# * Video 1: Closed & Maximal Frequent Itemset

* **Downward Closure Property**: Any subset of a frequent itemset must also be frequent

{A B C} ⟶ {A} {B} {C} {AB} {BC} {CA} {A,B C}

A,B,C N=3 ⟶ 7.

- For N distinct items, frequent itemset will be $2^N - 1$.

## * Closed & Maximal Frequent Itemsets:

### MAXIMAL FREQUENT ITEMSET:

| T_id | Items Purchased |
|---|---|
| 1 | Milk, Bread, Coffee |
| 2 | Butter, Bread, Egg |
| 3 | Milk, Butter, Bread, Egg |
| 4 | Butter, Egg. |

Using Apriori Algorithm, we get: Min. support = 2

Step 1: 1 Frequent Itemset

| Itemset | Sup. count |
|---|---|
| {M} | 2 |
| {Br} | 3 |
| {Bu} | 3 |
| {Eg} | 3 |

Step 2: 2 Freq Itemset

| Itemset | Sup. count |
|---|---|
| {M, Br} | 2 |
| {Br, Bu} | 2 |
| {Br, Eg} | 2 |
| {Bu, Eg} | 3 |

Step 3: 3 Freq Itemset

| Itemset | Sup. count |
|---|---|
| {Br, Bu, Eg} | 2 |
| {M, Br, Eg} | 1 |
| {M, Bu, Eg} | 1 |

- An itemset is called as maximal frequent itemset if, its ~~subsets~~ supersets are ~~in frame.~~ infrequent.

  Eg: {Br, Bu, Eg} ← maximal freq. itemset

Drawback

Maximal freq item sets do not have any information about the support count of the subsets.
Because of this it may give some redundant information.

* CLOSED FREQUENT ITEM SET : A frequent itemset is also known as closed frequent item set if its superset doesn't have same support count as of it, or greater sup. count.

{M} : 2 $\xrightarrow{\text{superset}}$ {M, Br} : 2

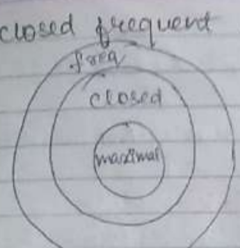
∴ not a closed freq. itemset.

{Br} : 3 (closed). → {M, Br} : 2 → Maximal, ∵ superset is infrequent ∴ ~~not~~ closed closed.
{Br, Bu} : 2
{Br, Eg} : 2

{E} : 3 → {Br, Eg} : 2 ∴ Not closed
{Bu, Eg} : 3 because of {Bu, Eg} = 3

NOTE:
A maximal frequent itemset can be a closed one, but closed cannot be a maximal one.

- Thus, there exists a relationship b/w frequent, closed frequent & maximal freq itemset.




∴ [maximal ⊂ closed ⊂ freq] ($\subset$ subset)

* Video 2: Generating Association Rules from frequent K-itemsets

Steps:

1). For each frequent itemset l, generate all non empty subsets of l. ∴ For {Br, Bu, Eg} :-
∴ {Br, Bu, Eg} {Br, Bu} {Bu, Eg} {Br, Eg}
{B} {Bu} {Eg}

2). For every non-empty subset s of l, output the rule "s → (l-s)" if : $\boxed{\frac{sc(l)}{sc(s)}} \geq$ min-confidence

support count

- Rule 1 : {Br, Bu} ⇒ Eg // The customer who purchase Br, Bu also purchase Eg.

$$\frac{\text{sup-count}(Br, Bu, Eg)}{\text{sup-count}(Br, Bu)} = \frac{2}{2} \times 100 = 100\%$$

Let us specify min-confidence = 70%.
∴ This rule is a Strong Rule.

Rule 2: {Bu, Eg} ⇒ {Br}

$\frac{\text{Sup-count}(Br, Bu, Eg)}{\text{Sup-count}(Bu, Eg)} = \frac{2}{3} \times 100 = 66.66\%$

∴ Not Strong / Not Interesting rules

## * Video 3: FP Growth Algorithm :-
↳ Frequent Item Set Mining Method

## * Drawbacks of Apriori :-

- Generates large candidate itemsets if itemset in the database is large.
- Needs multiple scans of database, to know the sup-count.
- So, the problem with Apriori is space & time. These drawbacks can be overcome by FP GROWTH ALGORITHM.

## FP GROWTH ALGORITHM :-
(Divide & Conquer Strategy)

→ Adv: generates only frequent itemsets & not the infrequent itemsets.

- It compresses the database into a frequent pattern tree.
- Divides the compressed database into a set of conditional databases.
- Each conditional database is associated with one frequent item.

E.g: min-sup = 3

| T-id | Items Purchased |
|---|---|
| 1 | I1, I2, I3 |
| 2 | I2, I3, I4 |
| 3 | I4, I5 |
| 4 | I1, I2, I4 |
| 5 | I1, I2, I3, I5 |
| 6 | I1, I2, I3, I4 |

Algorithm :-

- First, scans the database to find sup-count of each item.
- Then, list them in descending order by eliminating the items whose sup-count is less than min_supcount.

∴ {I2:5, I1:4, I3:4, I4:4}

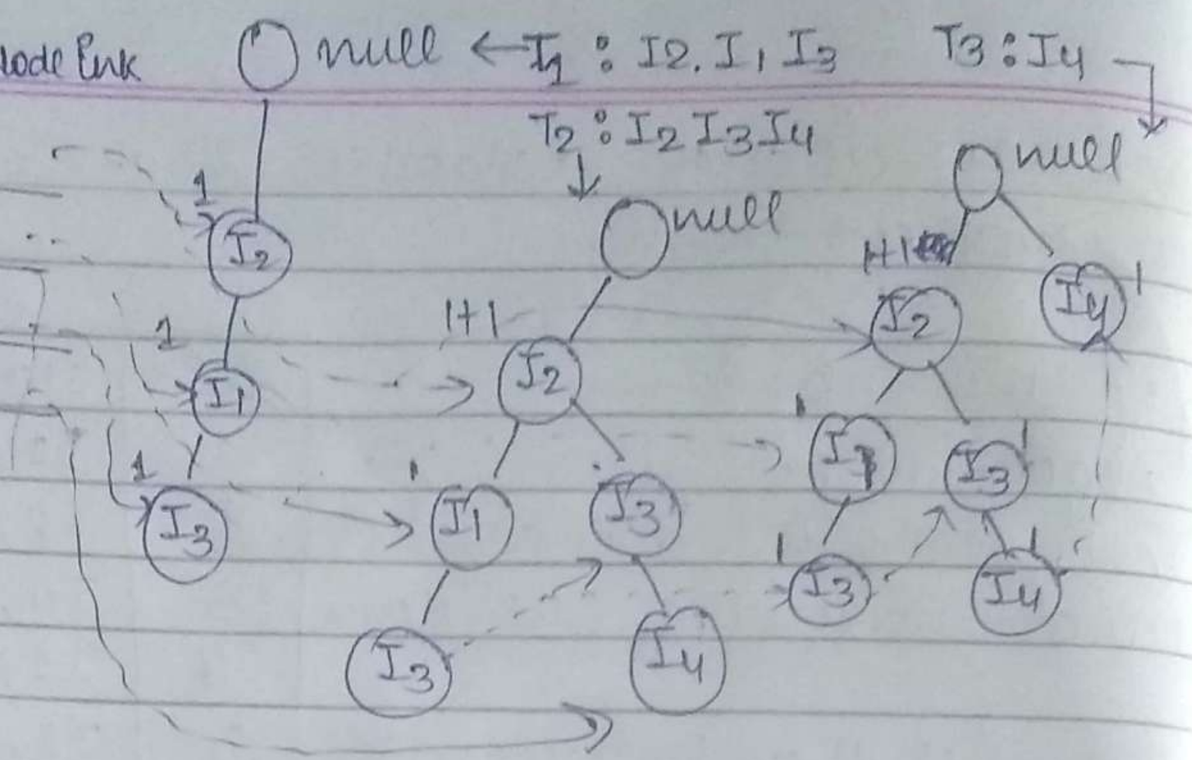
- Now, algorithm will reorganise the table :-

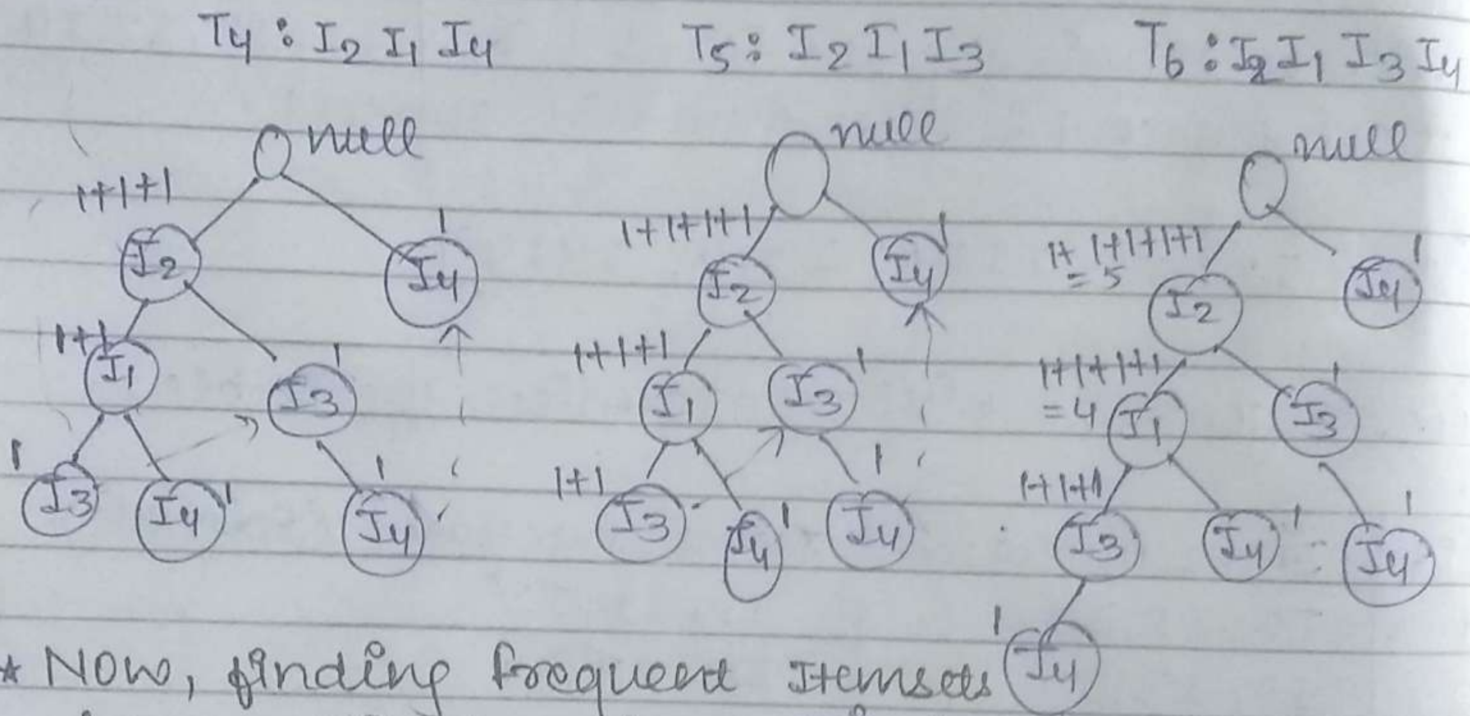
| Tid | Items Purchased | Items Purchased (Sorted) |
|---|---|---|
| 1 | I1, I2, I3 | I2, I1, I3 |
| 2 | I2, I3, I4 | I2, I3, I4 |
| 3 | I4, I5 | I4 |
| 4 | I1, I2, I4 | I2, I1, I4 |
| 5 | I1, I2, I3, I5 | I2, I1, I3 |
| 6 | I1, I2, I3, I4 | I2, I1, I3, I4 |

- Header Table :

| Item id | Sup. count | Node-link |
|---|---|---|
| I2 | 5 | |
| I1 | 4 | |
| I3 | 4 | |
| I4 | 4 | |

| Item id | S-c | Node link |
|---|---|---|
| $I_2$ | 5 | |
| $I_1$ | 4 | |
| $I_3$ | 4 | |
| $I_4$ | 4 | |




* Now, finding frequent Itemsets from FP Tree :- min_sup = 3

| Item | Conditional Pattern Base | Conditional FP Tree | Freq Patterns |
|---|---|---|---|
| $I_4$ | $I_2 I_1 I_3 : 1$, $I_2 I_1 : 1$, $I_2 I_3 : 1$, ~~$I_2 : 1$~~ (Root node) | $I_2 : 3$, $I_1 : 2$, $I_3 : 2$ ($\because < 3$) | $I_2 I_4 : 3$ |
| $I_3$ | $I_2 I_1 : 3$, $I_2 : 1$ | $I_2 : 4$, $I_1 : 3$ | $I_2 I_3$; $I_1 I_3 : 3$, $I_1 I_2 I_3 : 3$ |
| $I_1$ | $\{I_2 : 4\}$ | $\{I_2 : 4\}$ | $\{I_2 I_1\} : 4$ |

Scanned by CamScanner

# Cluster Analysis

* Video 1

## * Cluster Analysis:

- It is a data analysis mechanism that is used for grouping of similar objects.

- It groups the objects by looking at the similarities b/w object characteristics.

E.g: Grouping of docs related to sports into 1 cluster, medicines to other clusters etc.

- In order to find the similarities b/w the objects, we have different measures:-

One of the metric is:

MINKOWSKI DISTANCE:
↳ similarity metric
↳ used for numerical data

$$d(i,j) = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

If $p=1 \rightarrow$ Manhattan Distance

$p=2 \longrightarrow$ Euclidean Distance

* P=1 Manhattan Distance

$$\boxed{d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|}$$

* P=2 Euclidean Distance

$$\boxed{d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ip} - x_{jp})^2}}$$

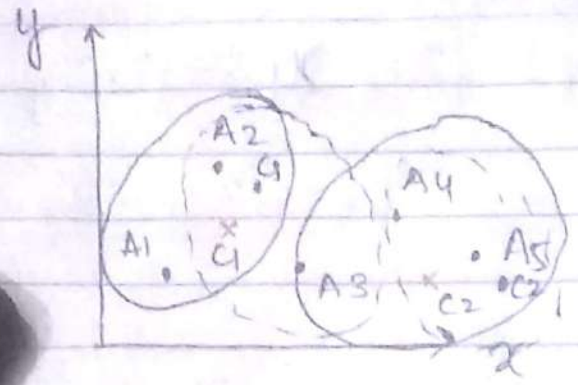
- $i = (x_{i1}, x_{i2}, \dots, x_{ip})$
- $j = (x_{j1}, x_{j2}, \dots, x_{jp})$

E.g. $O_1$ $O_2$ $O_3$
If dist b/w ($O_1$ & $O_2$) < dist b/w ($O_1$ & $O_3$)
then, $O_1$ & $O_2$ are put in 1 cluster.

* Cluster Methods:

1). Partitioning Method
2). Hierarchical Method
3). Density-Based Method

# * PARTITIONING METHOD




- We have five data points $A_1, A_2, A_3, A_4, A_5$.

- In this method, the centroid point will be given & based on that centroid point the data points are clustered.
  - ↳ Based on the distance b/w datapoints & centroid points

- Now, it will find the mean of centroid points ($C_1$ & $C_2$) & once again the data points neares to the new centroid points are clustered newly.

- So, in this way the cluster changes iteratively.

- This type of partioning method is used for small-to-medium size data.

- In this, the no. of centroid points are equal to the no. of clusters (K).
  Here, <u>K = 2</u>

Scanned by CamScanner

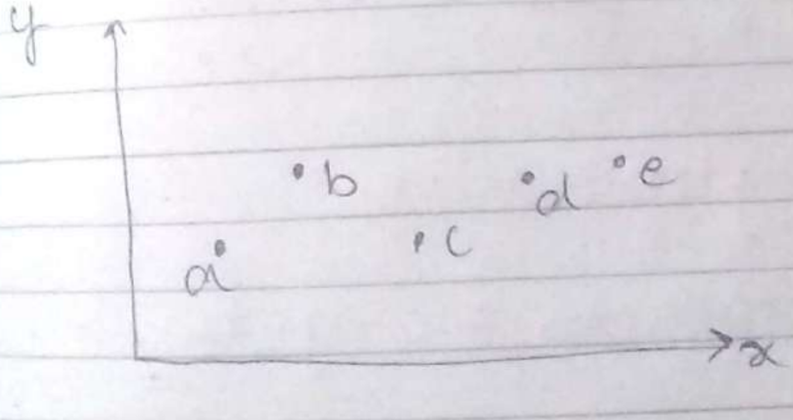
- Here, scaling is required. Thus, data transformation is the preprocessing required for this method since we are using distance metric.

- One of the popular algorithm in partitioning method is: K Means Algorithm

## * Hierarchical Method :

- In this method, the objects which are nearer to each other are grouped together.

- Two approaches → Bottom-Up, → Top-down




We have 05 data points a, b, c, d, e.

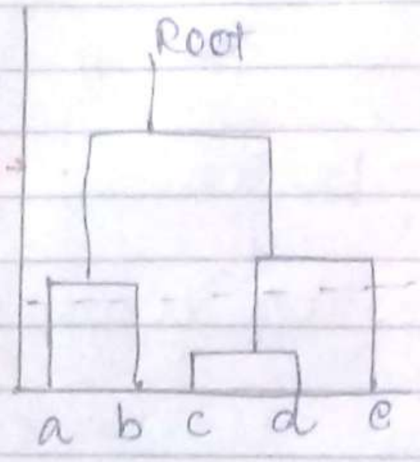
In Bottom Up Approach, it finds the distance from each data point to all other data points & finds out which are the two objects nearer to each other.
∴ (c, d) forms cluster.

- In the next, iteration it finds distance from this cluster to all other points & each point from other points.
  So (a, c) are clustered.

- Similarly, it continues.

- Finally all the objects are clustered together. It is also called as Nested clustering.

dendrogram →




In this method, we can cut it at any level.
However, once merging is done it cannot be undone.

- Algorithm: Agglomerative

In Top down approach, we go in reverse way.
↳ Algorithm is (Divisive).

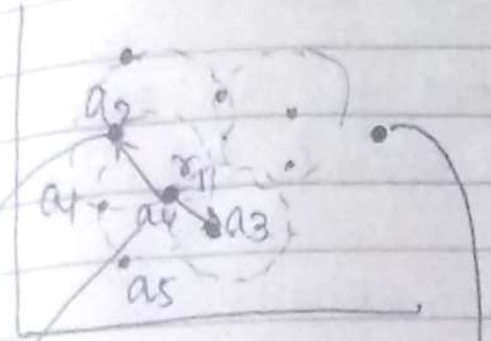
Drawback: They are preparing spherical clusters. ~~Aribtarit~~ Arbitrarily shaped clusters are not formed.

→ Algorithm : DBSCAN

# * Density Based Method :

- This method is based on the ~~ab~~ notion of density in neighbourhood of objects.

- It can generate arbitarily shaped clusters based on the density locations.




Two parameters :

- Epsilon (ε)
  ↳ refers to the radius

- Minpts
  ↳ refers to the min points in the area formed within radius 'r'.
  ↳ constraint

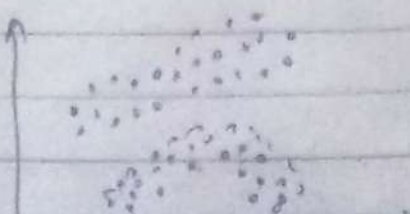
Core point, if it satisfies minpoint constraint

~~Boundary~~

Border point, it is in neighbourhood of core point.

Noise point, it is not in any cluster.

- It can find the neighbourhood datapoints that satisfies the constraint & then forms clusters, based on density.



# * Video 2: K-Means Clustering

- This is one of the popular ~~method~~, algorithm under partitioning method.

## K-Means Algorithm steps :-

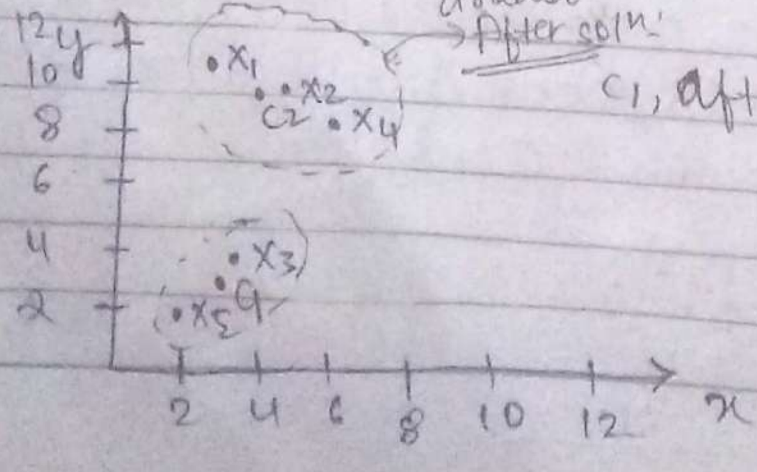
K, datapoints are given

i> cluster the objects as per the K-value.
ii> Finding mean of the each of the clusters. (centroids).
iii> Groups based on dist & centroid.
iv> Continue step 2 & 3 recursively till it converges (centroids remain same in the previous step & the next step or the data points don't change).

**Example:** $x_1 = (3,11)$ $x_2 = (5,10)$ $x_3 = (3,4)$
$x_4 = (6,9)$ $x_5 = (2,2)$

Assume $K=2$ & initial clusters are
$C_1 = \{x_1, x_3, x_5\}$ & $C_2 = \{x_2, x_4\}$.

drawn After soln:




c1, after soln:

$1+1$ $\sqrt{4+1}$ $\sqrt{5}$

$4+9$

Iteration 1

$C_1 = \{x_1, x_3, x_5\}$ $\quad C_2 = \{x_2, x_4\}$

$\therefore$ Centroid $C_1 = \left(\frac{3+3+2}{3}, \frac{11+4+2}{3}\right) = (2.67, 5.67)$

Centroid $C_2 = \left(\frac{5+6}{2}, \frac{10+9}{2}\right) = (5.5, 9.5)$

To compute distance:

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|
| $C_1$ | 5.66 | 6.66 | 2 | 6.66 | 4.34 |
| $C_2$ | 4 | 1 | 8 | 1 | 11 |
| Cluster | $C_2$ | $C_2$ | $C_1$ | $C_2$ | $C_1$ |

using Manhattam distance: // We can use any.

$d(C_1, x_1) = |2.67 - 3| + |5.67 - 11| = 5.66$

$d(C_2, x_1) = |5.5 - 3| + |9.5 - 11| = 4$

$d(C_1, x_2) = 6.66$ $\quad d(C_1, x_5) = 4.34$

$d(C_2, x_2) = 1$ $\quad d(C_2, x_5) = 11$

$d(C_1, x_3) = 2$

$d(C_2, x_3) = 8$

$d(C_1, x_4) = 6.66$

$d(C_2, x_4) = 1$

* New clusters:

Now, in $C_1 = \{x_3, x_5\}$ // Here Cluster changed so let's compute.

$C_2 = \{x_1, x_2, x_4\}$

$\therefore C_1 = \left(\frac{3+2}{2}, \frac{4+2}{2}\right) = (2.5, 3)$

$C_2 = \left(\frac{3+5+6}{3}, \frac{11+10+9}{3}\right) = (4.67, 10)$

Now, to compute ~~each~~ distn: b/w centroid to data points

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|
| $C_1$ | 8.5 | 9.5 | 1.5 | 9.5 | 1.5 |
| $C_2$ | 2.67 | 0.33 | 7.67 | 2.33 | 10.67 |
| Cluster | $C_2$ | $C_2$ | $C_1$ | $C_2$ | $C_1$ |

$d(C_1, x_1) = |2.5-3| + |3-11| = 8.5$
$d(C_2, x_1) = |4.67-3| + |3-10| = 2.67$

$d(C_1, x_2) = 9.5$
$d(C_2, x_2) = 0.33$

$d(C_1, x_3) = 1.5$
$d(C_2, x_3) = 7.67$

$d(C_1, x_4) = 9.5$
$d(C_2, x_4) = 2.33$

$d(C_1, x_5) = 1.5$
$d(C_2, x_5) = 10.67$

Now, new clusters :-

$C_1 = \{x_3, x_5\}$
$C_2 = \{x_1, x_2, x_4\}$

Here, we can see that these clusters are same as the previous cluster.

$\therefore$ centroid value is same & it converges.

$\therefore$ Algorithm converges here.

$\therefore$ final $C_1 = (2.5, 3)$
$C_2 = (4.67, 10)$

} see graph on previous page

* How value of k can be chosen?
  ↳ find sum of squared errors for cluster 1 & cluster 2.

$SSE_1 =$

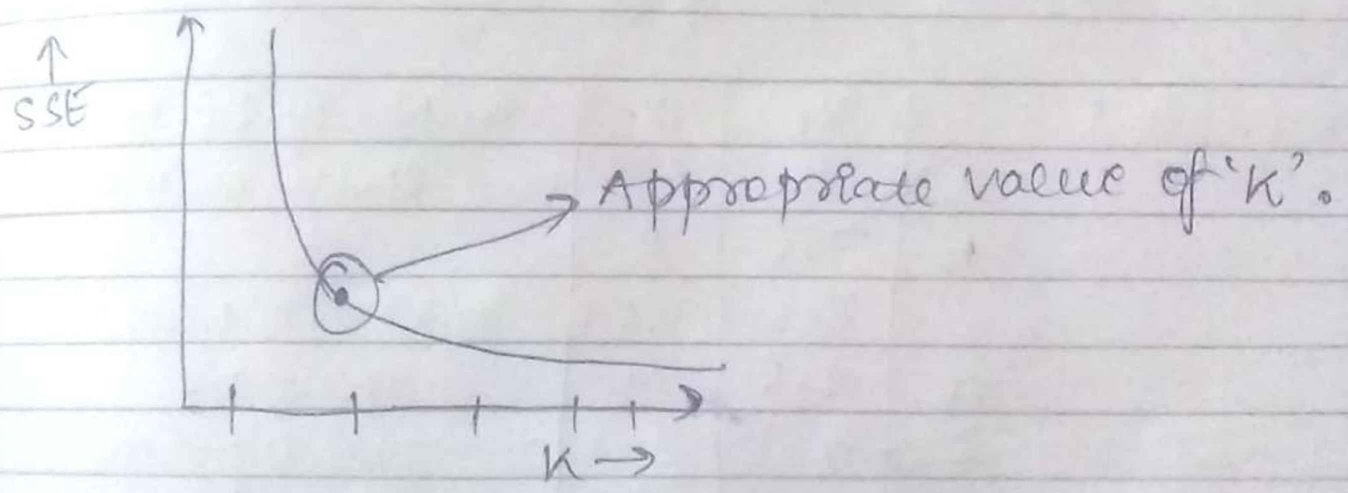
$SSE_2 =$

Then, find $SSE = SSE_1 + SSE_2$

Here, $SSE_1 = \sum_{l=1}^{n} (x_l - C_1)^2$

$SSE_2 = \sum_{l=1}^{n} (x_l - C_2)^2$

$\therefore SSE = SSE_1 + SSE_2$.




* Video 3: Hierarchical Clustering :-

* Hierarchical Agglomerative Clustering (HAC)

Scanned by CamScanner

(HAC)

# * Hierarchical Agglomerative Clustering

## Algorithm Steps-

read the data points

i) compute the distance b/w the data points. → Manhattam or Euclidean dist.

ii) Group ~~two~~ objects with ~~least~~ smallest distance. For each iteration

iii) Build the distance matrix

update distance matrix ↳ compute the dist to all other clusters.

iv). Repeat steps ~~to~~ (ii) & (iii).

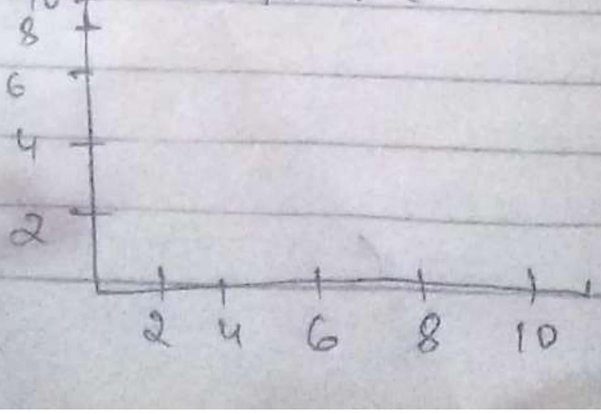
Example:

| | $x_1$ | $x_2$ |
|---|---|---|
| a | 2 | 4 |
| b | 7.5 | 2 |
| c | 7.5 | 3.5 |
| d | 1.5 | 4 |
| e | 9 | 1 |

$x_1$: No. of hours spent on reading

$x_2$: no. of hours spent on watching videos

We can go with Manhattam or Euclidean distance.



i> using Manhattan distance :-

$d(a,b) = |2-7.5| + |4-2| = 7.5$

$d(a,c) = |2-7.5| + |4-3.5| = 6$

$d(a,d) = |2-1.5| + |4-4| = 0.5$

$d(a,e)\ \ |2-9| + |4-1| = 10$

Step (ii) Distance Matrix :-

$d(b,c) = 1.5$

$d(b,d) = 8$

$d(b,e) = 2.5$

$d(c,d) = 6.5$

$d(c,e) = 4$

| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | 0 | 7.5 | 6 | 0.5 | 10 |
| b | 7.5 | 0 | 1.5 | 8 | 2.5 |
| c | 6 | 1.5 | 0 | 6.5 | 4 |
| d | 0.5 | 8 | 6.5 | 0 | 10.5 |
| e | 10 | 2.5 | 4 | 10.5 | 0 |

$d(a,e) = 10.5$

(iii) ∴ we need to group (a & d) ∵ they have smallest value.

Now, we find distance of this cluster from all other data points.

↳ Here, we use linkage criteria

• Single linkage: minimal pair linkage is taken $C_1$ $C_2$

∴ $ad \overset{4}{\frown} b$ ∴ 4 is taken
5

• Complete linkage take max. $C_1$ $C_2$

• Avg linkage $C_1$ $C_2$

$\frac{\text{Total } C_1 C_2}{N_{C_1} N_{C_2}}$

Now, updated matrix

| | ad | b | c | e |
|---|---|---|---|---|
| ad | 0 | | | |
| b | 7.5 | 0 | | |
| c | 6 | 1.5 | 0 | |
| e | 10 | 2.5 | 4 | 0 |

$d(\cancel{ad}, \cancel{b}) = \cancel{d(b,a)}\ \cancel{d(b,d)}$   $d(b,a)$   $d(b,d)$
$= 7.5 \quad = 8$

$d(ad,b) = d(a,b)\ \ d(d,b)$
$= 7.5$    8

∴ We cluster (b,c).

Updated Distance matrix :-

| | ad | bc | e |
|---|---|---|---|
| ad | 0 | | |
| bc | 6 | 0 | |
| e | 10 | 2.5 | 0 |

d (bc, ad) = d(b, ad) ~~and~~ d(c, ad)
= 7.5 6

d(e, ad) = 10

d(e, bc) = d(e, b) or d(c, e)
= 2.5 4

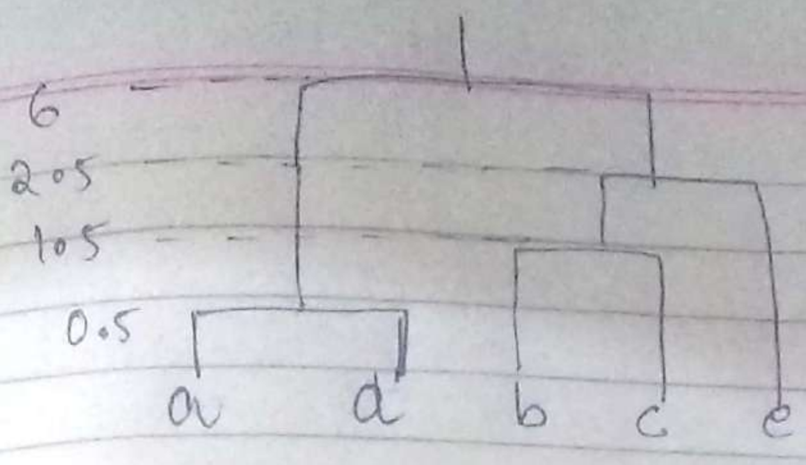
∴ We cluster (e, b, c)

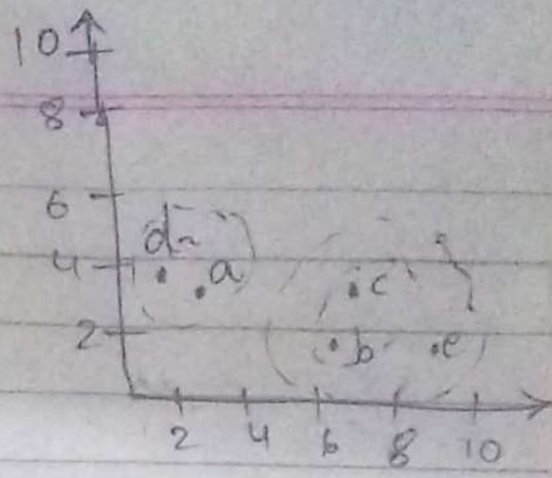
Updated distance matrix :-

| | ad | bce |
|---|---|---|
| ad | 0 | |
| bce | 6 | 0 |

d(bce, ad) = d(bc, ad) d(e, ad)
6 10

~~Now finally~~ (ad, bce)


6
2.5
1.5
0.5
a d b c e


10
8
6
4
2
d
a
c
b
e
2 4 6 8 10

* Cluster Analysis :

- It is a data analysis mechanism that is used for grouping of similar objects.

- It groups the objects by looking at the similarities b/w object characteristics.

E.g.: Grouping of docs related to sports into 1 cluster, medicines to other clusters etc.

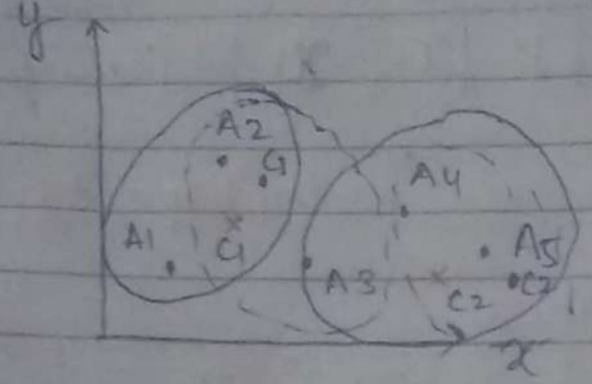
## * Cluster Methods :

1). Partitioning Method

2). Hierarchical Method

3). Density-Based Method

# * PARTITIONING METHOD




- We have five data points $A_1, A_2, A_3, A_4, A_5$.

- In this method, the centroid point will be given & based on that centroid point the data points are clustered.
  - ↳ Based on the distance b/w datapoints & centroid points

- Now, it will find the mean of centroid points ($C_1$ & $C_2$) & once again the data points nearer to the new centroid points are clustered newly.

- So, in this way the cluster changes iteratively.

- This type of partioning method is used for small-to-medium size data.

- In this, the no. of centroid points are equal to the no. of clusters (K).
  Here, K=2

- Here, scaling is required. Thus, data transformation is the preprocessing required for this method since we are using distance metric.

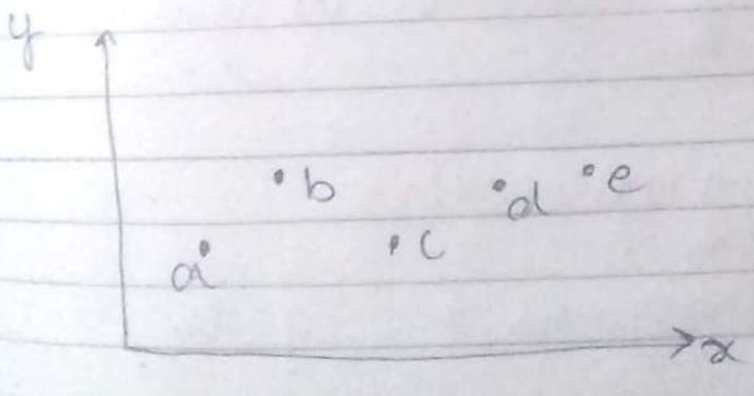
- One of the popular algorithm in partitioning method is: K Means Algorithm

* **Hierarchical Method:**

- In this method, the objects which are nearer to each other are grouped together.

- Two approaches → Bottom-Up, → Top-down

We have 05 data points a, b, c, d, e




In Bottom Up Approach, it finds the distance from each data point to all other data points & finds out which are the two objects nearer to each other.

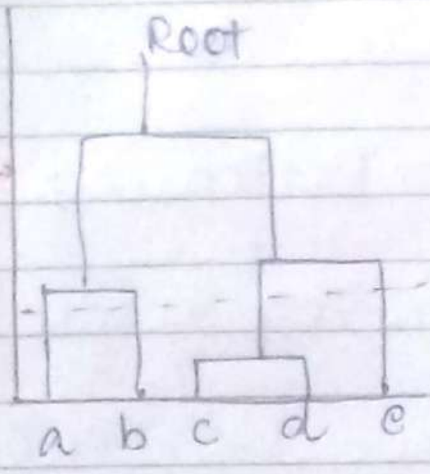
∴ (c,d) forms cluster.

- In the next, iteration: it finds distance from this cluster to all other points & each point from other points.
  So (a,c) are clustered.

- Similarly, it continues.

- Finally all the objects are clustered together. It is also called as Nested clustering.

dendrogram →




In this method, we can cut it at any level.
However, once merging is done it cannot be undone.

- Algorithm: Agglomerative

In Top down approach, we go in reverse way.

↳ Algorithm is (Divisive).

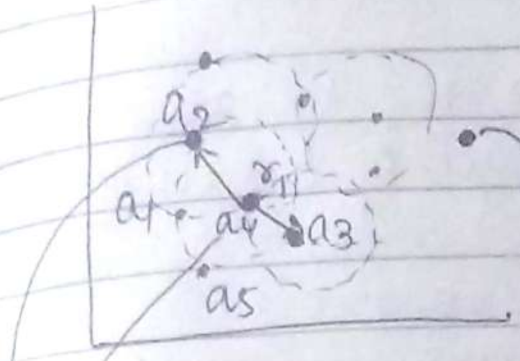
Drawback: They are preparing spherical clusters. ~~Arbitary~~ Arbitarily shaped clusters are not formed.

→ Algorithm : DBSCAN

# Density Based Method :

- This method is based on the ~~obe~~ notion of density in neighbourhood of objects.

- It can generate arbitarily shaped clusters based on the density locations.




Two parameters:
- Epsilon ($\varepsilon$)
  ↳ refers to the radius

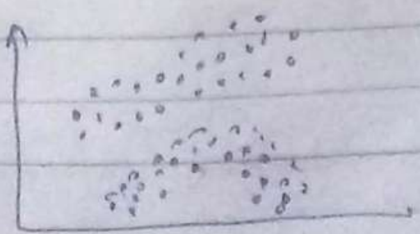
- Minpts
  ↳ refers to the min points in the area formed within radius 'r'.
  ↳ constraint

Core point, if it satisfies minpoint constraint

~~Boundary~~
Border point, it is in neighbourhood of core point.

Noise point, it is not in any cluster.

- It can find the neighbourhood datapoints that satisfies the constraint & then forms clusters, based on density.



Scanned by CamScanner

2). Suppose that the data mining task is to cluster following eight points with $(x, y)$ representing location into the clusters:-

$A_1(2,10)$ $A_2(2,5)$ $A_3(8,4)$

$B_1(5,8)$ $B_2(7,5)$ $B_3(6,4)$

$C_1(1,2)$ $C_2(4,9)$.

The distance function is Euclidean distance. Suppose we initially assign $A_1$, $B_1$ & $C_1$ as the center of each cluster, respectively. Use the K-Means Algo to show only:-

a). The 3 clusters after first round execution

b). The final 3 clusters.

Sol$^n$: Given centroids of each of the three clusters are :-

$c_1 =$ centroid $1 = (2,10)$

$c_2 =$ centroid $_2 = (5,8)$

$c_3 =$ centroid $_3 = (1,2)$

Now, compute the distance using Euclidean dist.

$d(\text{centroid}_1, A_2) = \sqrt{(2-2)^2 + (10-5)^2} = \sqrt{5^2} = 5$

$d(\text{centroid}_2, A_2) = 4.24$

$d(\text{centroid}_3, A_2) = 3.16$

$d(\text{centroid}_1, A_3) = 8.48$

$d(\text{centroid}_2, A_3) = 5$

$d(\text{centroid}_3, A_3) = 7.280$

$d(\text{centroid}_1, B_2) = 7.07$

$d(\text{centroid}_2, B_2) = 3.60$

$d(\text{centroid}_3, B_2) = 6.70$

$d(\text{centroid}_1, B_3) = 7.211$

$d(\text{centroid}_2, B_3) = 4.123$

$d(\text{centroid}_3, B_3) = 5.38$

d(centroid$_1$, $B_1$) = 3.60
d(centroid$_2$, $B_1$) = 0
d(centroid$_3$, $B_1$) = 7.211

d(centroid$_1$, $C_1$) = 8.0622
d(centroid$_2$, $C_1$) = 7.211
d(centroid$_3$, $C_1$) = 0

d(centroid$_1$, $C_2$) = 2.23
d(centroid$_2$, $C_2$) = 1.41
d(centroid$_3$, $C_2$) = 7.61

d(centroid$_1$, $A_1$) = 0
d(centroid$_2$, $A_1$) = 3.60
d(centroid$_3$, $A_1$) = 8.06

| | $A_1$ | $A_2$ | $A_3$ | $B_1$ | $B_2$ | $B_3$ | $C_1$ | $C_2$ |
|---|---|---|---|---|---|---|---|---|
| centroid 1 | 0 | 5 | 8.48 | 3.60 | 7.07 | 7.21 | 8.06 | 2.23 |
| centroid 2 | 3.60 | 4.24 | 5 | 0 | 3.60 | 4.123 | 7.21 | 1.41 |
| centroid 3 | 8.06 | 3.16 | 7.28 | 7.21 | 6.70 | 5.38 | 0 | 7.61 |
| Cluster | 1 | 3 | 2 | 2 | 2 | 2 | 3 | 2 |

∴ Initially clusters are

Cluster 1 = {$A_1$}
Cluster 2 = {$A_3, B_1, B_2, B_3, C_2$}
Cluster 3 = {$A_2, C_1$}

— (i)

∴ New centroids are:-

cent 1 = (2, 10)

$$\text{cent } 2 = \left(\frac{8+5+7+6+4}{5}, \frac{4+8+5+4+9}{5}\right)$$

$$= (6, 6)$$

$$\text{cent } 3 = \left(\frac{2+1}{2}, \frac{5+4}{2}\right) = (1.5, 4.5)$$

∴ New distance

| | $A_1$ | $A_2$ | $A_3$ | $B_1$ | $B_2$ | $B_3$ | $C_1$ | $C_2$ |
|---|---|---|---|---|---|---|---|---|
| | | | 8.48 | 3.60 | 7.07 | 7.21 | 8.06 | ~~2.23~~ |
| cent1 | 0 | 5 | 2.82 | 2.23 | 1.41 | 2 | 6.40 | 3.60 |
| cent2 | 5.65 | 4.12 | | | | | | |
| cent3 | 5.52 | 0.70 | 6.51 | 4.94 | 5.52 | 4.52 | 2.54 | 5.14 |
| cluster | 1 | 3 | 2 | 2 | 2 | 2 | 3 | 1 |

Scanned by CamScanner

Now, the new cluster is

Cluster 1 = {$A_1$, $C_2$}

Cluster 2 = {$A_3$, $B_1$, $B_2$, $B_3$}

Cluster 3 = {$A_2$, $C_1$}

$\therefore$ cent 1 = $\left(\frac{2+4}{2}, \frac{10+9}{2}\right) = (3, 9.5)$

cent 2 = $\left(\frac{8+5+7+6}{4}, \frac{4+8+5+4}{4}\right) = (6.5, 5.25)$

cent 3 = $(1.5, 4.5)$

$\therefore$ New distance

| | $A_1$ | $A_2$ | $A_3$ | $B_1$ | $B_2$ | $B_3$ | $C_1$ | $C_2$ |
|---|---|---|---|---|---|---|---|---|
| Cent 1 | 1.11 | 4.60 | 7.43 ~~4.48~~ | 2.5 ~~3.60~~ | 6.020 ~~[illegible]~~ | 6.26 ~~7.21~~ | 7.76 ~~8.06~~ | 1.11 ~~2.23~~ |
| Cent 2 | 6.54 | 4.50 | 1.95 | 3.13 | 0.55 | 1.34 | 6.38 | 4.50 |
| Cent 3 | 5.52 | 0.70 | 6.51 | 4.94 | 5.52 | 4.52 | 2.54 | 5.14 |
| Clusters | 1 | 3 | 2 | 1 | 2 | 2 | 3 | 1 |

Cluster 1 = {$A_1$, $B_1$, $C_2$} = (3.66, 9)

Cluster 2 = {$A_3$, $B_2$, $B_3$} = (7, 4.33)

Cluster 3 = {$A_2$, $C_1$} = (1.5, 4.5)

| | $A_1$ | $A_2$ | $A_3$ | $B_1$ | $B_2$ | $B_3$ | $C_1$ | $C_2$ |
|---|---|---|---|---|---|---|---|---|
| Cent 1 | 1.93 | 4.33 | 6.62 | 1.67 | 5.211 | 5.52 | 7.48 | 0.34 |
| cent 2 | 7.55 | 5.044 | 1.05 | 4.17 | 0.67 | 1.05 | 6.43 | 5.55 |
| cent 3 | 5.52 | 0.70 | 6.51 | 4.94 | 5.52 | 4.52 | 2.54 | 5.14 |
| Cluster | 1 | 3 | 2 | 1 | 2 | 2 | 3 | 1 |

$\therefore$ cluster 1 = {$A_1$, $B_1$, $C_2$}
cluster 2 = {$A_3$, $B_2$, $B_3$}
cluster 3 = {$A_2$, $C_1$}

$\Big\} \rightarrow$ (ii) Final clusters

$\because$ Converged $\therefore$ Terminated

3). Show dendrograms.

a). <u>Using single link</u> :-

| | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 1 | 4 | 5 |
| B | | 0 | 2 | 6 |
| C | | | 0 | 3 |
| D | | | | 0 |

→ club (A, B)

| | ~~AB~~ | ~~B~~ | ~~C~~ | ~~D~~ |
|---|---|---|---|---|
| ~~AB~~ | ~~0~~ | | | |
| ~~B~~ | | | | |
| ~~C~~ | | | | |
| ~~D~~ | | | | |

∴ Updated matrix :-

| | AB | C | D |
|---|---|---|---|
| AB | 0 | 2 | 5 |
| C | | 0 | 3 |
| D | | | 0 |

$d(AB, C) = d(A, C)$ or $d(B, C)$
$= 4$ or $2$ ✓

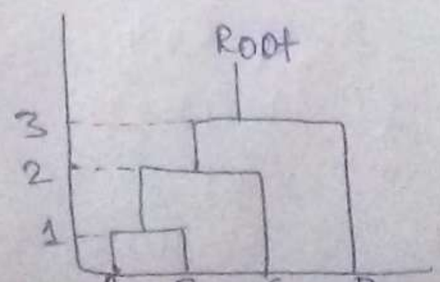
$d(AB, D) = d(A, D)$ or $d(B, D)$
$= 5$ ✓ or $6$

Now club (AB, C)

| | ABC | D |
|---|---|---|
| ABC | 0 | 3 |
| D | | 0 |

$d(ABC, D) = d(AB, D)$ or $(C, D)$
$= 5$ or $3$ ✓

Now, club all of them.
∴ Dendrogram is :



(5)

b). Using complete link:

| | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 1 | 4 | 5 |
| B | | 0 | 2 | 6 |
| C | | | 0 | 3 |
| D | | | | 0 |

→ (A,B) are clubbed.

| | AB | C | D |
|---|---|---|---|
| AB | 0 | 4 | 6 |
| C | | 0 | 3 |
| D | | | 0 |

d(AB,C) = d(A,C) or d(B,C)
= 4 or 2

d(AB,D) = d(A,D) or d(B,D)
= 5 or 6

∴ Club ~~(AB,C)~~ (C,D)

| | AB | CD |
|---|---|---|
| AB | 0 | 6 |
| CD | | 0 |

d(AB,CD) = d(AB,D) or d(AB,C)
= 6 or 4

Now Club all.

∴ Dendrogram:-