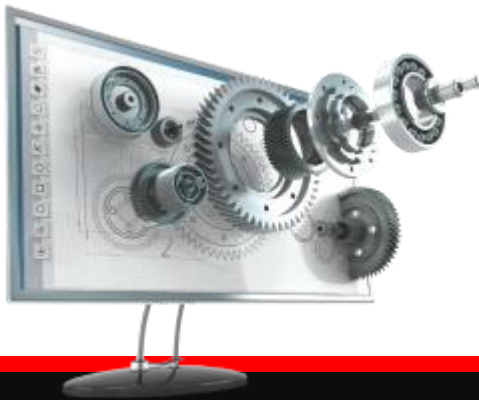




Python for Beginners

Archer Infotech , PUNE



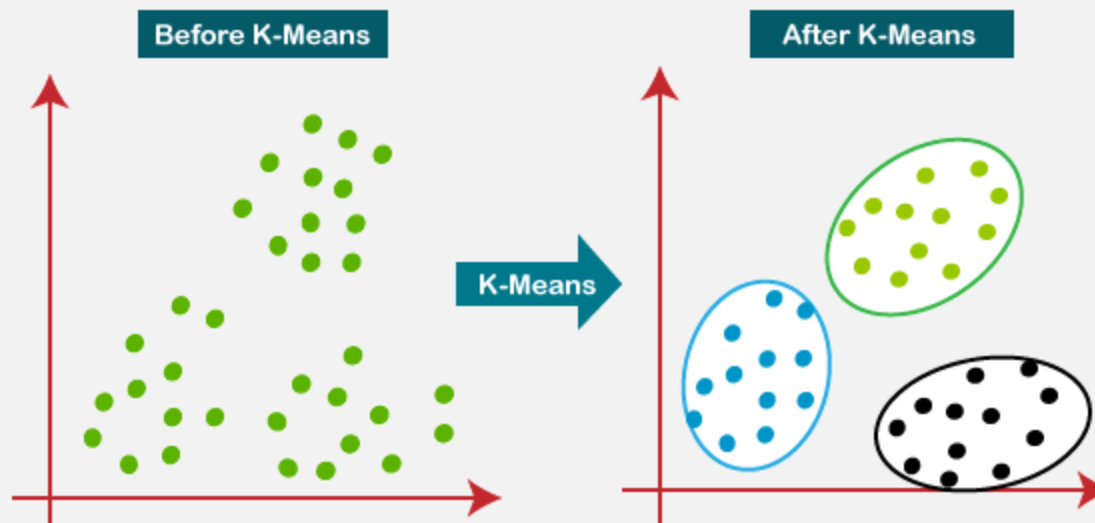


Python – Clustering

K-Means Clustering



- K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science.
- It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties



How does K-Means Works ?



Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

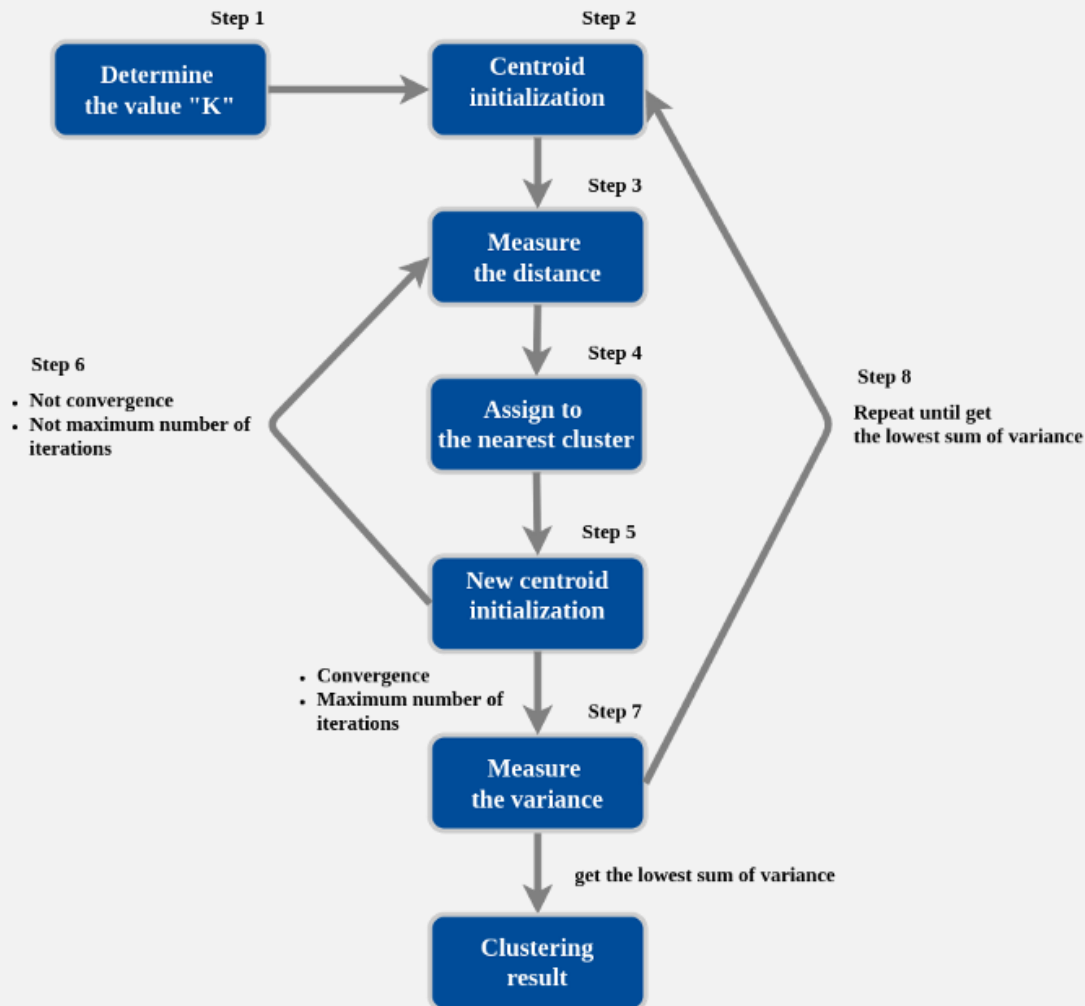
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

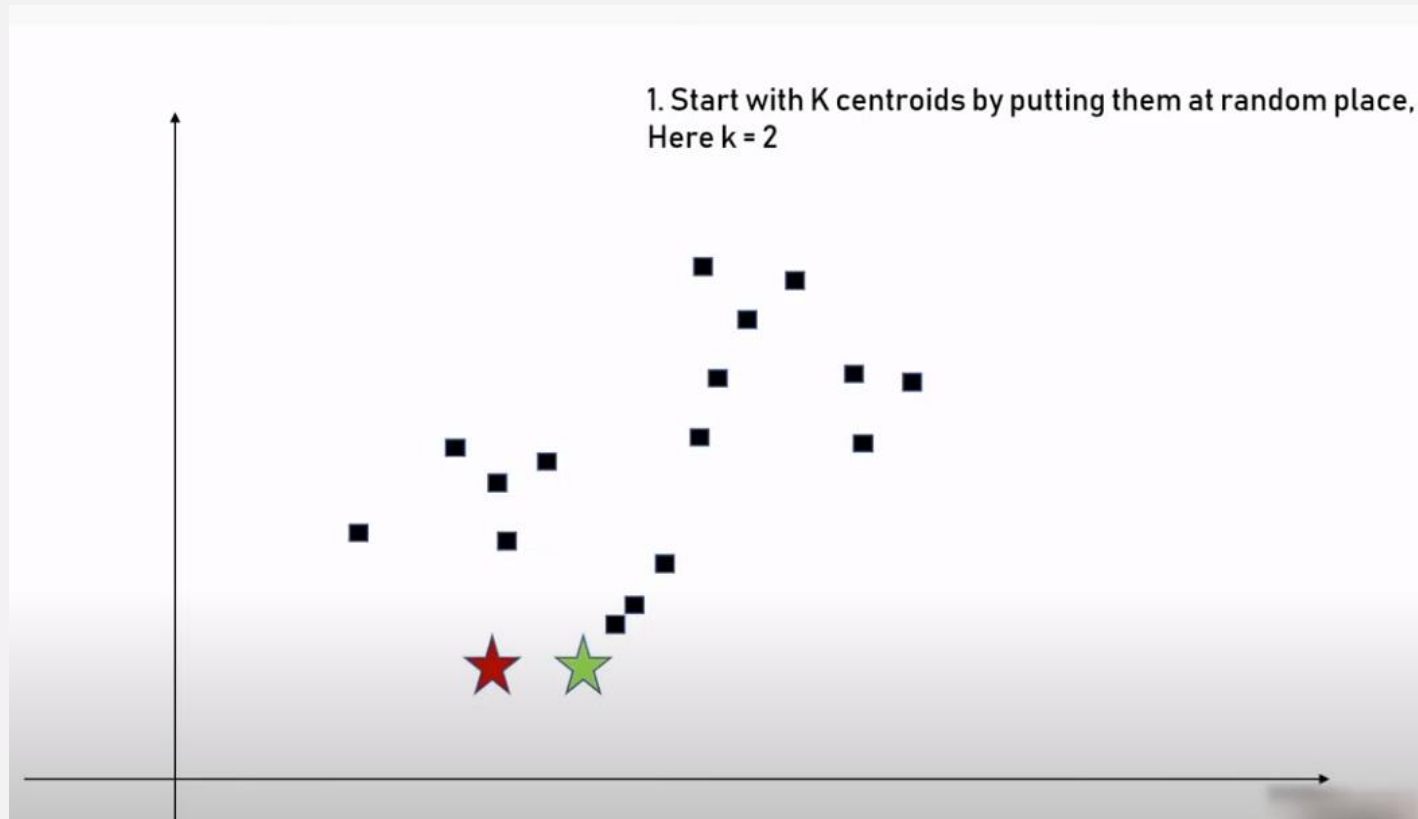
Step-7: The model is ready.



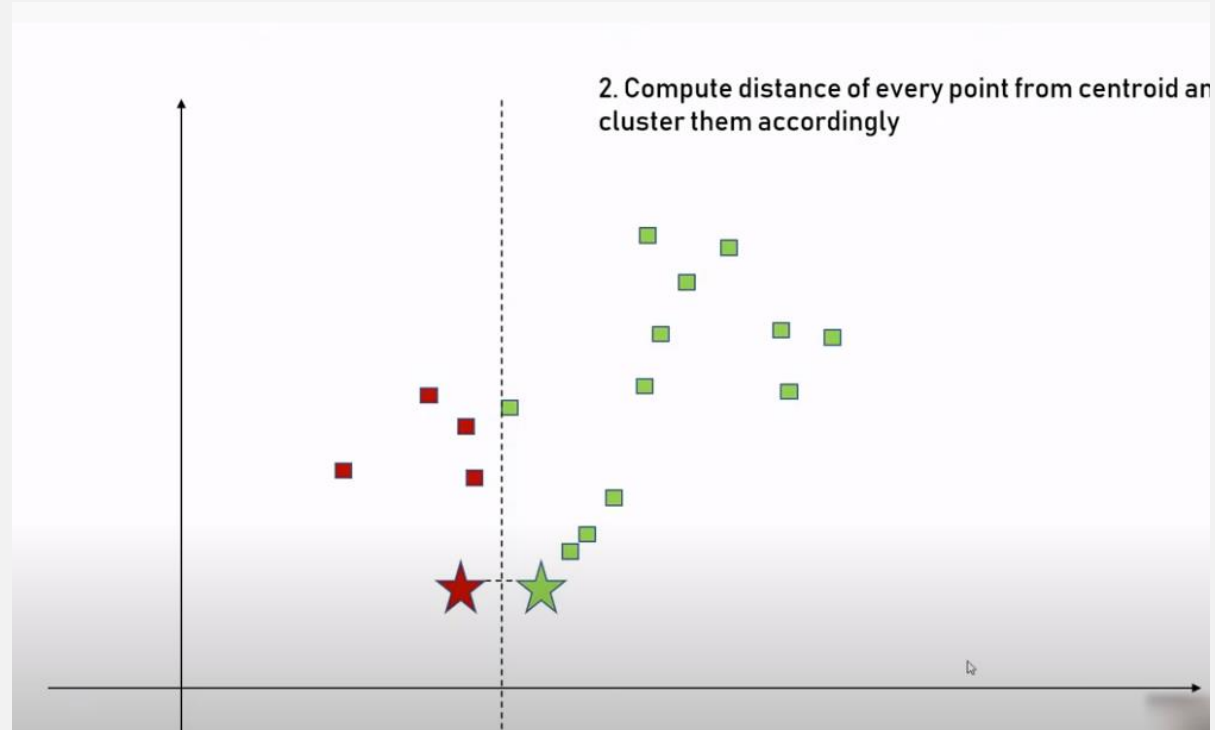
How does K-Means Works ?



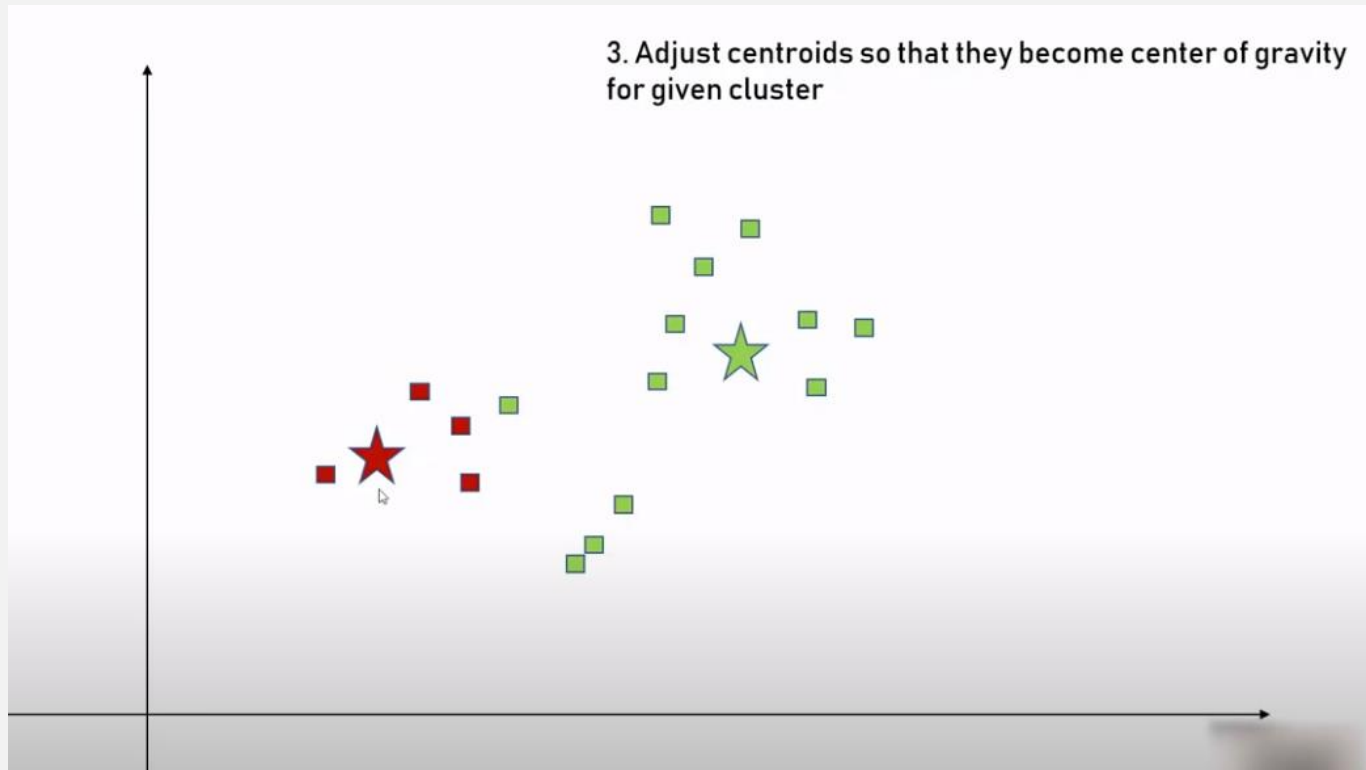
How does K-Means Works ?



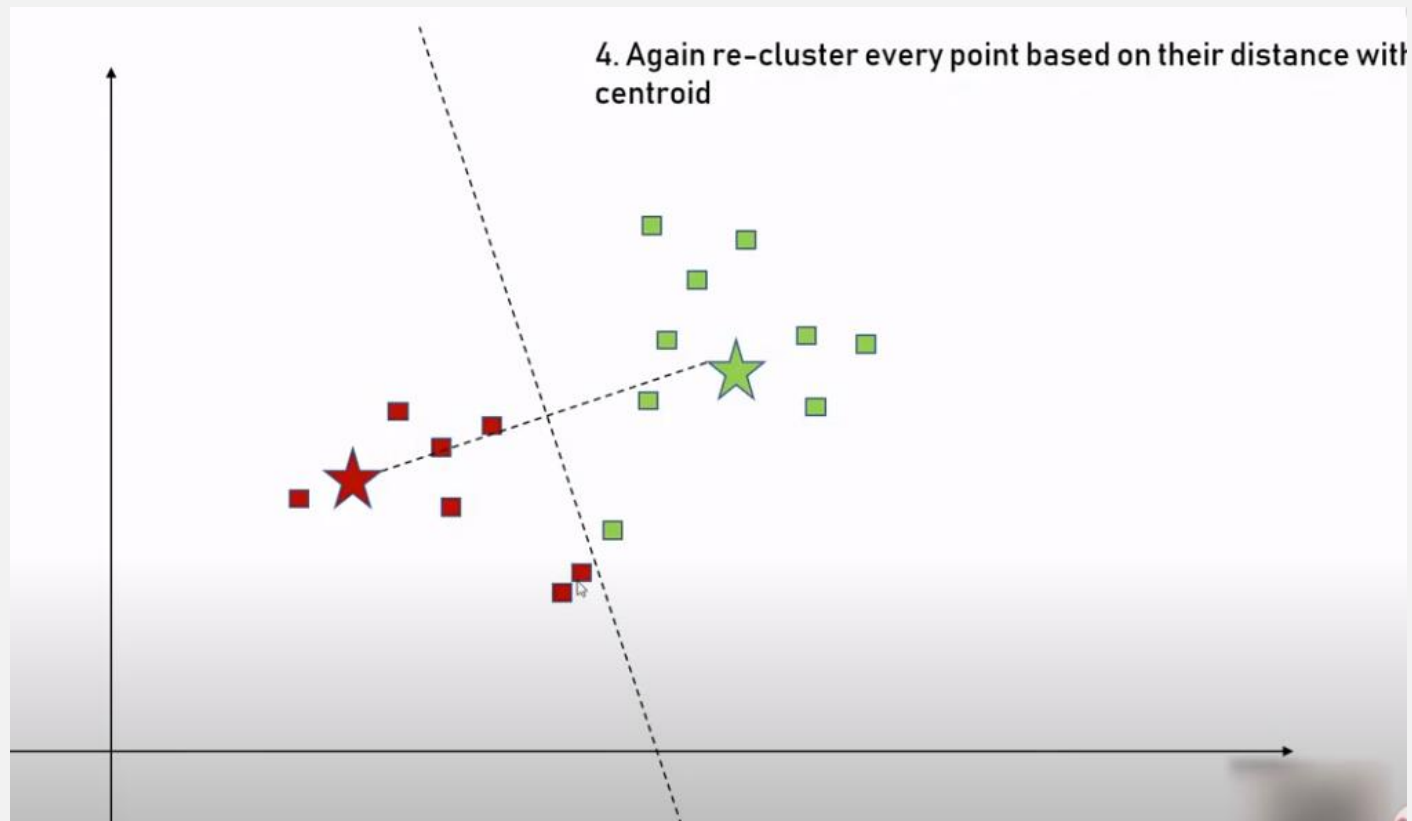
How does K-Means Works ?



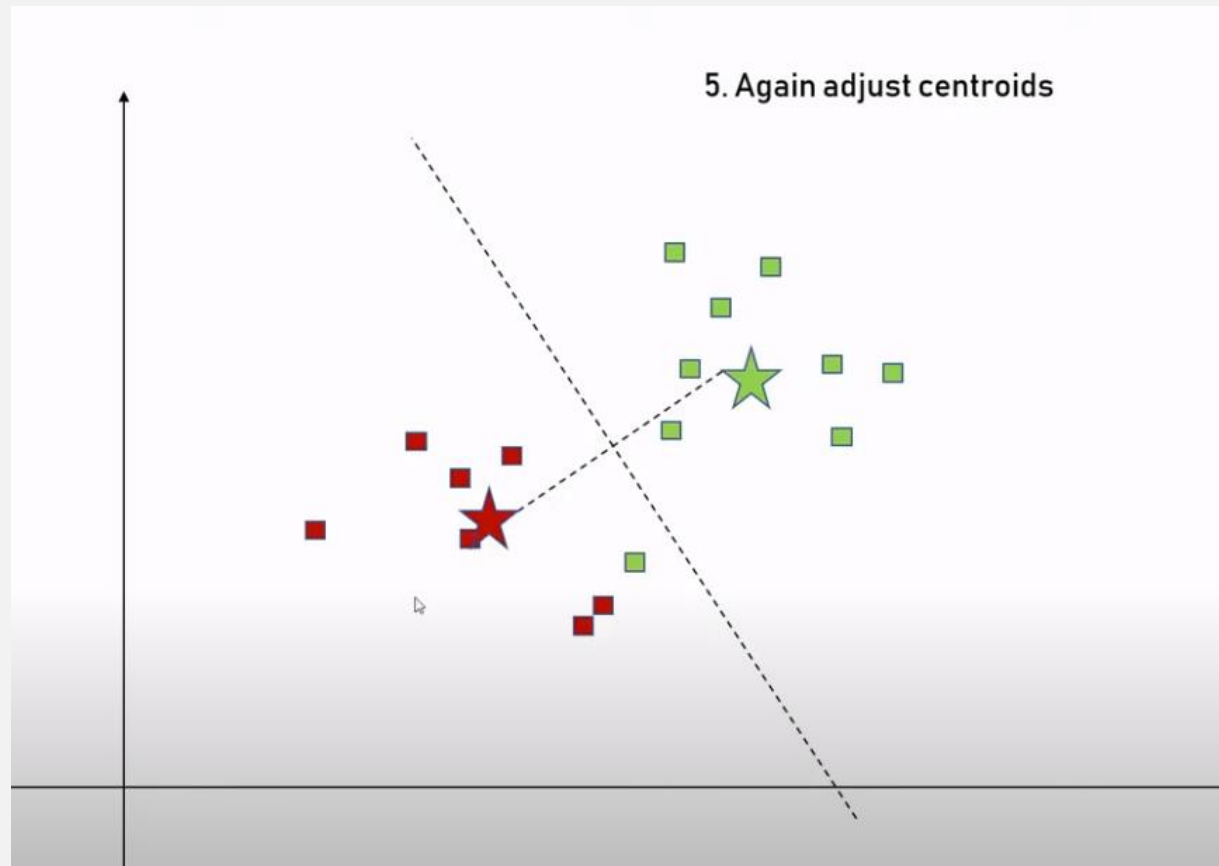
How does K-Means Works ?



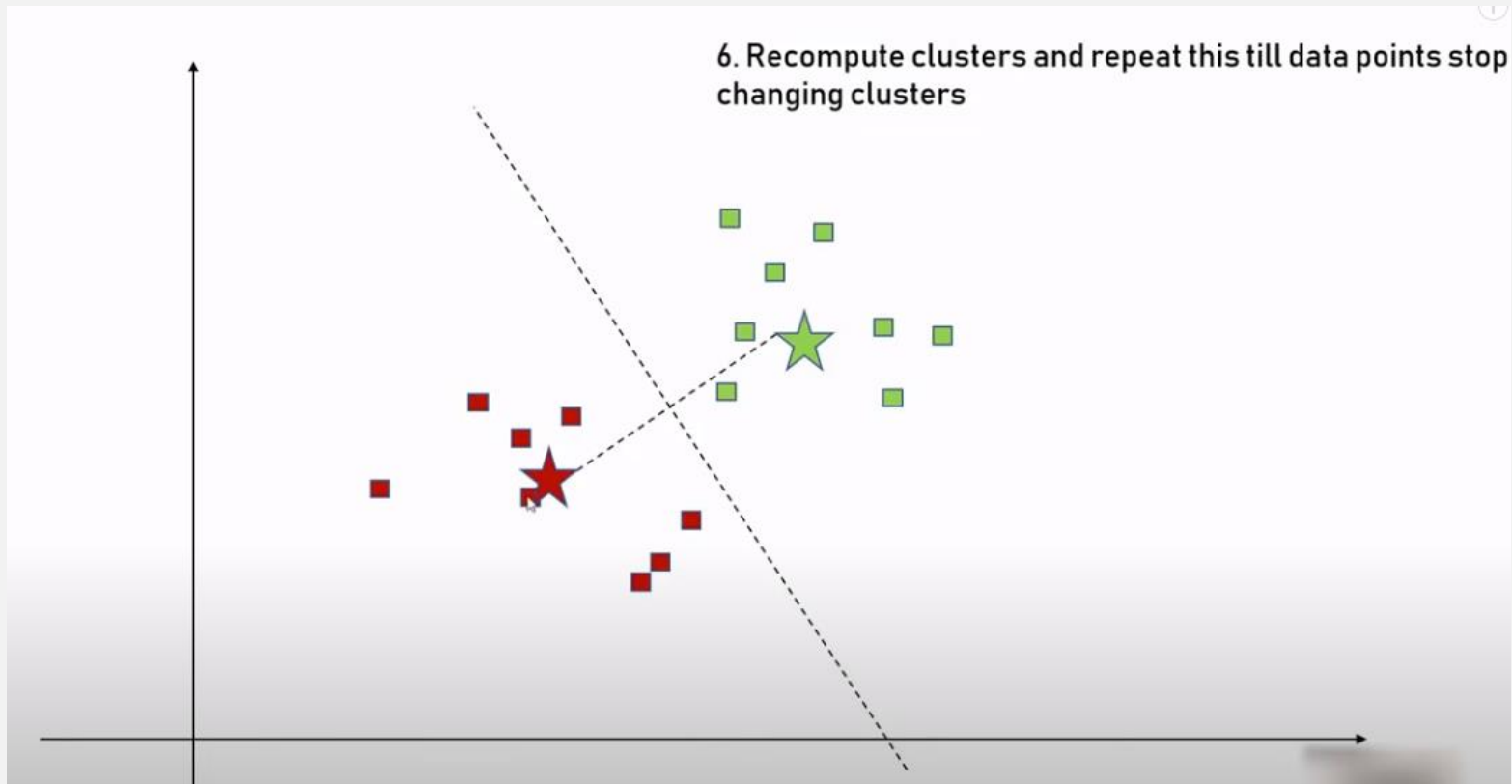
How does K-Means Works ?



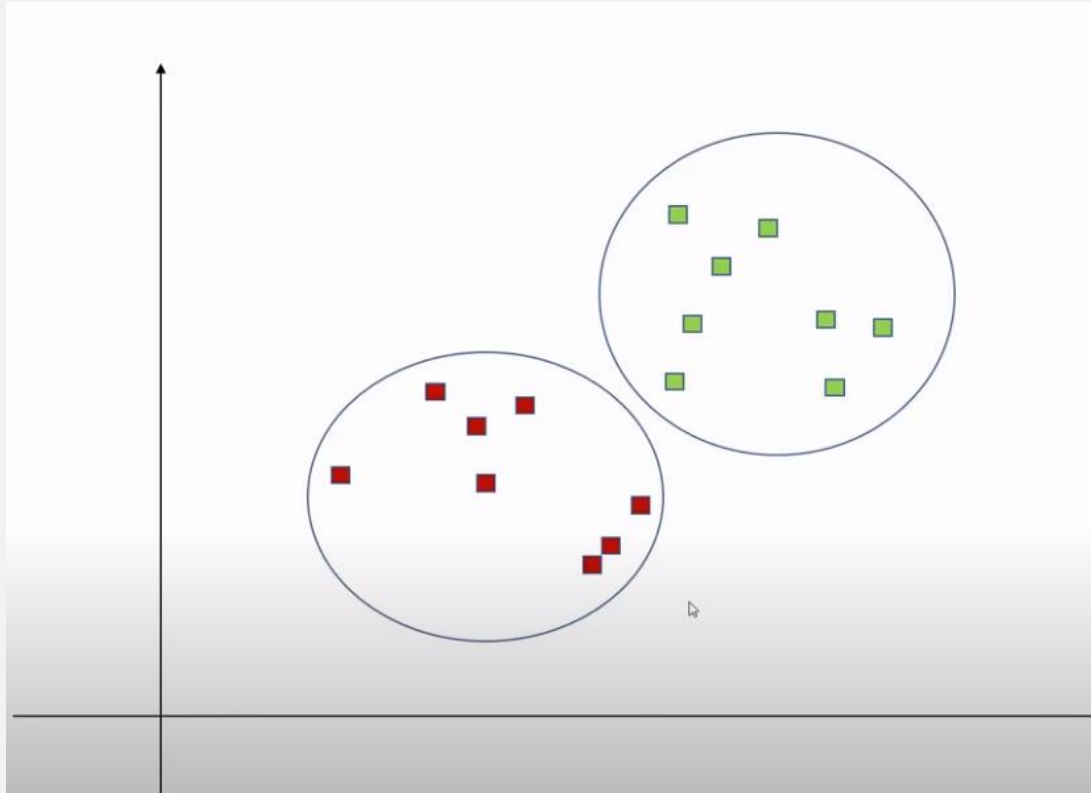
How does K-Means Works ?



How does K-Means Works ?



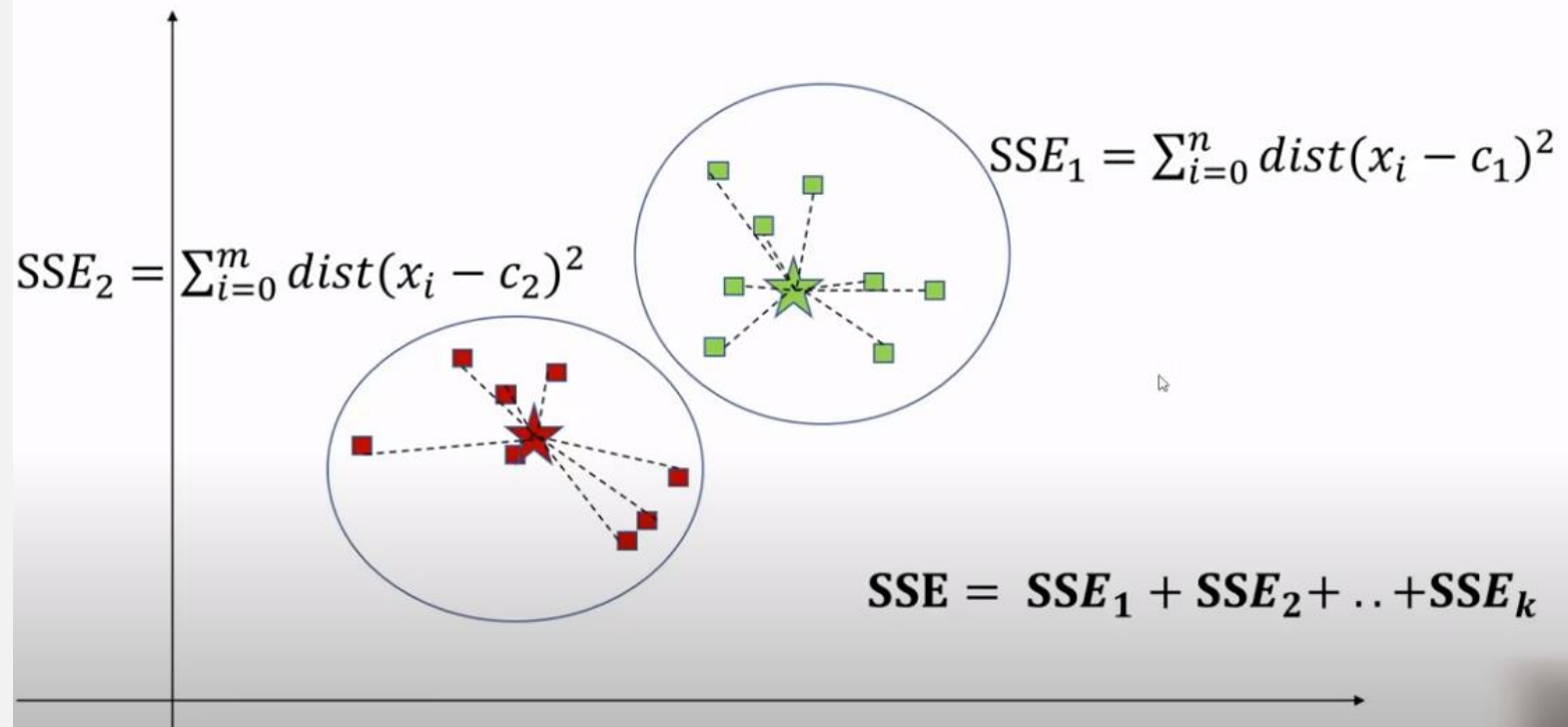
How does K-Means Works ?



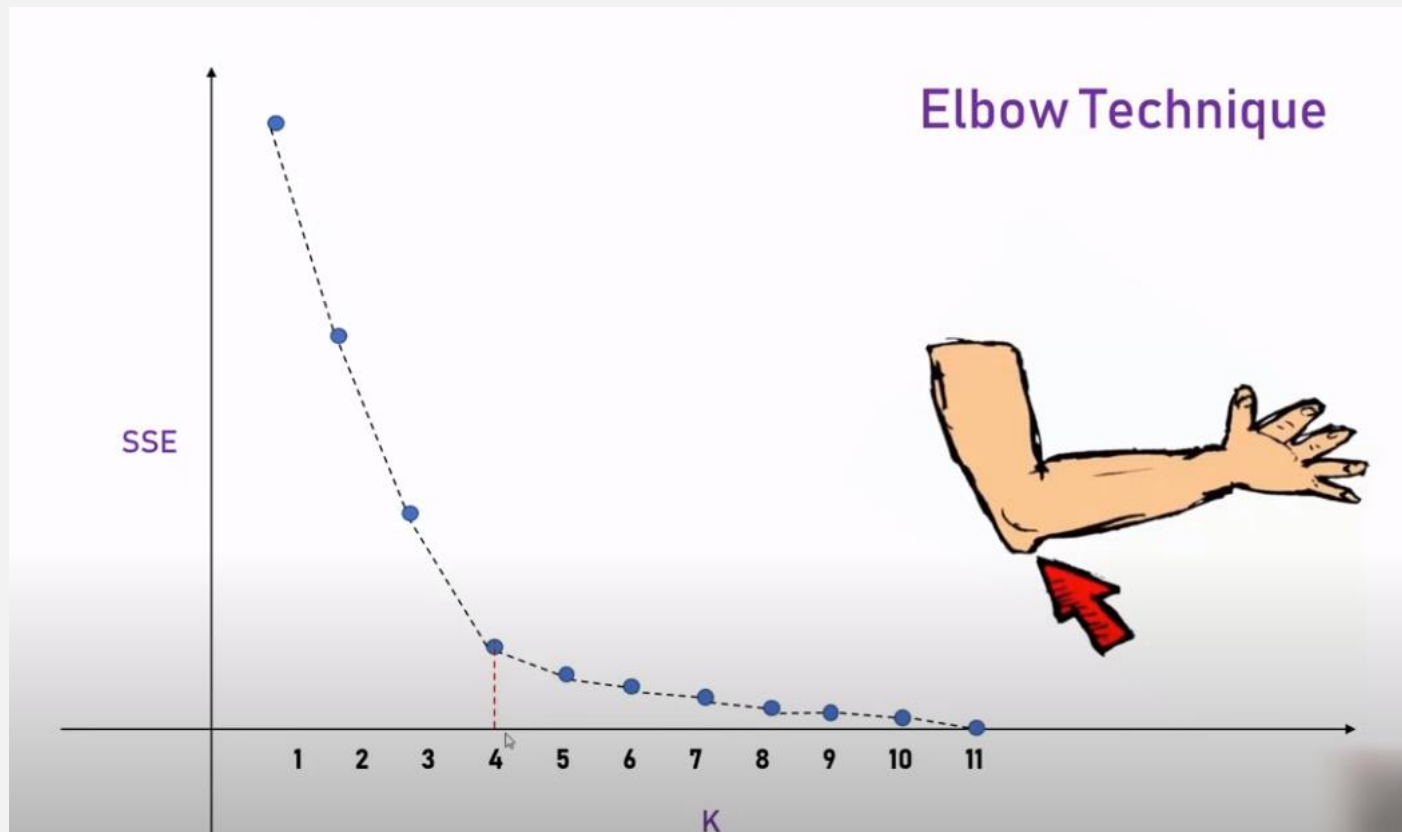
How to determine Correct k ?



SSE = Sum of Squared Errors



How to determine Correct k ?





Python – Hierarchical Clustering

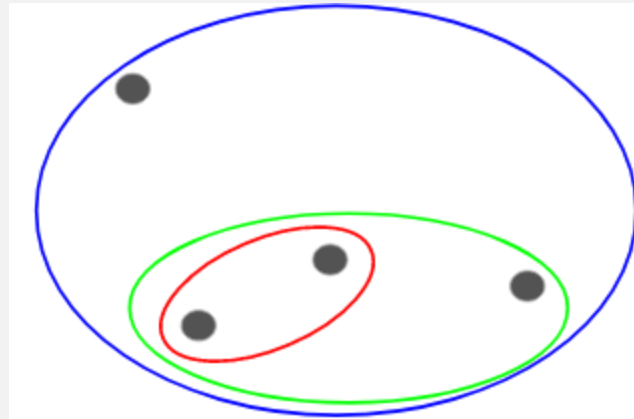
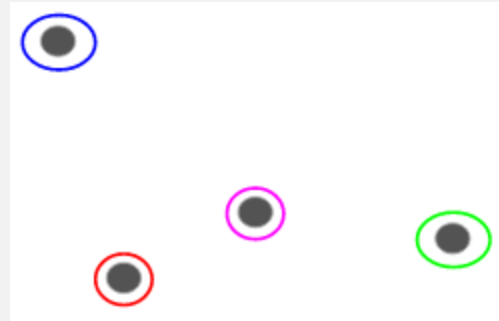
Hierarchical Clustering



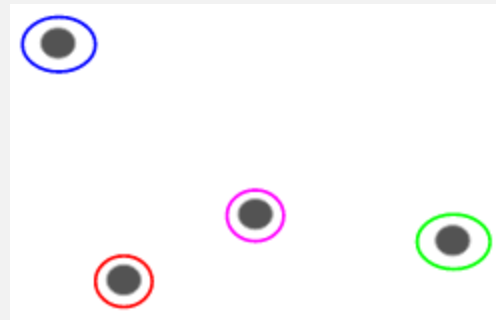
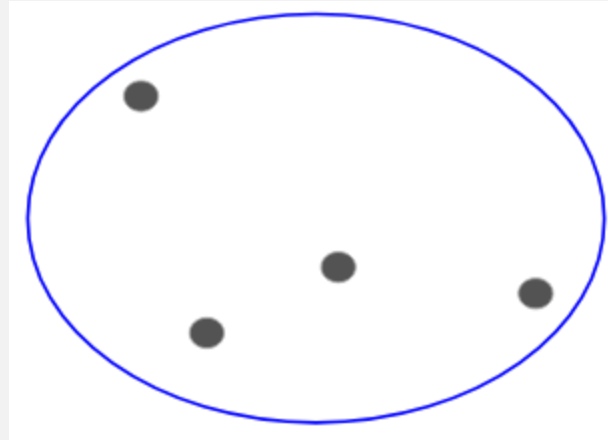
- **Hierarchical clustering is a type of unsupervised machine learning algorithm used to cluster unlabeled data points.**
- **Types of Hierarchical Clustering**
 - **Agglomerative hierarchical clustering**
 - **Divisive hierarchical clustering**



Agglomerative Hierarchical Clustering



Divisive Hierarchical Clustering



Steps to Perform Clustering



Student_ID	Marks
1	10
2	7
3	28
4	20
5	35

ID	1	2	3	4	5
1	0	3	18	10	25
2	3	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0



Steps to Perform Clustering



ID	1	2	3	4	5
1	0	3	18	10	25
2	3	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0

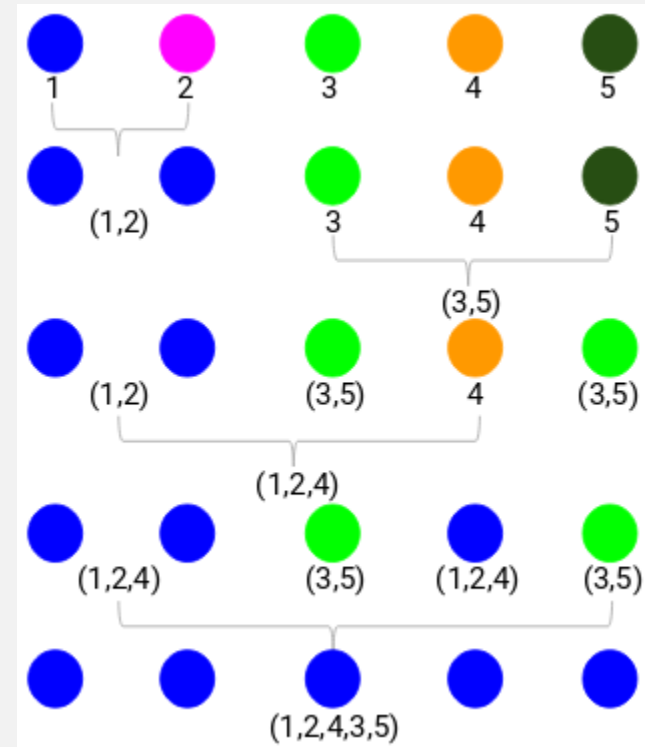


Steps to Perform Clustering



Student_ID	Marks
(1,2)	10
3	28
4	20
5	35

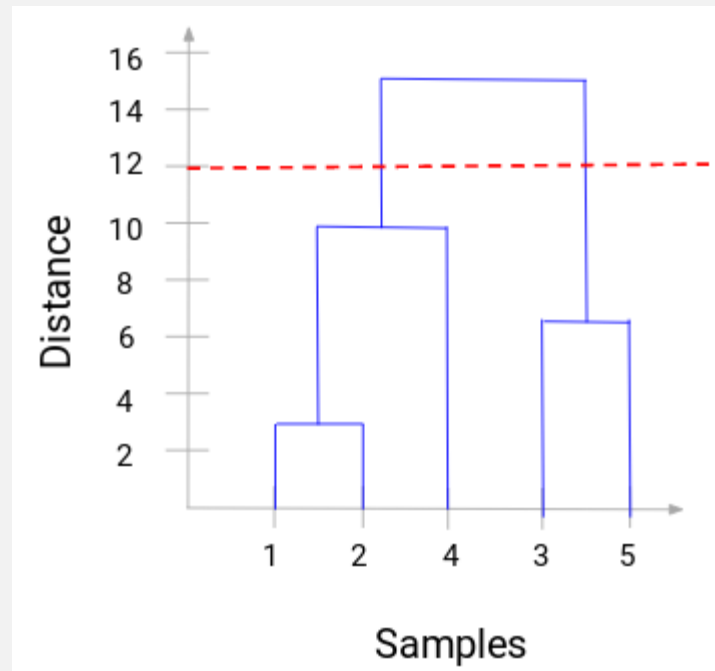
ID	(1,2)	3	4	5
(1,2)	0	18	10	25
3	18	0	8	7
4	10	8	0	15
5	25	7	15	0



Choosing No Of Clusters



A dendrogram is a tree-like diagram that records the sequences of merges or splits.





Python – Apriori

Apriori Algorithm



- **Apriori algorithm** is a machine learning model used in Association Rule Learning to identify frequent itemsets from a dataset
- With the help of these association rule, it determines how strongly or how weakly two objects are connected
- Apriori algorithm assumes that any subset of a frequent itemset must be frequent.

Say, a transaction containing {wine, chips, bread} also contains {wine, bread}. So, according to the principle of Apriori, if {wine, chips, bread} is frequent, then {wine, bread} must also be frequent.



Apriori Algorithm



Support : *Fraction of transactions that contain an itemset.*

For example, the support of item I is defined as the number of transactions containing I divided by the total number of transactions.

$$\text{support}(I) = \frac{\text{Number of transactions containing } I}{\text{Total number of transactions}}$$

Confidence : *Measures how often items in Y appear in transactions that contain X*























Confidence is the likelihood that item Y is also bought if item X is bought. It's calculated as the number of transactions containing X and Y divided by the number of transactions containing X.

$$\text{confidence}(X \rightarrow Y) = \frac{\text{Number of transactions containing } X \text{ and } Y}{\text{Number of transactions containing } X}$$



Apriori Algorithm - Support



Transaction 1	   
Transaction 2	  
Transaction 3	 
Transaction 4	 
Transaction 5	   
Transaction 6	  
Transaction 7	 
Transaction 8	 

Measure 1: Support.

This says how popular an itemset is, as measured by the proportion of transactions in which an itemset appears.

In Table the support of {apple} is 4 out of 8, or 50%.

Itemsets can also contain multiple items. For instance, the support of {apple, beer, rice} is 2 out of 8, or 25%.



Apriori Algorithm - Confidence



$$\text{Confidence } \{\text{🍏} \rightarrow \text{🍺}\} = \frac{\text{Support } \{\text{🍏}, \text{🍺}\}}{\text{Support } \{\text{🍏}\}}$$

Measure 2: Confidence.

This says how likely item Y is purchased when item X is purchased, expressed as {X -> Y}.

This is measured by the proportion of transactions with item X, in which item Y also appears. In Table , the confidence of {apple -> beer} is 3 out of 4, or 75%.



Apriori Algorithm - Lift



$$\text{Lift} \{ \text{🍏} \rightarrow \text{🍺} \} = \frac{\text{Support} \{ \text{🍏}, \text{🍺} \}}{\text{Support} \{ \text{🍏} \} \times \text{Support} \{ \text{🍺} \}}$$

Measure 3: Lift.

This says how likely item Y is purchased when item X is purchased, while controlling for how popular item Y is. In Table 1, the lift of {apple -> beer} is 1, which implies no association between items.

A lift value greater than 1 means that item Y is likely to be bought if item X is bought, while a value less than 1 means that item Y is unlikely to be



Apriori Algorithm Example



TID	ITEMSETS
T1	A, B
T2	B, D
T3	B, C
T4	A, B, D
T5	A, C
T6	B, C
T7	A, C
T8	A, B, C, E
T9	A, B, C

Given: Minimum Support= 2, Minimum Confidence= 50%



Apriori Algorithm Example



In the first step, we will create a table that contains support count (The frequency of each itemset individually in the dataset) of each itemset in the given dataset. This table is called the **Candidate set or C1**.

Itemset	Support_Count
A	6
B	7
C	5
D	2
E	1

Itemset	Support_Count
A	6
B	7
C	5
D	2



Apriori Algorithm Example



In this step, we will generate C2 with the help of L1. In C2, we will create the pair of the itemsets of L1 in the form of subsets. After creating the subsets, we will again find the support count from the main transaction table of datasets, i.e., how many times these pairs have occurred together in the given dataset. So, we will get the below table for C2

Itemset	Support_Count
{A, B}	4
{A,C}	4
{A, D}	1
{B, C}	4
{B, D}	2
{C, D}	0

Itemset	Support_Count
{A, B}	4
{A, C}	4
{B, C}	4
{B, D}	2

A, B, C, D



Apriori Algorithm Example



For C3, we will repeat the same two processes, but now we will form the C3 table with subsets of three itemsets together, and will calculate the support count from the dataset. It will give the below table:

Itemset	Support_Count
{A, B, C}	2
{B, C, D}	1
{A, C, D}	0
{A, B, D}	0



Apriori Algorithm Example



As the given threshold or minimum confidence is 50%, so the first three rules **$A \wedge B \rightarrow C$** , **$B \wedge C \rightarrow A$** , and **$A \wedge C \rightarrow B$** can be considered as the strong association rules for the given problem.

Rules	Support	Confidence
$A \wedge B \rightarrow C$	2	$\text{Sup}\{(A \wedge B) \wedge C\} / \text{sup}(A \wedge B) = 2/4 = 0.5 = 50\%$
$B \wedge C \rightarrow A$	2	$\text{Sup}\{(B \wedge C) \wedge A\} / \text{sup}(B \wedge C) = 2/4 = 0.5 = 50\%$
$A \wedge C \rightarrow B$	2	$\text{Sup}\{(A \wedge C) \wedge B\} / \text{sup}(A \wedge C) = 2/4 = 0.5 = 50\%$
$C \rightarrow A \wedge B$	2	$\text{Sup}\{(C \wedge (A \wedge B))\} / \text{sup}(C) = 2/5 = 0.4 = 40\%$
$A \rightarrow B \wedge C$	2	$\text{Sup}\{(A \wedge (B \wedge C))\} / \text{sup}(A) = 2/6 = 0.33 = 33.33\%$
$B \rightarrow B \wedge C$	2	$\text{Sup}\{(B \wedge (B \wedge C))\} / \text{sup}(B) = 2/7 = 0.28 = 28\%$



THANK YOU !!!

Amol Patil - 9822291613

