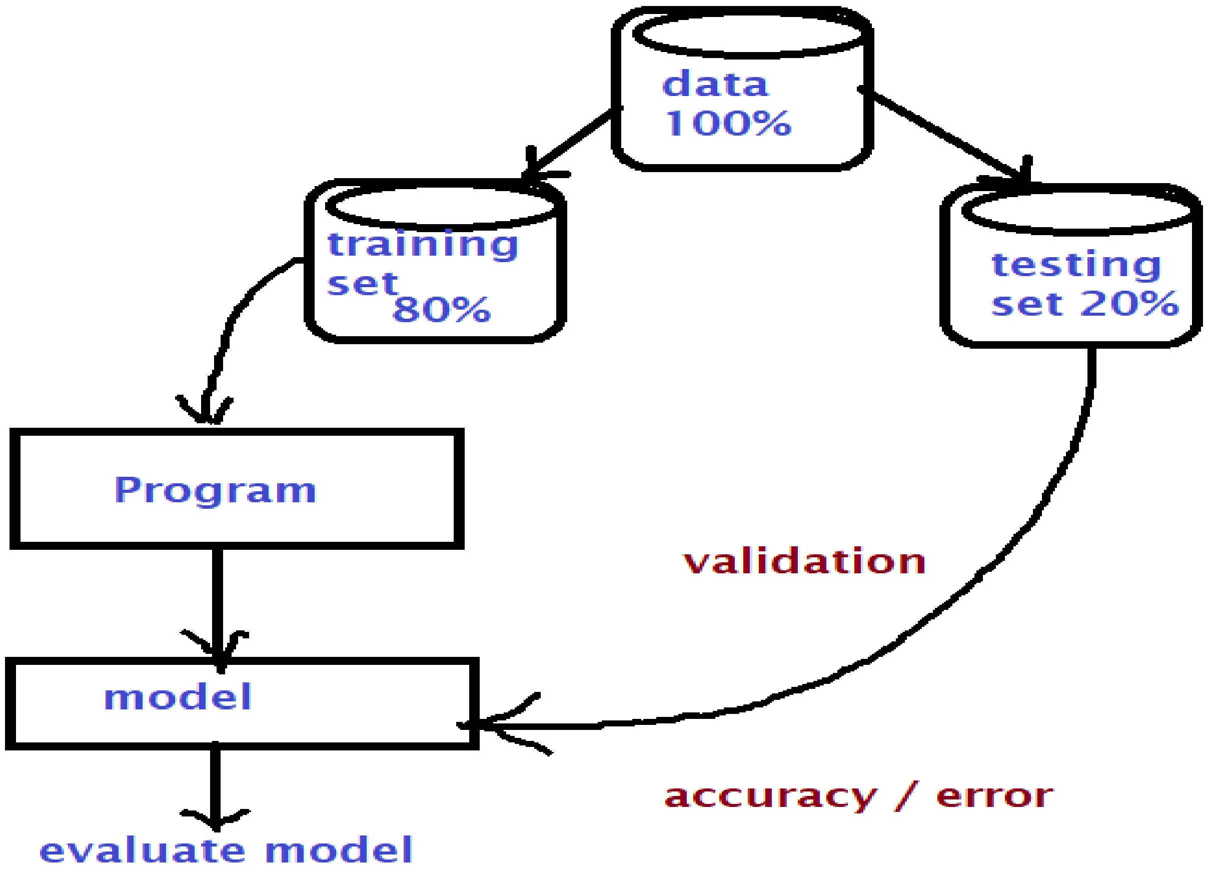
Embedded AI

Dr.Akshita Chanchlani

Agenda

- Clustering
- Association

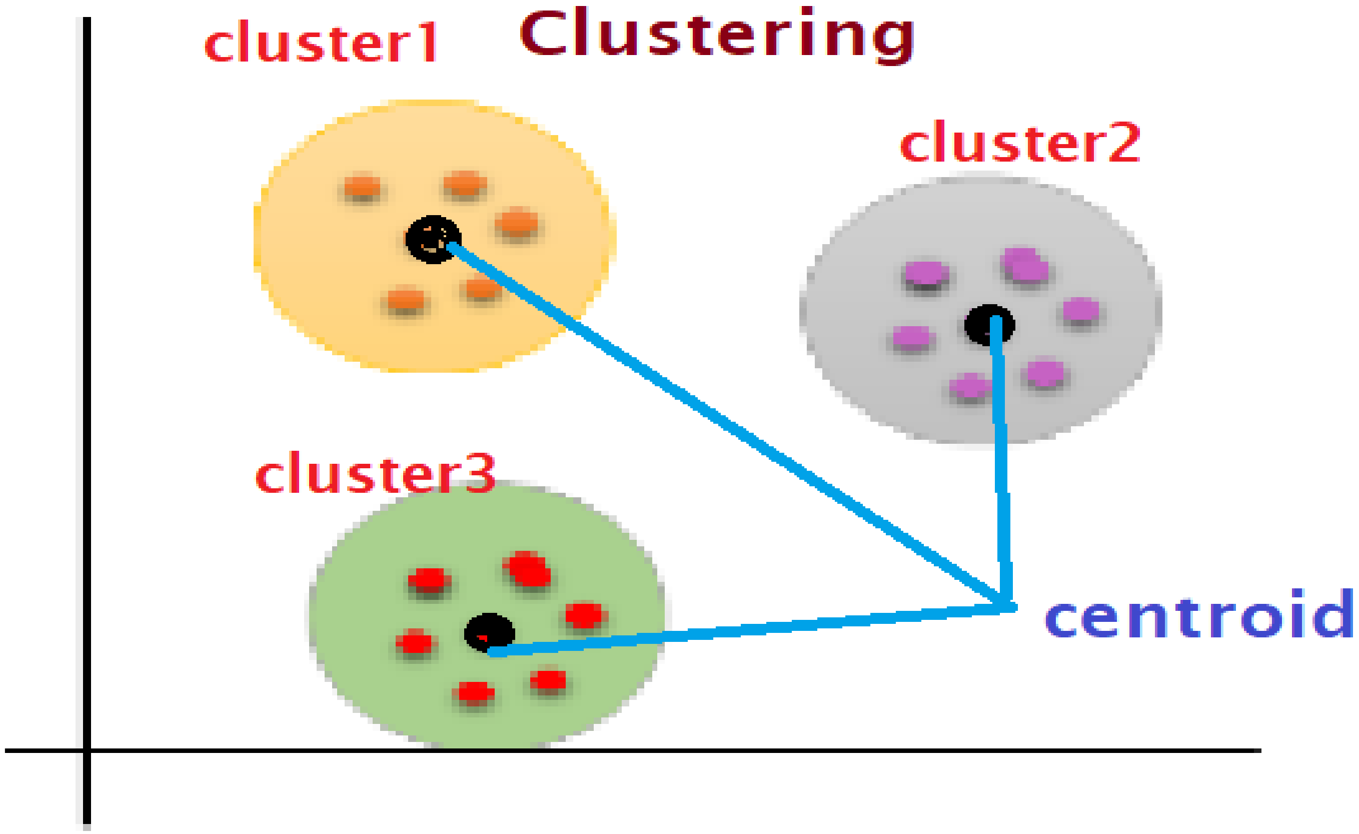
Supervised Learning



Model

- Linear Regression
- SVM
- KNN

Unsupervised Learning



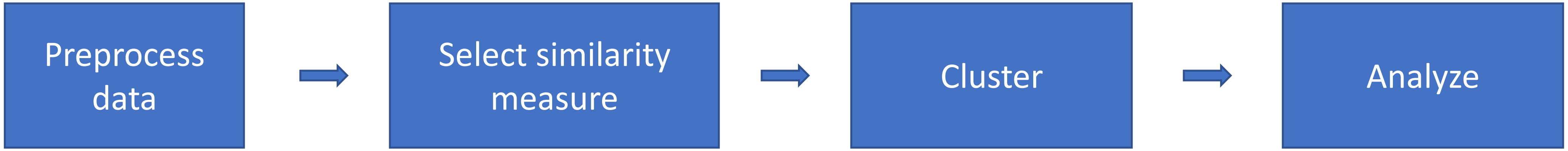
Overview

- **Clustering** is one of the most common **Exploratory Data Analysis(EDA)** technique used to get an intuition about the structure of the data
- It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different
- In other words, we try to find **homogeneous subgroups** within the data such that data points in each cluster are as similar as possible according to a similarity measure such as Euclidean-based distance or correlation-based distance
- The decision of which similarity measure to use is application-specific
- Unlike supervised learning, clustering is considered an unsupervised learning method since we don't have the ground truth (dependent variable) to compare the output of the clustering algorithm to the true labels to evaluate its performance

Overview

Based on
- distance
- correlation

K-Means
C-Means/ Fuzzy
Hierarchical



Preprocess data → Select similarity measure → Cluster → Analyze

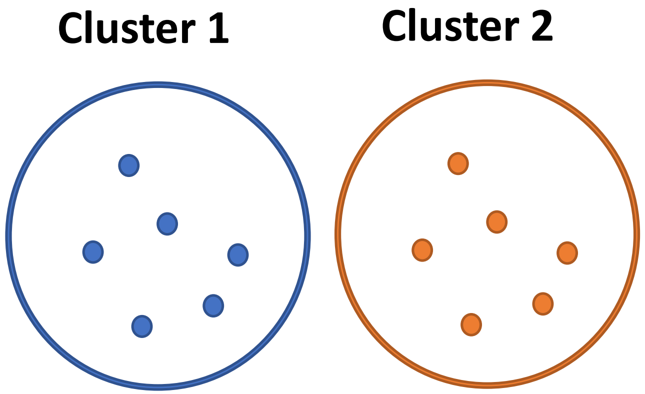
1. Exclusive
2. Overlapping
3. Hierarchical

Use Cases

- Marketing
  - Discovering groups in customer databases like who makes long-distance calls or who are earning more or who are spending more
- Insurance
  - Identifying groups of insurance policy holder with high claim rate
- Land use
  - Identification of areas of similar land use in GIS(Geographic Information System) database

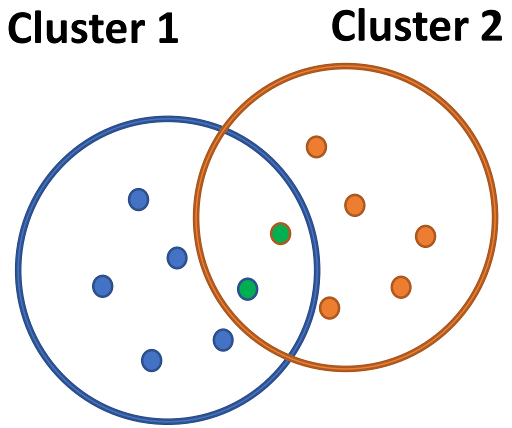
Types : Exclusive clustering

- **Exclusive clustering**
  - An item belongs exclusively to one cluster and not several
  - E.g. K-Means clustering



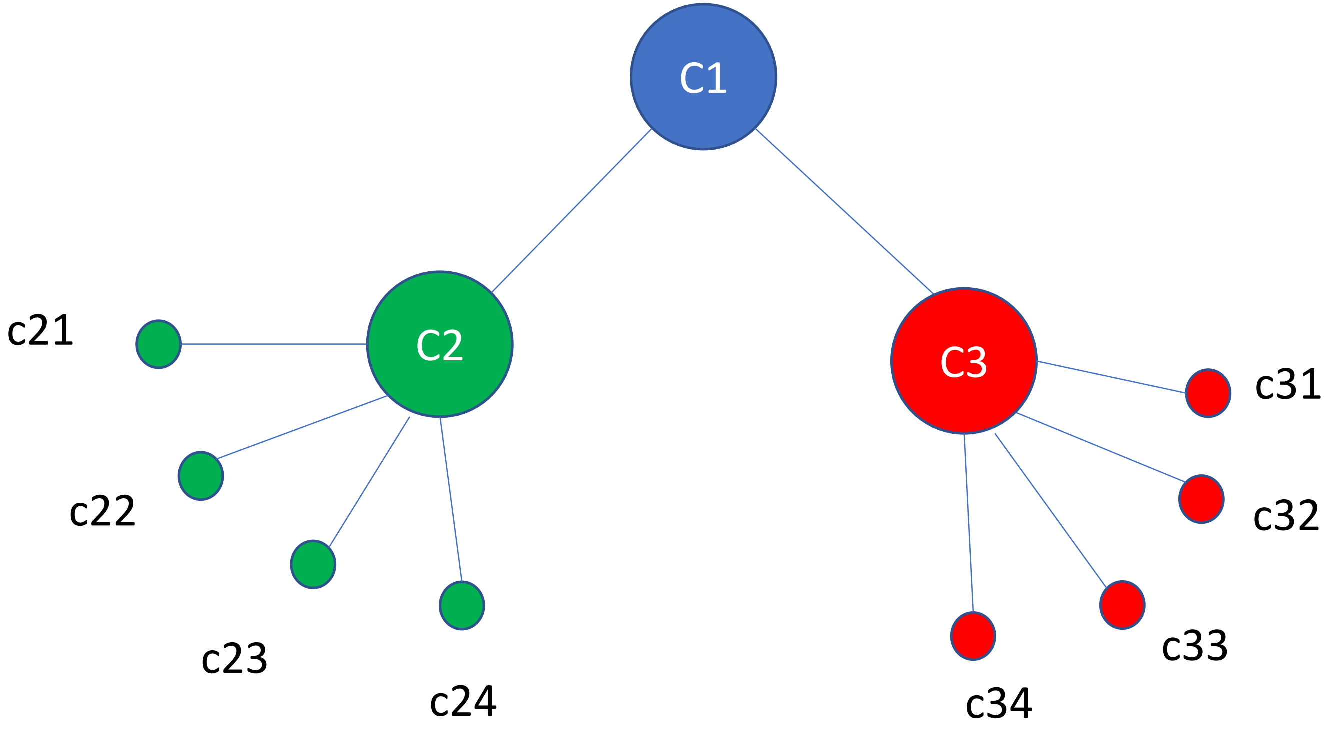
Types :Overlapping clustering

- **Overlapping clustering**
  - An Item can belong to multiple clusters
  - Its degree of association with each cluster is known
  - E.g. Fuzzy/C-means clustering



Types : Hierarchical clustering

- Hierarchical clustering
  - When two clusters have a parent child relationship
  - It forms a tree like structure
  - E.g. Hierarchical clustering



KMeans

Overview

- **Kmeans** algorithm is an iterative algorithm that tries to partition the dataset into distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**
- It tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different (far) as possible
- It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster’s centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum
- The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster

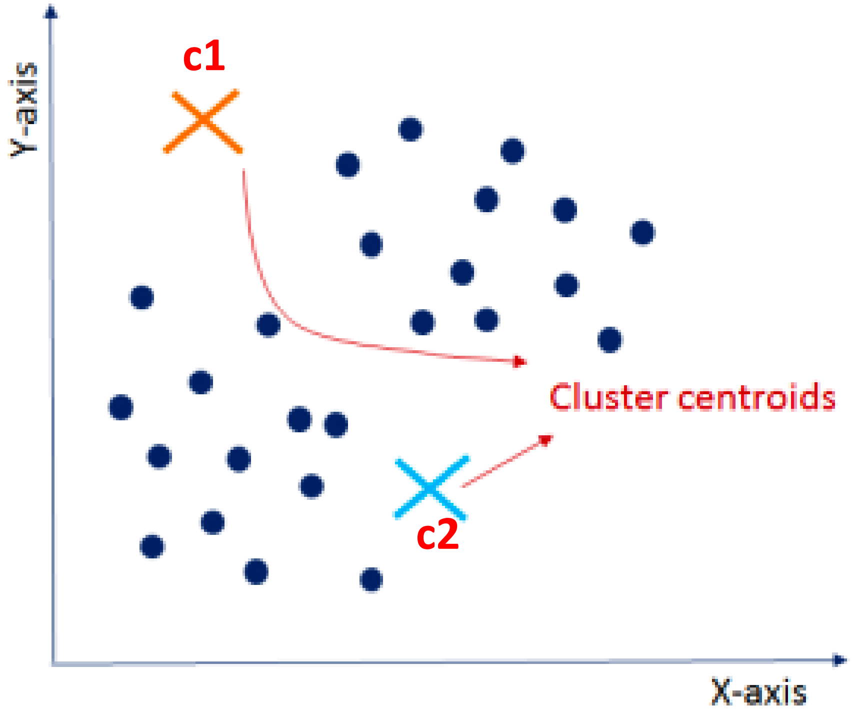
How does it work?

- Specify number of clusters *K*
- Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement
- Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn’t changing
- Compute the sum of the squared distance between data points and all centroids
- Assign each data point to the closest cluster (centroid)
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster

K-Means Clustering - Algorithm

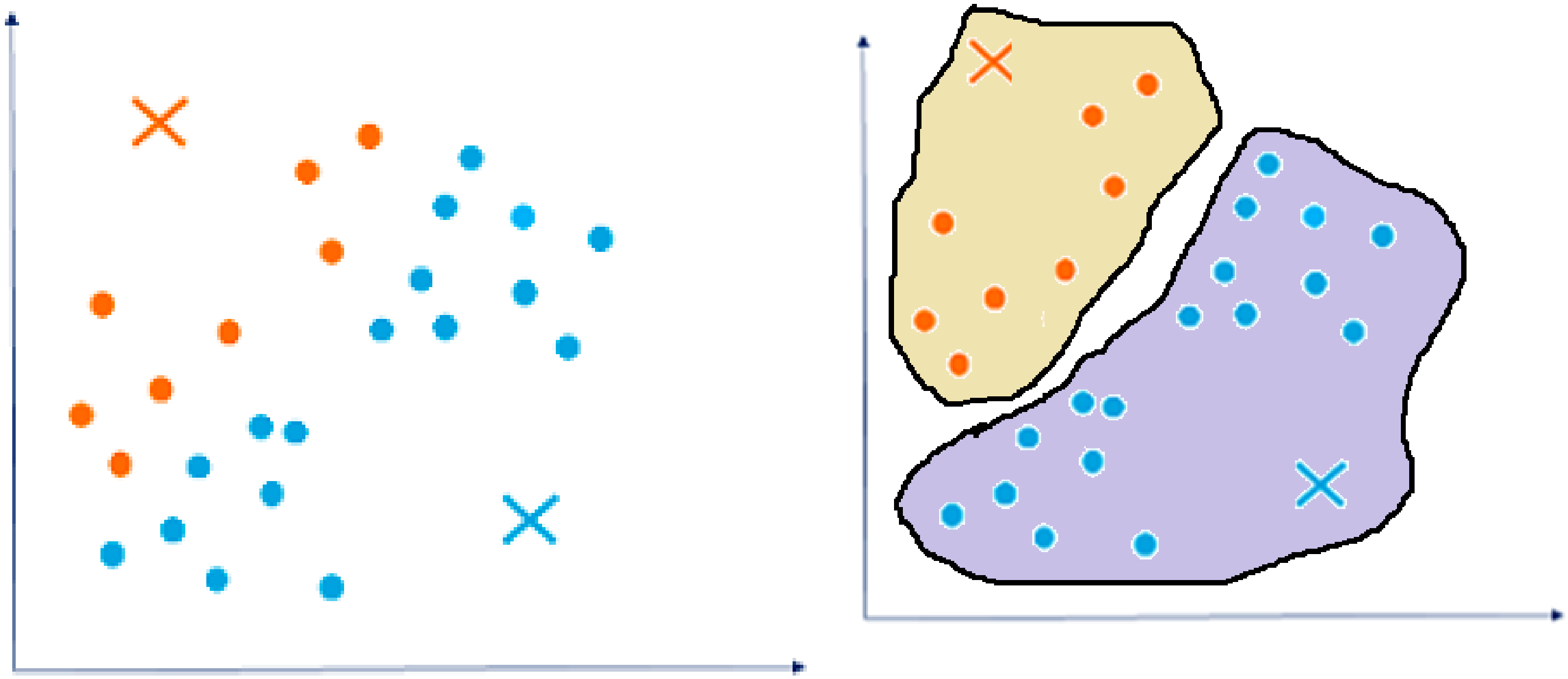
- **Initialization**
    - randomly initialise two points called the cluster centroids

Consider,
K = 2



K-Means Clustering - Algorithm

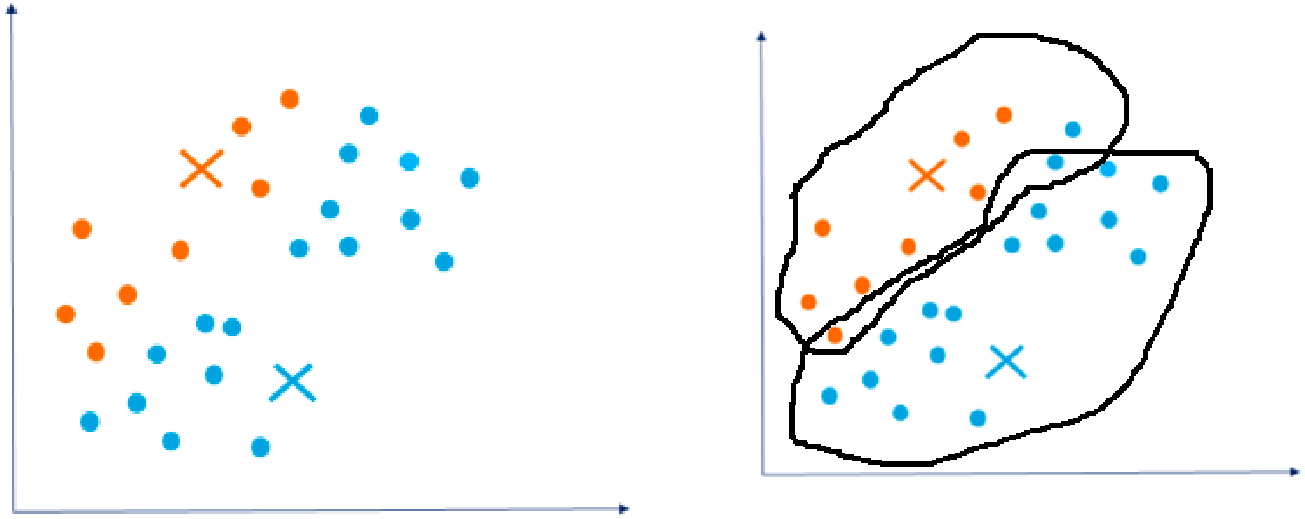
- **Cluster Assignment**
  - Compute the distance between both the points and centroids
  - Depending on the minimum distance from the centroid divide the points into two clusters



K-Means Clustering - Algorithm

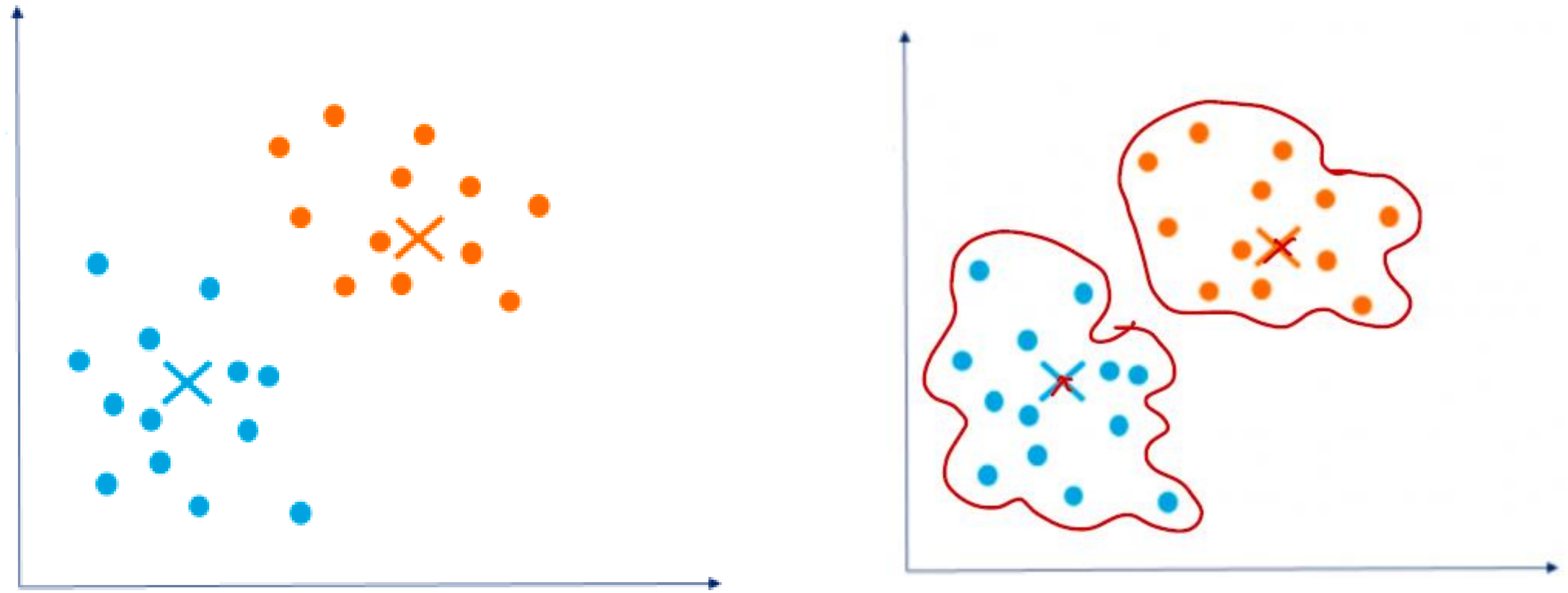
- **Move Centroid**
  - Consider the older centroids are data points
  - Take the older centroid and iteratively reposition them for optimization

- **Optimization**
  - Repeat the steps until the cluster centroids stop changing the position



K-Means Clustering - Algorithm

- **Convergence**
  - Finally, k-means clustering algorithm converges and divides the data points into two clusters clearly visible in multiple clusters



K-Means Clustering - Example

- Suppose we want to group the visitors to a website using just their age (one-dimensional space) as follows:

15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65

N = 19

K-Means Clustering - Example

- Initial clusters (random centroid or average)

$\boldsymbol{k = 2}$

$c_1 = 16$

$c_2 = 22$

$$\text{Distance } 1 = |x_i - c_1|$$

$$\text{Distance } 2 = |x_i - c_2|$$

K-Means Clustering - Example

- **Iteration I**

Before:

$c_1 = 16$

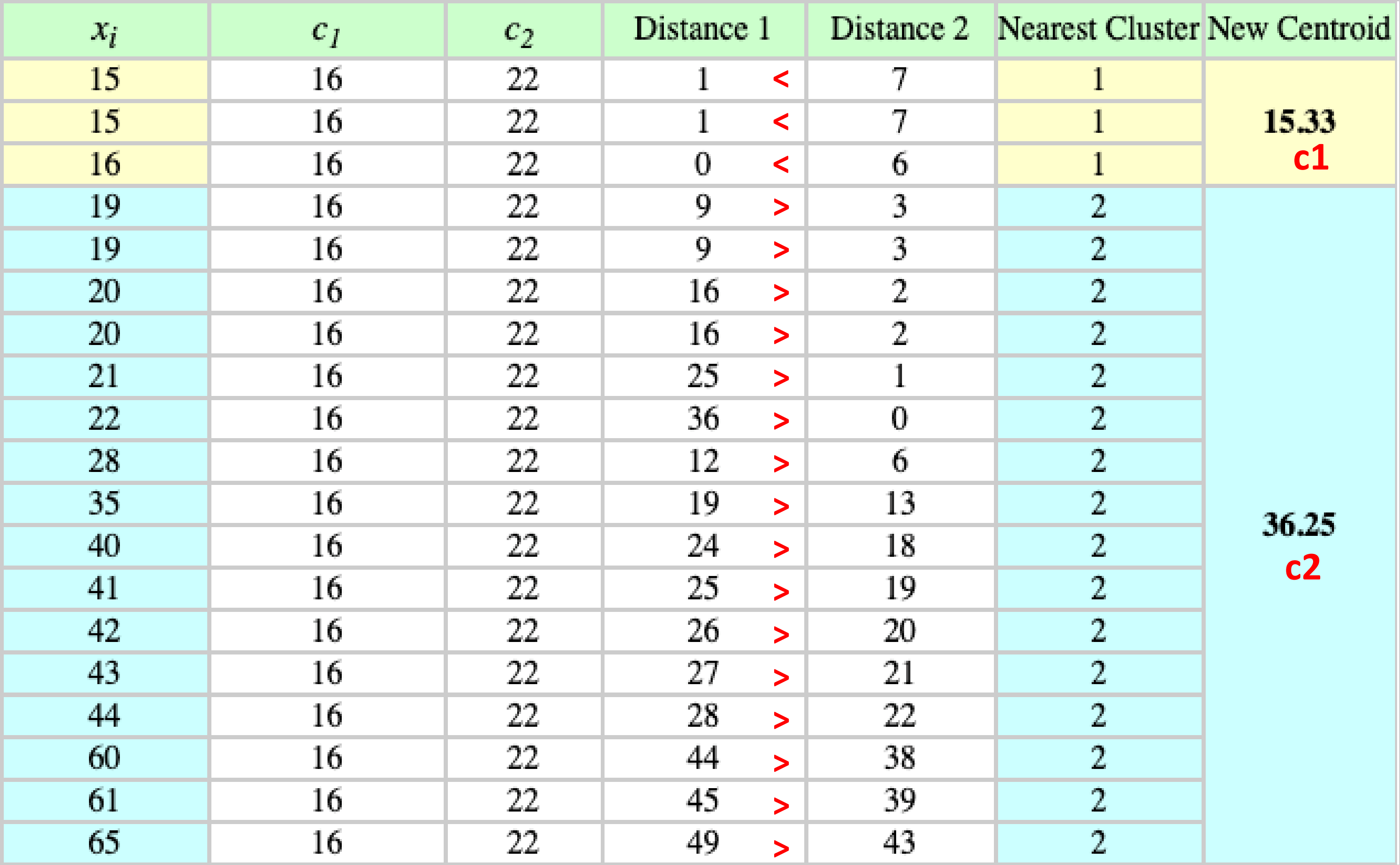
$c_2 = 22$

After:

$c_1 = 15.33$

$c_2 = 36.25$

X1→



x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	16	22	1 <	7	1	15.33 c1
15	16	22	1 <	7	1	
16	16	22	0 <	6	1	
19	16	22	9 >	3	2	36.25 c2
19	16	22	9 >	3	2	
20	16	22	16 >	2	2	
20	16	22	16 >	2	2	
21	16	22	25 >	1	2	
22	16	22	36 >	0	2	
28	16	22	12 >	6	2	
35	16	22	19 >	13	2	
40	16	22	24 >	18	2	
41	16	22	25 >	19	2	
42	16	22	26 >	20	2	
43	16	22	27 >	21	2	
44	16	22	28 >	22	2	
60	16	22	44 >	38	2	
61	16	22	45 >	39	2	
65	16	22	49 >	43	2	

K-Means Clustering - Example

- **Iteration II**

Before:

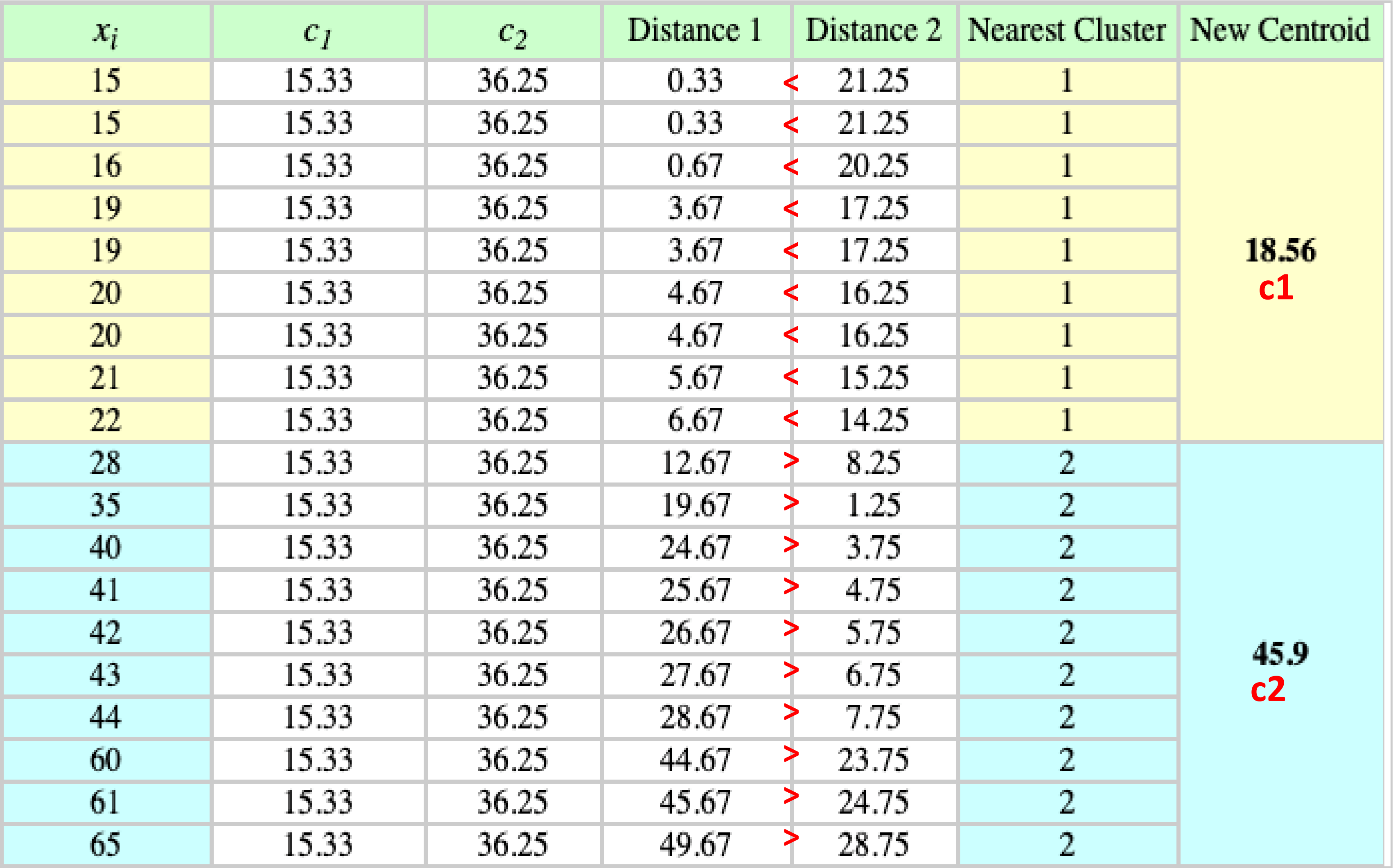
$c_1 = 15.33$

$c_2 = 36.25$

After:

$c_1 = 18.56$

$c_2 = 45.9$



x_i	c_1	c_2	Distance 1		Distance 2	Nearest Cluster	New Centroid
15	15.33	36.25	0.33	<	21.25	1	**18.56** c1
15	15.33	36.25	0.33	<	21.25	1	
16	15.33	36.25	0.67	<	20.25	1	
19	15.33	36.25	3.67	<	17.25	1	
19	15.33	36.25	3.67	<	17.25	1	
20	15.33	36.25	4.67	<	16.25	1	
20	15.33	36.25	4.67	<	16.25	1	
21	15.33	36.25	5.67	<	15.25	1	
22	15.33	36.25	6.67	<	14.25	1	
28	15.33	36.25	12.67	>	8.25	2	**45.9** c2
35	15.33	36.25	19.67	>	1.25	2	
40	15.33	36.25	24.67	>	3.75	2	
41	15.33	36.25	25.67	>	4.75	2	
42	15.33	36.25	26.67	>	5.75	2	
43	15.33	36.25	27.67	>	6.75	2	
44	15.33	36.25	28.67	>	7.75	2	
60	15.33	36.25	44.67	>	23.75	2	
61	15.33	36.25	45.67	>	24.75	2	
65	15.33	36.25	49.67	>	28.75	2	

K-Means Clustering - Example

- **Iteration III**

Before:

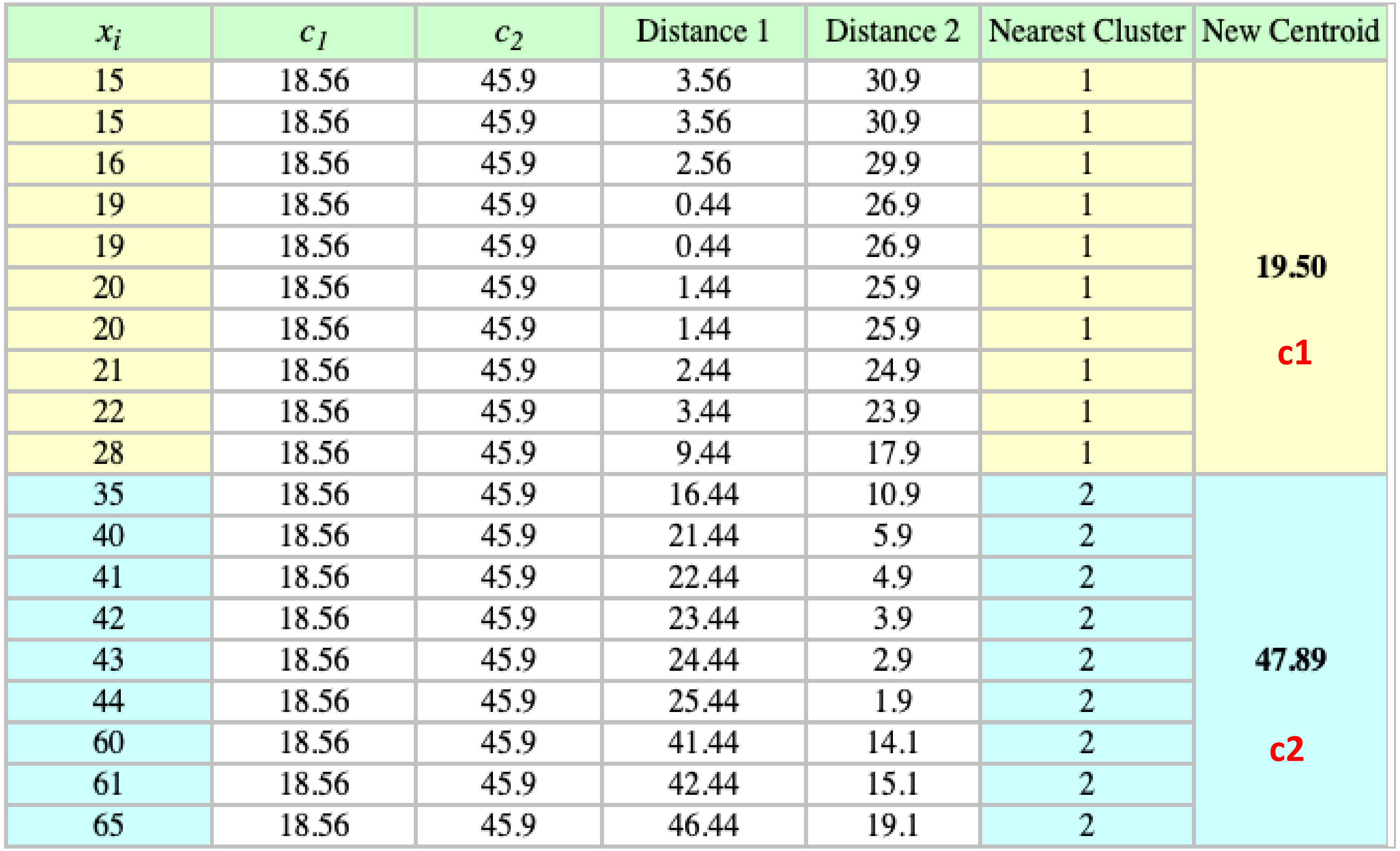
c_1 = 18.56

c_2 = 45.9

After:

c_1 = 19.50

c_2 = 47.89



x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	18.56	45.9	3.56	30.9	1	**19.50** c1
15	18.56	45.9	3.56	30.9	1	
16	18.56	45.9	2.56	29.9	1	
19	18.56	45.9	0.44	26.9	1	
19	18.56	45.9	0.44	26.9	1	
20	18.56	45.9	1.44	25.9	1	
20	18.56	45.9	1.44	25.9	1	
21	18.56	45.9	2.44	24.9	1	
22	18.56	45.9	3.44	23.9	1	
28	18.56	45.9	9.44	17.9	1	
35	18.56	45.9	16.44	10.9	2	**47.89** c2
40	18.56	45.9	21.44	5.9	2	
41	18.56	45.9	22.44	4.9	2	
42	18.56	45.9	23.44	3.9	2	
43	18.56	45.9	24.44	2.9	2	
44	18.56	45.9	25.44	1.9	2	
60	18.56	45.9	41.44	14.1	2	
61	18.56	45.9	42.44	15.1	2	
65	18.56	45.9	46.44	19.1	2	

K-Means Clustering - Example

- **Iteration IV**

Before:

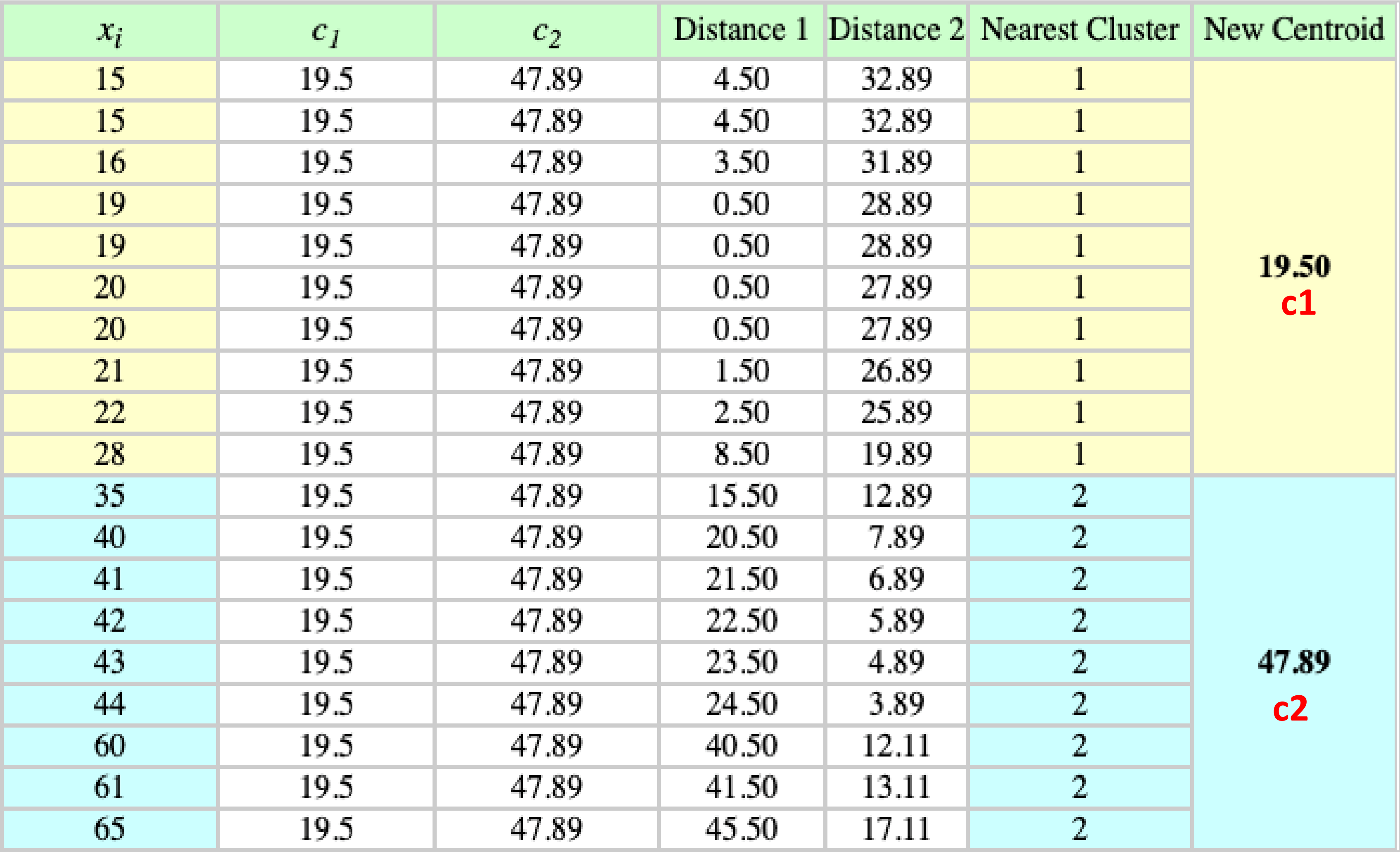
c_1 = 19.50

c_2 = 47.89

After:

c_1 = 19.50

c_2 = 47.89



x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	19.5	47.89	4.50	32.89	1	**19.50** c1
15	19.5	47.89	4.50	32.89	1	
16	19.5	47.89	3.50	31.89	1	
19	19.5	47.89	0.50	28.89	1	
19	19.5	47.89	0.50	28.89	1	
20	19.5	47.89	0.50	27.89	1	
20	19.5	47.89	0.50	27.89	1	
21	19.5	47.89	1.50	26.89	1	
22	19.5	47.89	2.50	25.89	1	
28	19.5	47.89	8.50	19.89	1	
35	19.5	47.89	15.50	12.89	2	**47.89** c2
40	19.5	47.89	20.50	7.89	2	
41	19.5	47.89	21.50	6.89	2	
42	19.5	47.89	22.50	5.89	2	
43	19.5	47.89	23.50	4.89	2	
44	19.5	47.89	24.50	3.89	2	
60	19.5	47.89	40.50	12.11	2	
61	19.5	47.89	41.50	13.11	2	
65	19.5	47.89	45.50	17.11	2	

Association Rule Mining

Enviro-Log
$3.99
Organic Cones

What are association rules?

- Association Rules is one of the very important concepts of machine learning being used in market **basket analysis**
- In a store, all vegetables are placed in the same aisle, all dairy items are placed together and cosmetics form another set of such groups
- Investing time and resources on deliberate product placements like this not only reduces a customer’s shopping time, but also reminds the customer of what relevant items (s)he might be interested in buying, thus helping stores cross-sell in the process
- Association rules help uncover all such relationships between items from huge databases

Applications

- Finding the set of items that has significant impact on business
- Collection information from numerous transactions (list of items)
- Generating rules from count in transactions

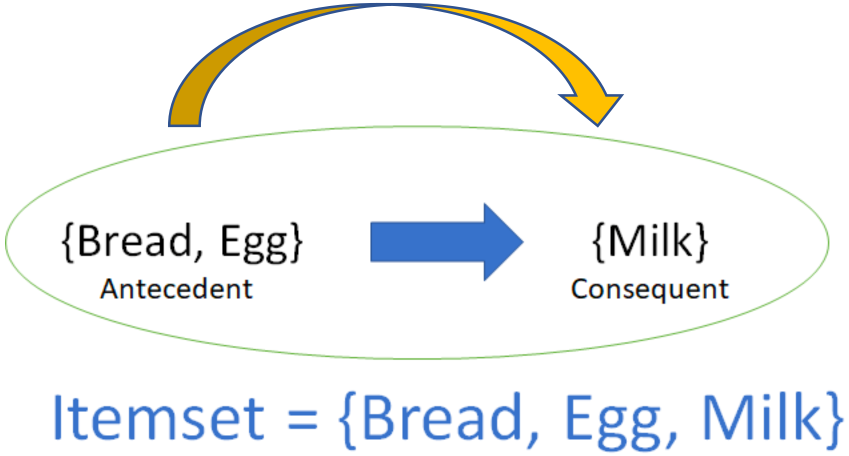
Apriori

Overview

- **Apriori algorithm** is given by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a dataset for boolean association rule
- Name of the algorithm is Apriori because 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
- To improve the efficiency of level-wise generation of frequent itemsets, an important property is used called *Apriori property* which helps by reducing the search space
- **Apriori Property:** All non-empty subset of frequent itemset must be frequent
- The key concept of Apriori algorithm is its anti-monotonicity of support measure

Terminology - Itemset

- It is a representation of the list of all items which form the association rule
- E.g.
    - Itemset = {Bread, Egg, Milk}



Terminology - Support

- This measure gives an idea of how frequent an *itemset* is in all the transactions
- E.g.
  - *itemset1* = {bread} and *itemset2* = {shampoo}
  - There will be far more transactions containing bread than those containing shampoo
  - So *itemset1* will generally have a higher support than *itemset2*
- *E.g.*
  - *itemset1* = {bread, butter} and *itemset2* = {bread, shampoo}
  - Many transactions will have both bread and butter on the cart but bread and shampoo are not so much
  - So in this case, *itemset1* will generally have a higher support than *itemset2*
- Mathematically support is the fraction of the total number of transactions in which the itemset occurs

$$Support(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Total\ number\ of\ transactions}$$

Terminology - Confidence

- This measure defines the likeliness of occurrence of consequent on the cart given that the cart already has the antecedents
- Technically, confidence is the conditional probability of occurrence of consequent given the antecedent

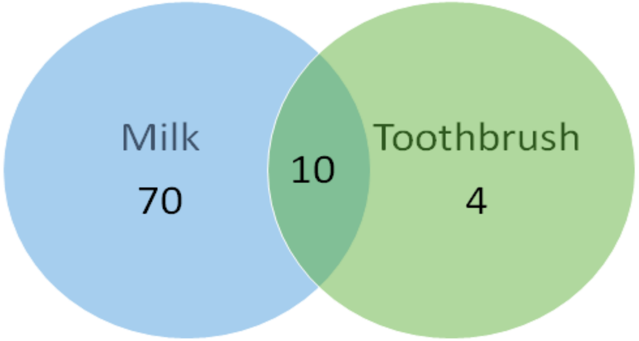
$$Confidence(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Transactions\ containing\ X}$$

- E.g.
  - Confidence for {Toothbrush} → {Milk} will be 10/(10+4) = 0.7

    $\underset{x}{\{Toothbrush\}} \rightarrow \underset{y}{\{Milk\}}$

- Confidence for {Milk} → {Toothbrush} will be 10/(10+70) = 10/80 = 1/8 = 0.13

  $\underset{x}{\{Milk\}} \rightarrow \underset{y}{\{Toothbrush\}}$



Terminology - Lift

- Lift controls for the *support* (frequency) of consequent while calculating the conditional probability of occurrence of {Y} given {X}
- Think of it as the *lift* that {X} provides to our confidence for having {Y} on the cart
- To rephrase, *lift* is the rise in probability of having {Y} on the cart with the knowledge of {X} being present over the probability of having {Y} on the cart without any knowledge about presence of {X}
- Mathematically

$$Lift(\{X\} \rightarrow \{Y\}) = \frac{(Transactions\ containing\ both\ X\ and\ Y)/(Transactions\ containing\ X)}{Fraction\ of\ transactions\ containing\ Y}$$

Summary

- **Association Rule**: Ex. {X → Y} is a representation of finding Y on the basket which has X on it
- **Itemset**: Ex. {X,Y} is a representation of the list of all items which form the association rule
- **Support**: Fraction of transactions containing the itemset
- **Confidence**: Probability of occurrence of {Y} given {X} is present
- **Lift**: Ratio of *confidence* to baseline probability of occurrence of {Y}