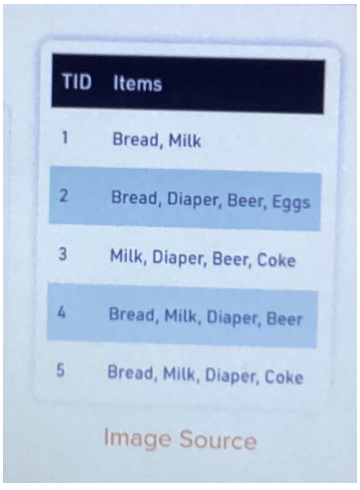
it is found in data

located in conjunction with the antecedent.

ex: "if a customer buys bread, he's 70% likely of buying milk"

antecedent = bread

consequent = milk



TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Image Source

ex of association rule mining.

(8) Association Rule Mining: Basic Definitions:

Support Count (σ):

It accounts for the frequency of occurrence of an itemset.

In the above example:-

$\sigma(\{Milk, Bread, Diaper\}) = 2$

Frequent Itemset:

→ It represents an itemset whose support is greater than (or) equal to the min threshold.

Association Rule:

→ It represents an implication expression of the form $X \rightarrow Y$

→ here, X and Y represent any '2' itemsets...

ex: $\{Milk, Diaper\} \rightarrow \{Beer\}$

Rule Evaluation Metrics:

Support(s): It is the no. of transactions that include items from the {X} and {Y} parts of the

rule as a percentage of total transactions.

Support = σ(X+Y) ÷ total:

It is a fraction of transactions that include both (X) and (Y)

Confidence (C):

This ratio represents the total no. of transactions of all of the items in {A} and {B} to the no. of transactions of the items in {A}

Conf(X ⇒ Y) = Supp(X ∪ Y) ÷ Supp(X)

It counts the no. of items each time in (Y) appears in transactions that also include items in (X)

Lift (L): The lift of the rule X ⇒ Y is the confidence of the rule divided by the expected confidence.

Lift $(X \Rightarrow Y)$ = Conf $(X \Rightarrow Y)$ ÷ Supp (Y):

Lift values near 1 indicate that 'X' and 'Y' almost always appear together as expected.

[Lift > 1] they appear together more than expected....

[Lift < 1] less than expected....

Applications of Association Rule Mining:

1. Market - Basket Analysis
2. Medical Diagnosis
3. Census Data

Algorithms of Association Rule Mining:

1. Apriori Algo
2. Eclat Algo
3. FP - Growth Algo

Apriori Algo:

→ It is for finding frequent itemsets in a dataset for boolean Association rule.

→ It uses prior knowledge of frequent itemset properties.

→ we apply an iterative approach (or) level-wise search where k-frequent itemsets are used to find (k+1) itemsets.

→ we use Apriori property which helps by reducing the search space.