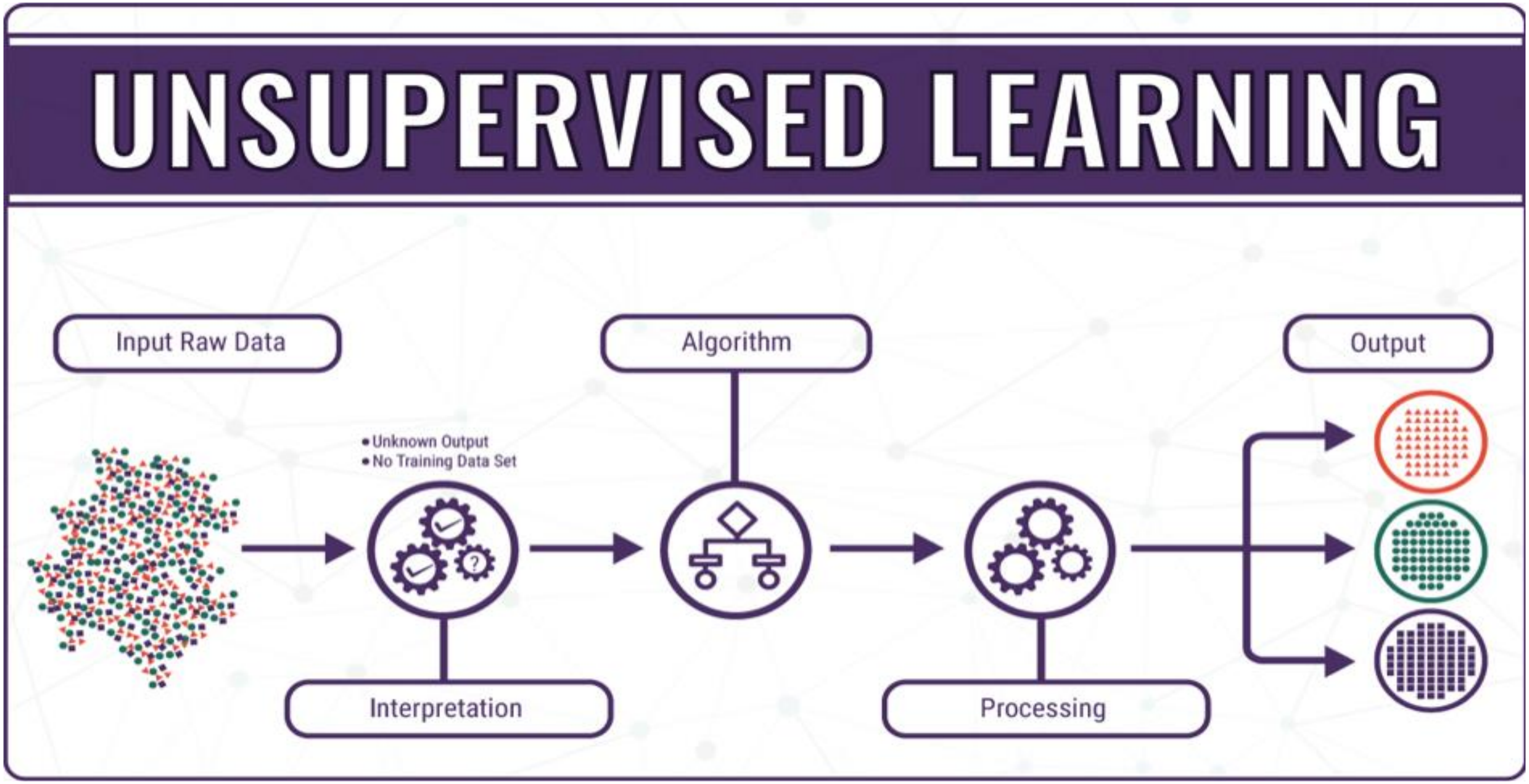
Unsupervised Learning

By

Prof(Dr) Premanand P Ghadekar

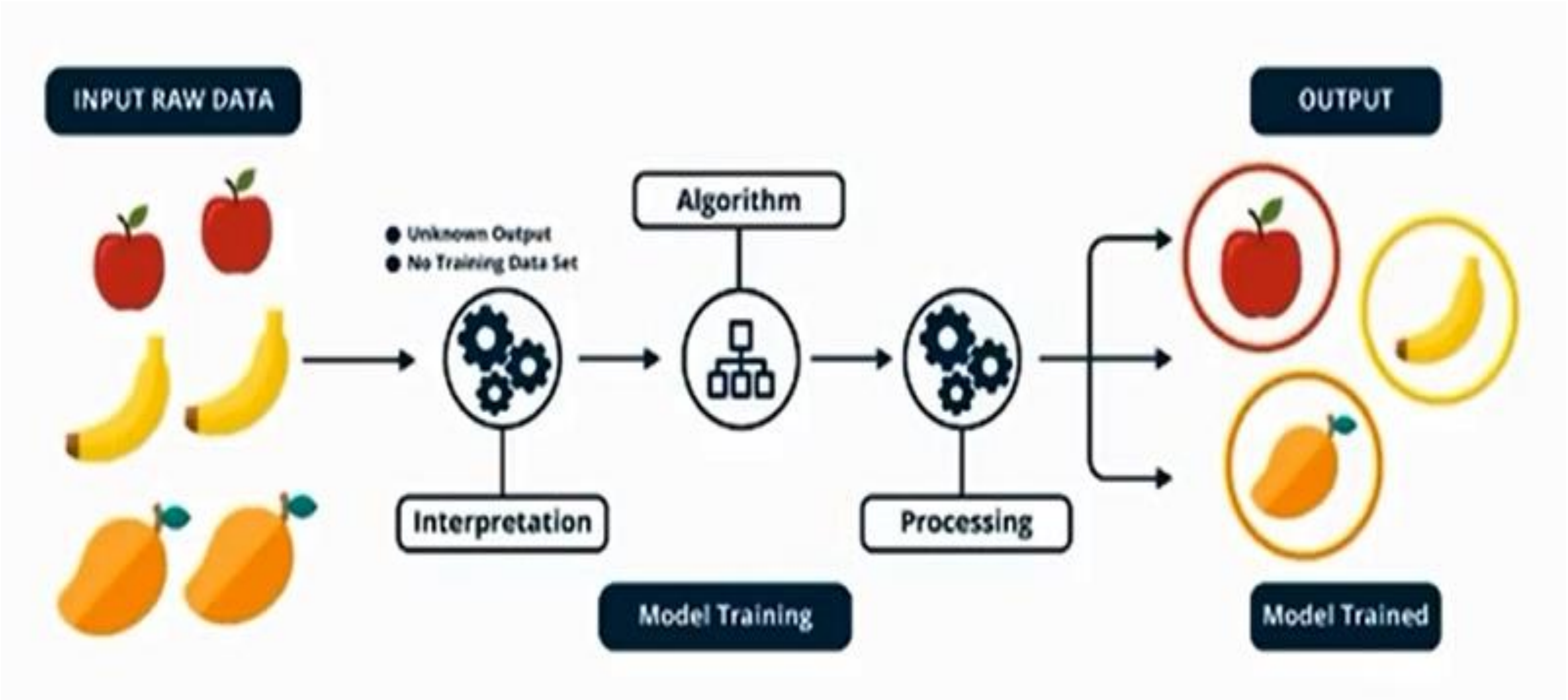


Unsupervised Learning

- K-Means Clustering
- C-Means Clustering
- Associated Rule Mining

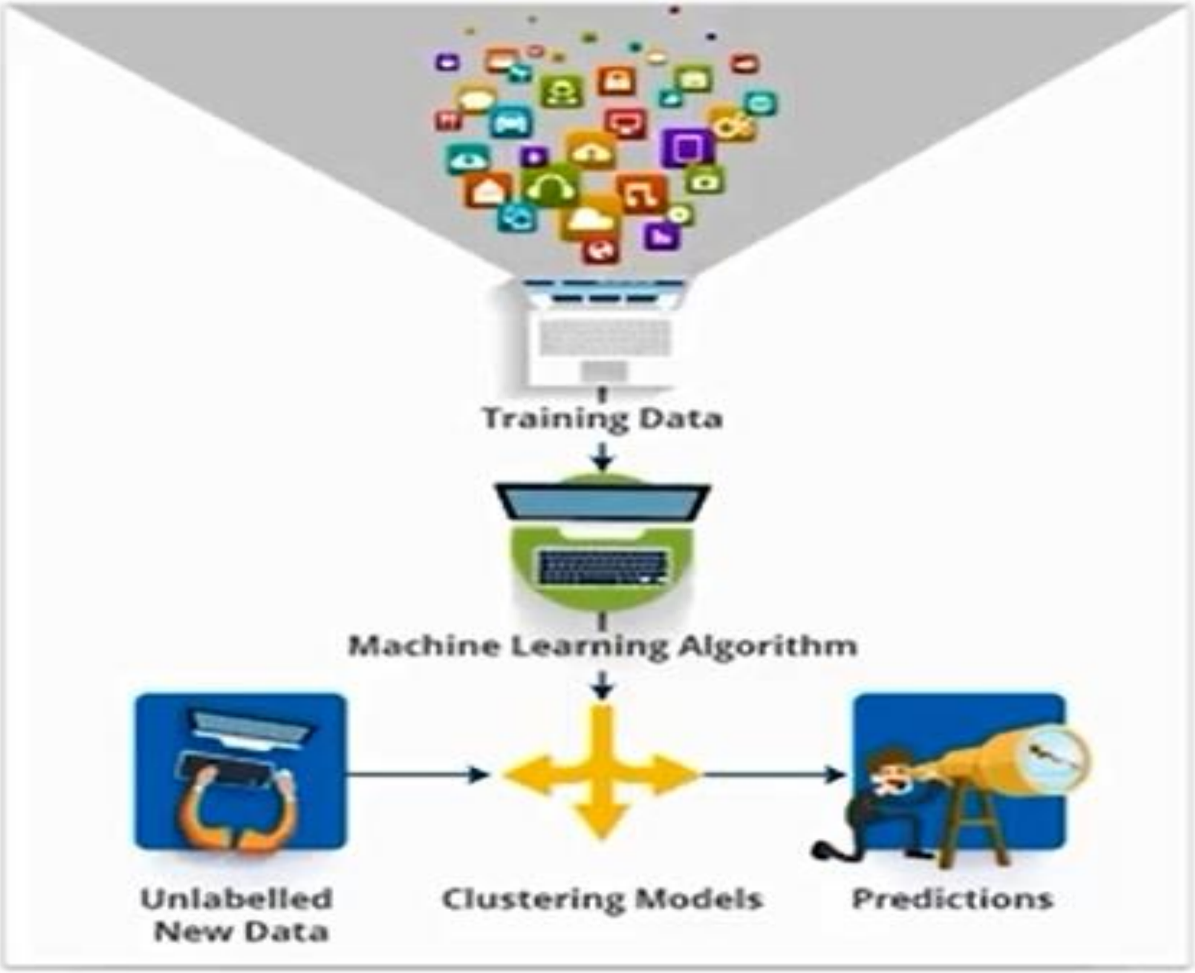
Unsupervised Learning

Unsupervised Learning is a type of Machine Learning Algorithm used to draw inferences From data sets consisting of input data without labelled responses



Unsupervised Learning-Process Flow

Training Data is collections of information without any label



What is Clustering

Clustering is the process of dividing the datasets into groups consisting of similar data points.

- It merge grouping of objects based on the information found in the data, describing objects or their relationship.
- Points in the same groups are as similar as possible.
- Points in different groups are as dissimilar as possible.

Why is Clustering used

The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data.

Sometimes Partitioning is the goal.

Where it is used

Retail Store

Banking

Insurance Companies

amazon

NETFLIX

Recommended Movies

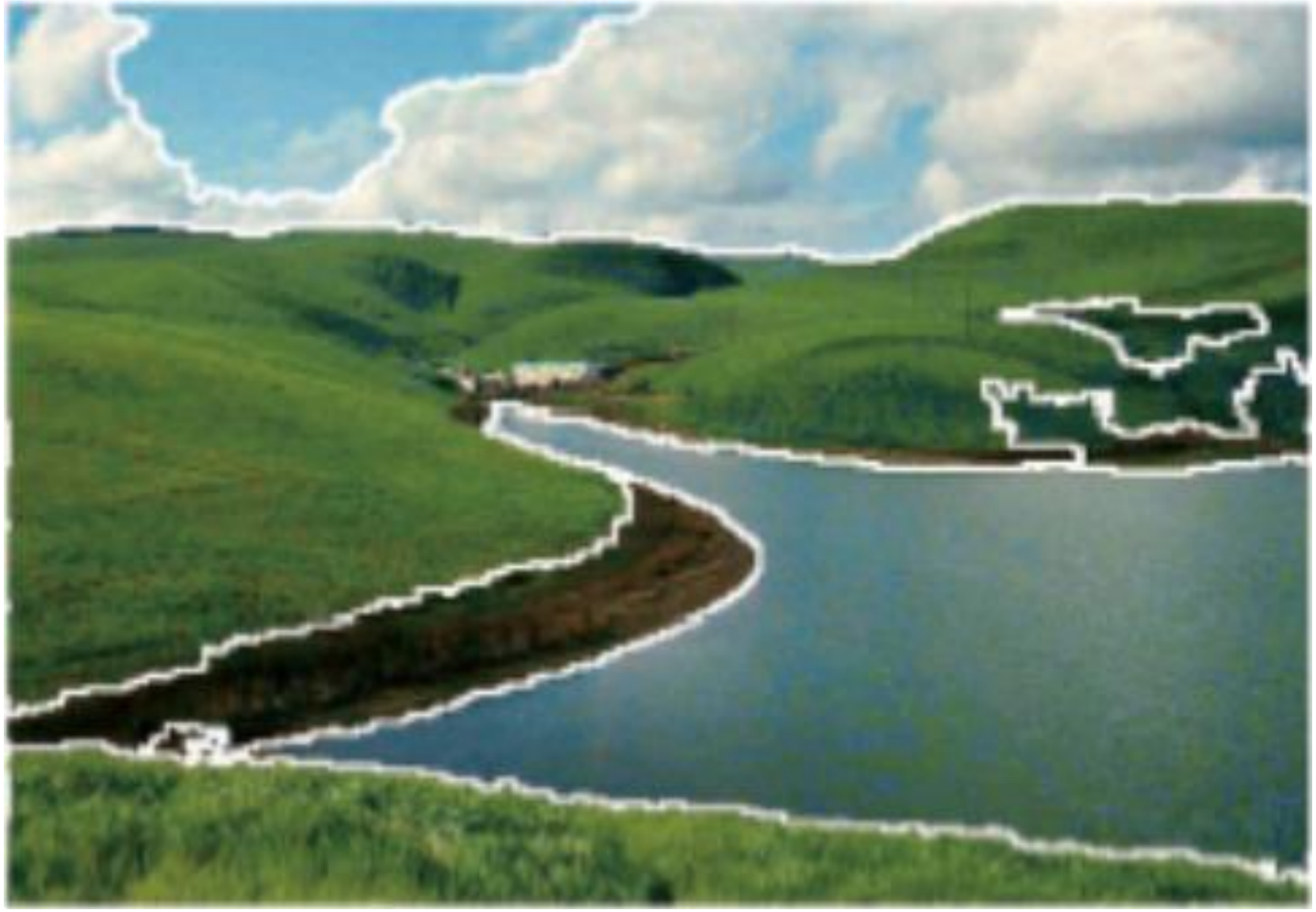
flickr

Flickr's Photos

Clustering Example

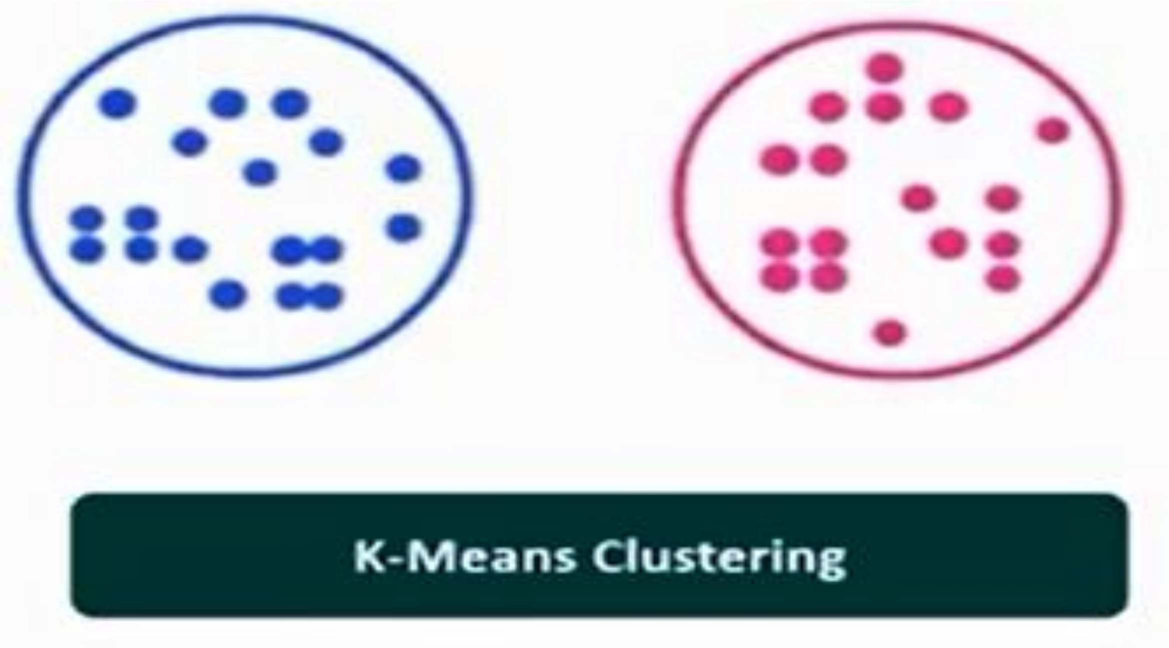
Image segmentation

Goal: Break up the image into meaningful or perceptually similar regions



Types of Clustering

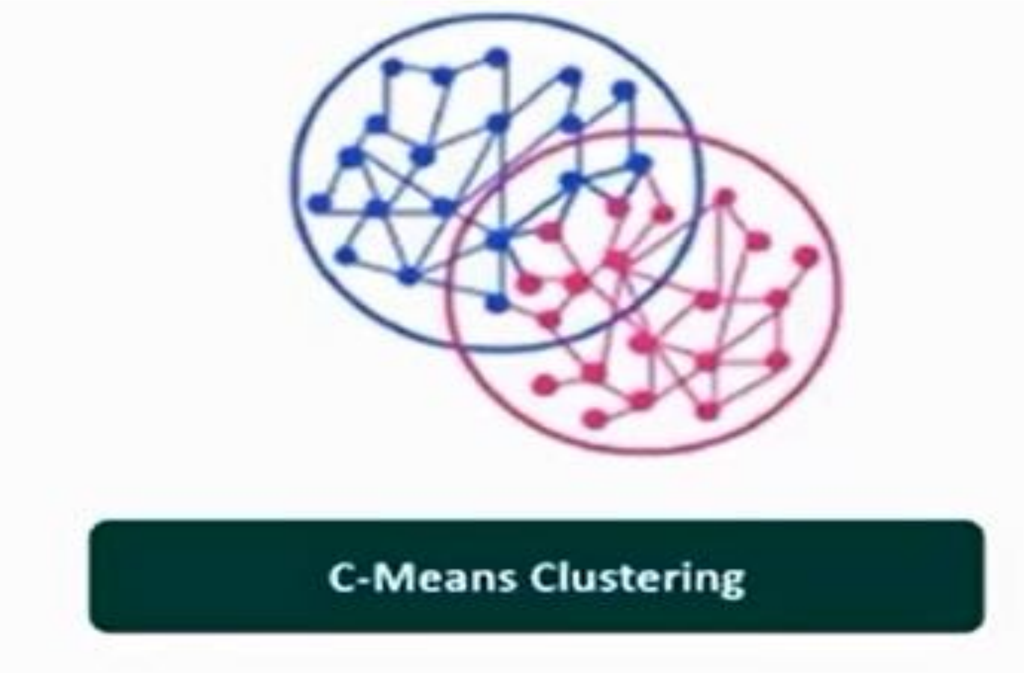
- **Exclusive Clustering-K-Means** Clustering



Division of Objects into clusters such that each object is in exactly **one cluster not Several clusters**

Types of Clustering

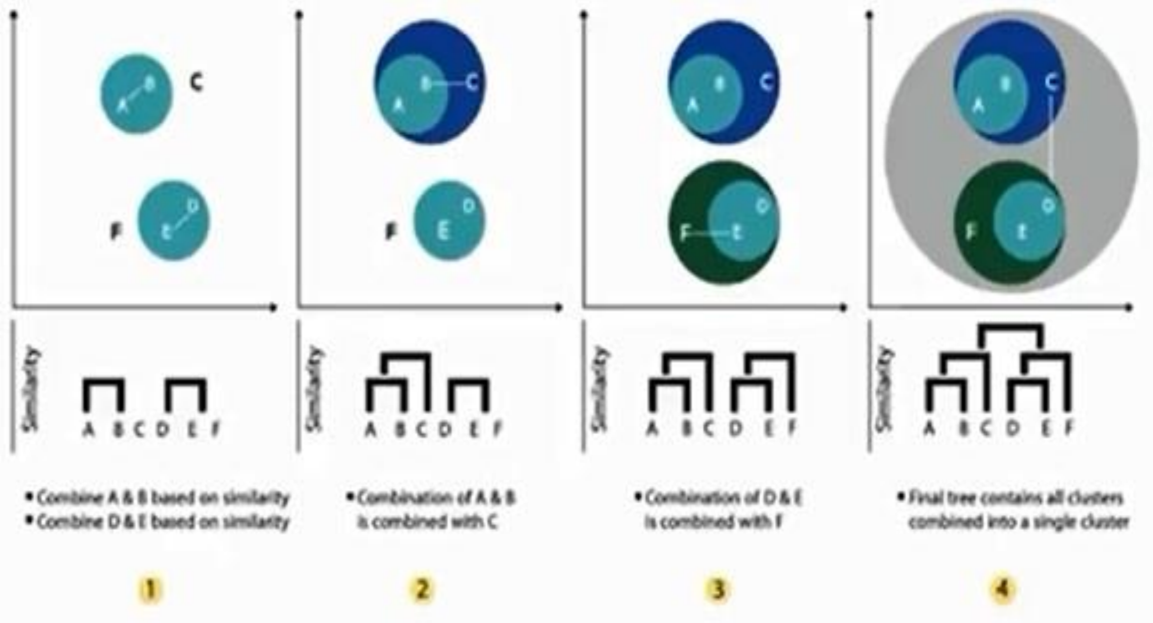
- **Overlapping Clustering-C Means Clustering**



C-Means Clustering- Division of Objects into clusters such that each object can belongs to Multiple clusters

Types of Clustering

- Exclusive Clustering
- Overlapping Clustering
- **Hierarchical Clustering- Agglomerative and Divisive**


C
A
B
D
E
F
Similarity
A B C D E F
• Combine A & B based on similarity
• Combine D & E based on similarity
• Combination of A & B is combined with C
• Combination of D & E is combined with F
• Final tree contains all clusters combined into a single cluster
1
2
3
4

Clusters have a tree like structure or a Parent Child Relationship

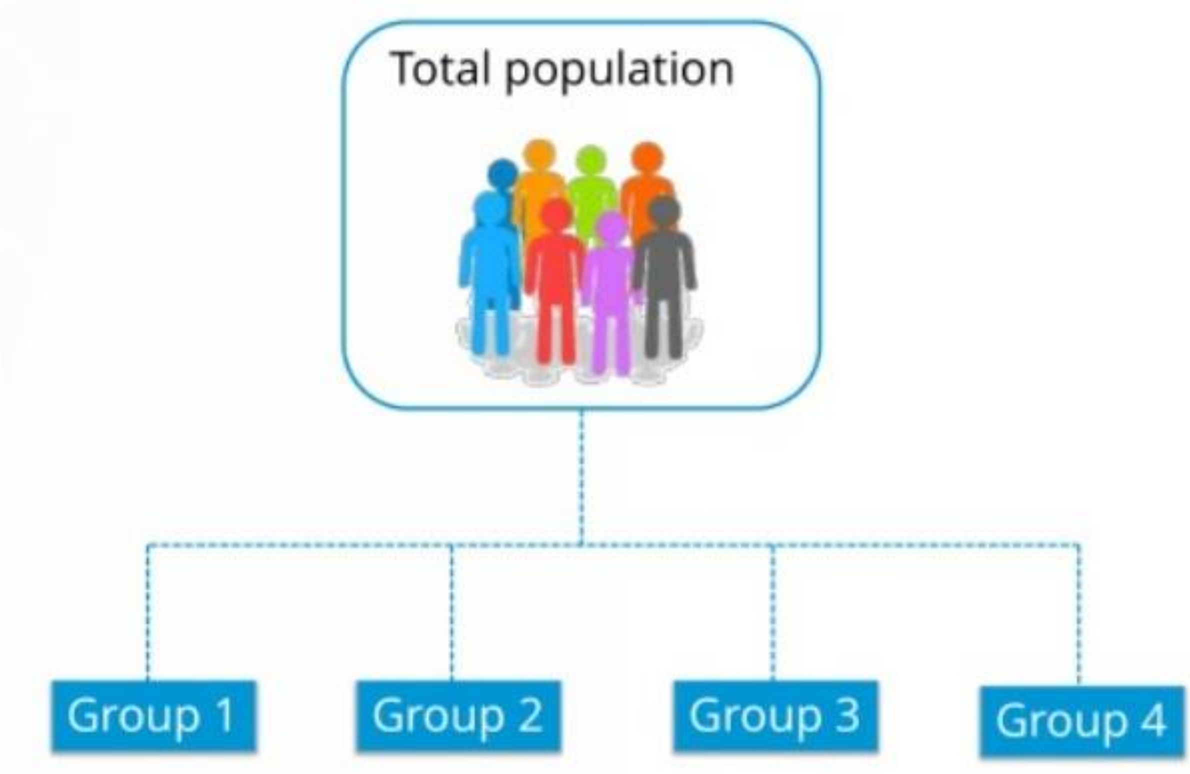
Agglomerative or Bottom up Approach -Begin with each element as a separate cluster and merge them into successively larger clusters.

Divisive or Top down Approach- Approach begin with the whole set and proceed to divide it into smaller clusters.

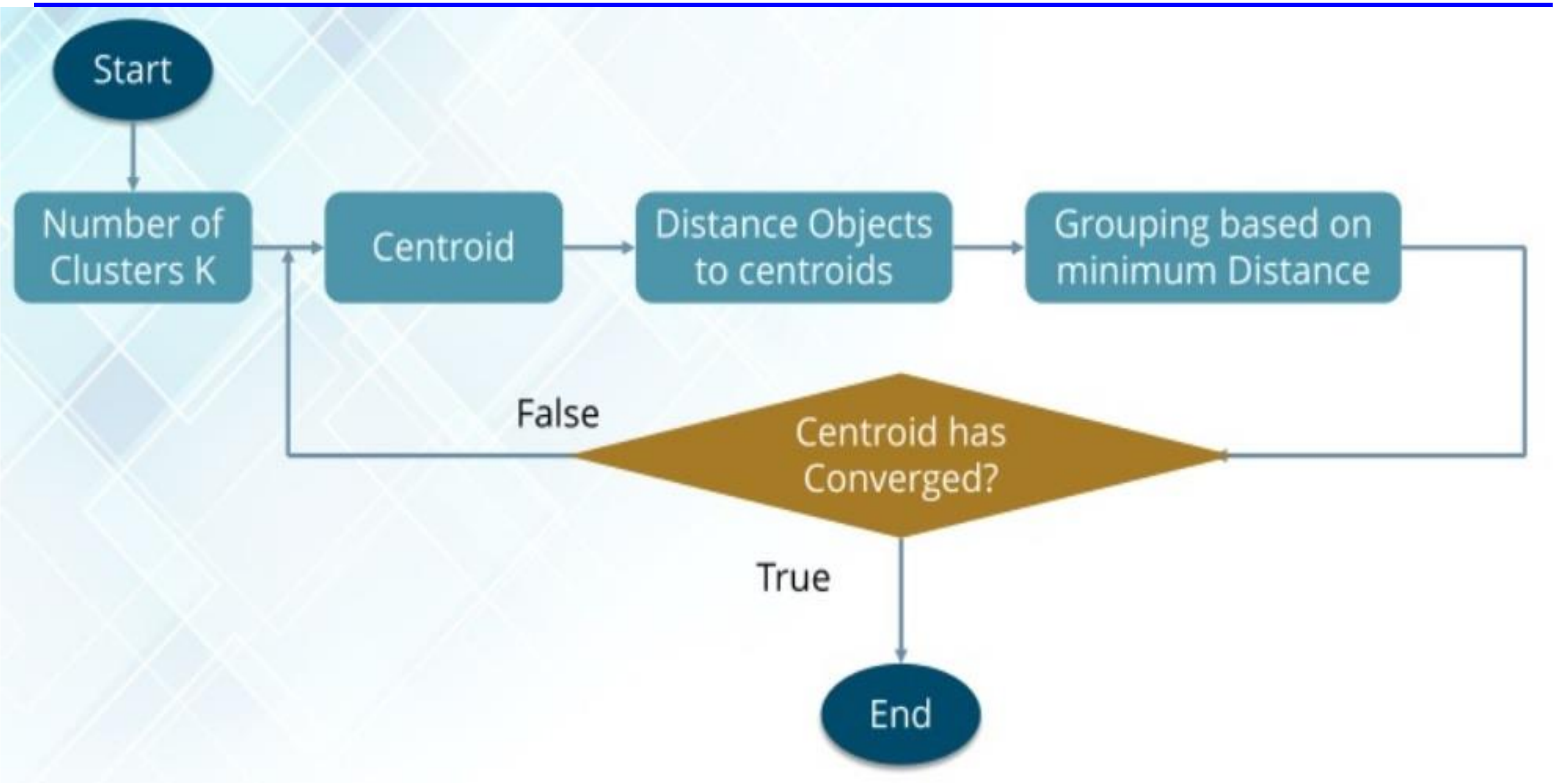
K-Means Clustering

The process by which objects are classified into a predefined number of groups so that they are **as much dissimilar as possible from one group to another group, but as much similar as possible within each group.**

‘K’ in K-Means represent t the number of Clusters



K-Means Algorithm WORKING

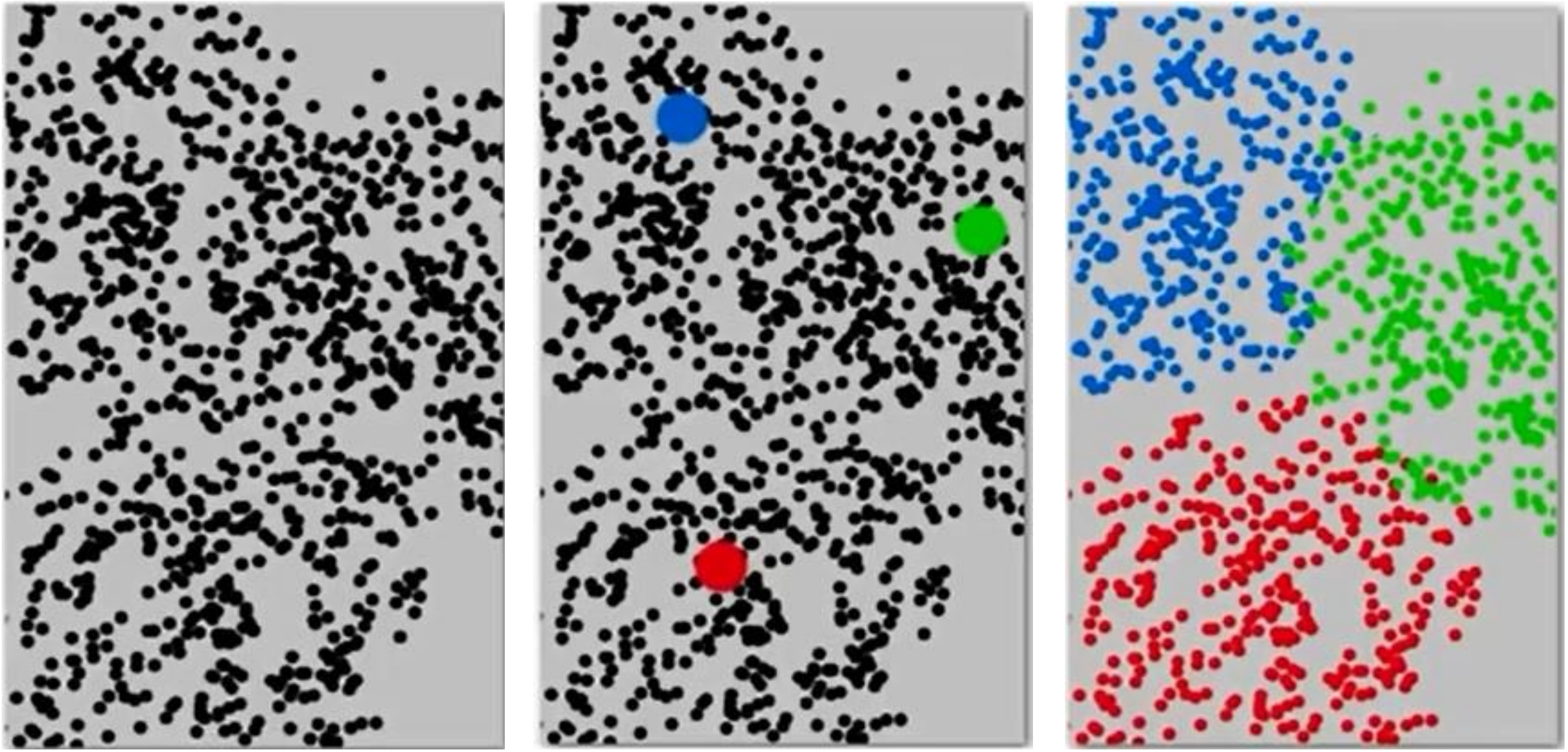


Distance Measure determines the similarity between two elements, and it influences the shape of the Clusters.

Euclidean Distance Measure, Manhattan Distance Measure, Squared Euclidean Distance Measure, Cosine distance measure

K-Means Clustering Steps

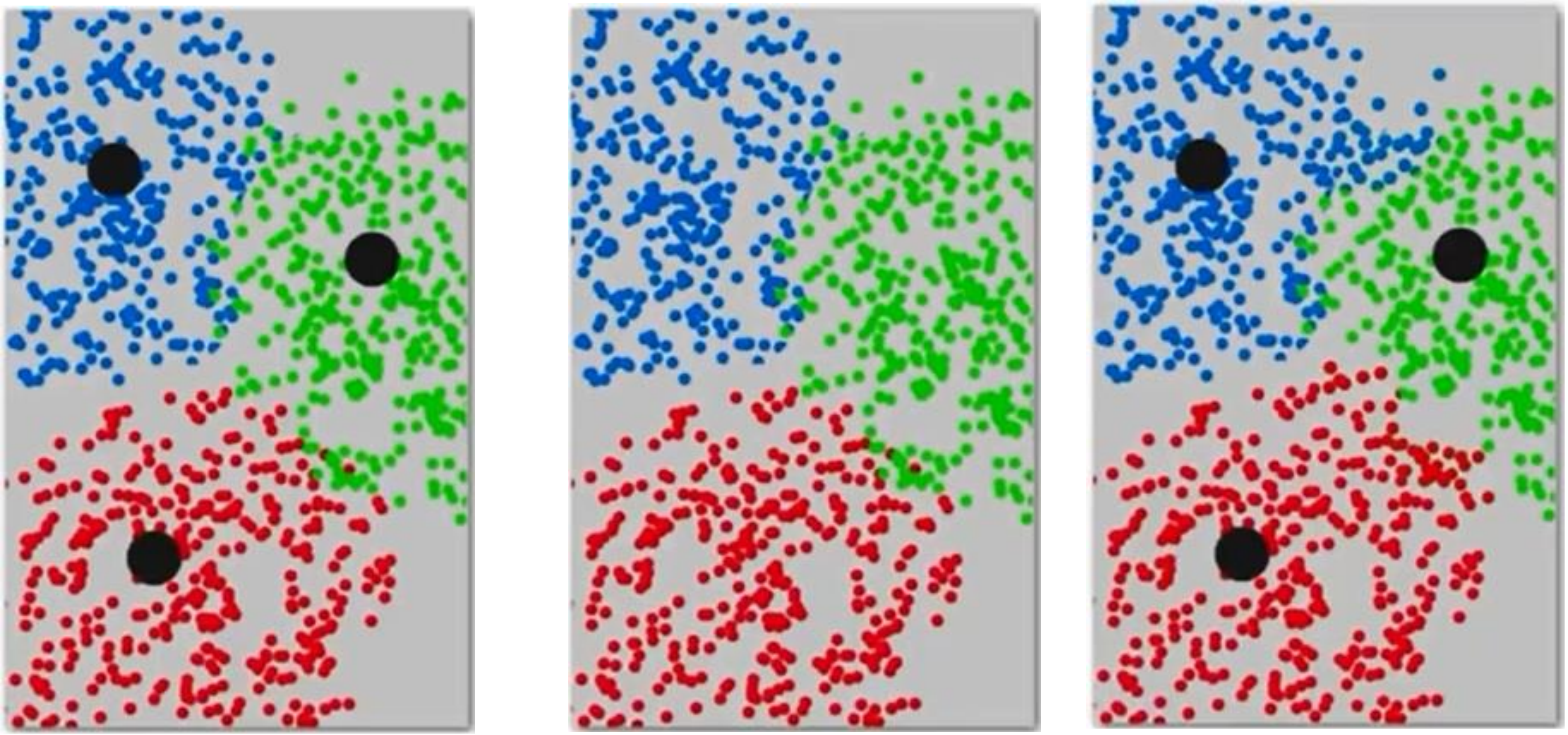
1. First, we need to decide number of clusters to be made (Guessing).
2. Then we provide centroid of all the clusters (Guessing)
3. The Algorithm calculates Euclidian distance of the points from each centroid and assign the point to the closest cluster.



(a) (b) (c)

K-Means Clustering Steps

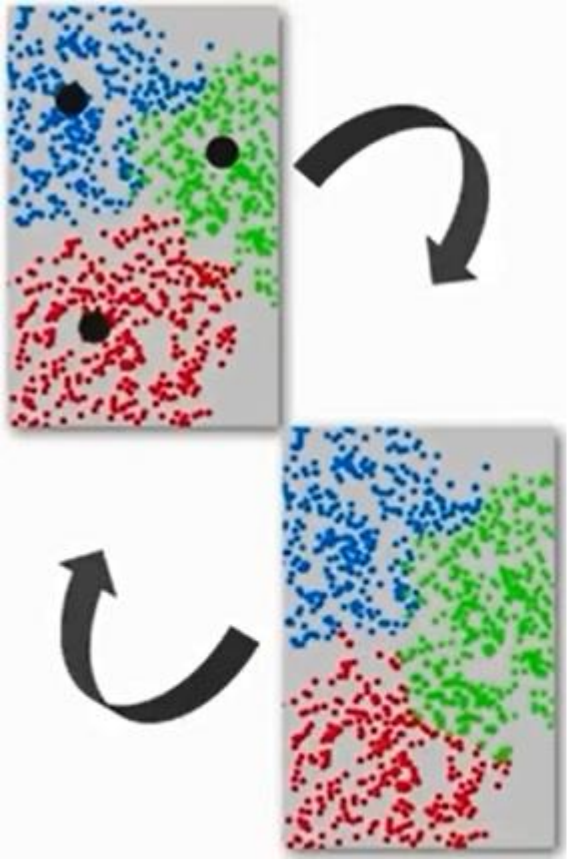
4. Next the Centroids are calculated again, when we have our new cluster.
5. The Distance of the points from the center of Clusters are calculated again and points are assigned to the closet cluster.
6. And then again, the new centroid for the cluster is calculated.



(a) (b) (c)

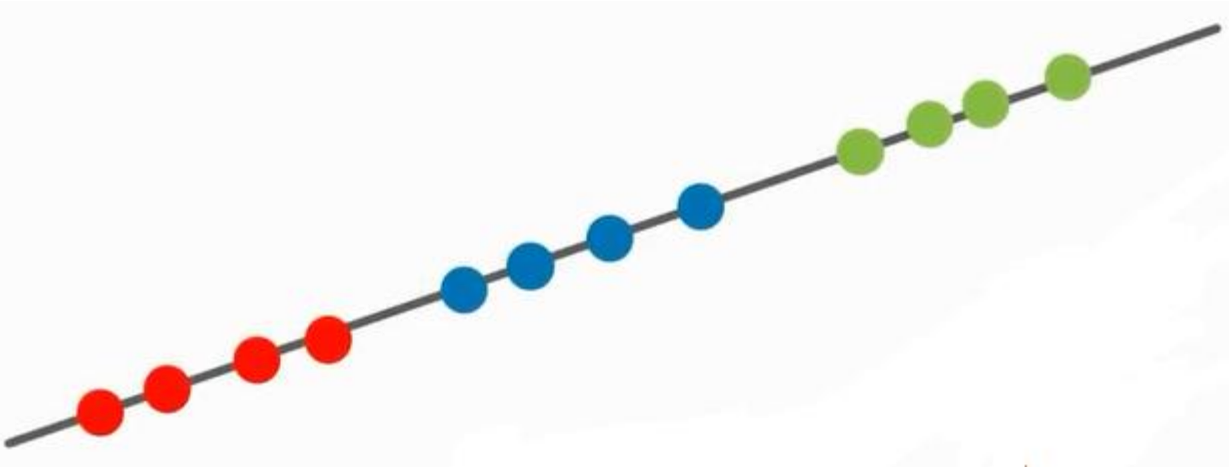
K-Means Clustering Steps

7. These steps are repeated until we have a repetition in Centroids or new centroids are very close to the previous one.



K-Means Clustering Algorithm

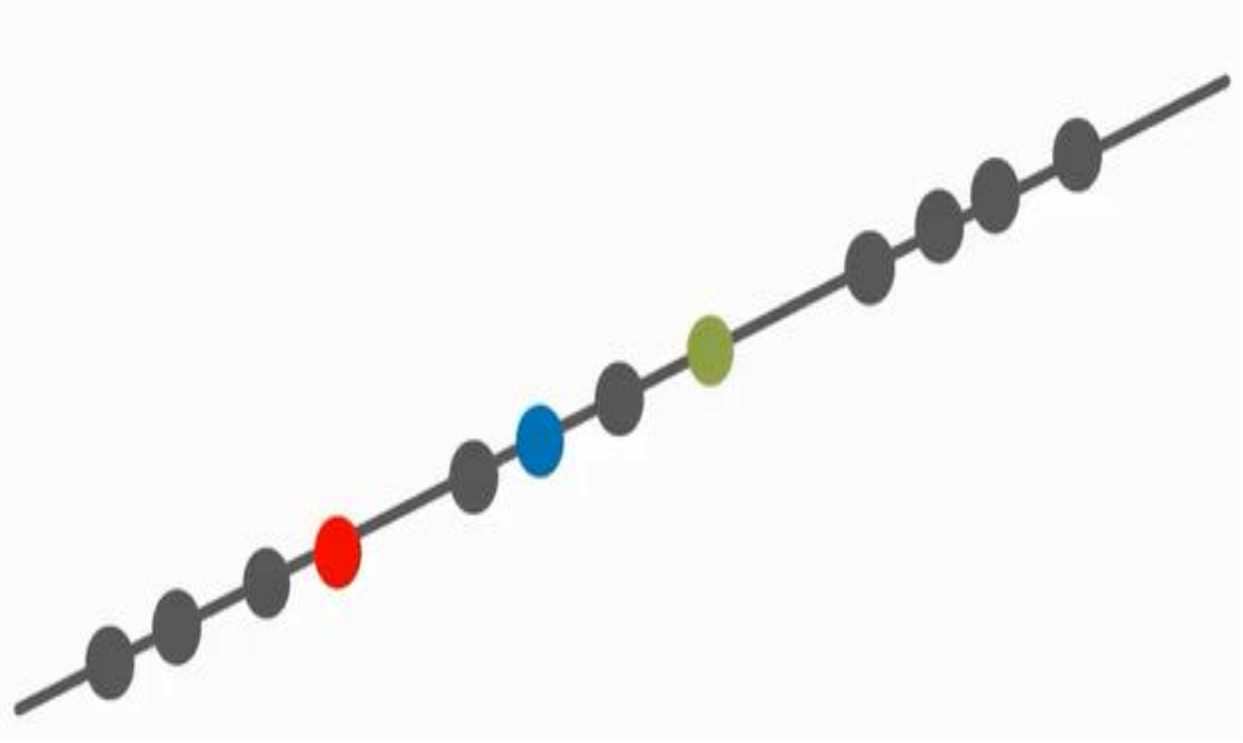
- Divide the point into three Clusters. Where K=3



K-Means Clustering Algorithm

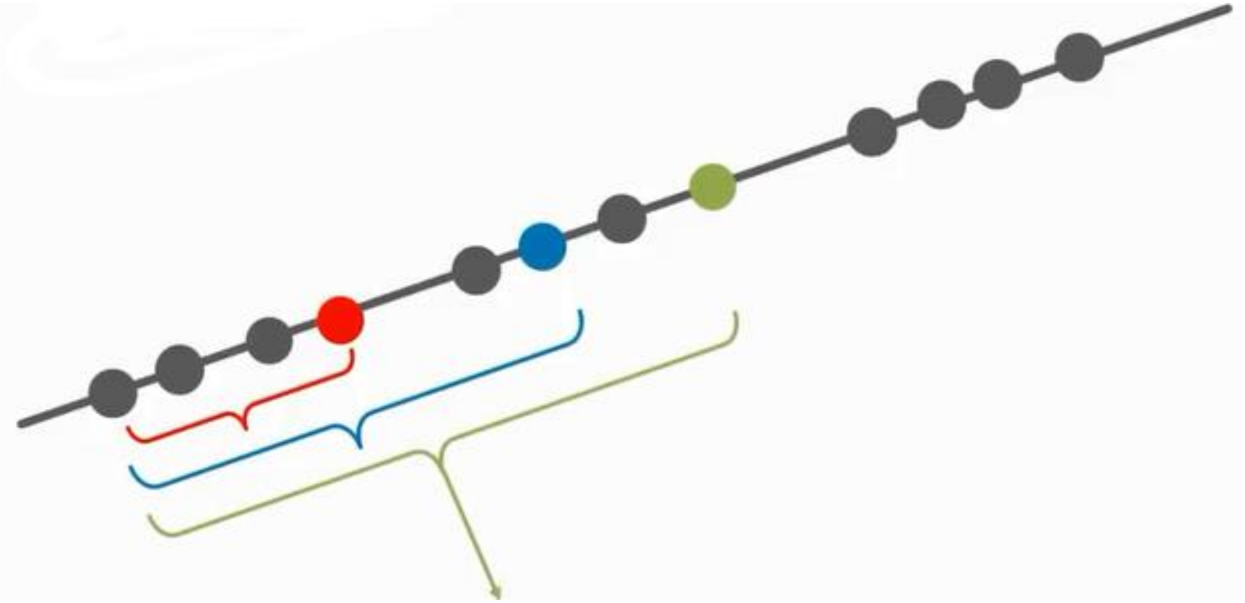
Step-1: Select the number of Clusters to be identified, i.e. select a value of k=3 in this case.

Step-2: Randomly select three distinct data points.



K-Means Clustering Algorithm

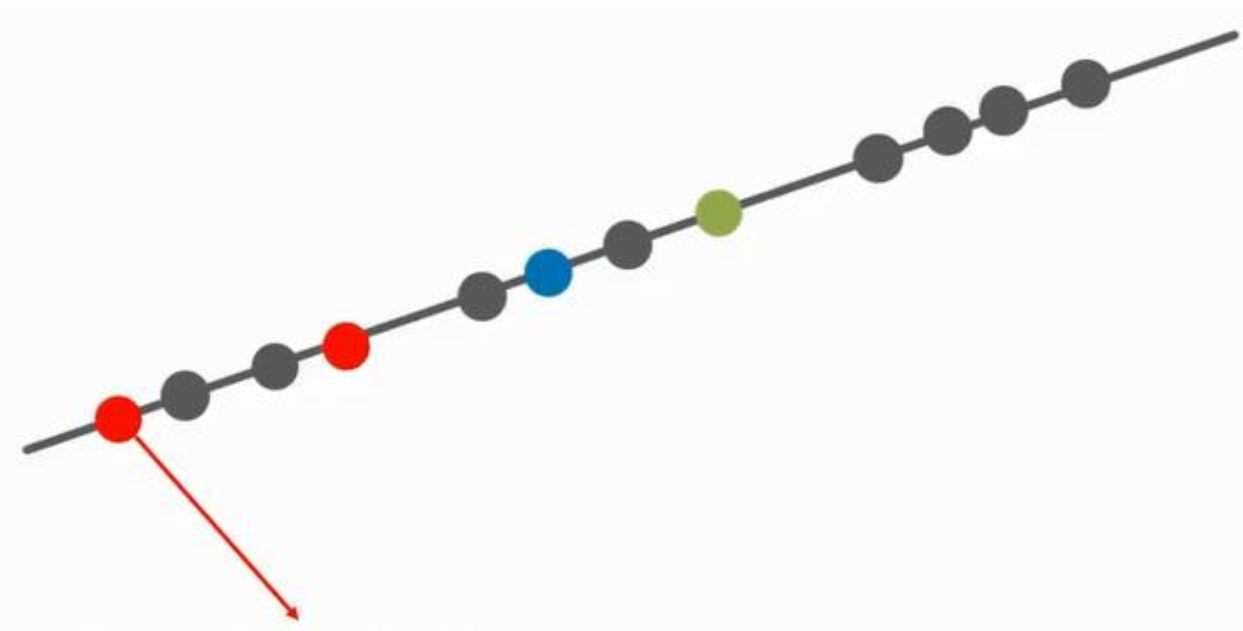
Step-3: Measure the distance between the 1st point and selected three Clusters.



Distance from 1st Point to Cluster 1, 2 and 3.

K-Means Clustering Algorithm

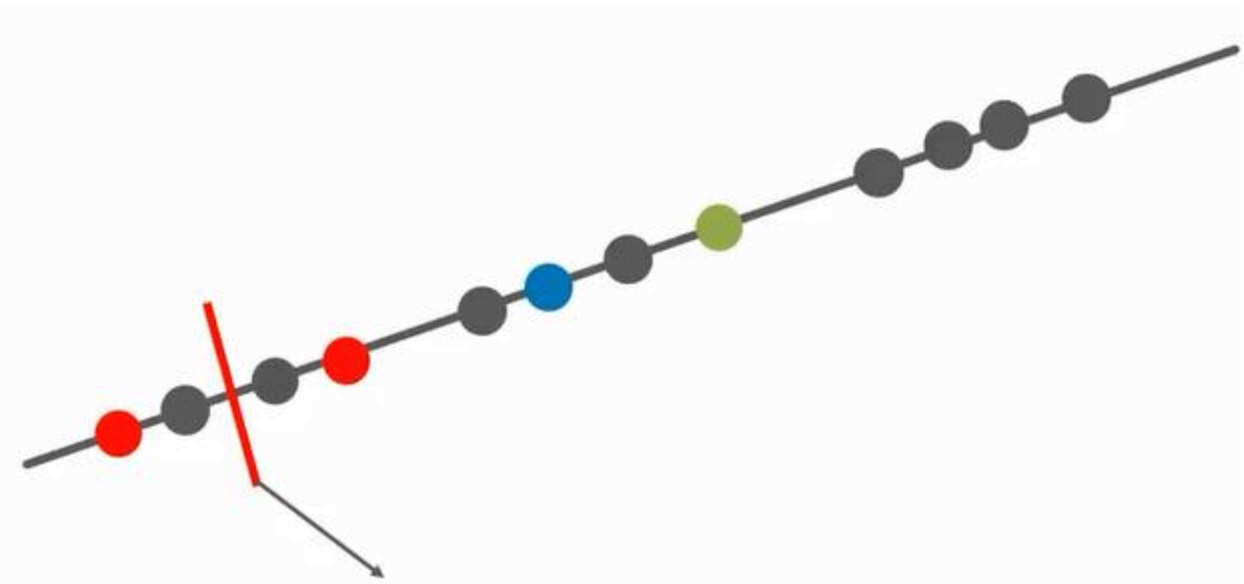
Step-4:Assign 1st point to nearest cluster.



Assign 1st point to nearest cluster. (Red in this case)

K-Means Clustering Algorithm

Step-5:Calculate the mean value including the new point for the red cluster.

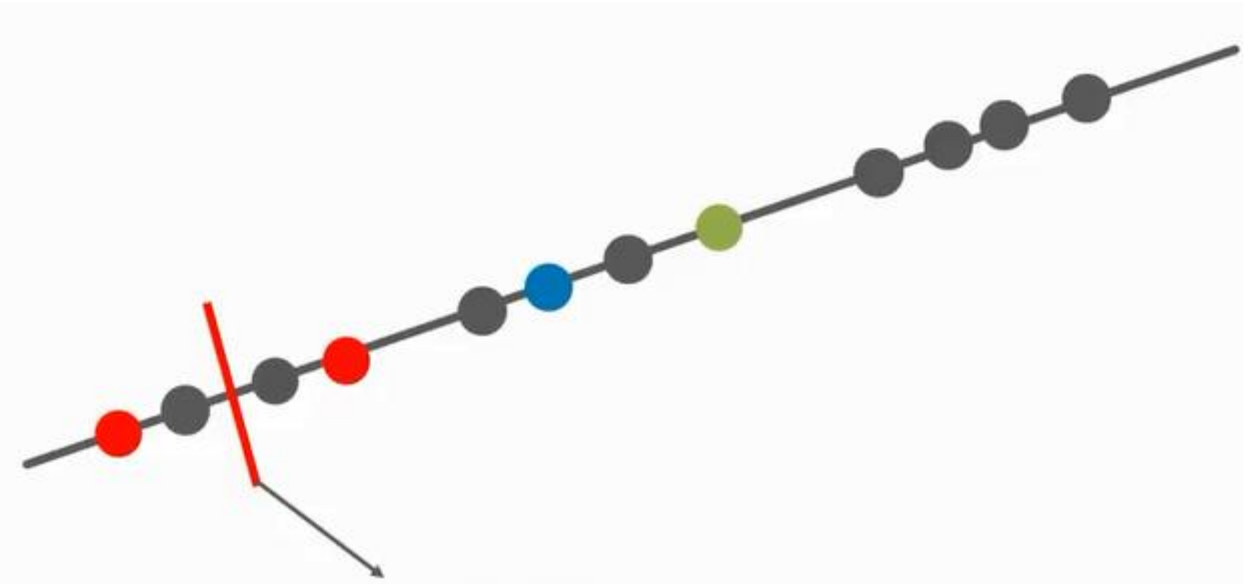


Calculate the mean value including the new point for the red cluster.

K-Means Clustering Algorithm

Step-6: Find to which cluster does point 2 belongs to, how?

Repeat the same procedure but measure distance to the red mean

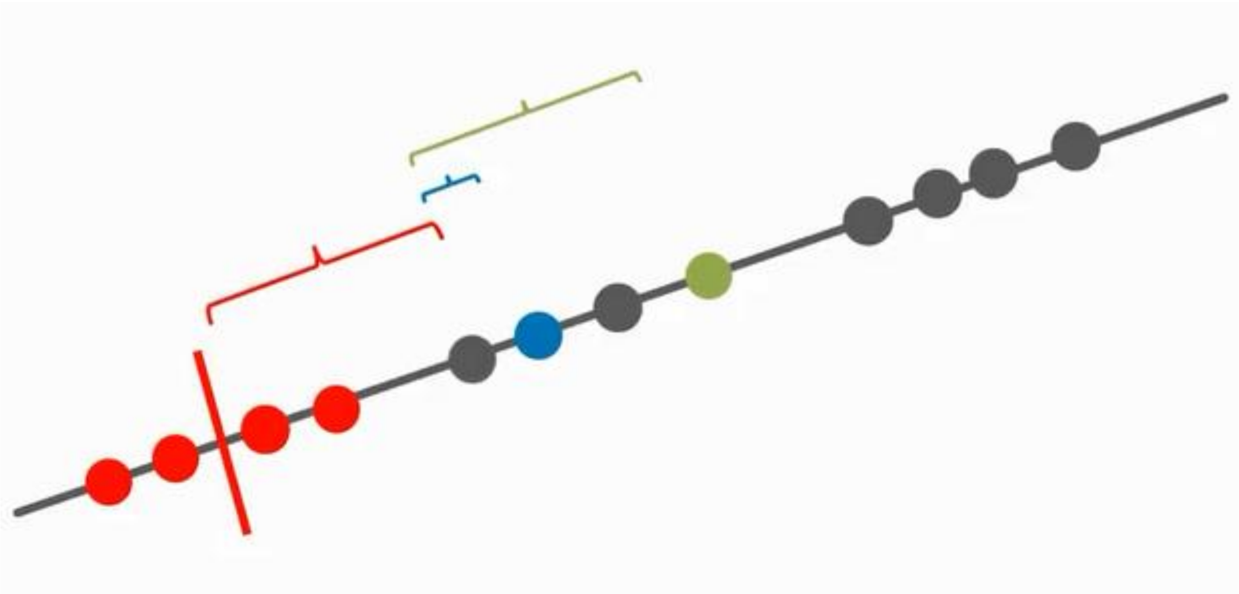


Calculate the mean value including the new point for the red cluster.

K-Means Clustering Algorithm

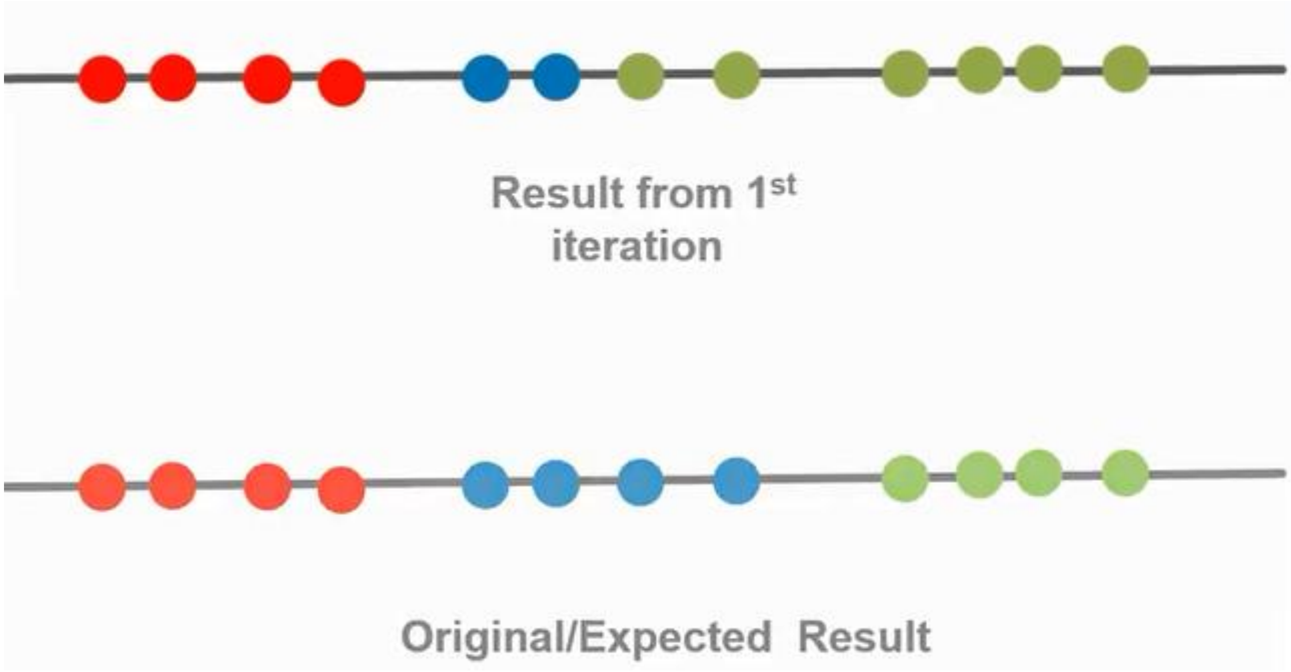
Step-7:

- Measure the distance
- Assign the point to the nearest Cluster
- Calculate the Cluster mean using the new point.

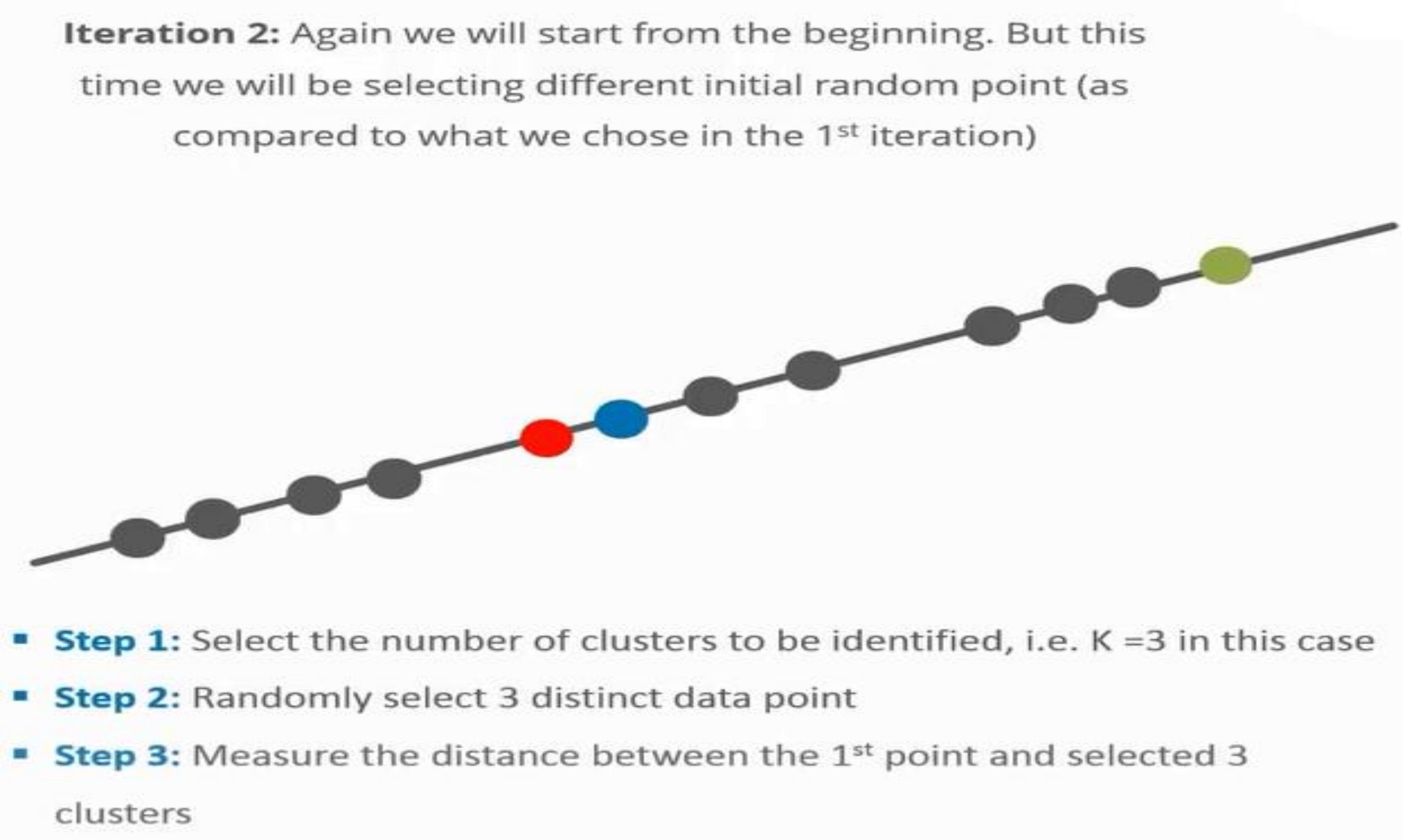


K-Means Clustering Algorithm

According to K-means Algorithm it iterates over again and again unless and until the data points within each cluster stop changing.



K-Means Clustering Algorithm



Iteration 2: Again we will start from the beginning. But this time we will be selecting different initial random point (as compared to what we chose in the 1st iteration)

- **Step 1:** Select the number of clusters to be identified, i.e. K =3 in this case
- **Step 2:** Randomly select 3 distinct data point
- **Step 3:** Measure the distance between the 1st point and selected 3 clusters

K-Means Clustering Algorithm

```
Randomly chose K examples as initial Centroids
While True:
Create K Clusters by assigning each Example to closest Centroid
Compute k new Centroids by Averaging Examples in each Cluster
If centroids don’t change :
break
```

K-Means Clustering Example

Sr No	Height	Weight
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76

K1 Cluster-Centroid-185, 72

K2 Cluster-Centroid-170, 56

Centroid & Euclidean Distance technique

Euclidean Distance Row 3- For K1=20.68

for K2= 4.48

Row 3 belongs to K2 (As ED=4.48)

New Centroid Calculations For K2-

((170+168)/2, (56+60)/2)= **(169, 58)**

Euclidean Distance Row 4 for K1=6.32

For K2=14.14

Row 4 belongs to K1 (As ED=6.32)

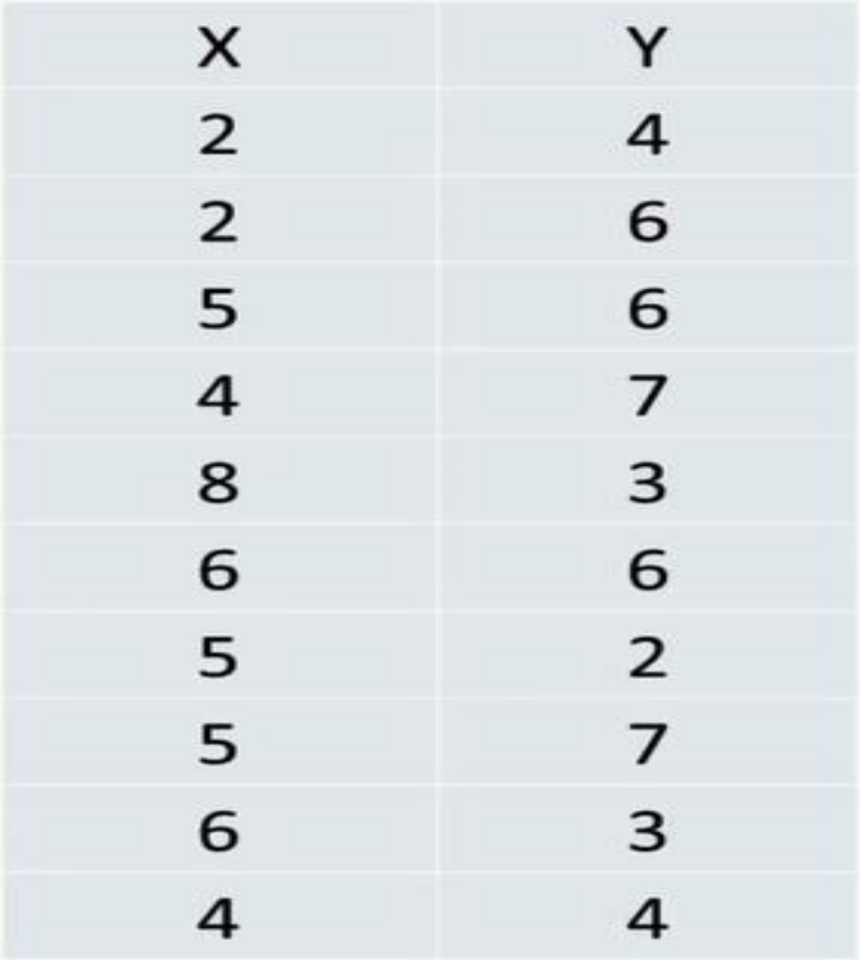
Calculate New Centroid for K1-

((185+179)/2, (72+68/2))= **(182, 70)**

K1= {1,4, 5,6,7,8,9,10,11,12}

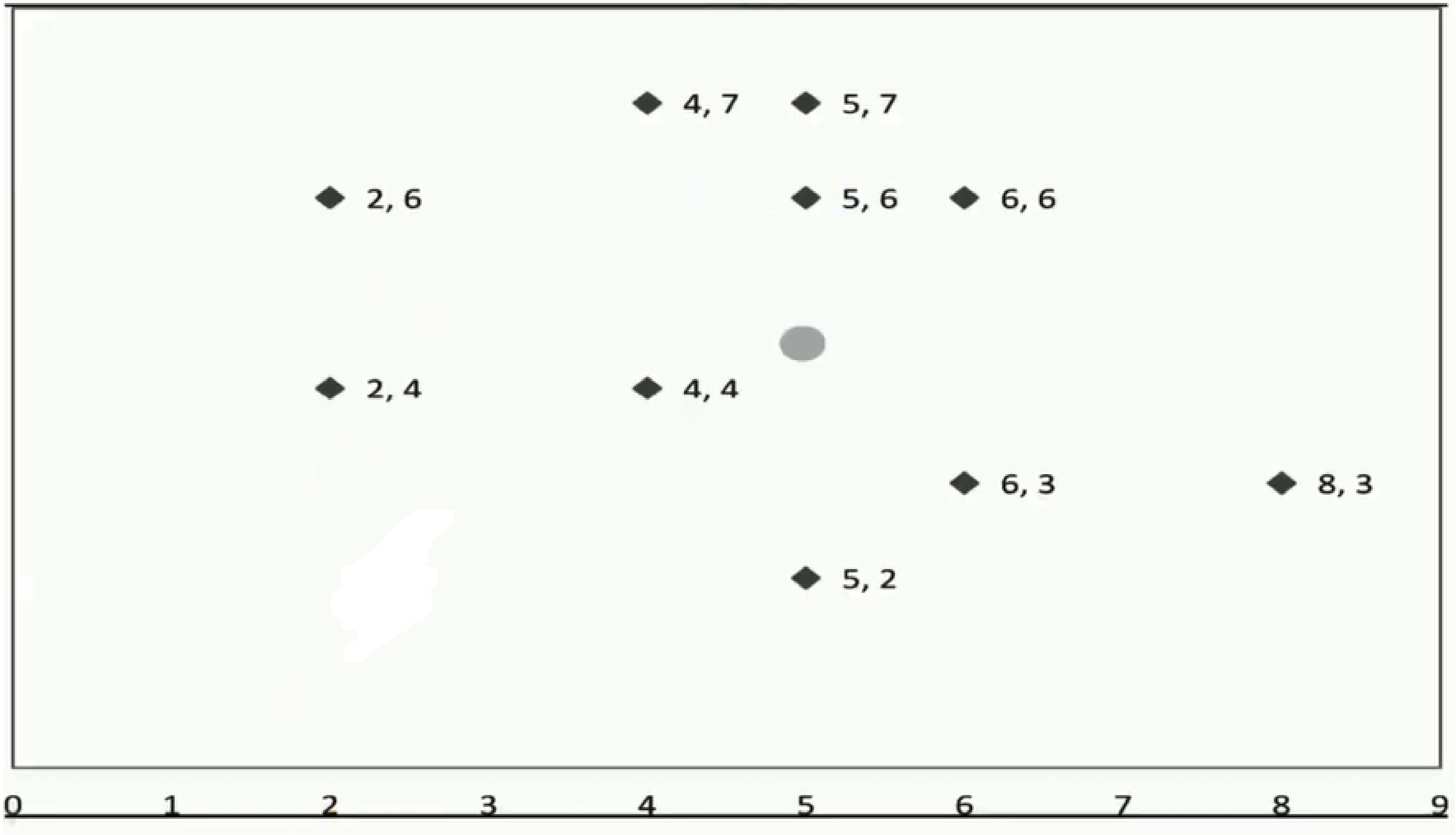
K2= {2, 3}

K-Means Clustering Example

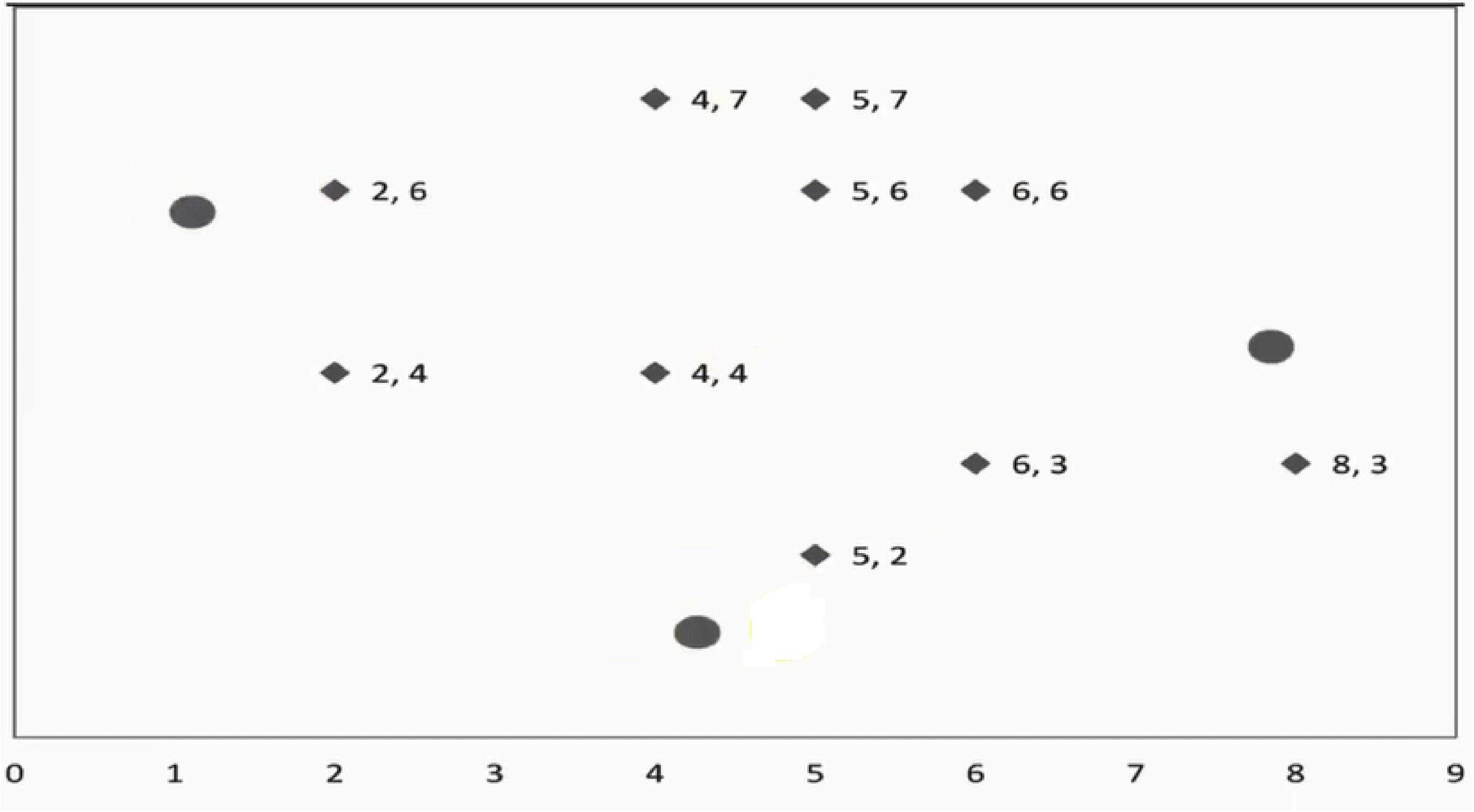


X	Y
2	4
2	6
5	6
4	7
8	3
6	6
5	2
5	7
6	3
4	4

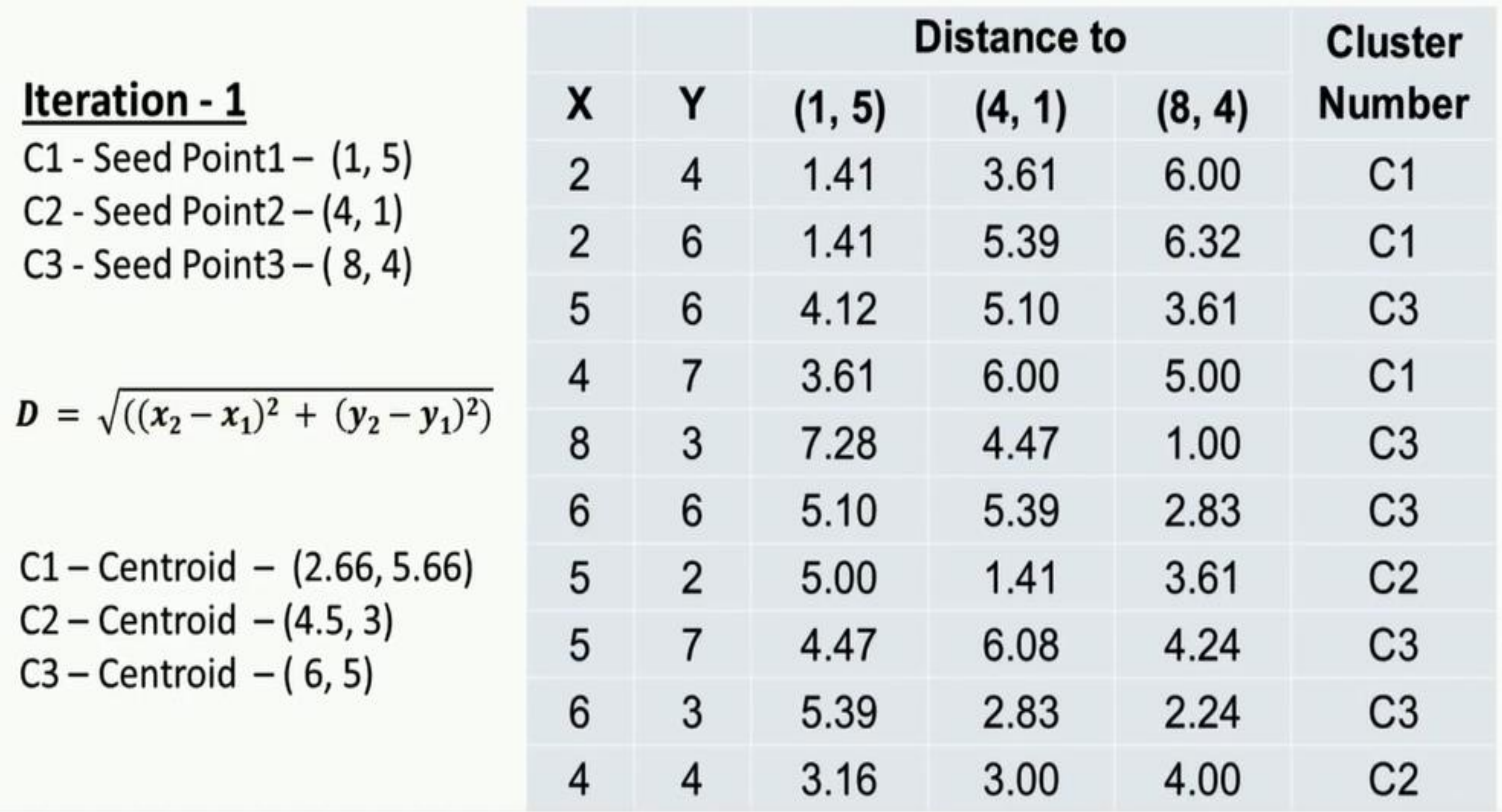
K-Means Clustering Example



K-Means Clustering Example



K-Means Clustering Example



Iteration - 1

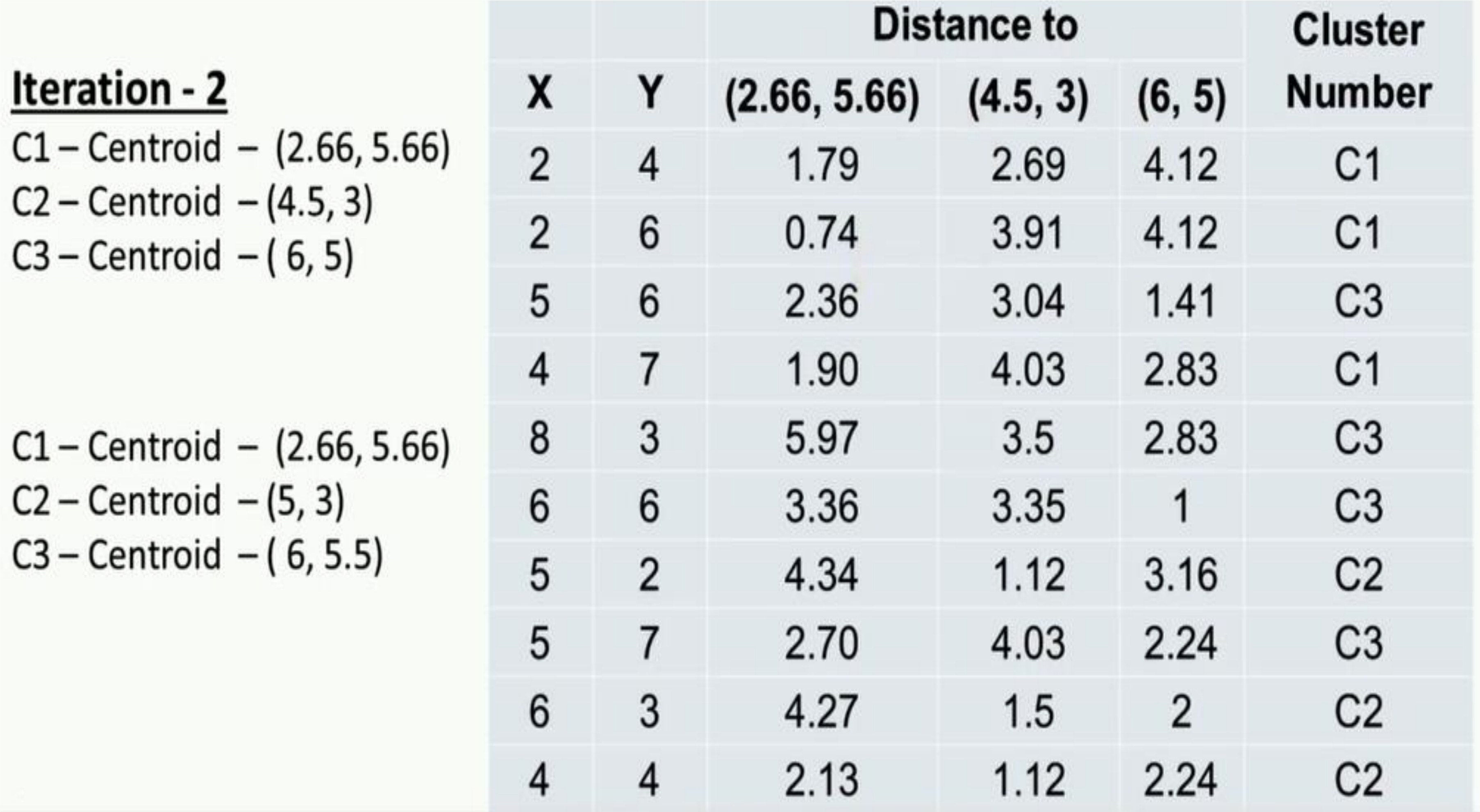
C1 - Seed Point1 – (1, 5)
C2 - Seed Point2 – (4, 1)
C3 - Seed Point3 – (8, 4)

$$D = \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)}$$

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (4.5, 3)
C3 – Centroid – (6, 5)

		Distance to			Cluster
X	Y	(1, 5)	(4, 1)	(8, 4)	Number
2	4	1.41	3.61	6.00	C1
2	6	1.41	5.39	6.32	C1
5	6	4.12	5.10	3.61	C3
4	7	3.61	6.00	5.00	C1
8	3	7.28	4.47	1.00	C3
6	6	5.10	5.39	2.83	C3
5	2	5.00	1.41	3.61	C2
5	7	4.47	6.08	4.24	C3
6	3	5.39	2.83	2.24	C3
4	4	3.16	3.00	4.00	C2

K-Means Clustering Example



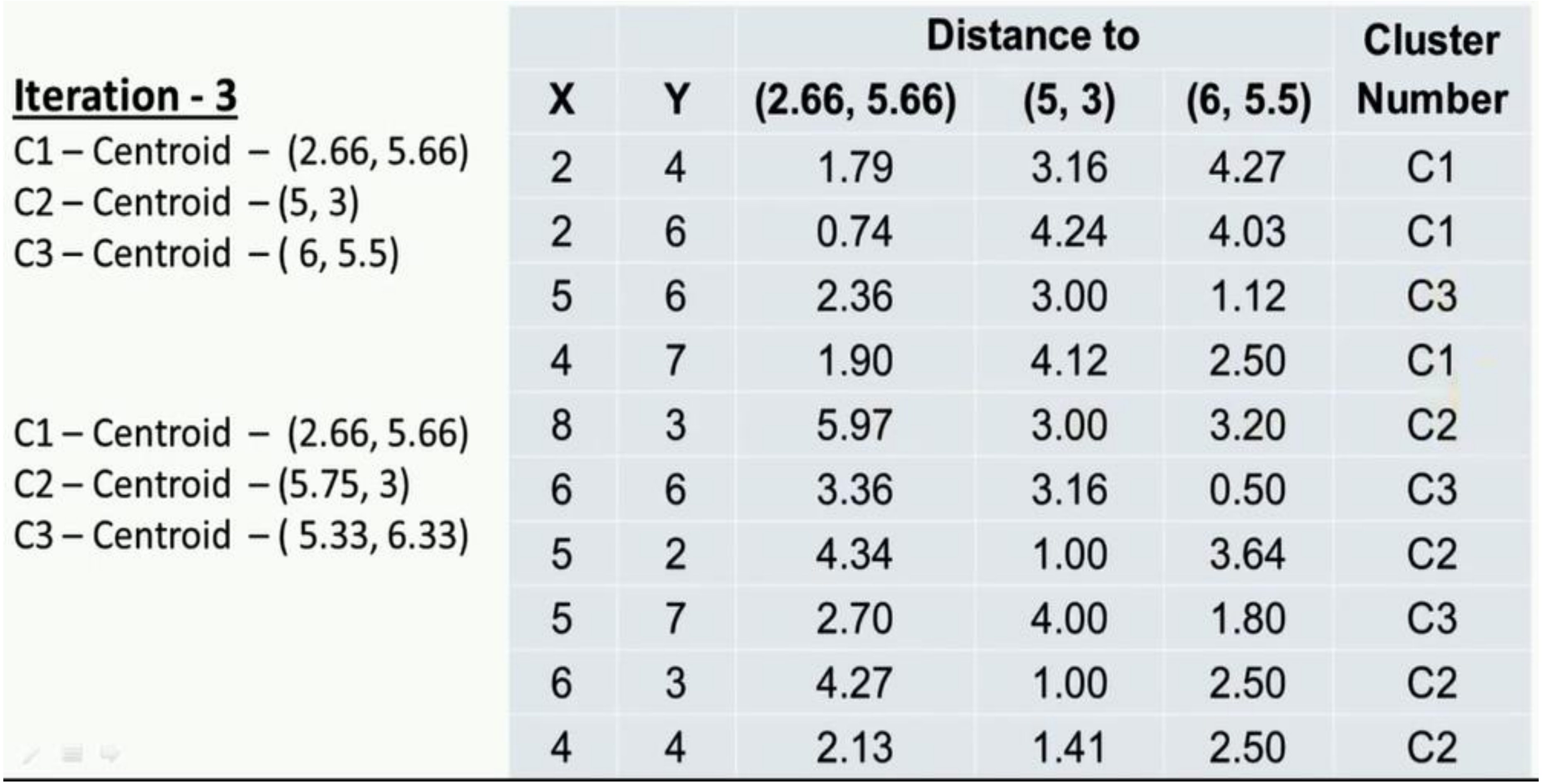
Iteration - 2

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (4.5, 3)
C3 – Centroid – (6, 5)

		Distance to			Cluster
X	Y	(2.66, 5.66)	(4.5, 3)	(6, 5)	Number
2	4	1.79	2.69	4.12	C1
2	6	0.74	3.91	4.12	C1
5	6	2.36	3.04	1.41	C3
4	7	1.90	4.03	2.83	C1
8	3	5.97	3.5	2.83	C3
6	6	3.36	3.35	1	C3
5	2	4.34	1.12	3.16	C2
5	7	2.70	4.03	2.24	C3
6	3	4.27	1.5	2	C2
4	4	2.13	1.12	2.24	C2

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (5, 3)
C3 – Centroid – (6, 5.5)

K-Means Clustering Example



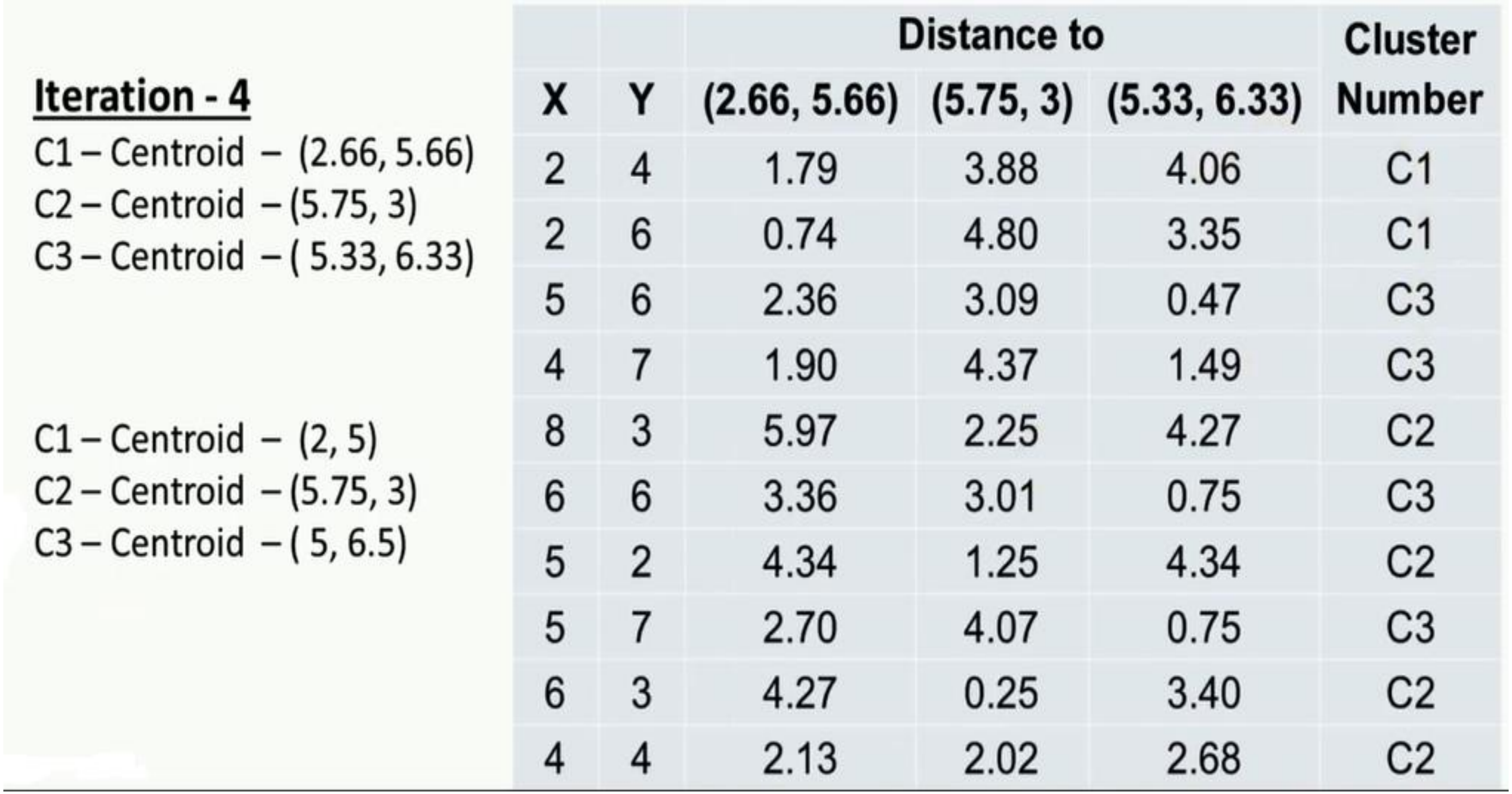
Iteration - 3

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (5, 3)
C3 – Centroid – (6, 5.5)

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (5.75, 3)
C3 – Centroid – (5.33, 6.33)

		Distance to			Cluster
X	Y	(2.66, 5.66)	(5, 3)	(6, 5.5)	Number
2	4	1.79	3.16	4.27	C1
2	6	0.74	4.24	4.03	C1
5	6	2.36	3.00	1.12	C3
4	7	1.90	4.12	2.50	C1
8	3	5.97	3.00	3.20	C2
6	6	3.36	3.16	0.50	C3
5	2	4.34	1.00	3.64	C2
5	7	2.70	4.00	1.80	C3
6	3	4.27	1.00	2.50	C2
4	4	2.13	1.41	2.50	C2

K-Means Clustering Example



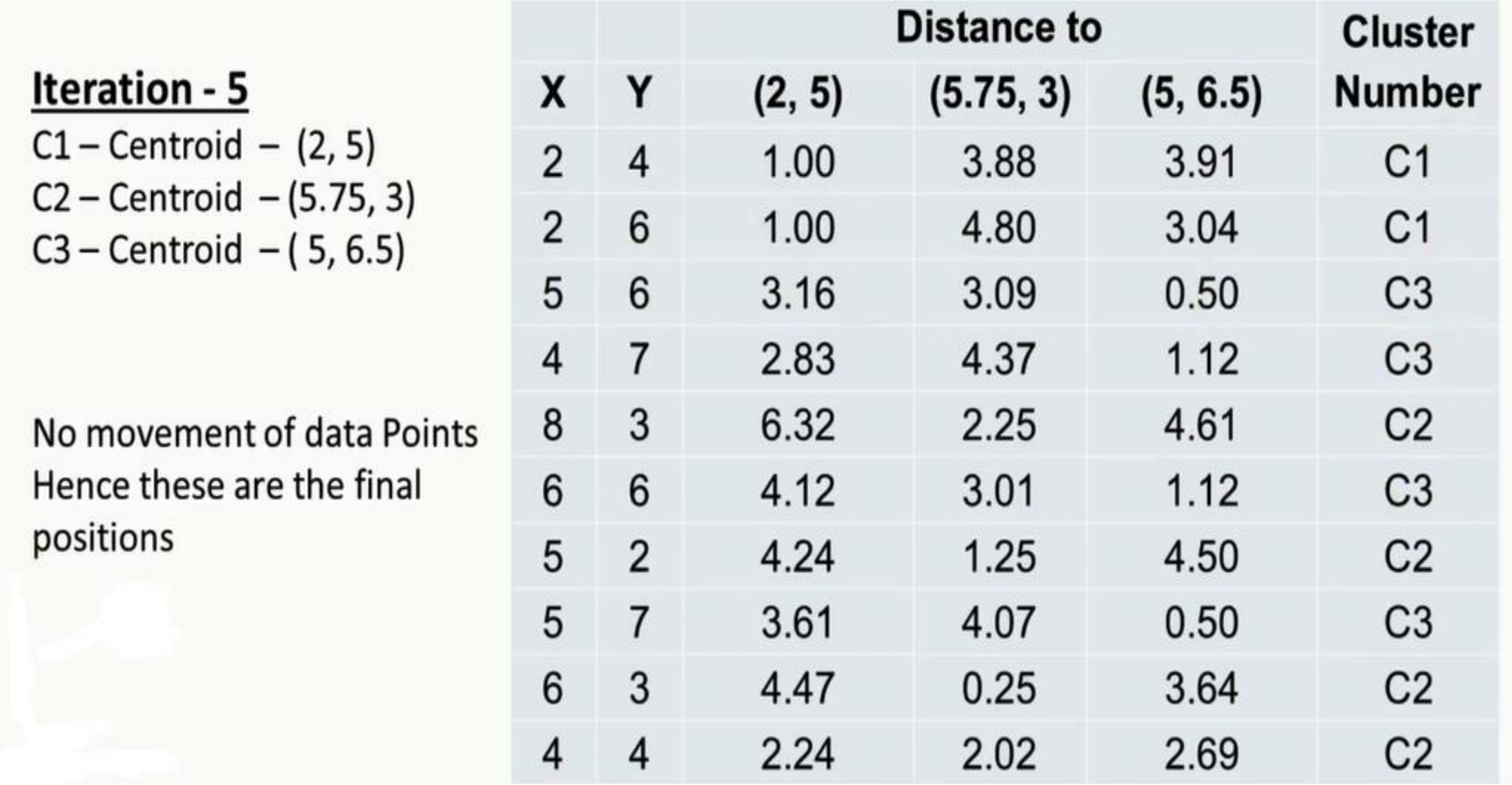
Iteration - 4

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (5.75, 3)
C3 – Centroid – (5.33, 6.33)

		Distance to			Cluster
X	**Y**	**(2.66, 5.66)**	**(5.75, 3)**	**(5.33, 6.33)**	**Number**
2	4	1.79	3.88	4.06	C1
2	6	0.74	4.80	3.35	C1
5	6	2.36	3.09	0.47	C3
4	7	1.90	4.37	1.49	C3
8	3	5.97	2.25	4.27	C2
6	6	3.36	3.01	0.75	C3
5	2	4.34	1.25	4.34	C2
5	7	2.70	4.07	0.75	C3
6	3	4.27	0.25	3.40	C2
4	4	2.13	2.02	2.68	C2

C1 – Centroid – (2, 5)
C2 – Centroid – (5.75, 3)
C3 – Centroid – (5, 6.5)

K-Means Clustering Example



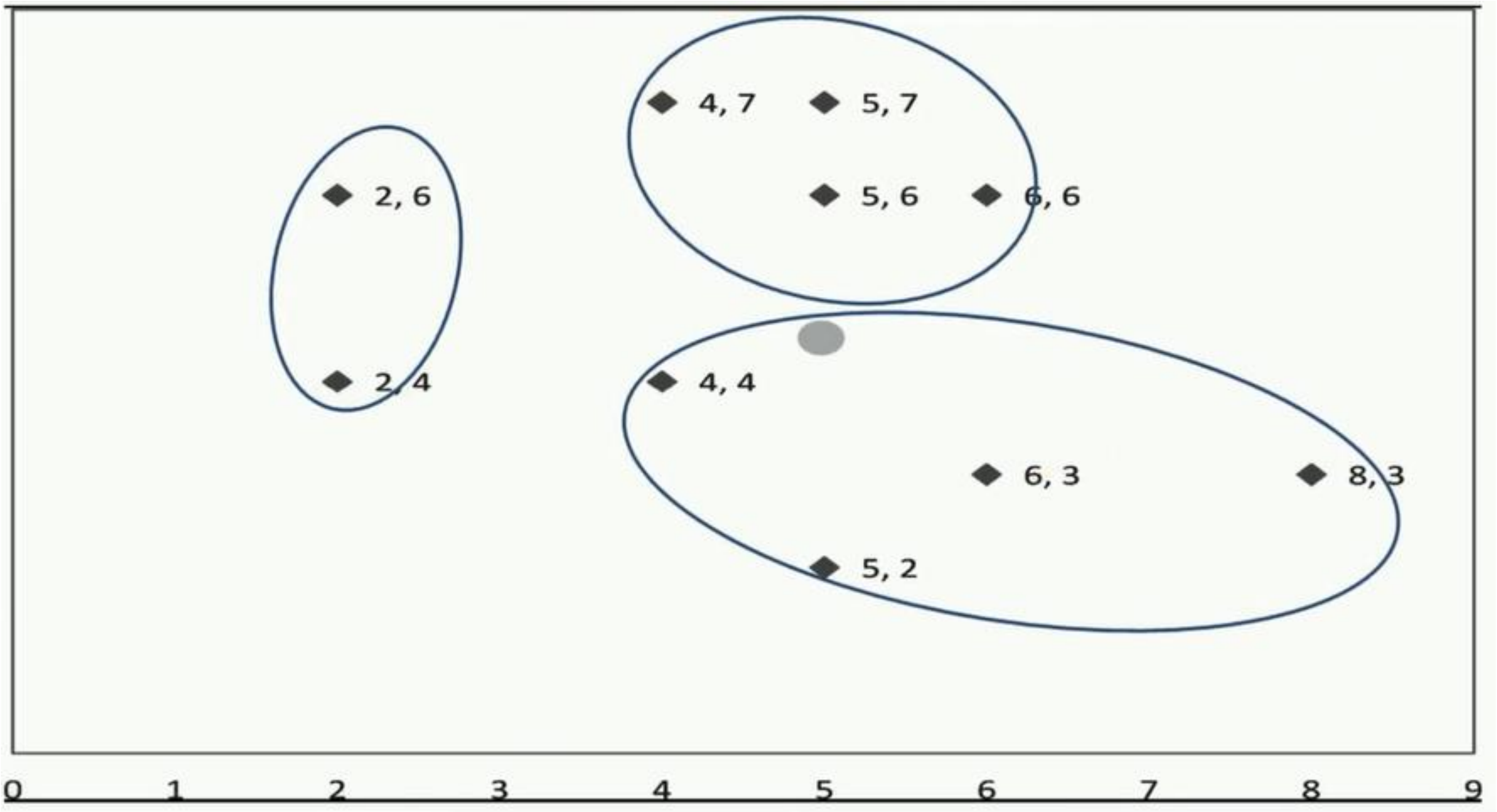
Iteration - 5

C1 – Centroid – (2, 5)
C2 – Centroid – (5.75, 3)
C3 – Centroid – (5, 6.5)

No movement of data Points Hence these are the final positions

		Distance to			Cluster
X	Y	(2, 5)	(5.75, 3)	(5, 6.5)	Number
2	4	1.00	3.88	3.91	C1
2	6	1.00	4.80	3.04	C1
5	6	3.16	3.09	0.50	C3
4	7	2.83	4.37	1.12	C3
8	3	6.32	2.25	4.61	C2
6	6	4.12	3.01	1.12	C3
5	2	4.24	1.25	4.50	C2
5	7	3.61	4.07	0.50	C3
6	3	4.47	0.25	3.64	C2
4	4	2.24	2.02	2.69	C2

K-Means Clustering Example

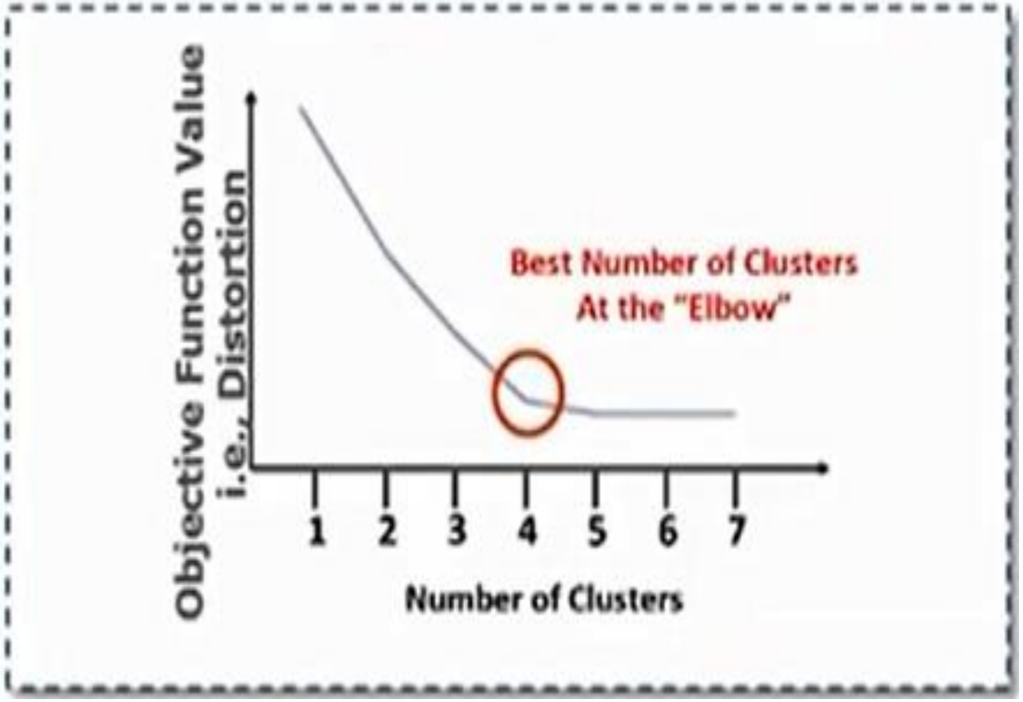


How to decide the number of Clusters

The Elbow Method

First of all, Compute the **Sum of Squared Error (SSE)** for some value of k (For Ex-2,4,6,8 etc.). The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid. Mathematically-

$$\text{SSE}=\sum_{i=1}^{K}\sum_{z=ci} dist(x,ci)^2$$



The idea of the elbow method is to choose the ‘k’ after which the SSE decrease is Almost constant.

Pros and Cons of K-means Clustering

Pros

- Simple Understandable
- Items automatically assigned to clusters

Cons

- Must define number of clusters
- All items forced into clusters
- Unable to handle noisy data and outliers

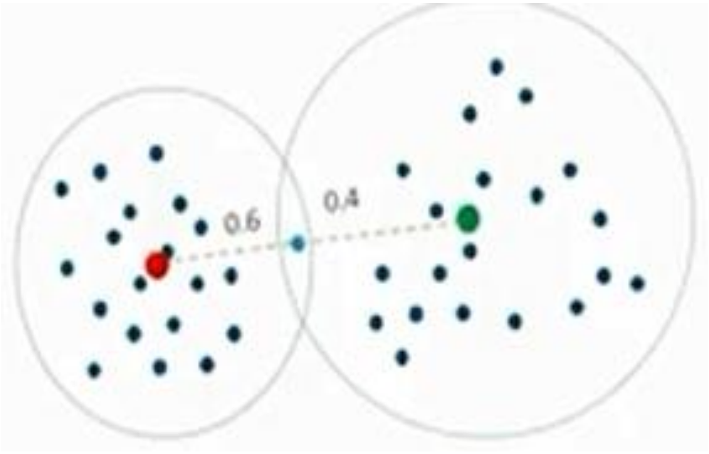
Applications of K-Means Clustering

- Academic Performance
- Diagnostic System
- Search Engine
- Wireless Sensor Network

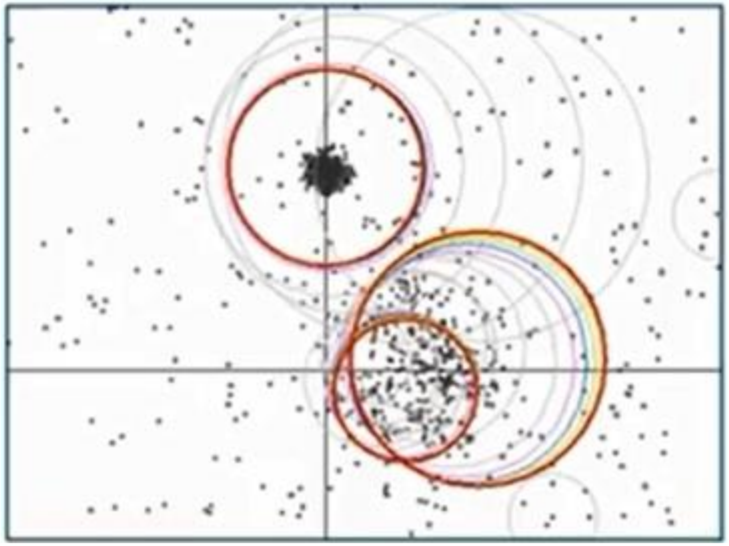
Fuzzy C-means Clustering

Fuzzy C-Means is the extension of K-means, the popular simple Clustering technique.

Fuzzy clustering (also referred to as soft clustering) is a form of clustering in which each data point can belong to more than one cluster.



a



b

Fuzzy C-means Clustering

- Fuzzy Logic was proposed by scientist Lotfi Zadeh
- Represent Uncertainty
- Represent with degree
- Represent the belongingness of a number of a crips set to fuzzy set
- It's a mathematical Language
- Relational Logic + Boolean Logic + Predicate Logic=Fuzzy Language
- Fuzzy Logic deals with Fuzzy Set/ Fuzzy Algebra

Pros and Cons of C-means Clustering

Pros

- Allows a data point to be in multiple clusters.
- A more natural representation of the behavior of genes.
- Genes usually are involved in multiple functions.

Cons

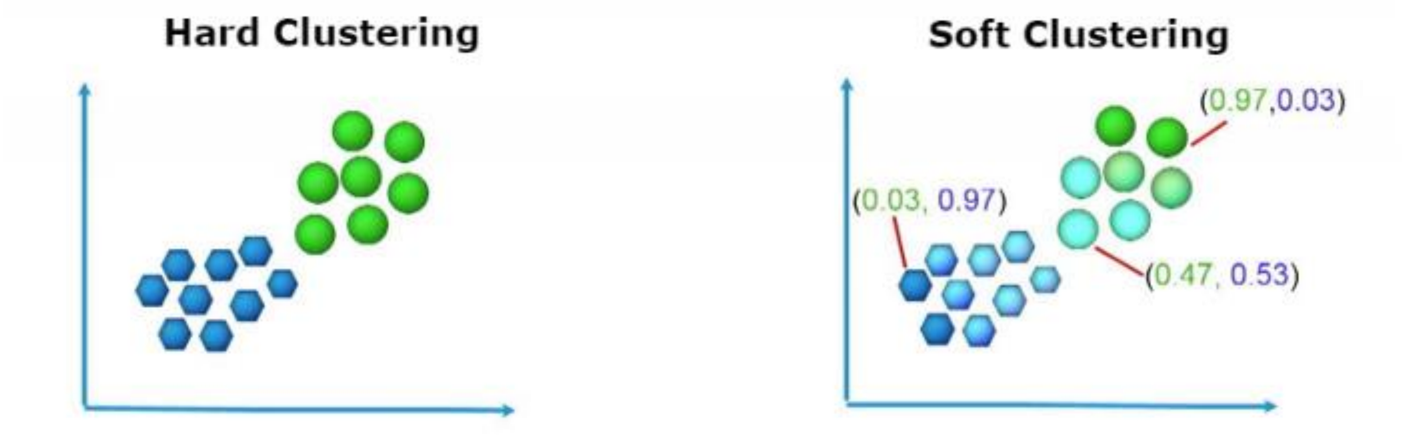
- Need to define c, the number of clusters.
- Need to determine membership cut-off value.
- Clusters are sensitive to initial assignment of centroids.
- Fuzzy C-Means is not a deterministic algorithm.

K-Means versus Fuzzy C-Means

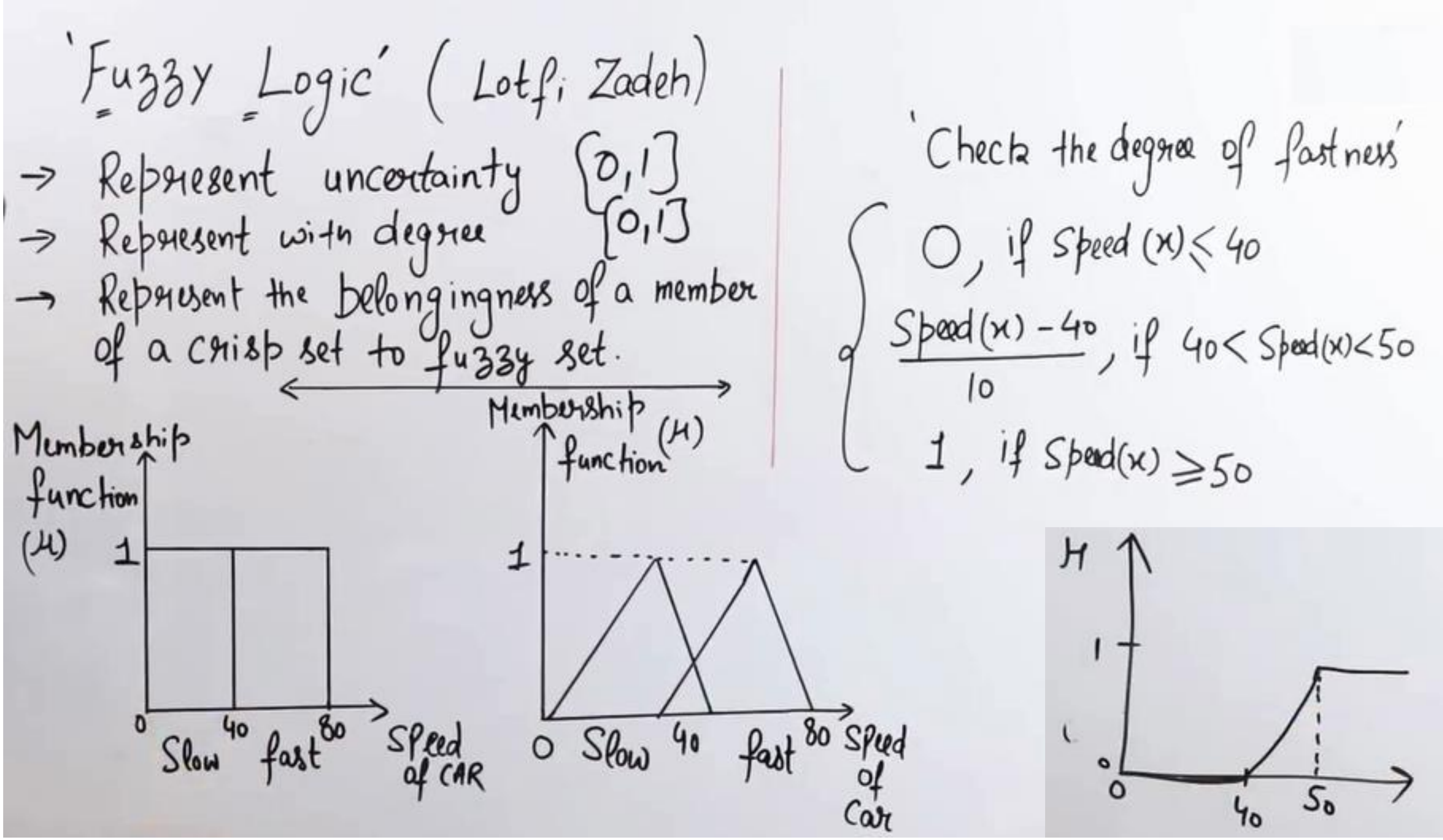
Attribution to a Cluster- In fuzzy clustering, each point has a probability of belonging to each cluster, rather than completely belonging to just one cluster as it is the case in the traditional k-means.

Speed-Fuzzy-C means will tend to run slower than K means, since it's actually doing more work. Each point is evaluated with each cluster, and more operations are involved in each evaluation.

Personal Opinion-Soft-K-Means is "less stupid" than Hard-K-Means when it comes to elongated clusters



Fuzzy C-means Clustering



'Fuzzy Logic' (Lotfi Zadeh)

→ Represent uncertainty $\{0,1\}$ $[0,1]$

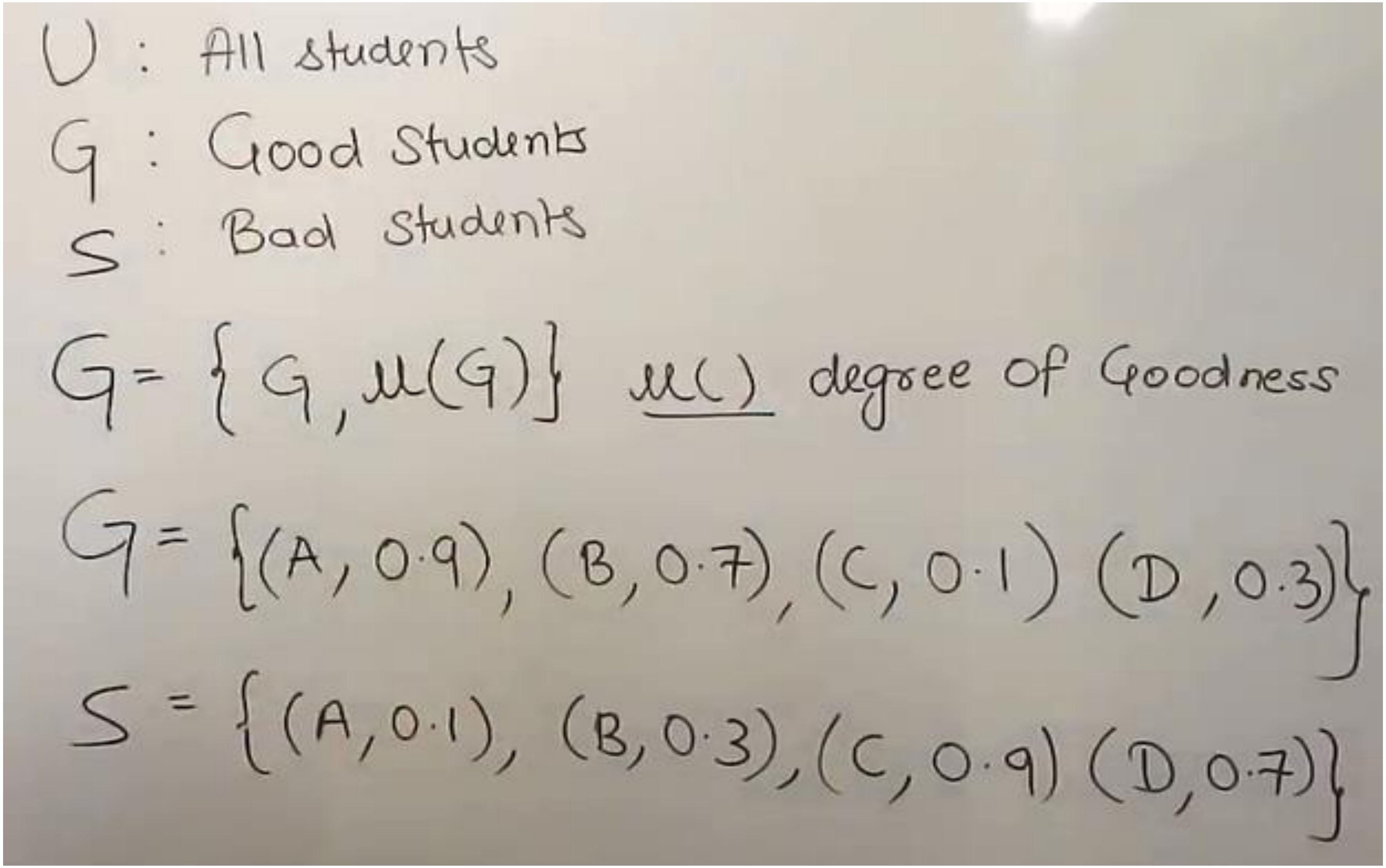
→ Represent with degree

→ Represent the belongingness of a member of a crisp set to fuzzy set.

'Check the degree of fastness'

$$\begin{cases} 0, & \text{if Speed}(x) \leq 40 \\ \dfrac{\text{Speed}(x) - 40}{10}, & \text{if } 40 < \text{Speed}(x) < 50 \\ 1, & \text{if Speed}(x) \geq 50 \end{cases}$$

Fuzzy Sets



U : All students

G : Good Students

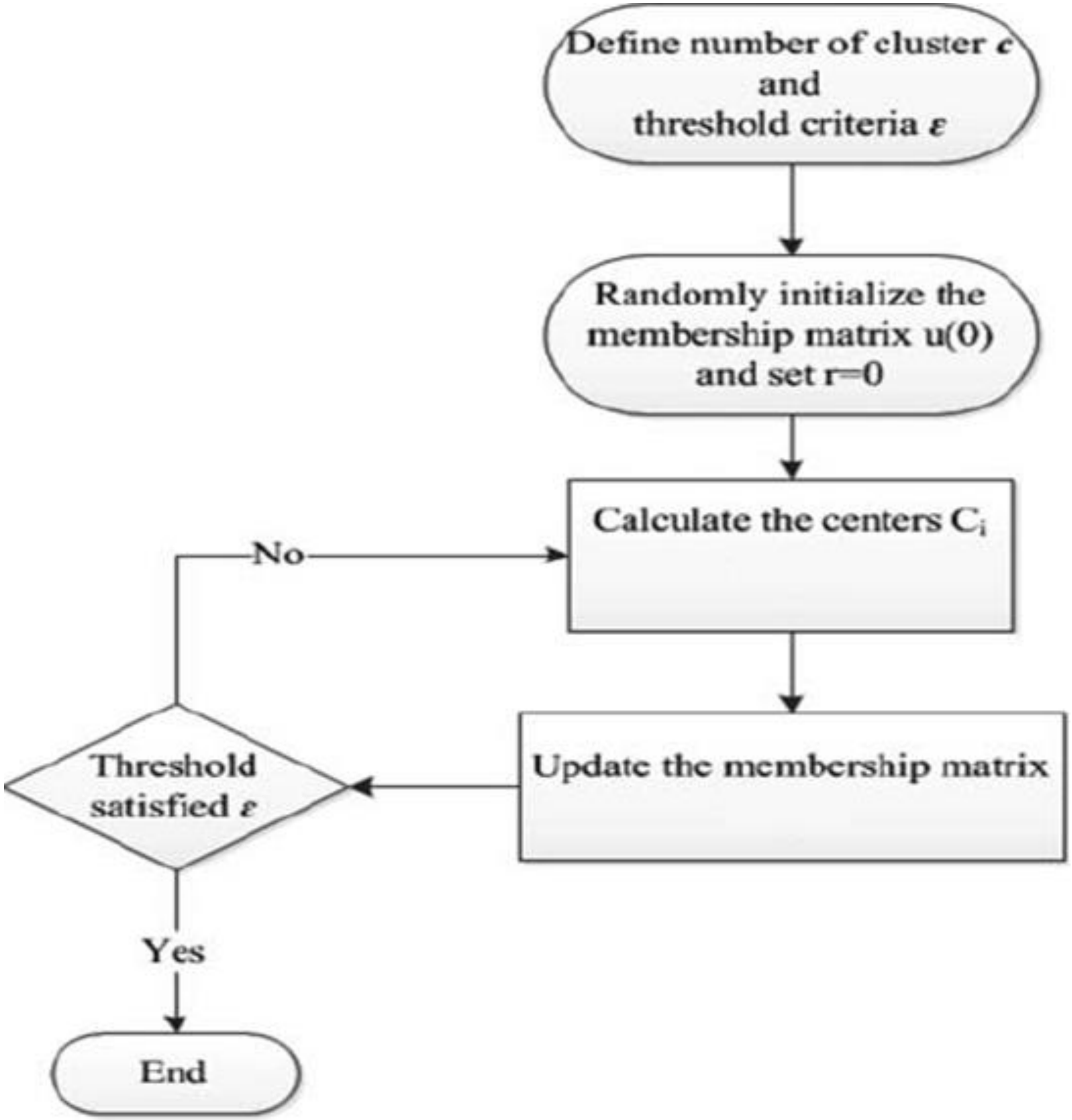
S : Bad students

$G = \{G, \mu(G)\}$ $\underline{\mu(\,)}$ degree of Goodness

$G = \{(A, 0.9), (B, 0.7), (C, 0.1)\ (D, 0.3)\}$

$S = \{(A, 0.1), (B, 0.3), (C, 0.9)\ (D, 0.7)\}$

Steps in Fuzzy C-Means



The process flow of fuzzy C-Means

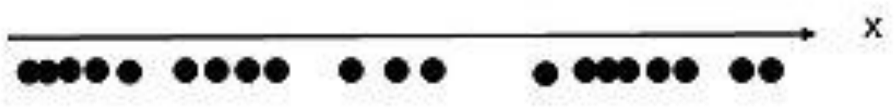
1. Assume a fixed number of clusters k.
2. Initialization: Randomly initialize the k-means μk associated with the clusters and compute the probability that each data point xi is a member of a given cluster k, P(point xi has label k|xi, k).
3. Iteration: Recalculate the centroid of the cluster as the weighted centroid given the probabilities of membership of all data points xi:

$$\mu_k(n+1) = \frac{\sum_{x_i \epsilon k} x_i * P(\mu_k|x_i)^b}{\sum_{x_i \epsilon k} P(\mu_k|x_i)^b}$$

4. Termination: Iterate until convergence or until a user-specified number of iterations has been reached (the iteration may be trapped at some local maxima or minima).

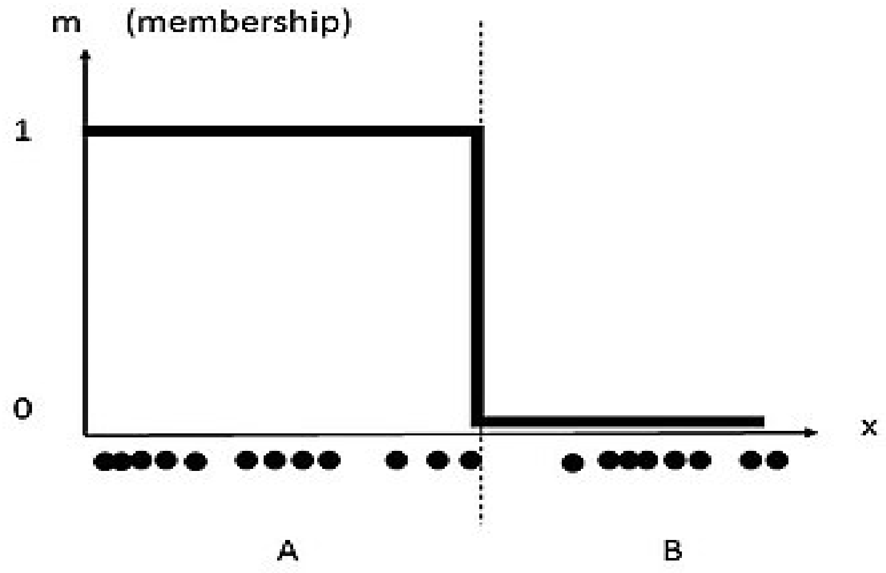
The Fuzzy C-Means Example

To better understand this principle, a classic example of mono-dimensional data is given below on an x axis.



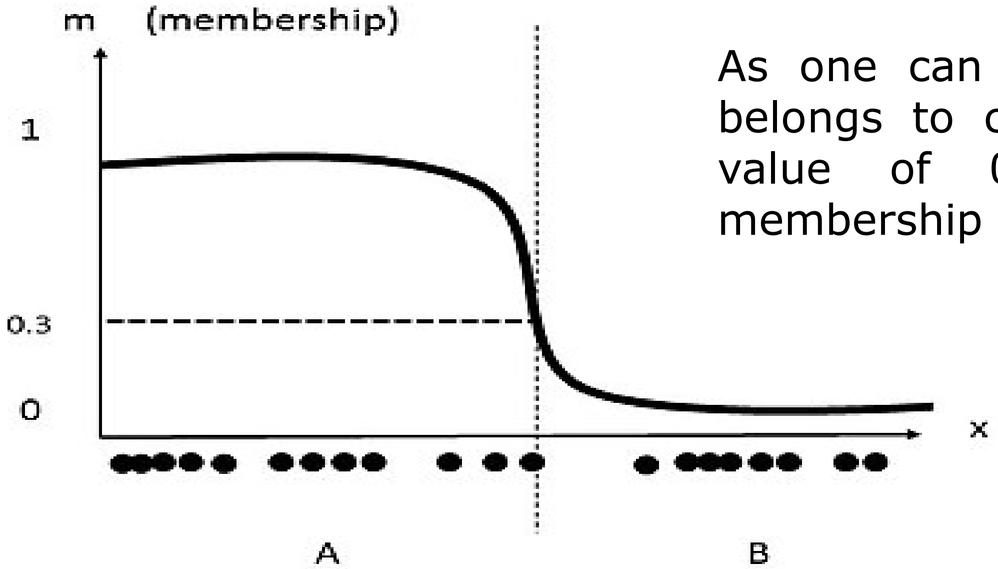
The Fuzzy C-Means Example

- This data set can be traditionally grouped into two clusters.
- By selecting a threshold on the x-axis, the data is separated into two clusters.
- The resulting clusters are labelled 'A' and 'B', as seen in the following image. Each point belonging to the data set would therefore have a membership coefficient of 1 or 0.
- This membership coefficient of each corresponding data point is represented by the inclusion of the y-axis.



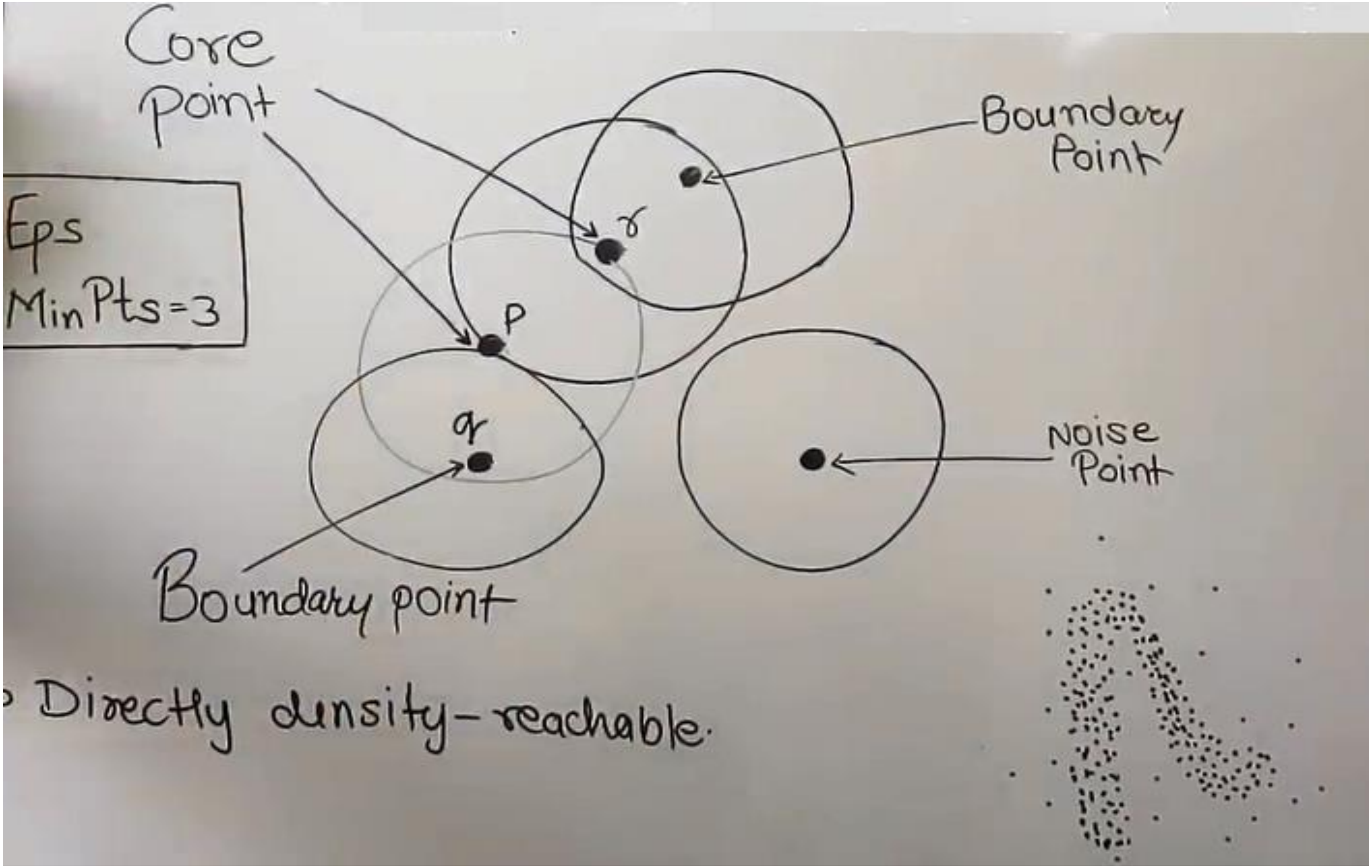
The Fuzzy C-Means Example

- In fuzzy clustering, each data point can have membership to multiple clusters.
- By relaxing the definition of membership coefficients from strictly 1 or 0, these values can range from any value from 1 to 0.
- The following image shows the data set from the previous clustering, but now fuzzy c-means clustering is applied.
- First, a new threshold value defining two clusters may be generated.
- Next, new membership coefficients for each data point are generated based on clusters centroids, as well as distance from each cluster centroid.



As one can see, the middle data point belongs to cluster A and cluster B. the value of 0.3 is this data point's membership coefficient for cluster A.

Density Based Spatial Clustering of Application with Noise



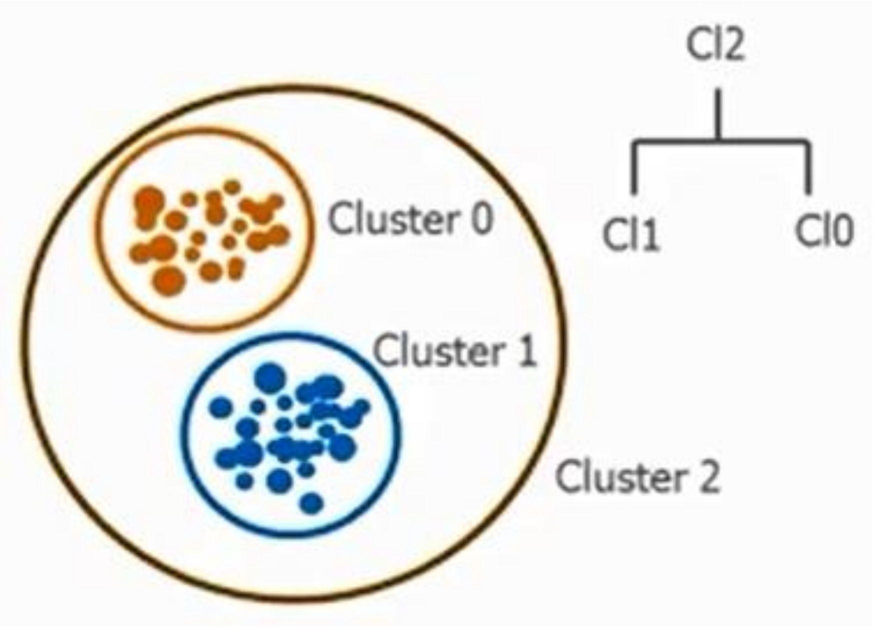
Evaluation Metrics for Clusters

Some popular measures used to evaluate the C-Means clusters :

1. Homogeneity analysis of the clusters formed.
2. The clusters thus formed using Fuzzy C-Means, need to homogeneous and separated from other clusters.
3. Coefficient of Variance analysis for each cluster.
4. Pearson Correlation can be used for validating the quality of clusters.
5. If we have ground truth cluster values, precision, recall, and f-score can also be considered.
6. Elbow Method and Silhouette are also some statistical measures for evaluating your clusters (I would rather use them to in pre-definition of cluster number).
7. Entropy-based methods

Hierarchical Clustering

Hierarchical Clustering is an alternative approach which builds a hierarchy from the bottom up, and doesn't require us to specify the number of clusters beforehand.



Pros and Cons : Hierarchical Clustering

Pros

- No assumption of a particular number of clusters.
- May corresponds to meaningful taxonomies.

Cons

- Once a decision is made to combine two clusters, it cant be undone.
- TO slow for large datasets.

Why to use Market Basket Analysis

Market Basket Analysis

In order to understand why Market Basket Analysis is important we need to understand the objective of MBA

The primary objectives of Market Basket Analysis is to

○ Improve the effectiveness of Marketing and

○ Improve the Sales tactics using customer data collected (During the Sales Transaction)

Market Basket Analysis is a modelling technique based upon the theory that if you buy a certain group of items, you are more (or less) likely to buy another group of items.

What Questions Market Basket Analysis ?

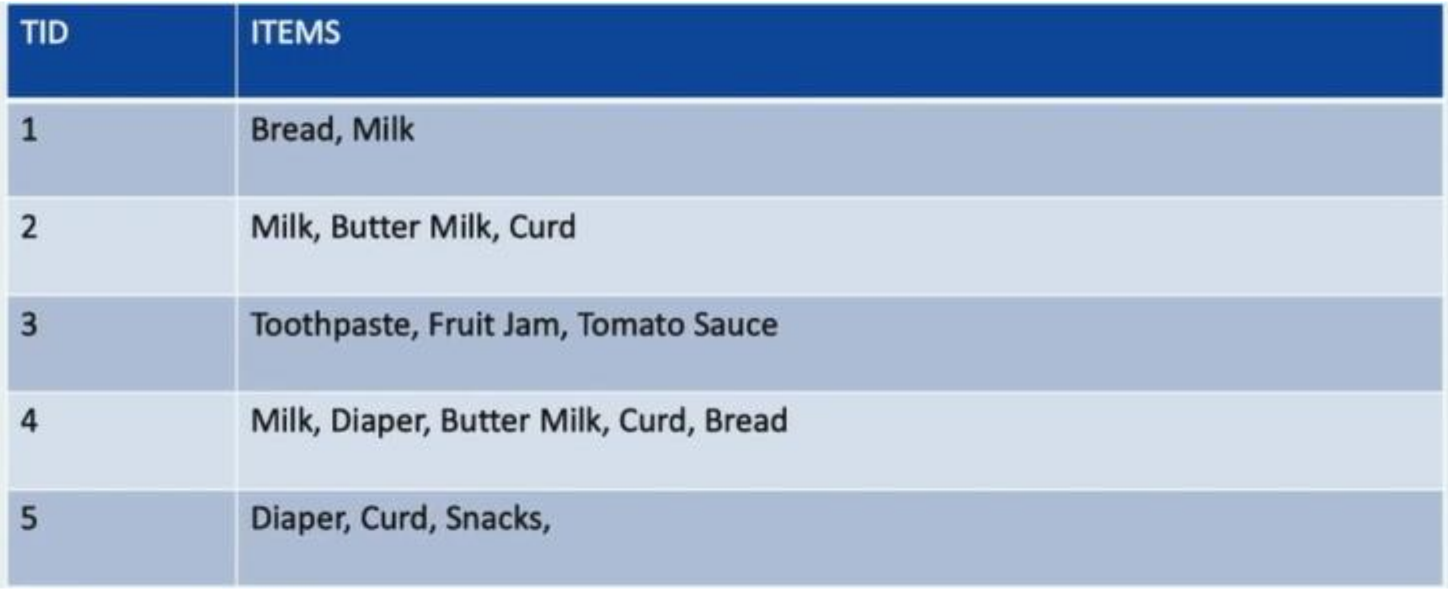
- What products are customers really interested in?
- What products are sold well and which products can be combined with them?
- Which combinations are working well in terms of products?
- Other Random Observations or hidden Pattern if any?

What is Market Basket Analysis ?

- Market Basket Analysis(MBA) is a technique or algorithm of Data Mining to file association rules from given Data or available data.
- The Mathematical Concept behind this algorithm is simple
- Support
- Confidence

Example

Given is the data of the transaction table of the shop/Super Market



TID	ITEMS
1	Bread, Milk
2	Milk, Butter Milk, Curd
3	Toothpaste, Fruit Jam, Tomato Sauce
4	Milk, Diaper, Butter Milk, Curd, Bread
5	Diaper, Curd, Snacks,

Example

A customer coming to a shop buys a Milk, also buys a Bread.

So in the given scenario lets calculate Support and Confidence to understand:

- Support 60%(60% Customers buying at least 1 product from the Shop bought Milk in the list.
- Confidence 66%(66% customers who bought Milk also bought Bread)

Market Basket Analysis

Market Basket Analysis explains the combinations of products that frequently co-occur in transactions.

Market Basket Analysis Algorithms

1. **Association Rule Mining**
2. **Apriori**

Association Rule Mining

- Mining Frequent Pattern and rules
- Association Rules:-Conditional Dependencies
- Two Stages
- Find Frequent Patterns
- Derive Associations (A → B) from frequent Patterns
- Find Patterns in
- Sequences (time Series data, Fault Analysis)
- Transactions (Market basket data)
- Graphs(Social network analysis)

Association Rule Mining

Association Rule Mining is a technique that shows how items are associated with each other.

Examples

1. Customer who purchase bread have a 60% likelihood of also purchasing Jam.

2. Customer who purchase laptop are more likely to purchase laptop bags.

Association Rule Mining

Example of Association Rule

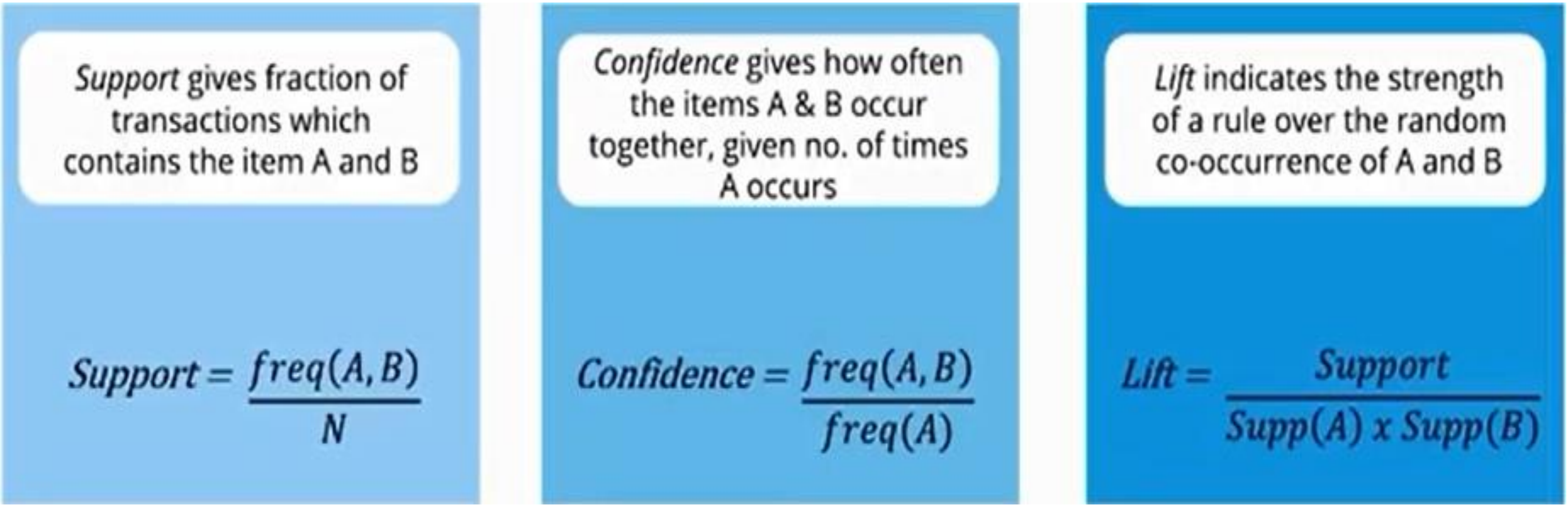
A → B

It means that if a person buys item A then he will also buy item B

Three common ways to measure association:-

- **Support**
- **Confidence**
- **Lift**

Association Rule Mining



Support gives fraction of transactions which contains the item A and B

$$Support = \frac{freq(A,B)}{N}$$

Confidence gives how often the items A & B occur together, given no. of times A occurs

$$Confidence = \frac{freq(A,B)}{freq(A)}$$

Lift indicates the strength of a rule over the random co-occurrence of A and B

$$Lift = \frac{Support}{Supp(A)\ x\ Supp(B)}$$

Market Basket Analysis

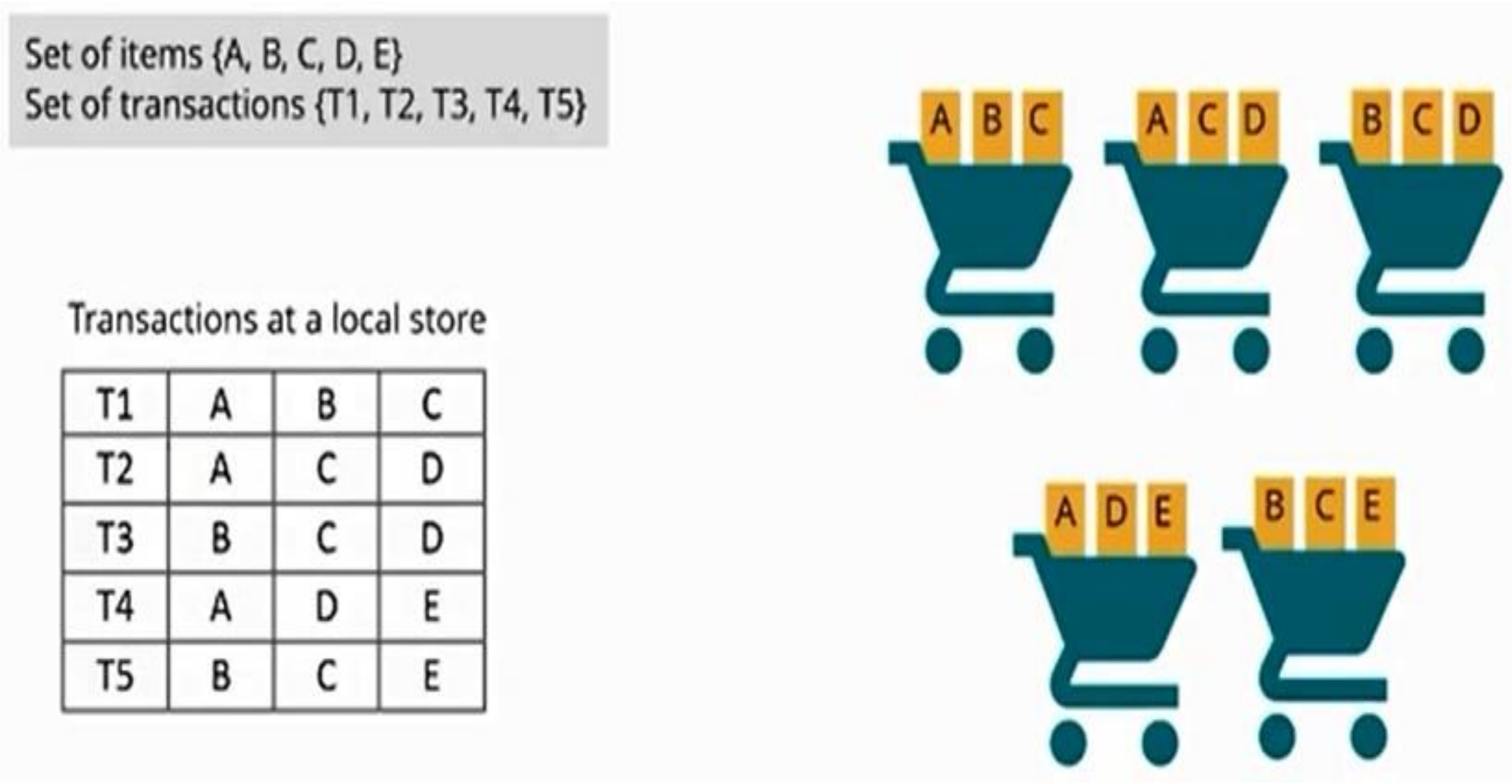
Market Basket Analysis

- Immediate Extension to Association Rules.
- Association Rules with “Lot of Business” outcome.
- Very highly used in Retail Scenarios.
- **Typical Input**
  - List of purchases by customers over different visits
- **Output**
  - What items purchased together?
  - What items purchased sequentially?
  - What items purchased in seasons?
- **Association Rules-Generate Rules**
- **Example-(X→Y)**
  - Market Basket → Assigns Business
  - Outcome to those rules
  - Example-X, Y Could be Sold together.

Market Basket Analysis-Lift Measure

- **Confidence :** How confident that Y is present, in the presence of X?
- **Expected Confidence :** How confident that Y is present, in the absence of X?
- **Lift = Confidence/(Expected Confidence)** =(Y in presence of X)/(Y in absence of X)
- Explains the change in probability of Y over (presence of X) and (absence of X).
- Lift =1 implies that X actually makes no impact on Y.
- Lift>1 implies that the relationship between X and Y is more significant.
- The Larger the **lift ratio**, the more significant the association.

Association Rule Mining-Example

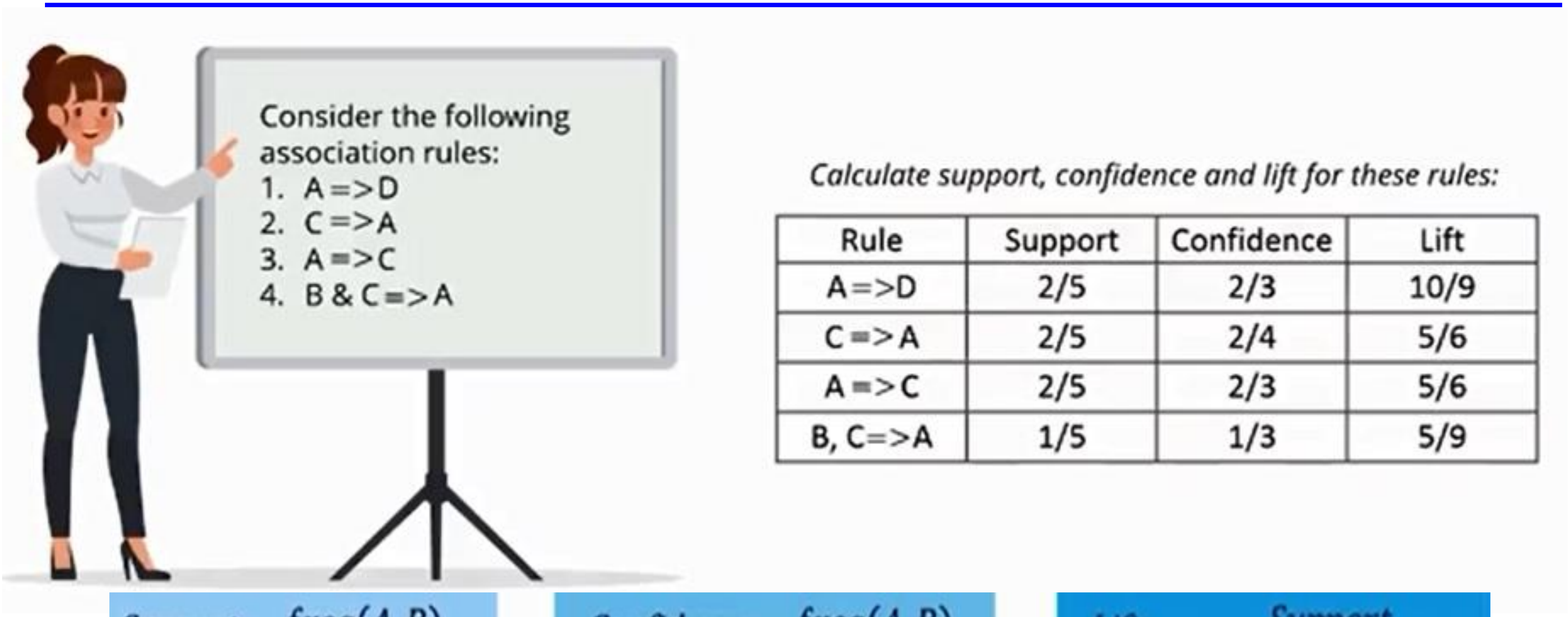


Set of items {A, B, C, D, E}
Set of transactions {T1, T2, T3, T4, T5}

Transactions at a local store

T1	A	B	C
T2	A	C	D
T3	B	C	D
T4	A	D	E
T5	B	C	E

Association Rule Mining-Example



Consider the following association rules:

1. A=>D
2. C=>A
3. A=>C
4. B & C=>A

Calculate support, confidence and lift for these rules:

Rule	Support	Confidence	Lift
A=>D	2/5	2/3	10/9
C=>A	2/5	2/4	5/6
A=>C	2/5	2/3	5/6
B, C=>A	1/5	1/3	5/9

$$Support = \frac{freq(A,B)}{N}$$

$$Confidence = \frac{freq(A,B)}{freq(A)}$$

$$Lift = \frac{Support}{Supp(A)\ x\ Supp(B)}$$

Support P (A U B)

Confidence P(B|A) P(A U B) / P(A)

Lift Confidence / P(B) => P(A U B) / P(A).P(B)

Apriori Algorithm-Example

Example- For the following Given Transaction Data-Set, Generate Rules using Apriori Algorithm. Consider Values as **Support=50%, and Confidence=75%.**

Data Set **Frequent Item Set Support(Bread)=nBread/n**

Trans. ID	Items Purchased
1	Bread, Cheese, Egg, Juice
2	Bread, Cheese, Juice
3	Bread, Milk, Yogurt
4	Bread, Juice, Milk
5	Cheese, Juice, Milk

Items	Frequency	Support
Bread	4	4/5=80%
Cheese	3	3/5=60%
Egg	**1**	**1/5=20%**
Juice	4	4/5=80%
Milk	3	3/5=60%
Yogurt	**1**	**1/5=20%**

Remove Egg and Yogurt from the list as Support value is less than 50%

Apriori Algorithm-Example

Make 2-Items Candidate Set and Write their Frequency

Support=50%, and Confidence=75%.

Item Pairs	Frequency	Support
Bread, Cheese	2	2/5=40%
Bread, Juice	**3**	**3/5=60%**
Bread, Milk	2	2/5=40%
Cheese, Juice	**3**	**3/5=60%**
Cheese, Milk	1	1/5=20%
Juice, Milk	2	2/5=40%

For Rules –

I. Bread, Juice (1)

II. Cheese, Juice (2)

1. (Bread, Juice-)

Bread → Juice

Juice → Bread

Confidence (A → B)= Support(A U B)/S(A)

I. Bread, Juice –Calculate Confidence Level

1. Bread → Juice = S(B U J) /S(B) = 3/5 * 5/4 =3/4= **75%**
2. Juice → Bread = S(J U B) /S(J) = 3/5 * 5/4 =3/4=**75 %**

II. Cheese, Juice-Calculate Confidence Level

1. Cheese → Juice = S(C U J)/S(C) = 3/5 * 5/3 =**100%**
2. Juice → Cheese = S(J U C)/S(J) =3/5 * 5/4= 3/4 = **75 %**

As Confidence Level of all the Rules are equal to 75% -Means all the Rules are Good.

Apriori Algorithm-Example

Example- For the following Given Transaction Data-Set, Generate Rules using Apriori Algorithm. Consider Values as **Support=50%, and Confidence=70%.**

Data Set

Trans. ID	Items Purchased
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

I Find out the Frequency of Items

Item Set	Support
1	2/4 = 50%
2	3/4 = 75%
3	3/4 = 75%
4	**1/4 = 25%**
5	3/4 = 75%

Itemset-1,2,3,5 For 4 Support is 25

Apriori Algorithm-Example

Example- For the following Given Transaction Data-Set, Generate Rules using Apriori Algorithm. Consider Values as **Support=50%, and Confidence=70%.**

II Find out the Support/ Frequency of Pair of Items

Item Sets	**Support**
{1,2}	**1/4 = 25%**
{1,3}	2/4 = 50%
{1,5}	**1/4 = 25%**
{2,3}	2/4 = 50%
{2,5}	3/4 = 75%
{3,5}	2/4 = 50%

III Find out the Support/Frequency of three Items

Item Sets	**Support**
{1,3,5}	**1/4 = 25%**
{2,3,5}	2/4 = 50%
{1, 2,3}	**1/4 = 25%**

In {2,3, 5} Support is 50%. We have to prepare Rules.

Apriori Algorithm-Example

Consider Values as **Support=50%, and Confidence=70%.**

Define Rules and Calculate Confidence Values.

II Rules

Confidence=S(A U B)/S(A)

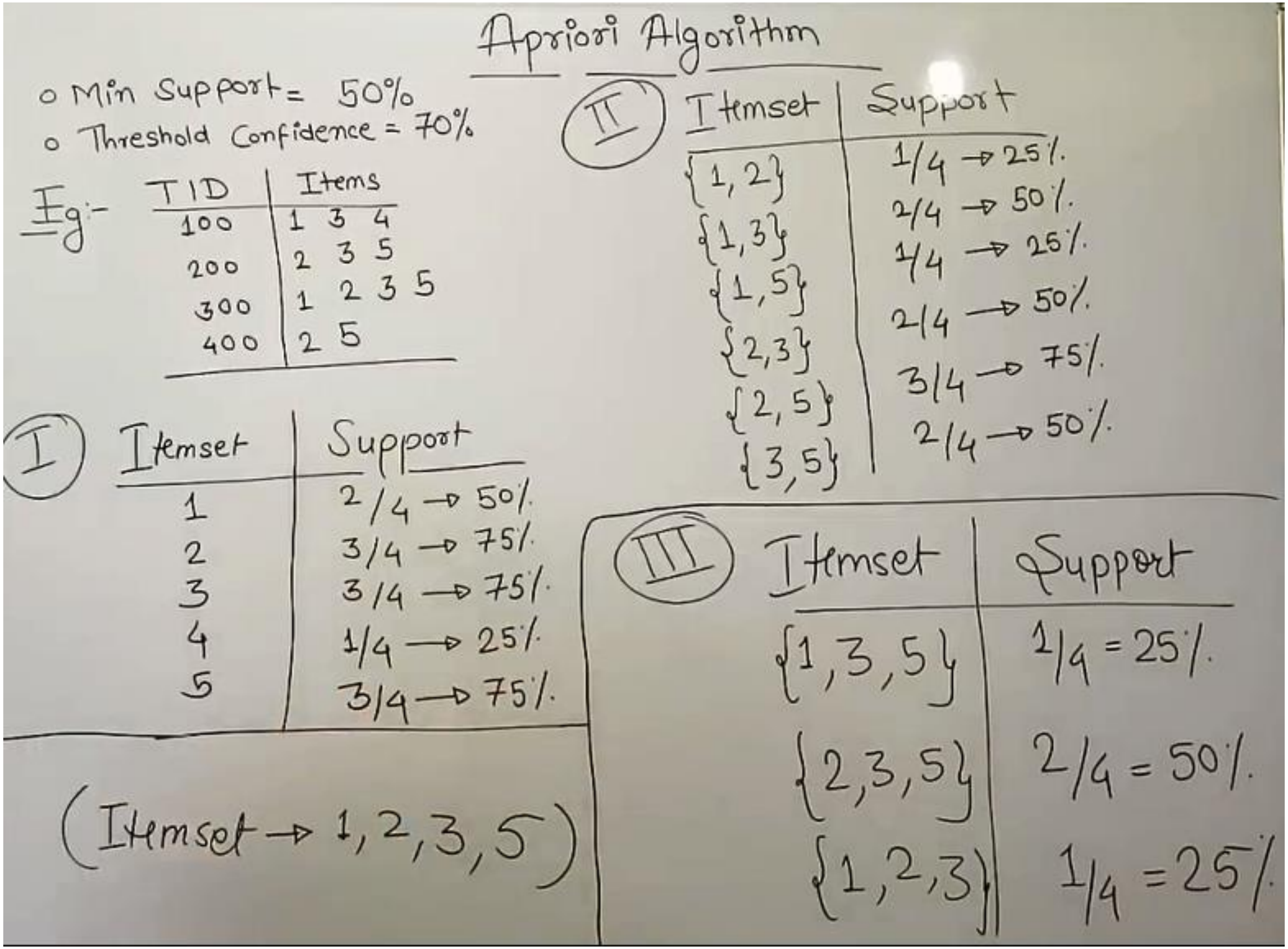
Ex-(2 ^ 3) → 5 =2/2=100%

Sr No	Rules	Support
1	(2 ^ 3) → 5	2
2	(3 ^ 5) → 5	2
3	(2 ^ 5) → 3	2
4	2 → (3 ^ 5)	2
5	5 → (2 ^ 3)	2
6	3 → (2 ^ 5)	2

Sr No	Rules	Confidence
1	(2 ^ 3) → 5	2/2=100%
2	(3 ^ 5) → 5	2/2=100%
3	(2 ^ 5) → 3	2/3=66%
4	2 → (3 ^ 5)	2/3=66%
5	5 → (2 ^ 3)	2/3=66%
6	3 → (2 ^ 5)	2/3=66%

Rules 1 & 2 are Valid as Confidence Level is greater than 70%

Apriori Algorithm-Example



Apriori Algorithm

- Min Support = 50%
- Threshold Confidence = 70%

Eg:-

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

I

Itemset	Support
1	2/4 → 50%
2	3/4 → 75%
3	3/4 → 75%
4	1/4 → 25%
5	3/4 → 75%

(Itemset → 1, 2, 3, 5)

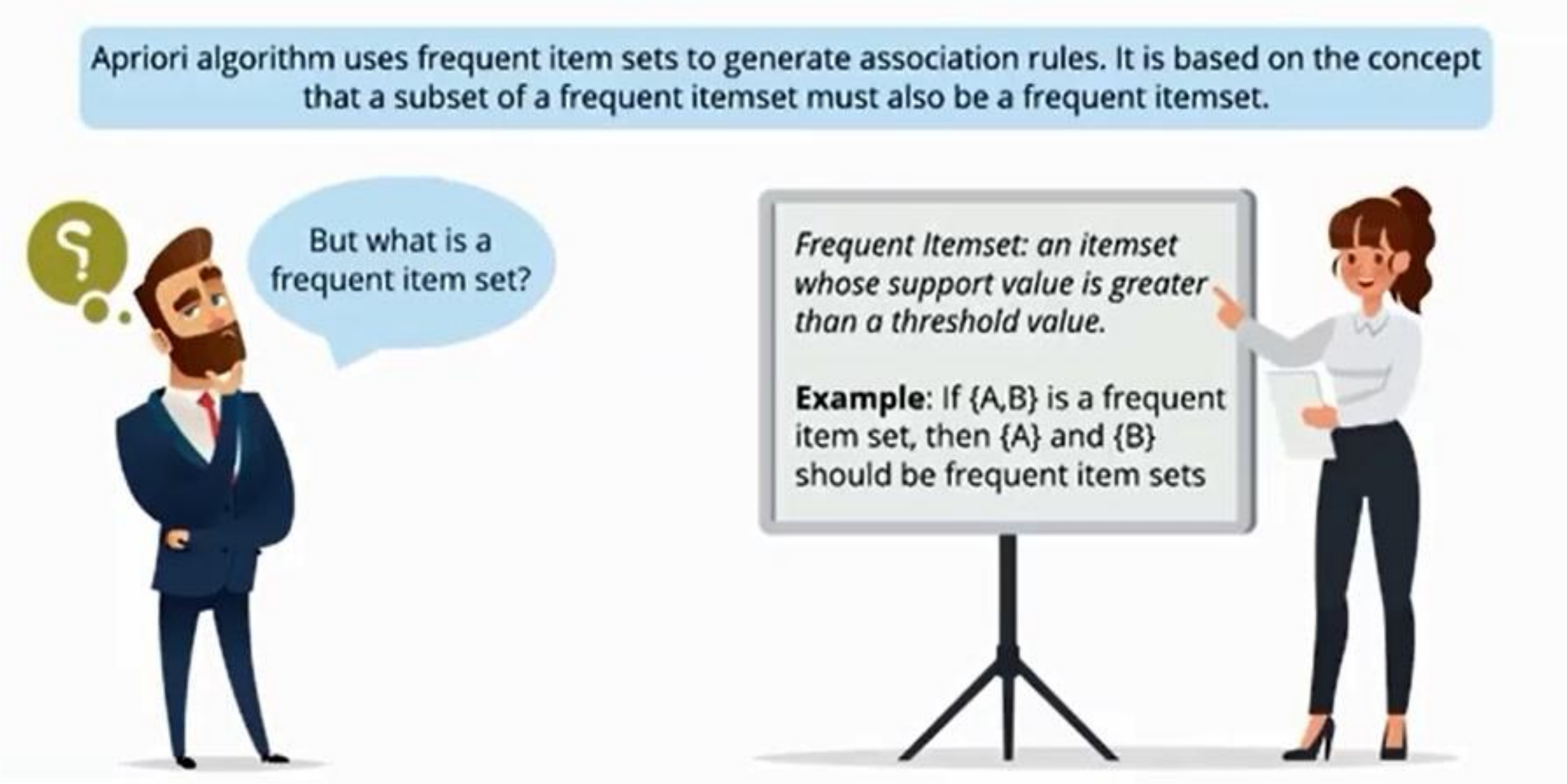
II

Itemset	Support
{1, 2}	1/4 → 25%
{1, 3}	2/4 → 50%
{1, 5}	1/4 → 25%
{2, 3}	2/4 → 50%
{2, 5}	3/4 → 75%
{3, 5}	2/4 → 50%

III

Itemset	Support
{1, 3, 5}	1/4 = 25%
{2, 3, 5}	2/4 = 50%
{1, 2, 3}	1/4 = 25%

Apriori Algorithm

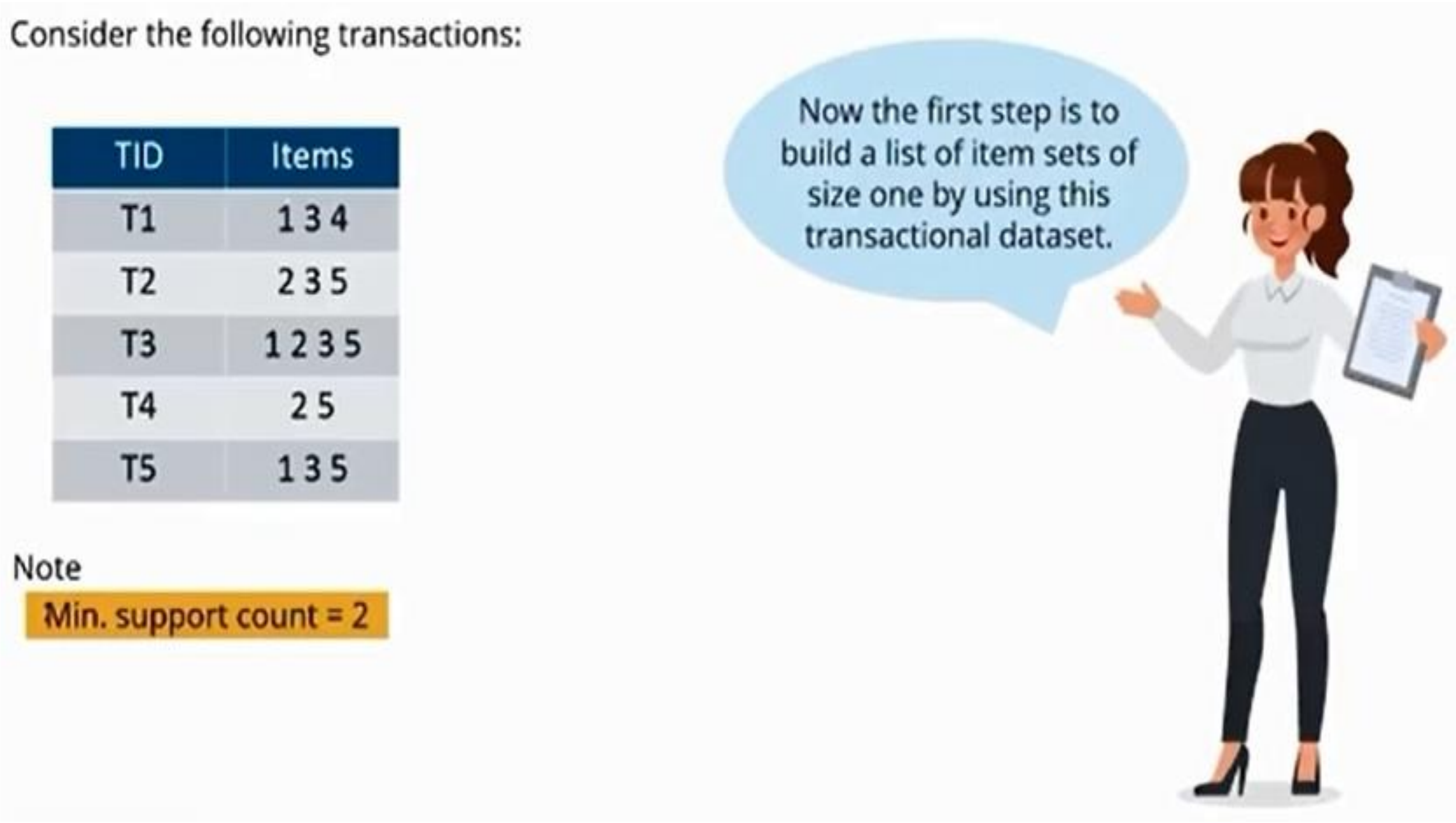


Apriori algorithm uses frequent item sets to generate association rules. It is based on the concept that a subset of a frequent itemset must also be a frequent itemset.

Frequent Itemset: an itemset whose support value is greater than a threshold value.

Example: If {A,B} is a frequent item set, then {A} and {B} should be frequent item sets

Apriori Algorithm



Consider the following transactions:

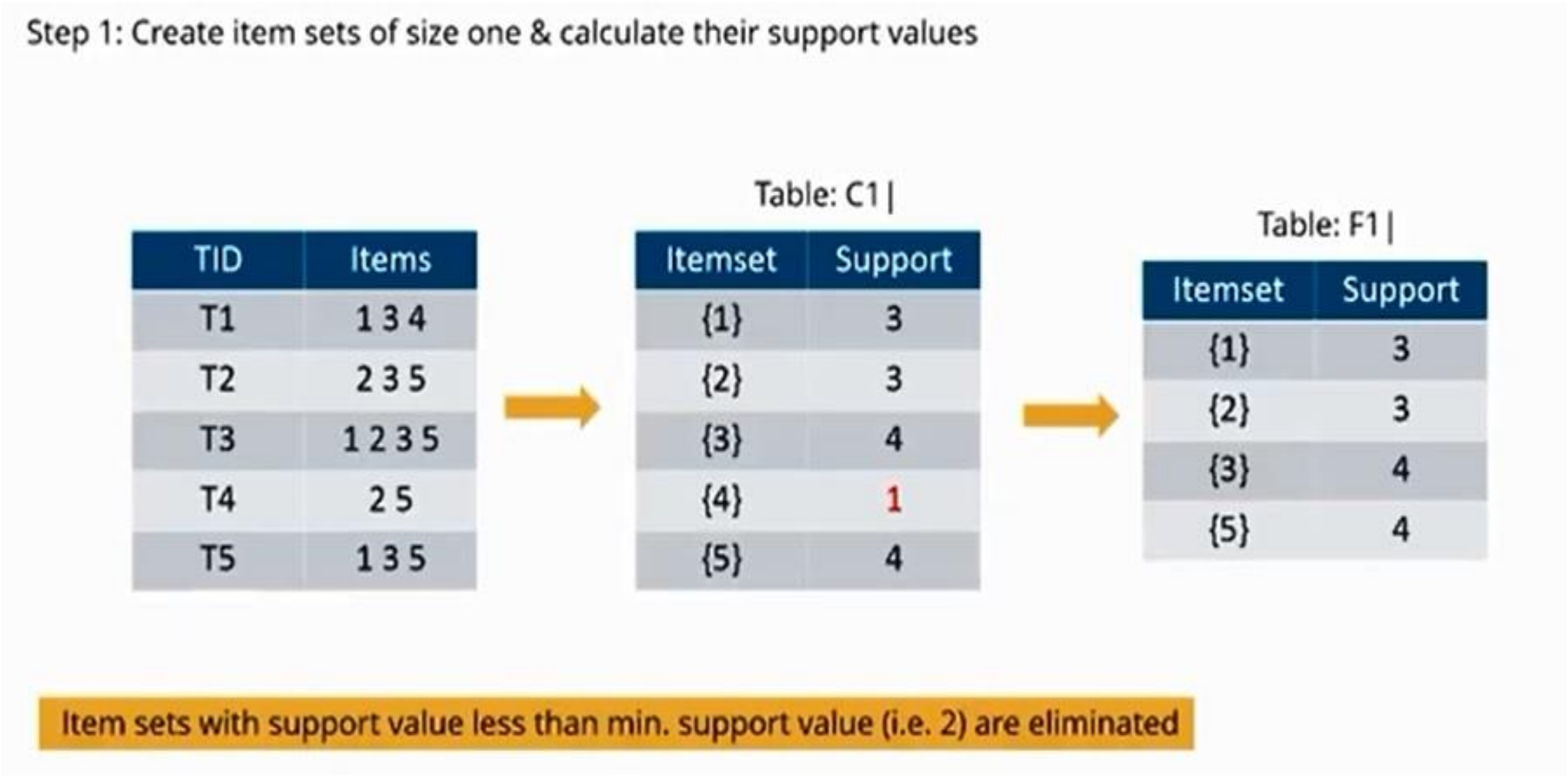
TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Note

Min. support count = 2

Now the first step is to build a list of item sets of size one by using this transactional dataset.

Apriori Algorithm-First Iteration



Step 1: Create item sets of size one & calculate their support values

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Table: C1 |

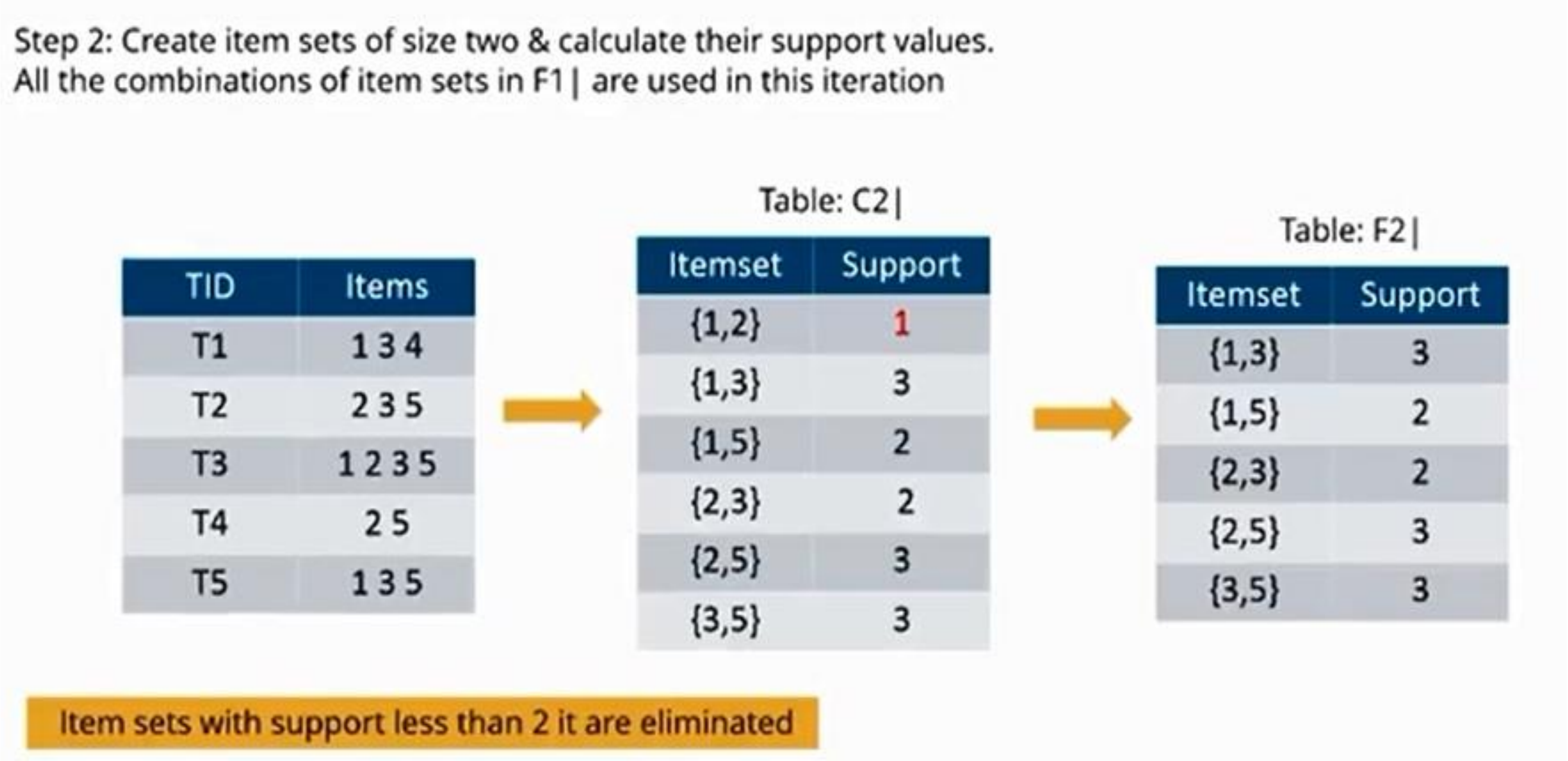
Itemset	Support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

Table: F1 |

Itemset	Support
{1}	3
{2}	3
{3}	4
{5}	4

Item sets with support value less than min. support value (i.e. 2) are eliminated

Apriori Algorithm- Second Iteration



Step 2: Create item sets of size two & calculate their support values.
All the combinations of item sets in F1| are used in this iteration

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Table: C2|

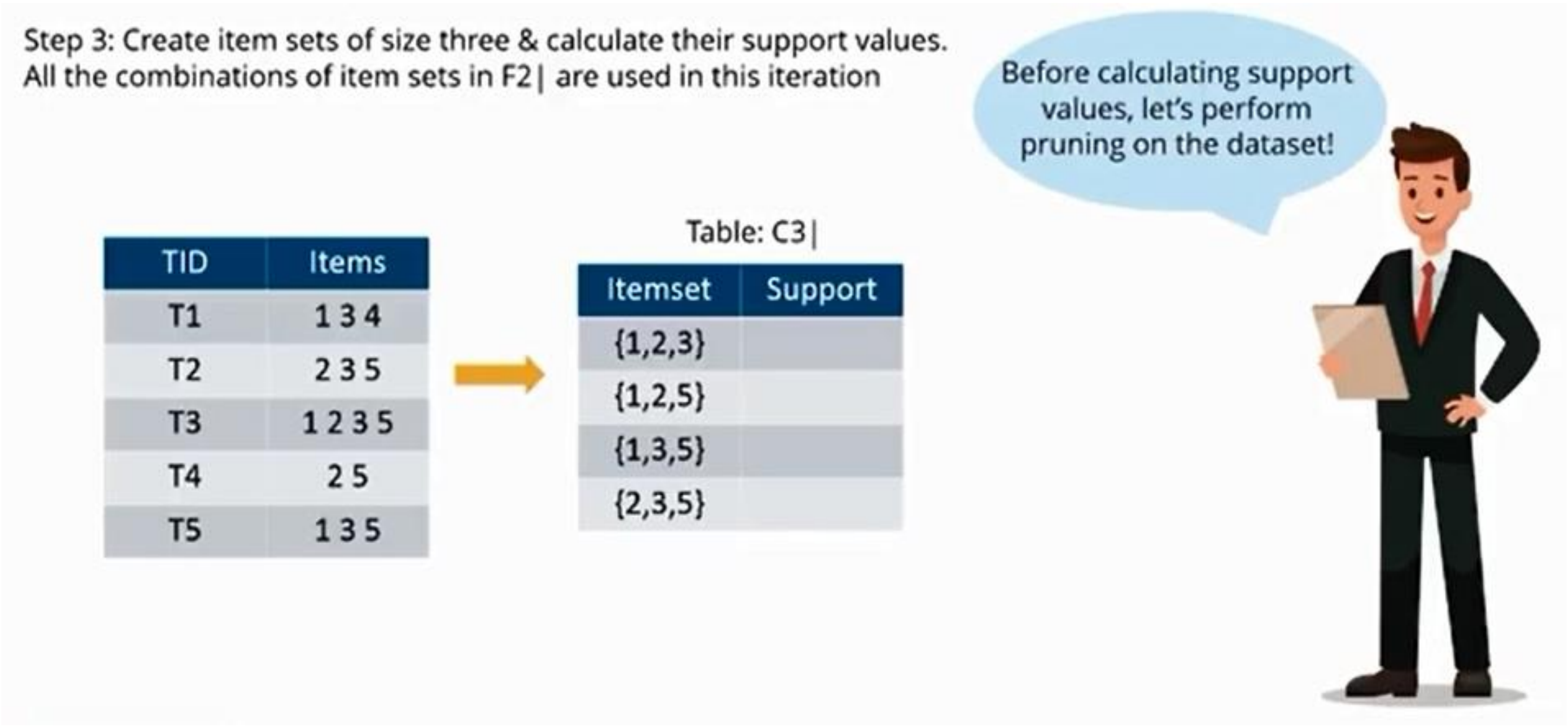
Itemset	Support
{1,2}	1
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

Table: F2|

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

Item sets with support less than 2 it are eliminated

Apriori Algorithm- Third Iteration



Step 3: Create item sets of size three & calculate their support values.
All the combinations of item sets in F2| are used in this iteration

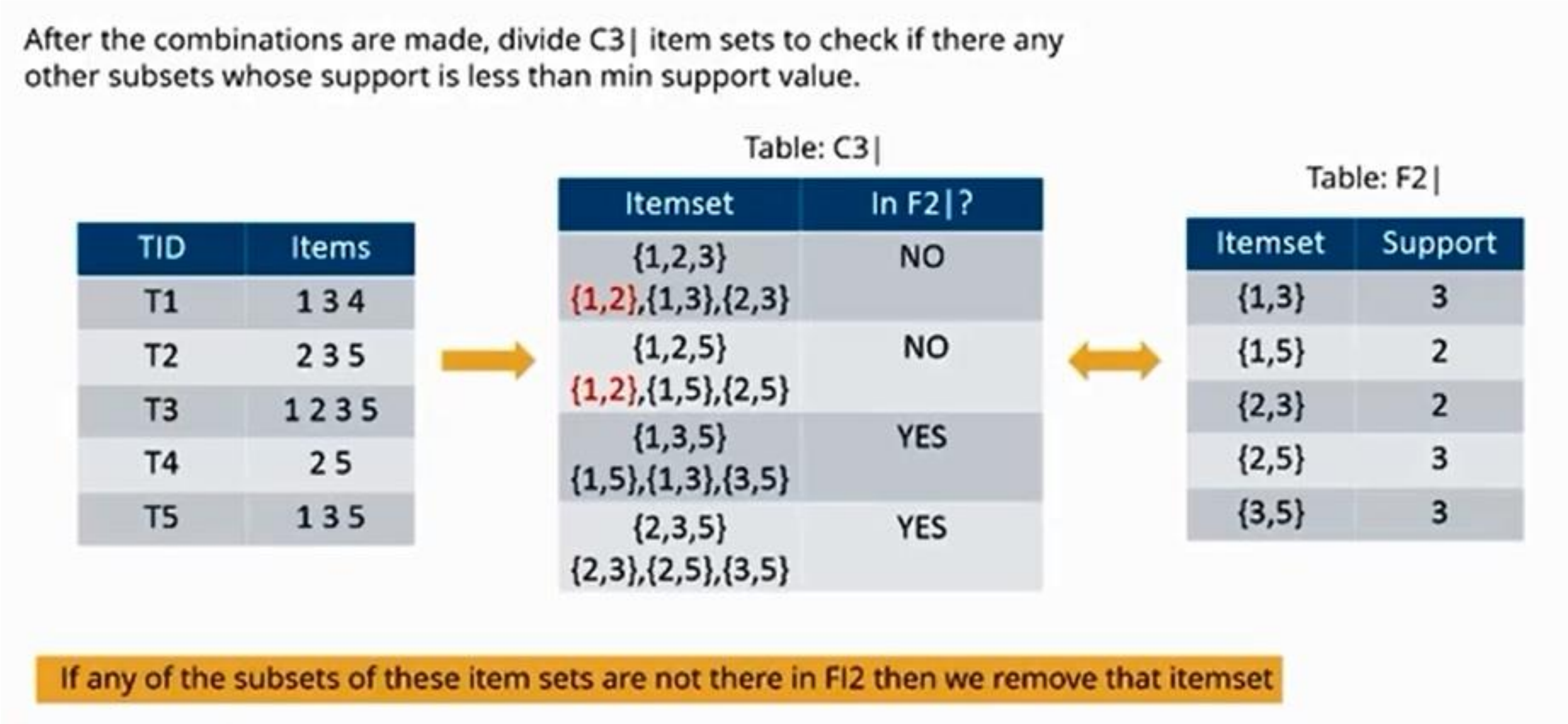
TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Table: C3|

Itemset	Support
{1,2,3}	
{1,2,5}	
{1,3,5}	
{2,3,5}	

Before calculating support values, let's perform pruning on the dataset!

Apriori Algorithm-Pruning



After the combinations are made, divide C3| item sets to check if there any other subsets whose support is less than min support value.

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Table: C3|

Itemset	In F2|?

{1,2,3}
{1,2},{1,3},{2,3}
	NO

{1,2,5}
{1,2},{1,5},{2,5}
	NO

{1,3,5}
{1,5},{1,3},{3,5}
	YES

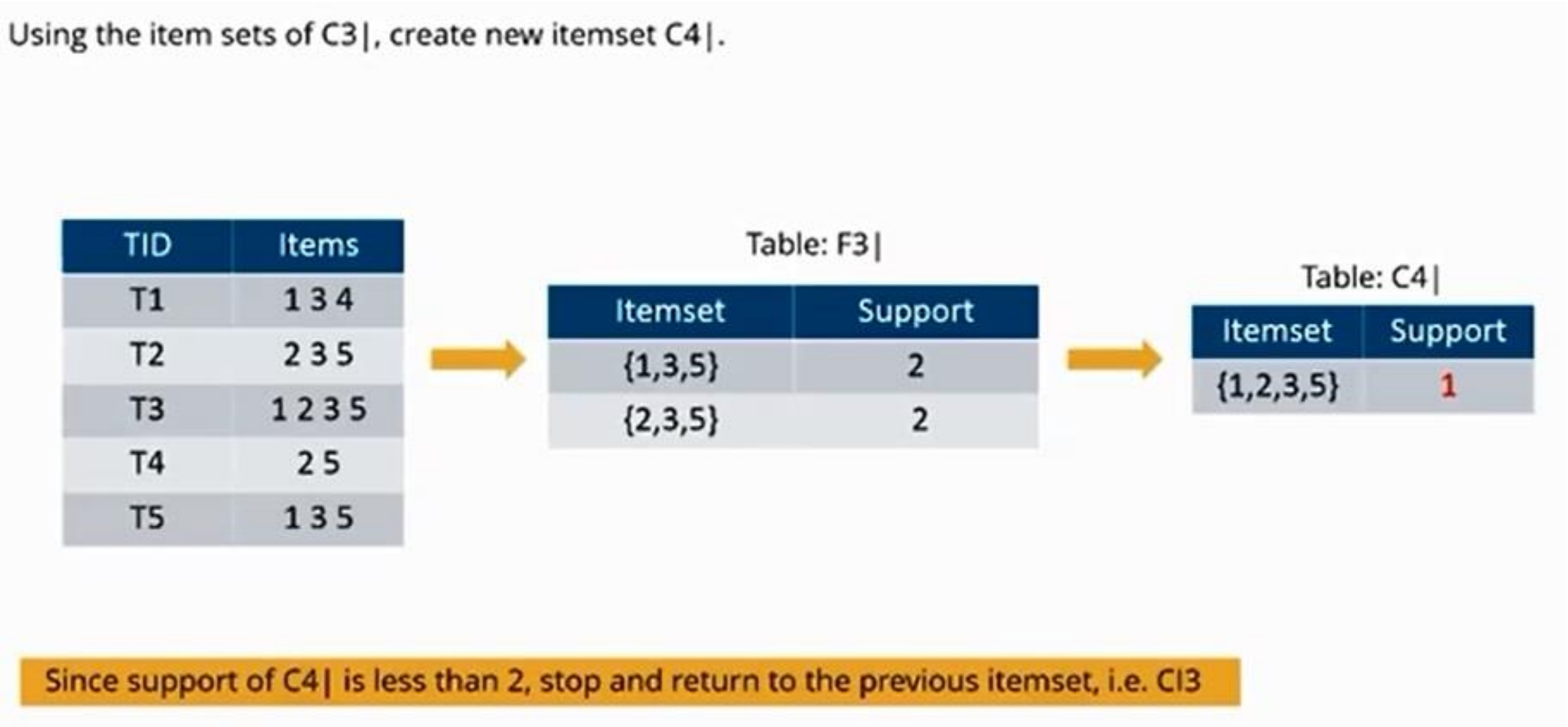
{2,3,5}
{2,3},{2,5},{3,5}
	YES

Table: F2|

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

If any of the subsets of these item sets are not there in FI2 then we remove that itemset

Apriori Algorithm-Fourth Iteration



Using the item sets of C3|, create new itemset C4|.

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Table: F3|

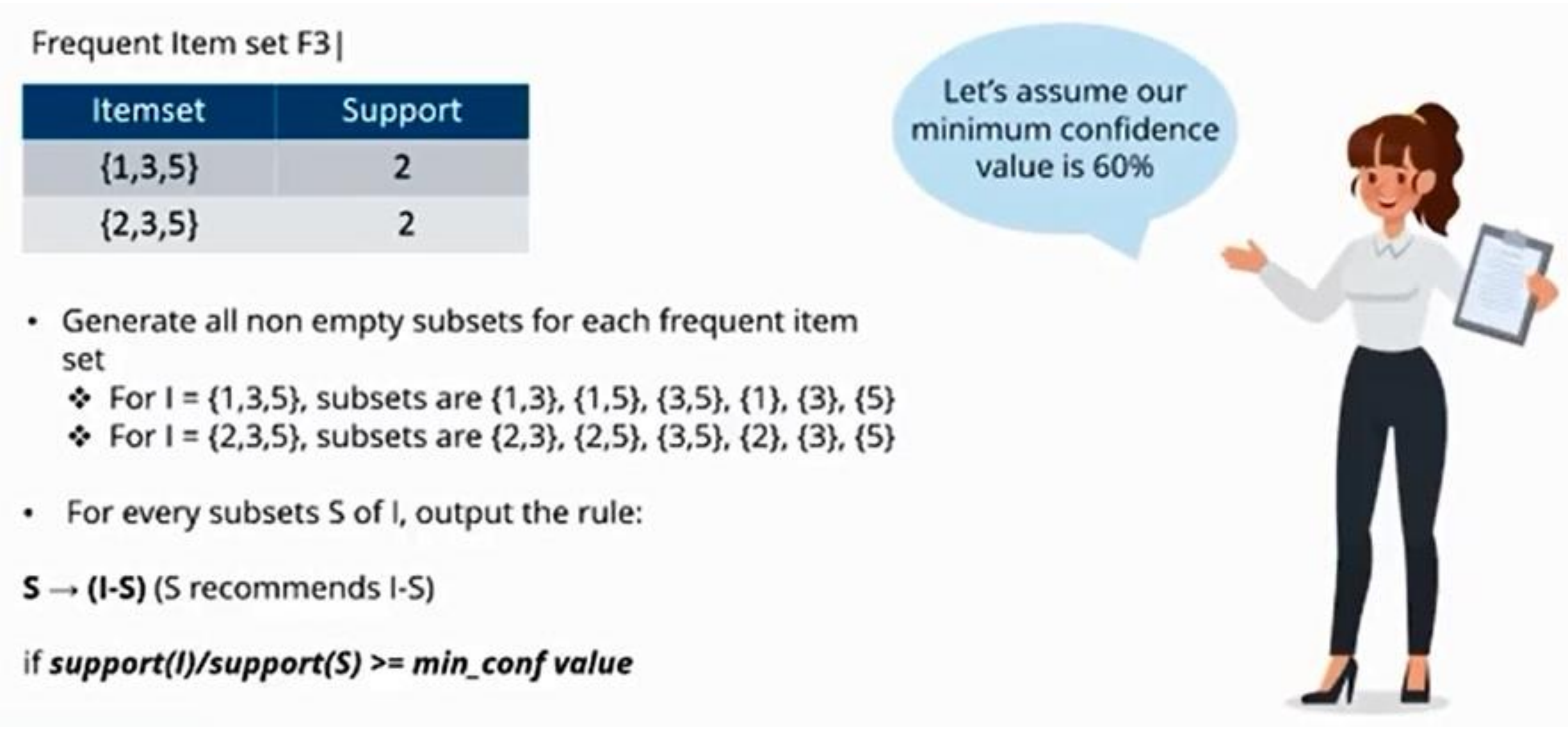
Itemset	Support
{1,3,5}	2
{2,3,5}	2

Table: C4|

Itemset	Support
{1,2,3,5}	1

Since support of C4| is less than 2, stop and return to the previous itemset, i.e. CI3

Apriori Algorithm-Subset Creation



Frequent Item set F3 |

Itemset	Support
{1,3,5}	2
{2,3,5}	2

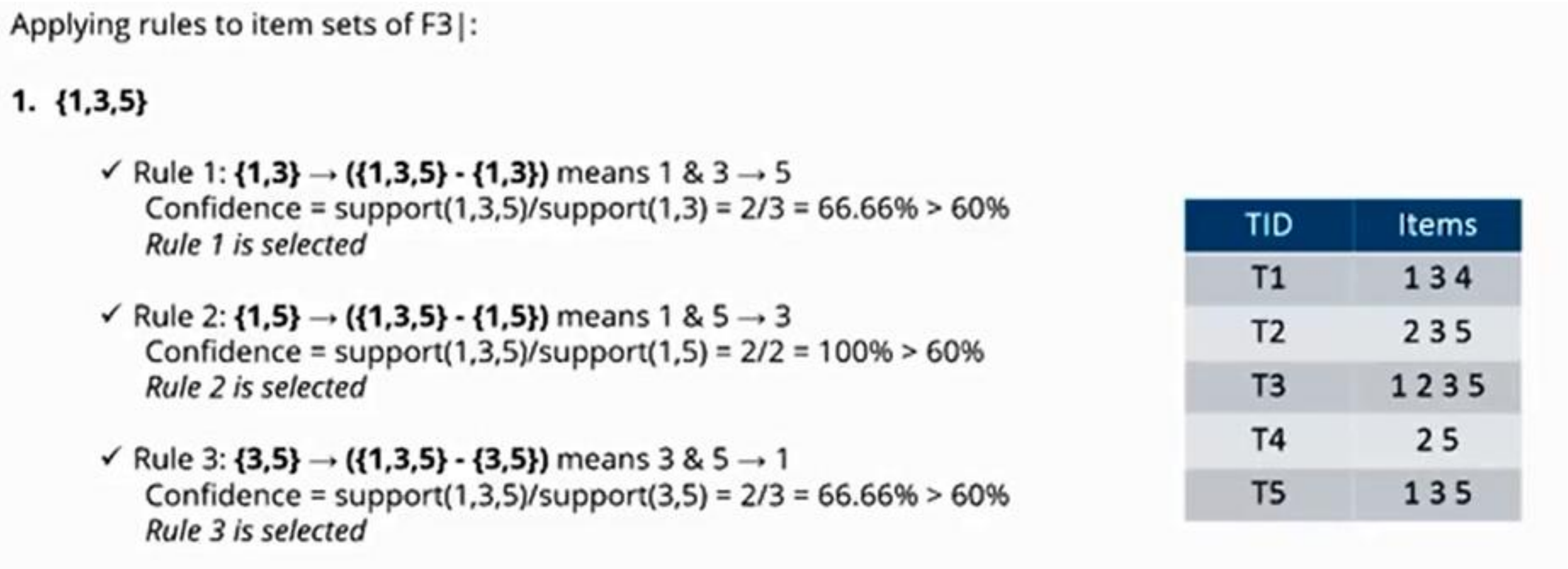
Let's assume our minimum confidence value is 60%

- Generate all non empty subsets for each frequent item set
  - ❖ For I = {1,3,5}, subsets are {1,3}, {1,5}, {3,5}, {1}, {3}, {5}
  - ❖ For I = {2,3,5}, subsets are {2,3}, {2,5}, {3,5}, {2}, {3}, {5}
- For every subsets S of I, output the rule:

S → (I-S) (S recommends I-S)

if ***support(I)/support(S) >= min_conf value***

Apriori Algorithm-Applying Rules



Applying rules to item sets of F3|:

1. {1,3,5}

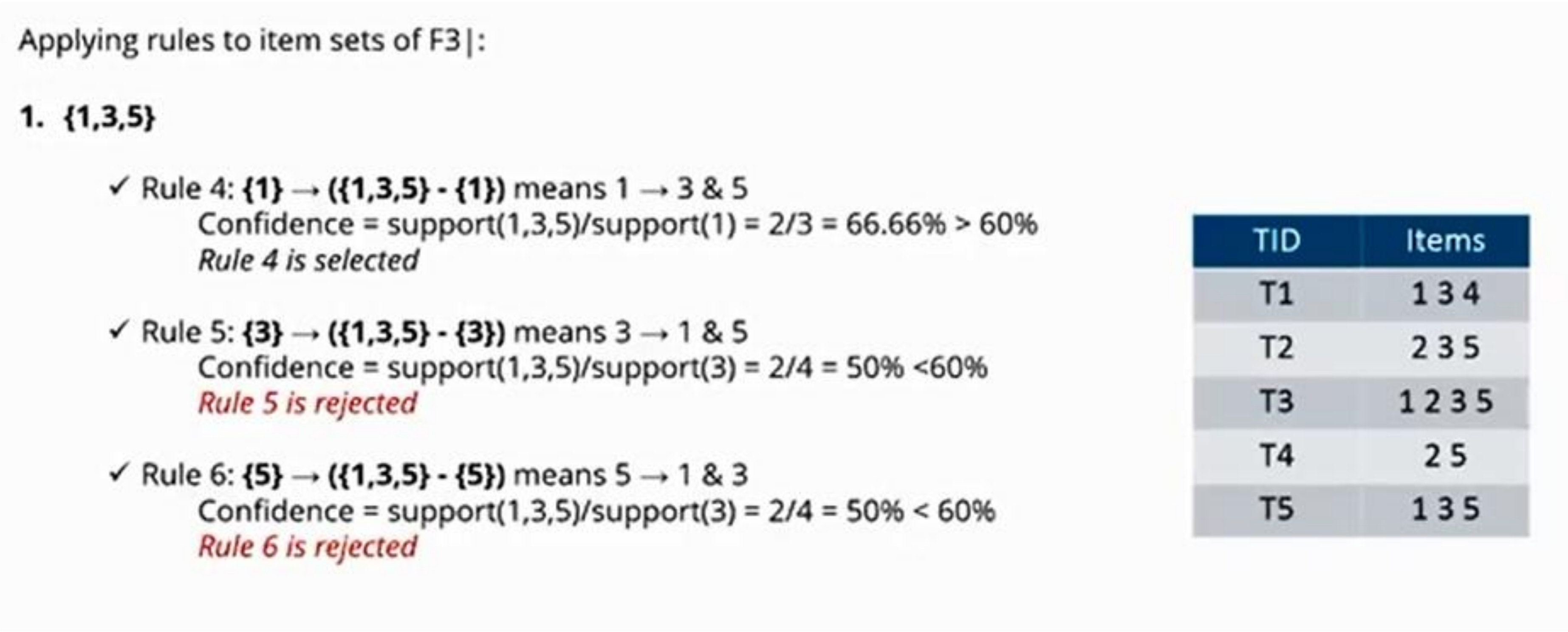
- ✓ Rule 1: **{1,3} → ({1,3,5} - {1,3})** means 1 & 3 → 5
  Confidence = support(1,3,5)/support(1,3) = 2/3 = 66.66% > 60%
  Rule 1 is selected

- ✓ Rule 2: **{1,5} → ({1,3,5} - {1,5})** means 1 & 5 → 3
  Confidence = support(1,3,5)/support(1,5) = 2/2 = 100% > 60%
  Rule 2 is selected

- ✓ Rule 3: **{3,5} → ({1,3,5} - {3,5})** means 3 & 5 → 1
  Confidence = support(1,3,5)/support(3,5) = 2/3 = 66.66% > 60%
  Rule 3 is selected

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Apriori Algorithm-Applying Rules



Applying rules to item sets of F3|:

1. **{1,3,5}**

✓ Rule 4: **{1}** → **({1,3,5} - {1})** means 1 → 3 & 5
Confidence = support(1,3,5)/support(1) = 2/3 = 66.66% > 60%
Rule 4 is selected

✓ Rule 5: **{3}** → **({1,3,5} - {3})** means 3 → 1 & 5
Confidence = support(1,3,5)/support(3) = 2/4 = 50% <60%
Rule 5 is rejected

✓ Rule 6: **{5}** → **({1,3,5} - {5})** means 5 → 1 & 3
Confidence = support(1,3,5)/support(3) = 2/4 = 50% < 60%
Rule 6 is rejected

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Diolch
Kiitos
Sheun
umesc
Dekuju/Dekujeme
Kasih
Todah
Mamnoon
Shnorhakalutiun
Teşekkür
Hvala
Shokriya
Dziękuję
Shokrun
Spaas
Mul
Ači
Gamsahapnida
Ngiyabonga
Xie
Dank
Takk
Cam
Gra
or
Dhanyavaad
Dhanyavadalu
Waad
Kop
Salamat
Merci
al
Dakujem
Dankie
Go
Grazie
Faleminderit
Daw
Dhanyavad
Khopjai
Kruthagnathalu
ThankYou
kun
Arigatou
krap
Tack
Shukriya
Or
Dhonnobaad
ederim
Asante
Grazzi
raibh
Gracias
Nandree
Blagodariya
Gomapsupnida
Euxaristo
Kun
Hain
Dhan
Fyrir
Terima
Enkosi
Danke
dank
daa