

# BIG DATA & DATA ANALYTICS




# CLUSTERING MODEL

## Basic Concept & Algorithms

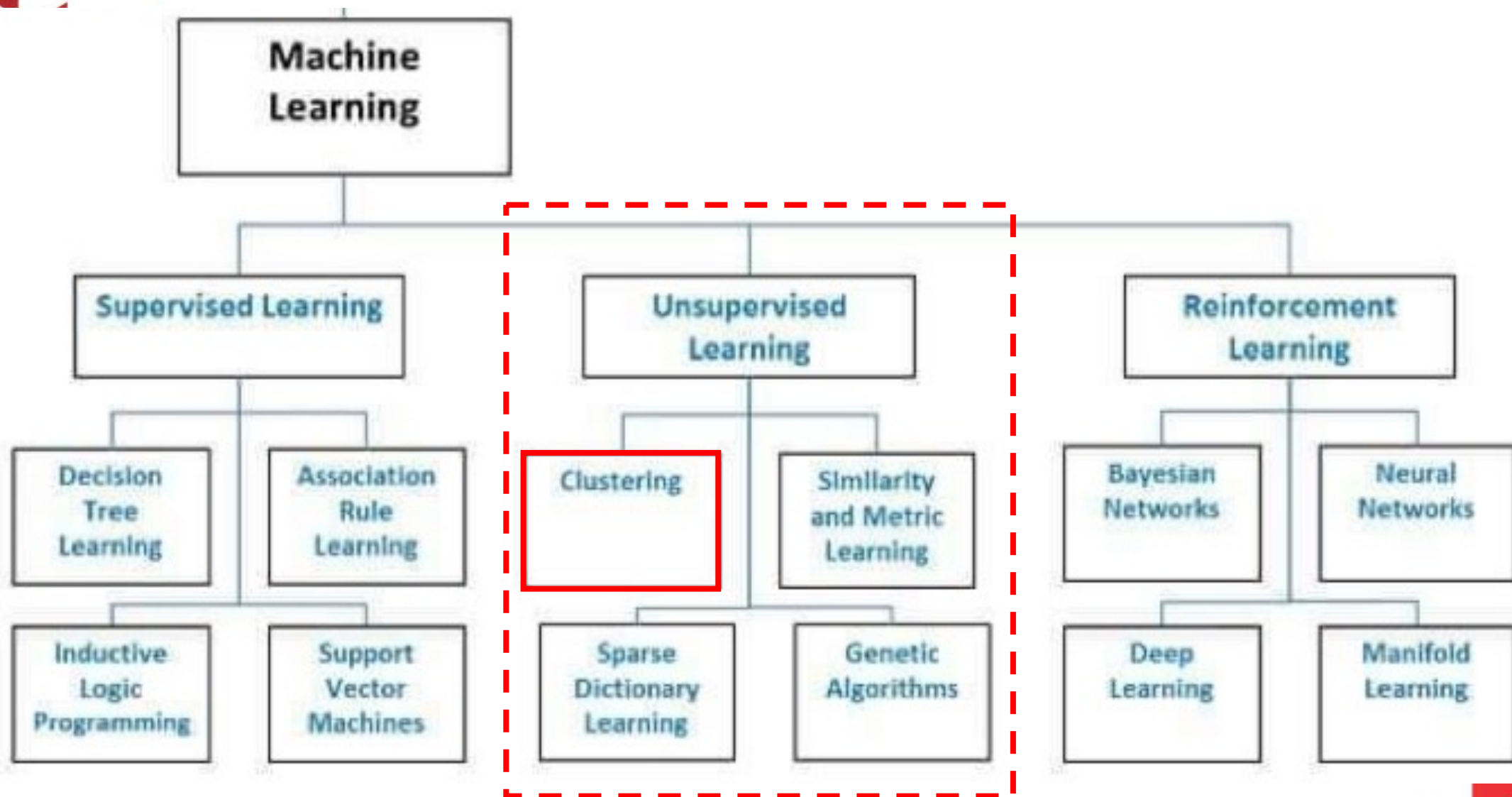
**Week 3 – EBI3B4 Big Data & Data Analytics**



# OUTLINE

1. Cluster Analysis: Basic Concepts
  2. Partitioning Methods
  3. Hierarchical Methods
  4. Density Methods
  5. Evaluation of Clustering
  6. Summary
- 

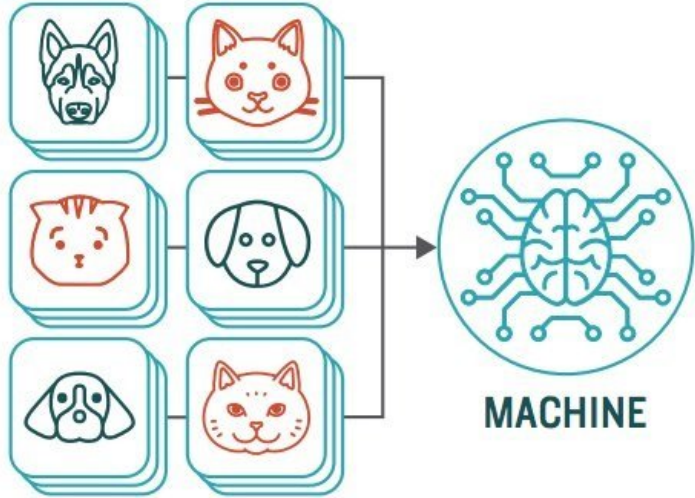
# Clustering Model



# Unsupervised Learning

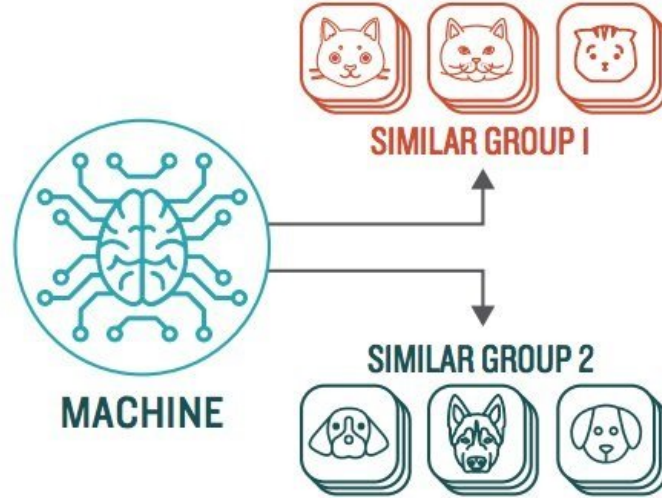
## STEP 1

Provide the machine learning algorithm uncategorized, unlabeled input data to see what patterns it finds



## STEP 2

Observe and learn from the patterns the machine identifies



- Unsupervised learning techniques serve a different process than supervised learning
- Unsupervised Learning designed to identify patterns inherent in the structure of the data.

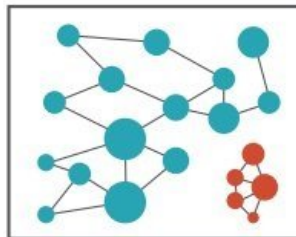
Unsupervised Learning draws inferences from datasets without labels. It is best used if you want to find patterns but don't know exactly what you're looking for.

## TYPES OF PROBLEMS TO WHICH IT'S SUITED

### CLUSTERING

Identifying similarities in groups

For Example: Are there patterns in the data to indicate certain patients will respond better to this treatment than others?



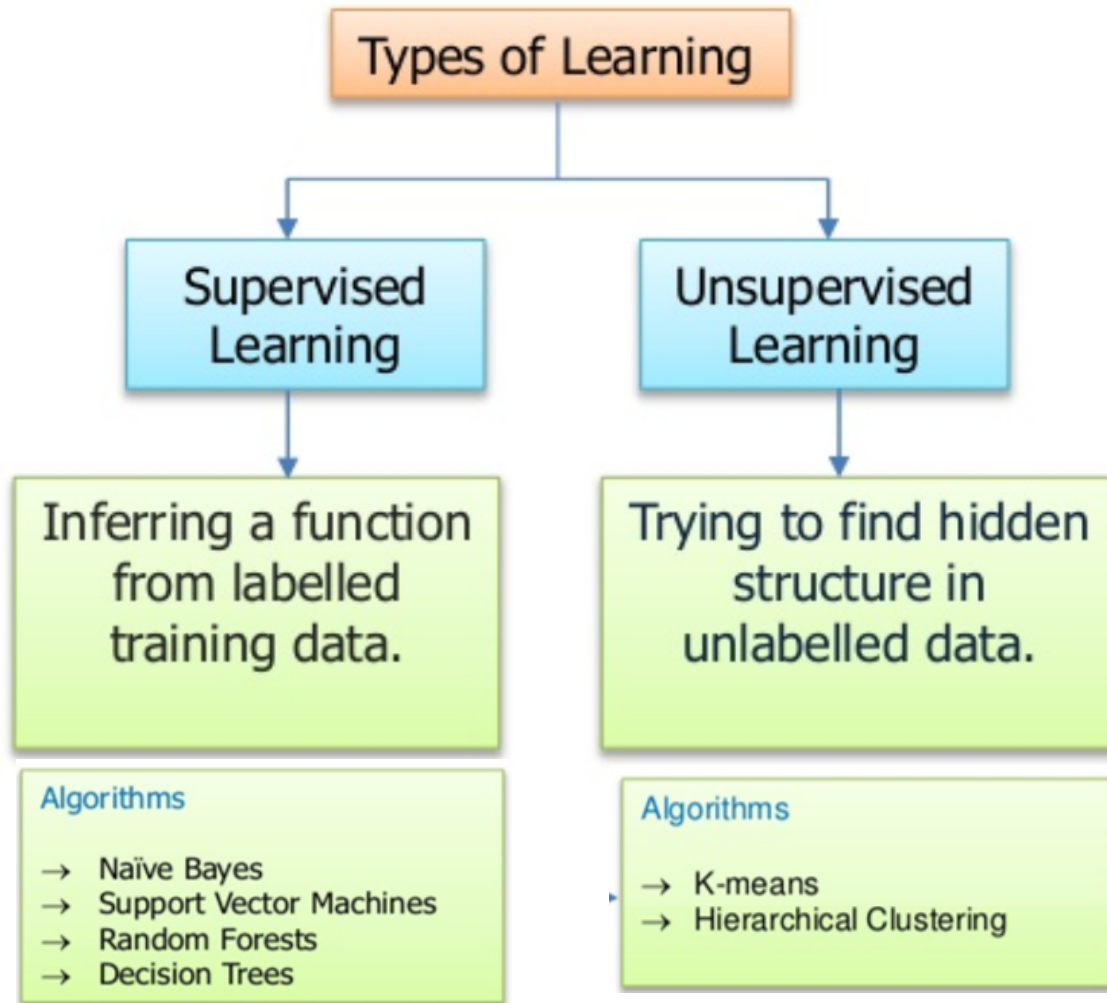
### ANOMALY DETECTION

Identifying abnormalities in data

For Example: Is a hacker intruding in our network?



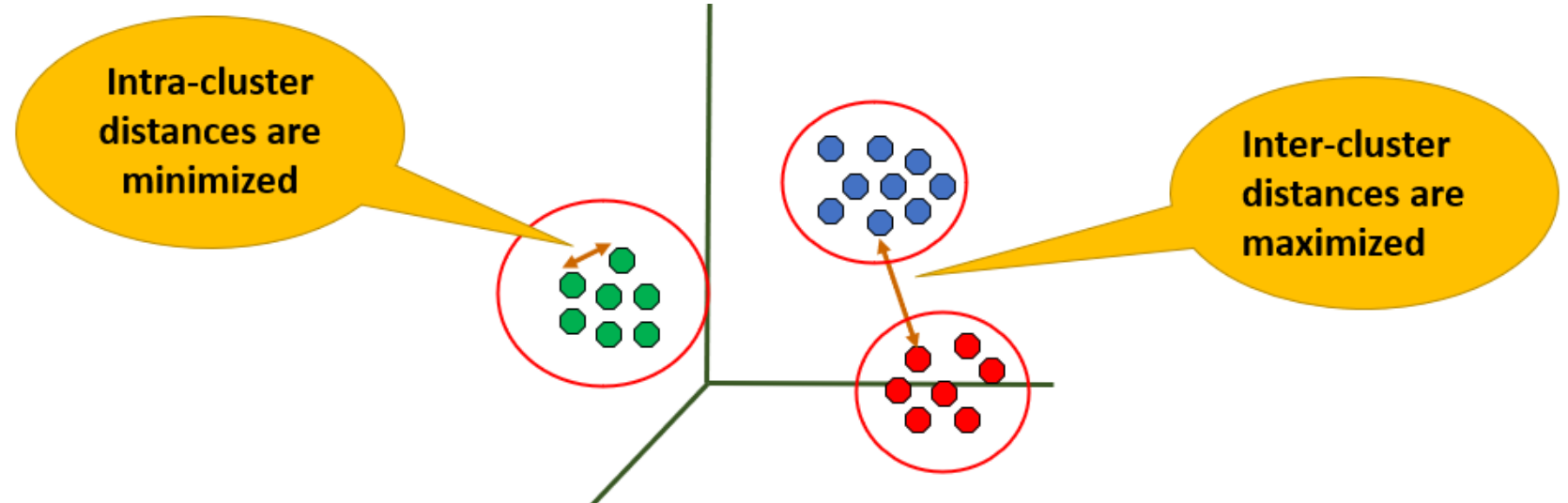
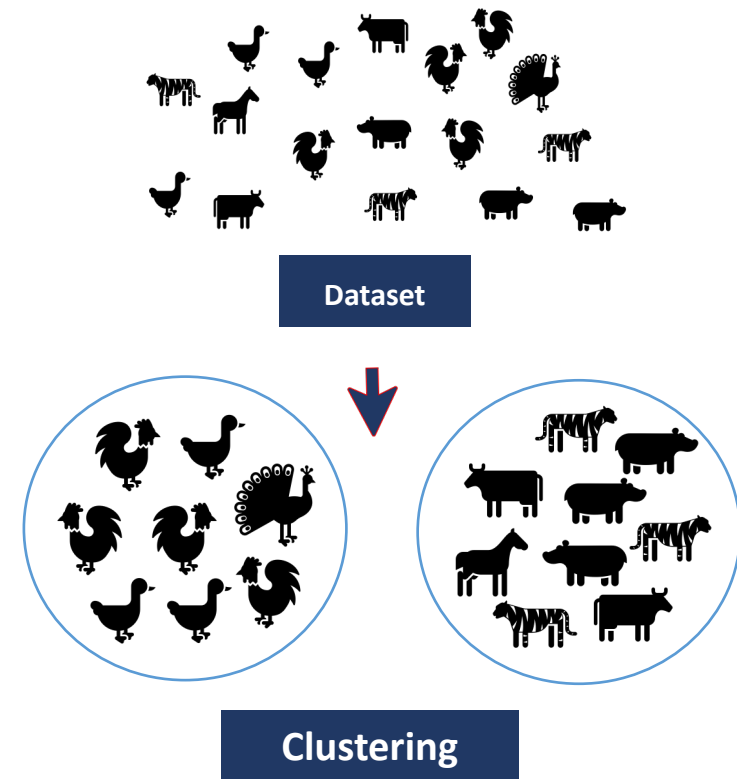
# Classification and Clustering



1. In general, in classification you have a set of predefined classes (label) and want to know which class a new object belongs to.
2. Clustering tries to group a set of objects and find whether there is *some* relationship between the objects. No predefined classes (label).
3. In the context of machine learning, classification is supervised learning and clustering is unsupervised learning.

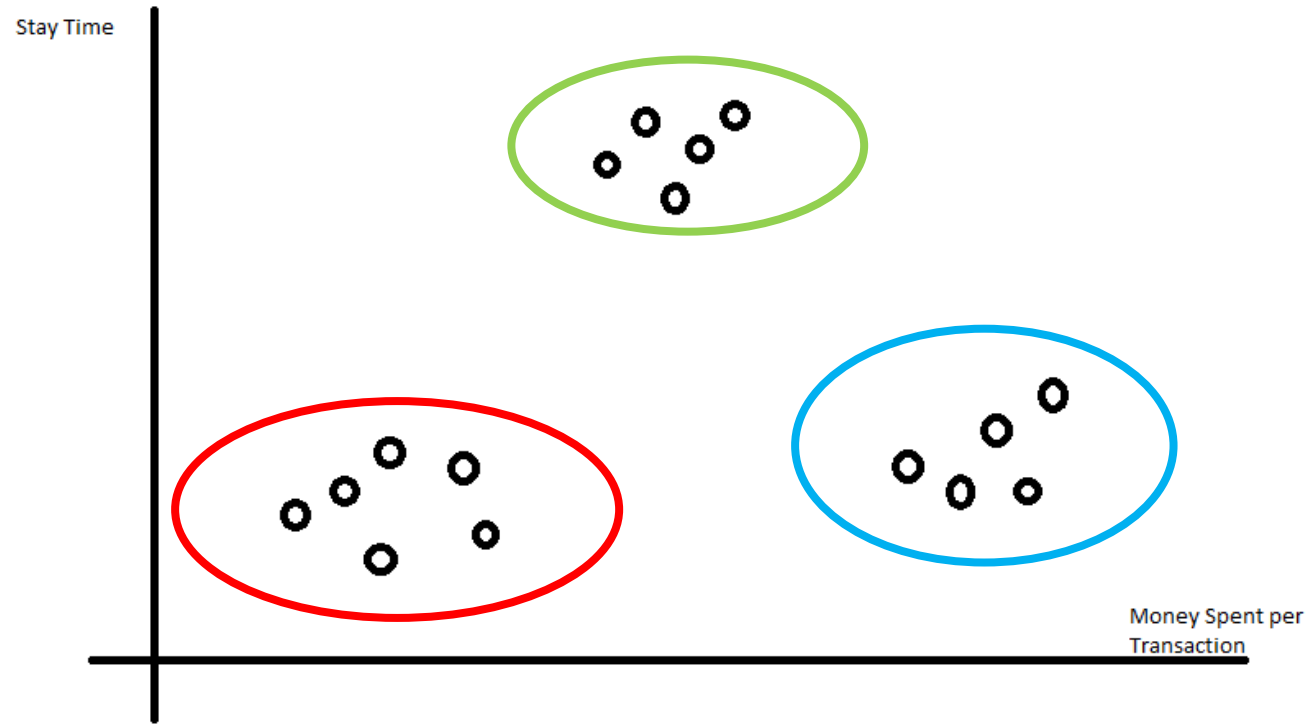
# What is Cluster Analysis

Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



1. **Clusters** are considered as groups containing data objects that are similar to each other
2. **Clustering** is a technique of organizing a group of data into classes and clusters where the objects reside inside a cluster will have high similarity and the objects of two clusters would be dissimilar to each other.
3. The main target of clustering is to divide the whole data into multiple clusters.
4. The similarity between two objects is measured by the **similarity function**, in which generally represented by distance metric

# Clustering Example: Pizza Hut Transaction




Pizza Hut Transaction

- **Red:** group of workers (lunch hour)
- **Green:** group of college students (after hour – night hour)
- **Blue:** group of families (mostly day)






# Clustering: Application Examples

1. **Biology:** taxonomy of living things: kingdom, phylum, class, order, family, genus and species
  2. **Information Retrieval:** document clustering
  3. **Land Use:** Identification of areas of similar land use in an earth observation database
  4. **Marketing:** Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
  5. **City Planning:** Identifying groups of houses according to their house type, value, and geographical location
  6. **Earthquake Studies:** Observed earthquake epicenters should be clustered along continent faults
  7. **Climate:** understanding earth climate, find patterns of atmospheric and ocean
  8. **Economic:** market research
- 



# What is not Cluster Analysis?

- **Supervised classification**
    - Have class label information
  - **Simple segmentation**
    - Dividing students into different registration groups alphabetically, by last name
  - **Results of a query**
    - Groupings are a result of an external specification
  - **Graph partitioning**
    - Some mutual relevance and synergy, but areas are not identical
- 

# Types of Clustering

## 1. Partitional Clustering (K-means and its variants)

- A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset

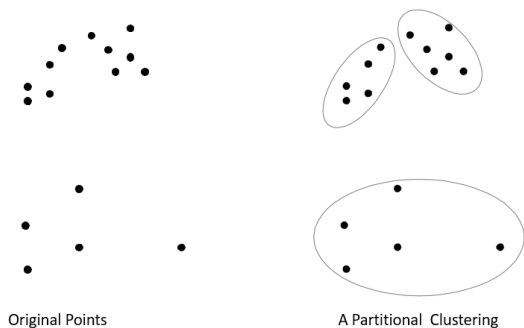
## 2. Hierarchical clustering (Connectivity based)

- A set of nested clusters organized as a hierarchical tree, usually depicted by a binary tree or dendrogram

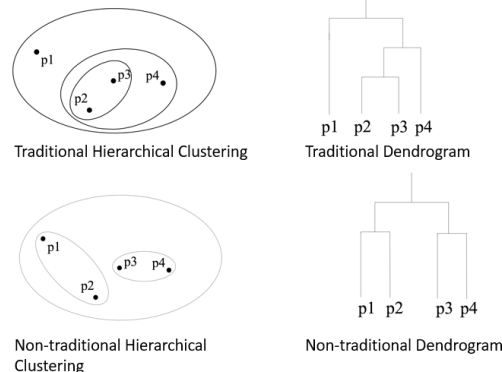
## 3. Density based Clustering

- Refers to unsupervised learning methods that identify distinctive groups/clusters in the data, works by detecting areas where points are concentrated and where they are separated by areas that are empty or sparse. Points that are not part of a cluster are labeled as noise.

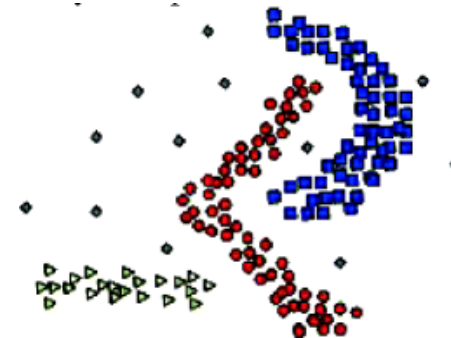
### Partitional



### Hierarchical



### Density Based





# Partitioning Clustering

## K-Means Algorithm

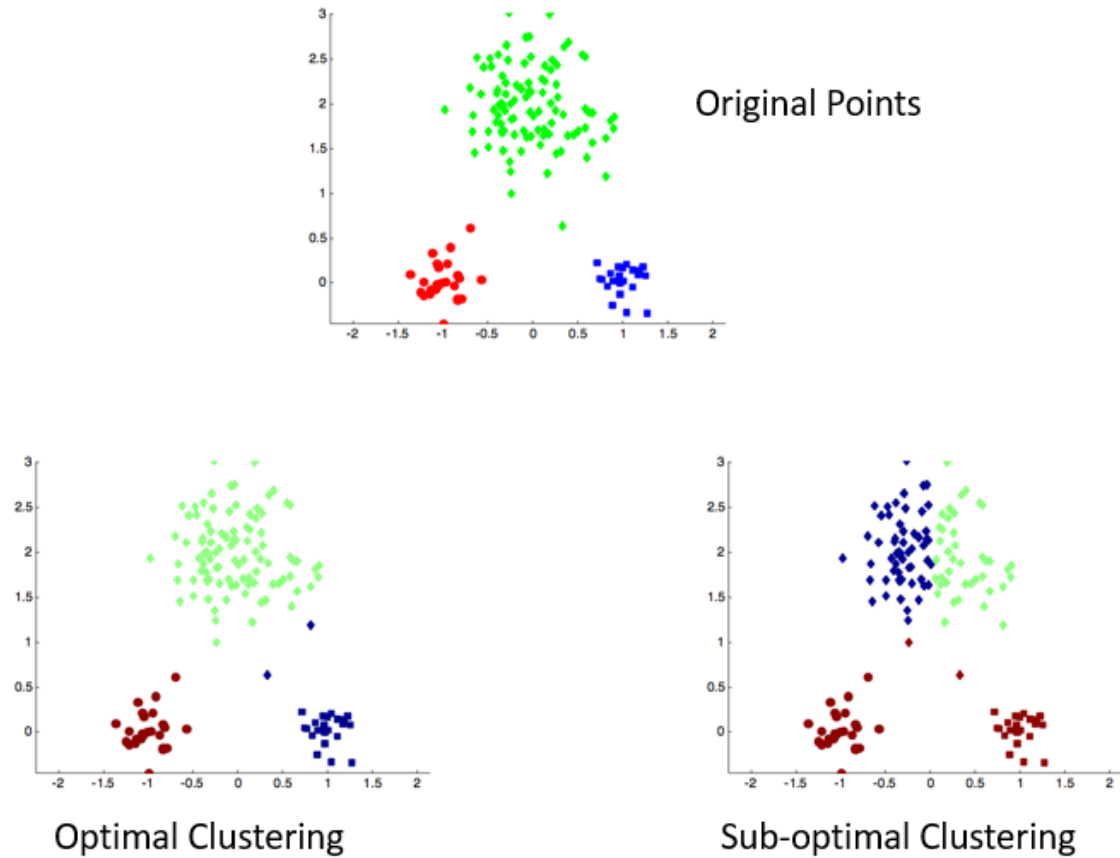
K-Means clustering is used to divide or distribute  $n$  observations into  $k$  clusters in which each observation belongs to the cluster with the nearest centroid

- 
- 1: Select  $K$  points as the initial centroids.
  - 2: **repeat**
  - 3:   Form  $K$  clusters by assigning all points to the closest centroid.
  - 4:   Recompute the centroid of each cluster.
  - 5: **until** The centroids don't change
- 

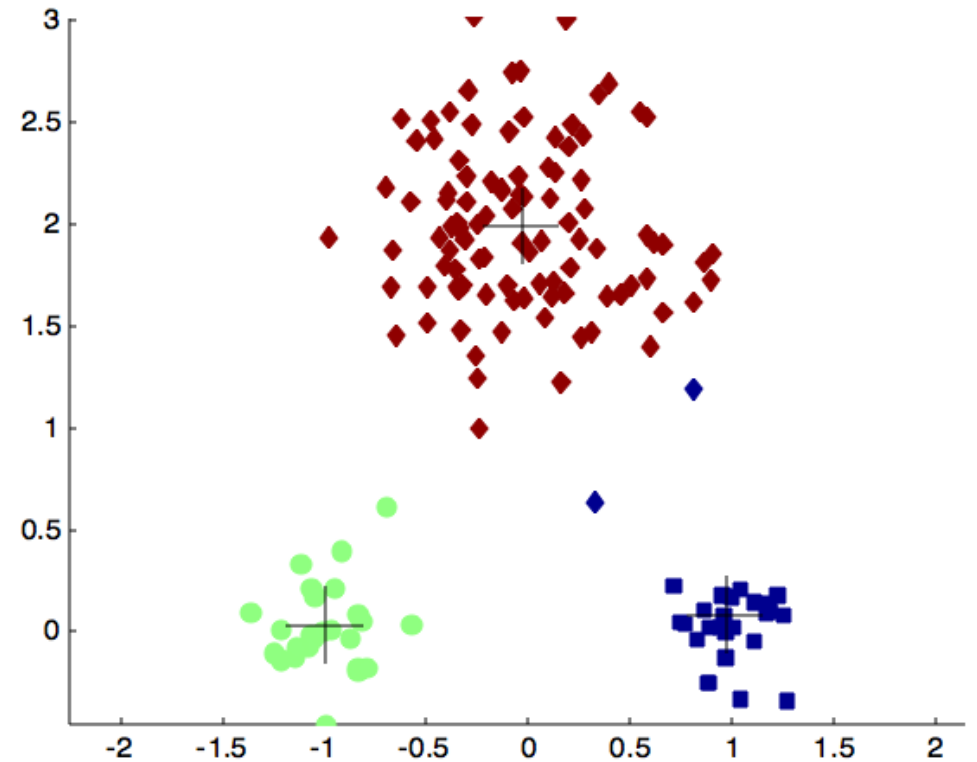
1. Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
2. The centroid is (typically) the mean of the points in the cluster.
3. 'Closeness' is measured by *Euclidean distance*, *cosine similarity*, *correlation*, etc.
4. K-means will converge for common similarity measures mentioned above.
5. Most of the convergence happens in the first few iterations.
6. Complexity is  $O(n * K * I * d)$ 
  - $n$  = number of points,
  - $K$  = number of clusters,
  - $I$  = number of iterations,
  - $d$  = number of attributes



# Optimal K-means Clustering



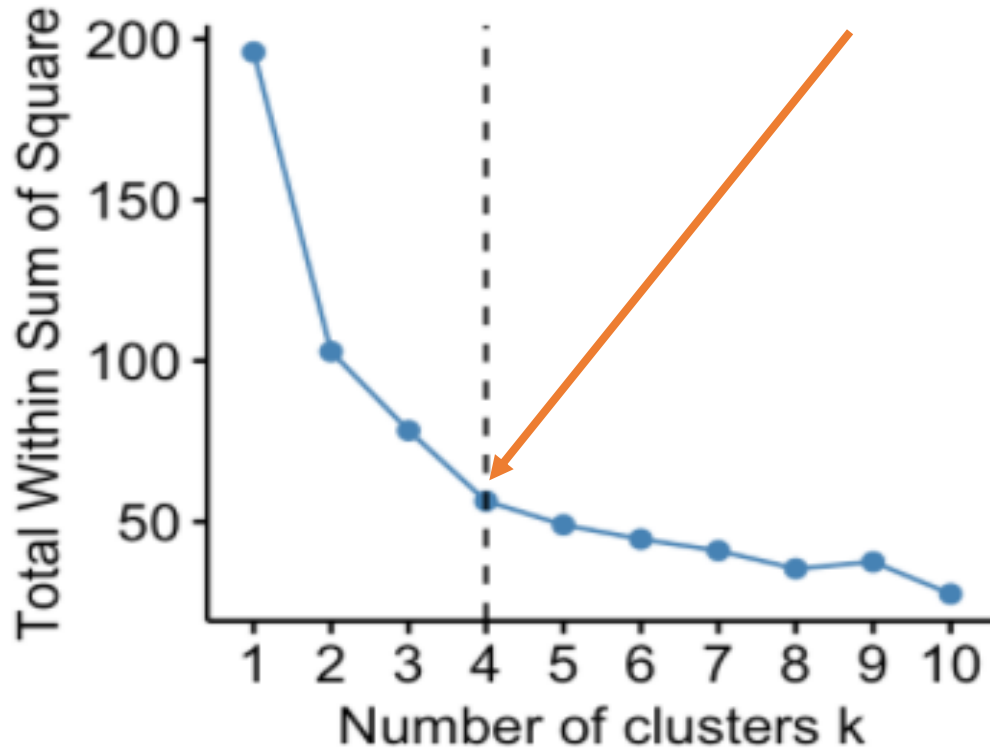
# Choosing Initial Centroids





# Elbow Method

Finding the Optimal Number of Clusters

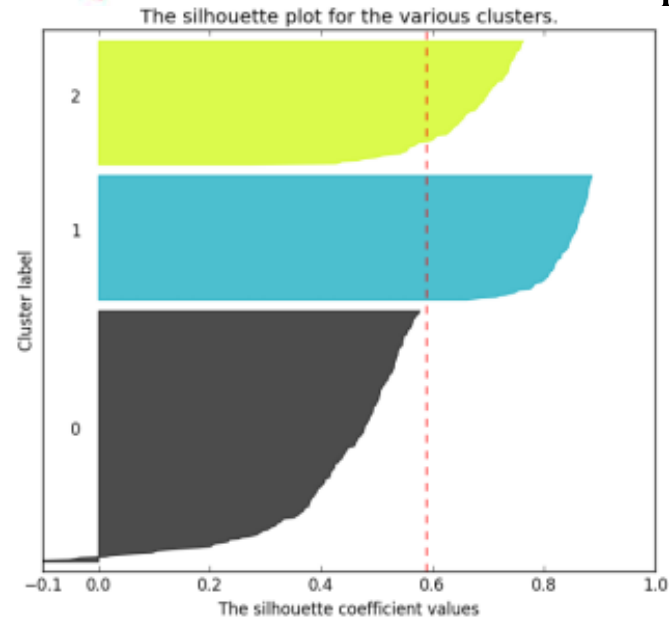


Elbow method: 4 clusters solution suggested

1. The variance (within-cluster sum of squared errors/SSE) is plotted against the number of clusters.
2. The first few clusters will introduce a lot of variance and information, but at some point, the information gain will become low, thus imparting an angular structure to the graph.
3. The optimal number of clusters is found out from the elbow point; therefore, this is known as the “elbow criterion.”

# Silhouette Method

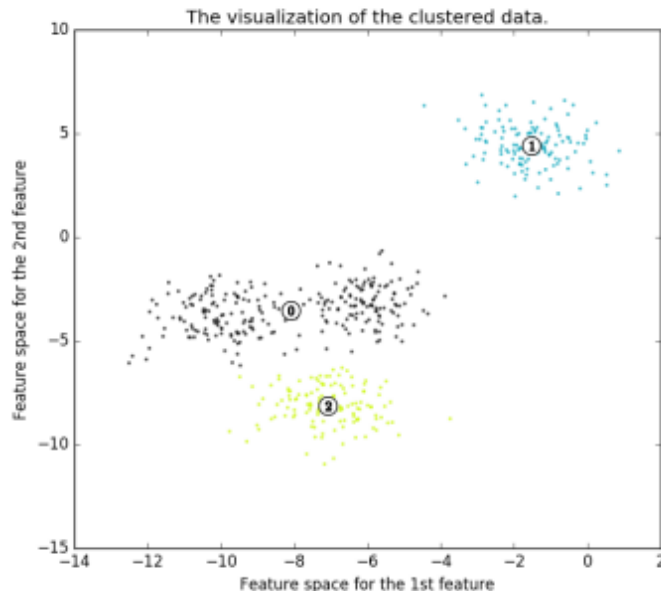
## Finding the Optimal Number of Clusters



- **The silhouette value** is a measure of *how similar an object is to its own cluster (cohesion) compared to other clusters (separation)*.
- The silhouette ranges from  $-1$  to  $+1$ , where a high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.
- If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

$$\text{Silhouette Coefficient} = (x-y) / \max(x,y)$$

where,  $y$  is the mean of intra cluster distance.  $x$  is the mean to nearest cluster distance.



## Elbow vs Silhouette

<https://www.youtube.com/watch?v=AtxQOrvdQIA&t=401s>

<https://www.youtube.com/watch?v=qs8nfzUsW5U>




# Limitations of K-means

## Advantages

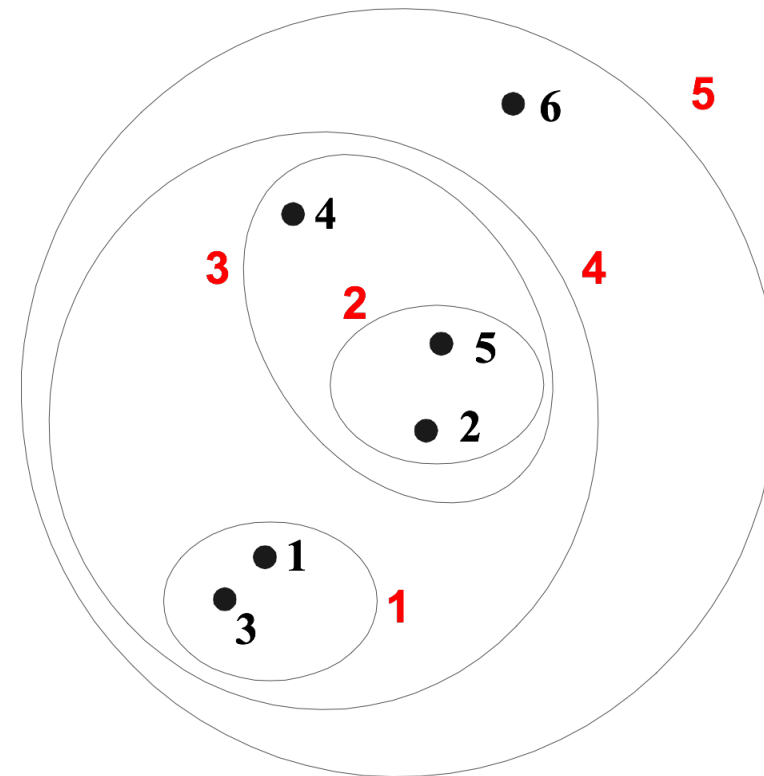
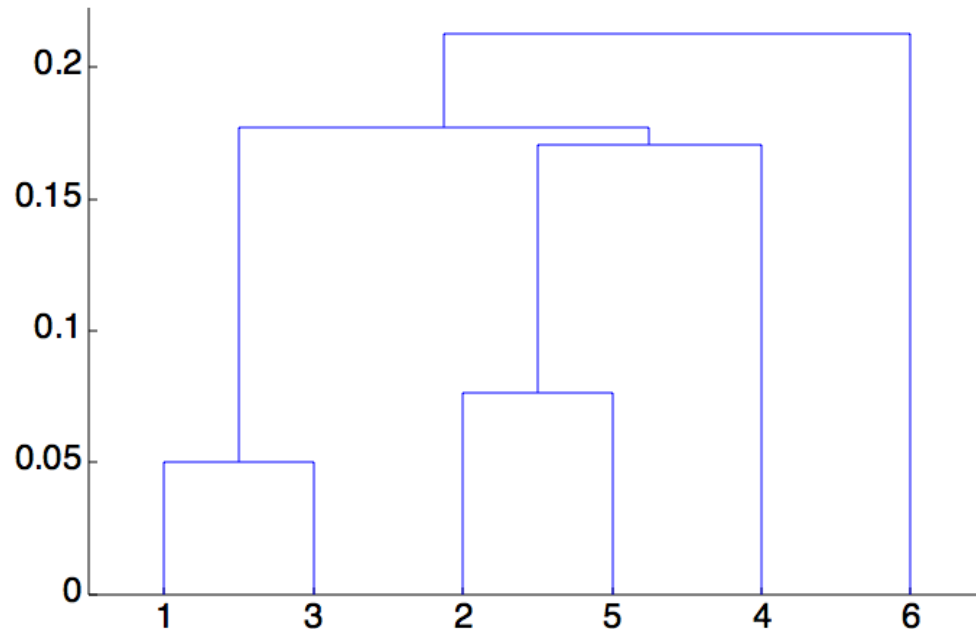
1. If variables are huge, then K-Means most of the times computationally faster than hierarchical clustering, if we keep  $k$  small.
2. K-Means produce tighter clusters than hierarchical clustering, especially if the clusters are globular.

## Disadvantages

1. Difficult to predict K-Value.
  2. With global cluster, it didn't work well.
  3. Different initial partitions can result in different final clusters.
  4. It does not work well with clusters (in the original data) of Different size and Different density
- 

# Hierarchical Clustering

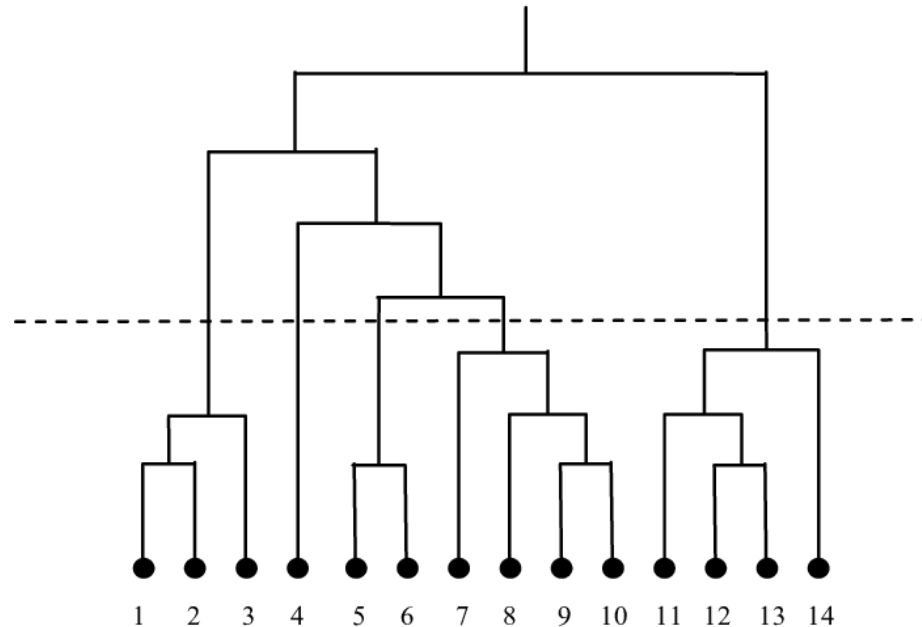
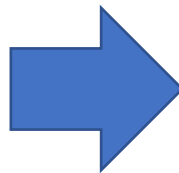
- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits



# Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters ( $k$ )
  - Any desired number of clusters can be obtained by 'cutting' the dendrogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, ...)

Cutting Dendrogram





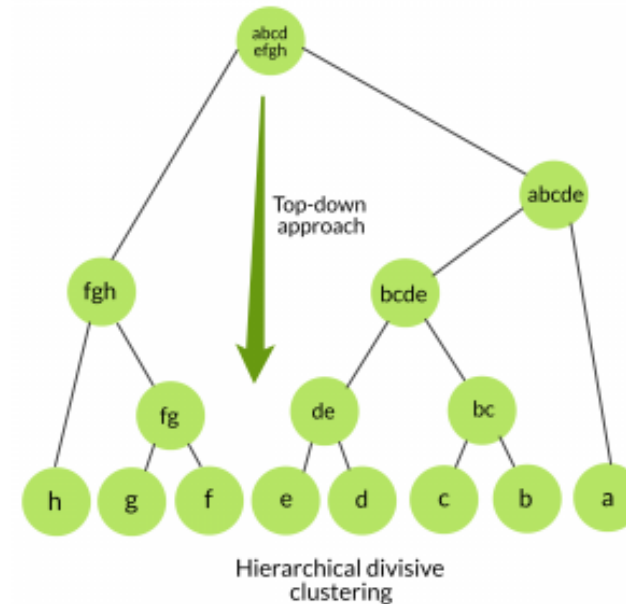
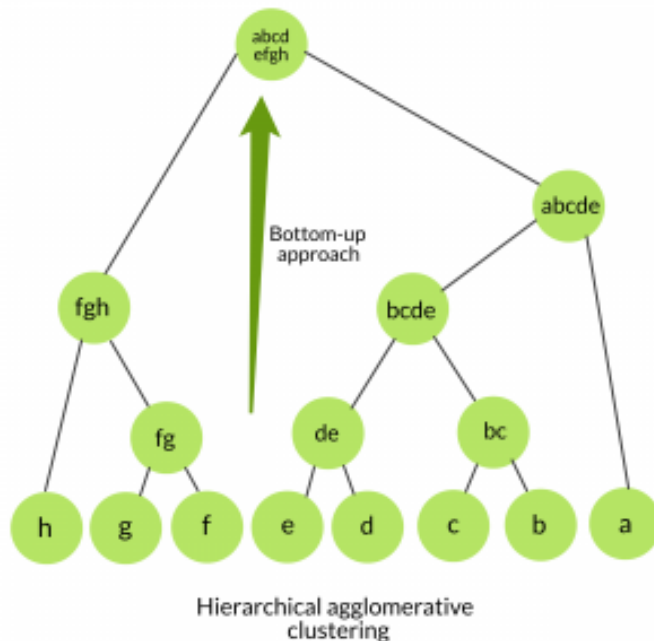
# Hierarchical Clustering Types

## 1. Agglomerative (bottom to top, small to big, merging):

- Start with the points as individual clusters
- At each step, merge the closest pair of clusters until only one cluster (or k clusters) left

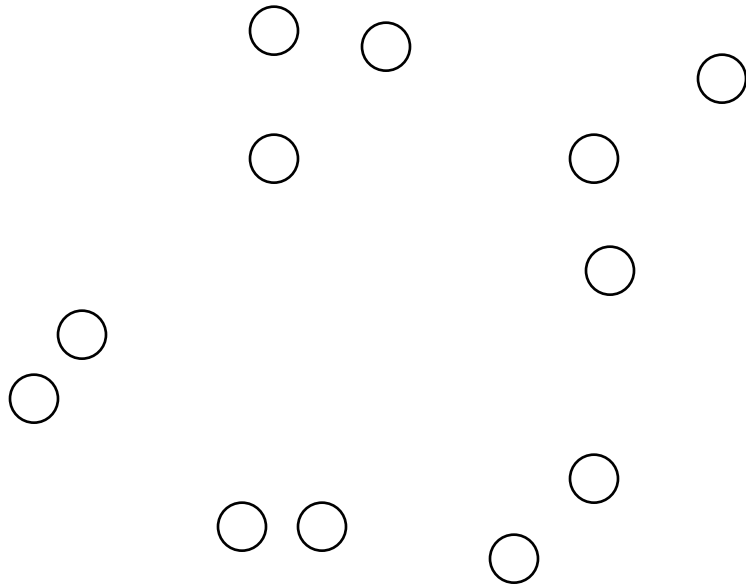
## 2. Divisive (top to bottom, big to small, splitting):

- Start with one, all-inclusive cluster
- At each step, split a cluster until each cluster contains a point (or there are k clusters)



# Initial Step (Algorithm)

- Start with clusters of individual points and a proximity matrix



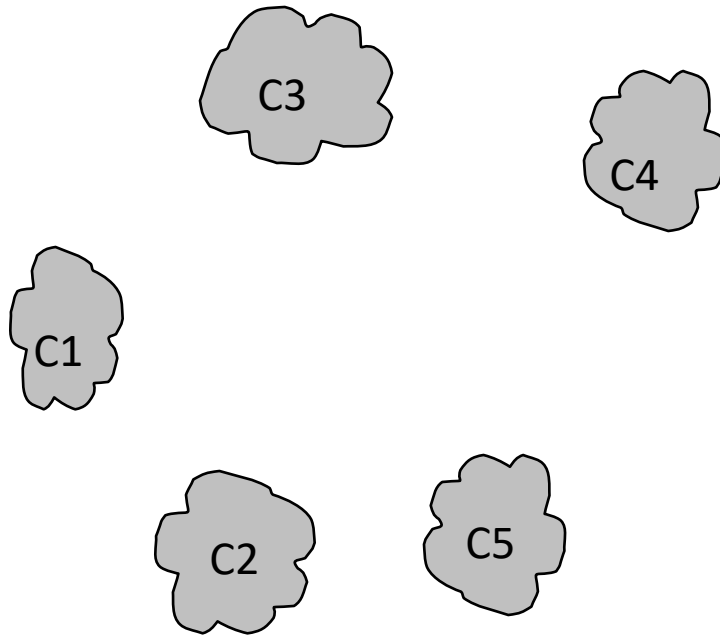
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

p1 p2 p3 p4 ... p9 p10 p11 p12

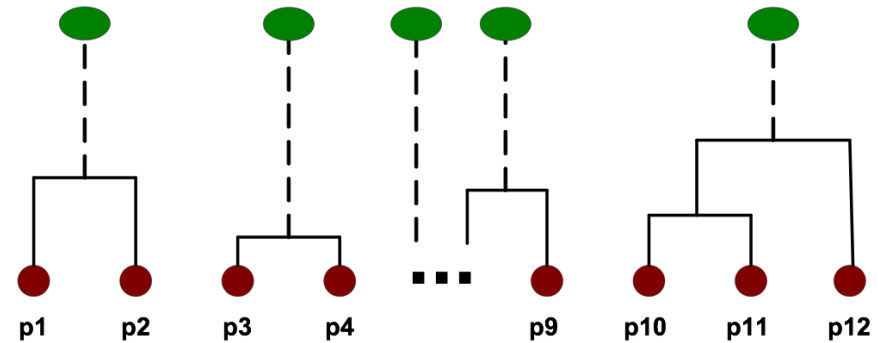
# Merging Step

- After some merging steps, we have some clusters



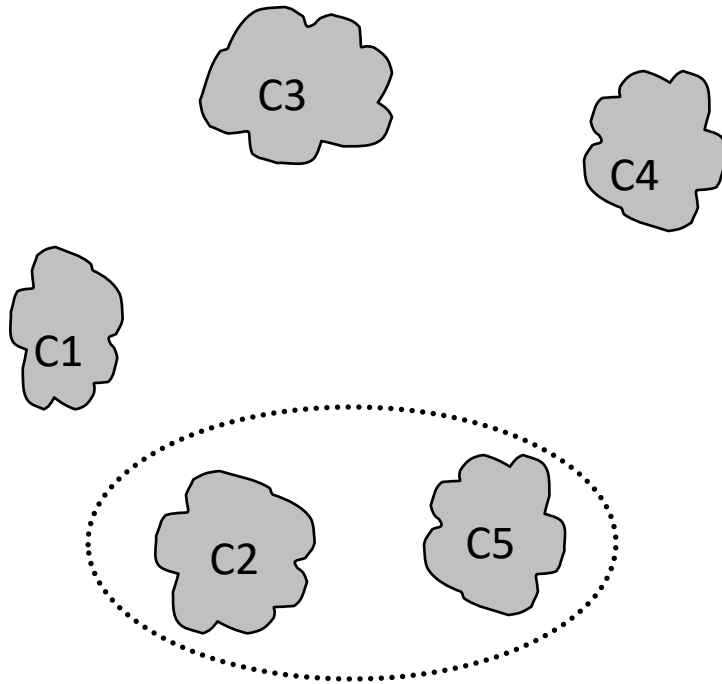
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



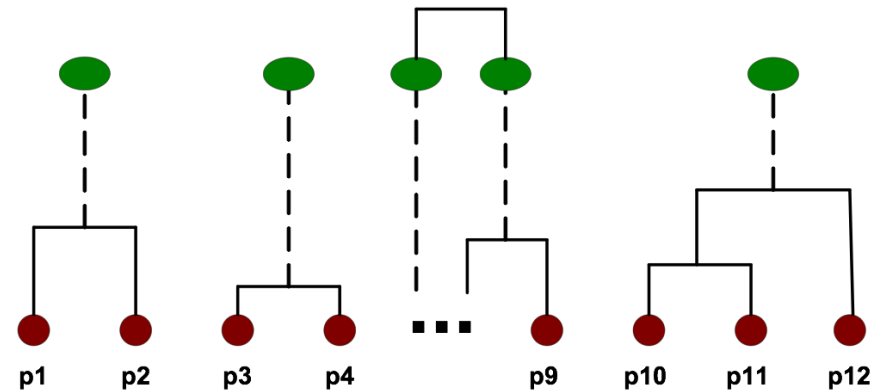
# Merging Step

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



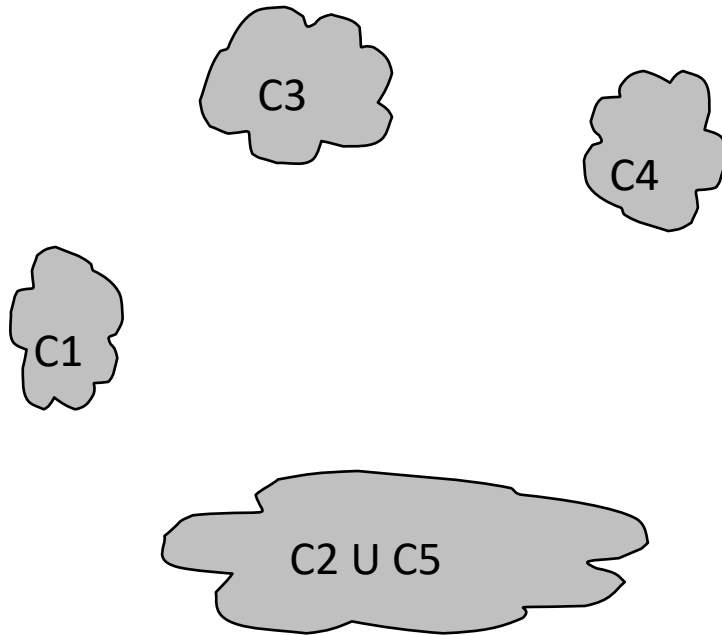
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



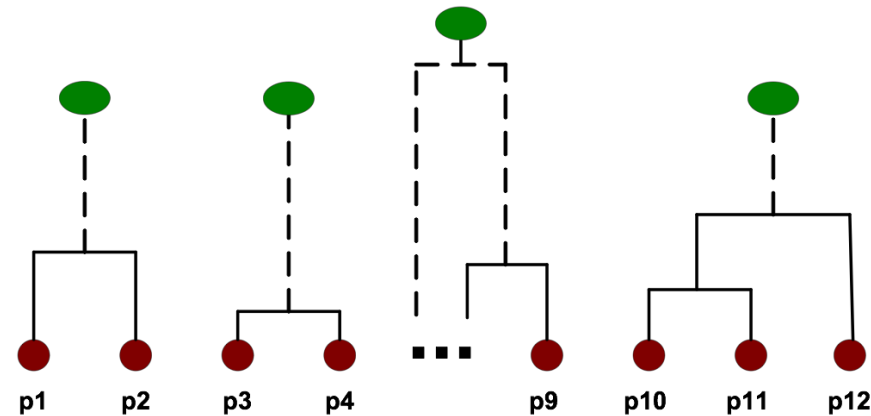
# After Merging Step

- The question is “How do we update the proximity matrix?”



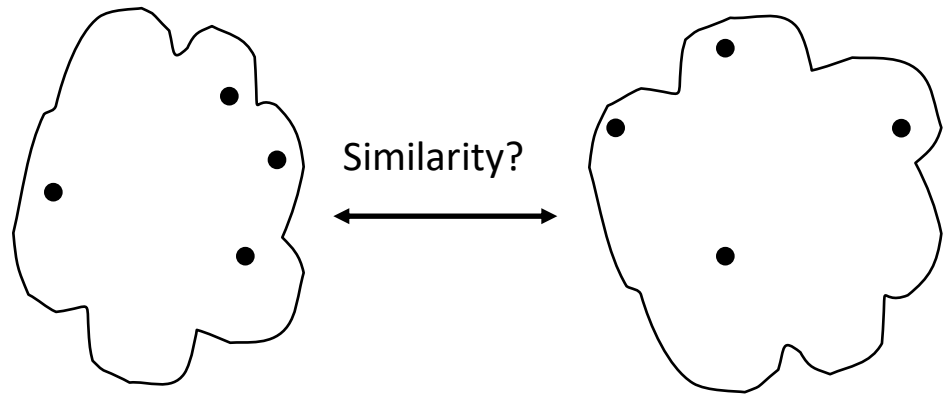
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix

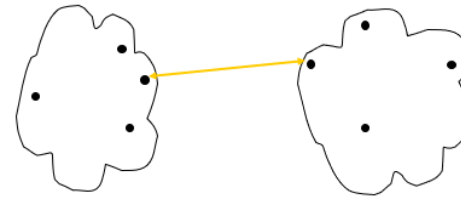




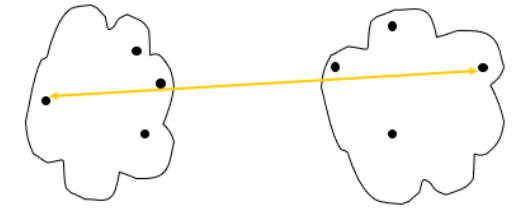
# How to Define Inter-Cluster Similarity



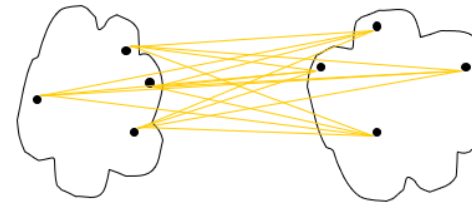
- MIN
- MAX
- Group Average
- Distance Between Centroids



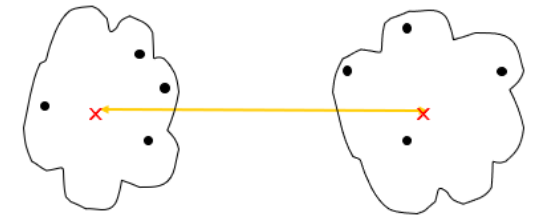
MIN



MAX

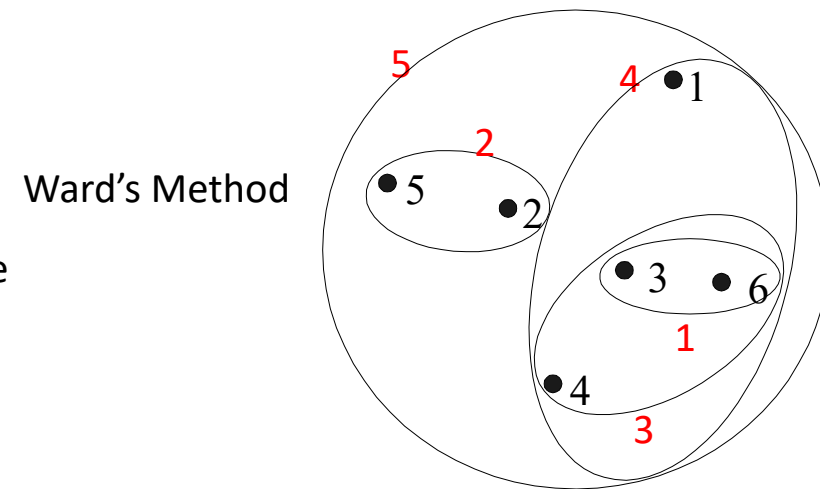
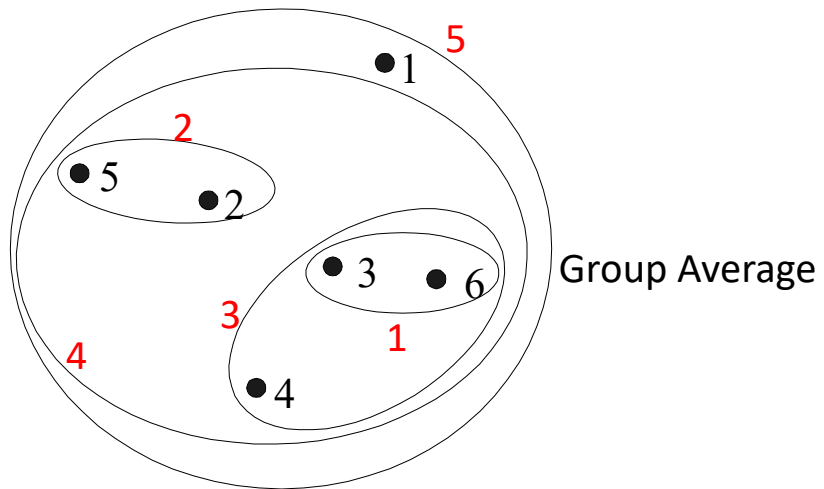
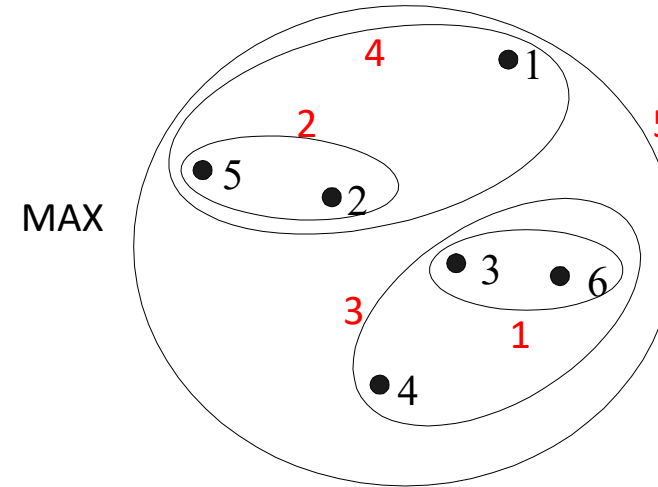
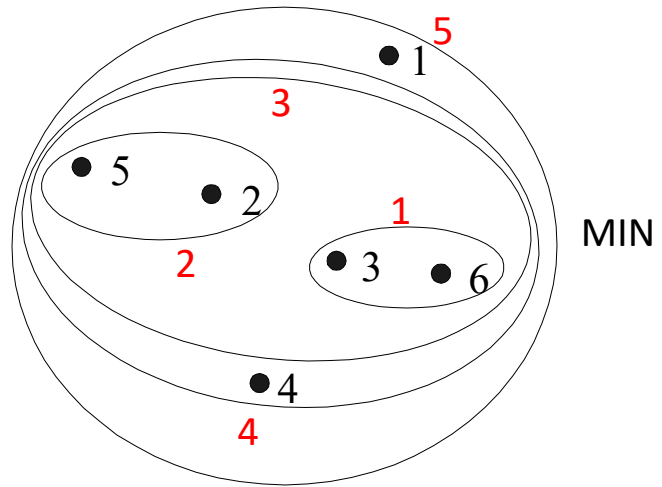


Group Average



Distance Between Centroids

# Hierarchical Clustering: Comparison





# More About Hierarchical Clustering

## Time and Space Requirements


- $O(N^2)$  space since it uses the proximity matrix.  $N$  number of data
- $O(N^3)$  time in many cases
  - There are  $N$  steps and at each step the size,  $N^2$ , proximity matrix must be updated and searched
- Complexity can be reduced to  $O(N^2 \log(N))$  time for some approaches

## Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized (Example: No SSE to minimize)
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

## Partitioning VS Hierarchical Clustering

Partitioning (K-Means)	Hierarchical
needs the number of clusters to be specified	doesn't need the number of clusters to be specified
usually more efficient run-time wise	can be slow (has to make several merge/split decisions)
good for large dataset	good for small datasets

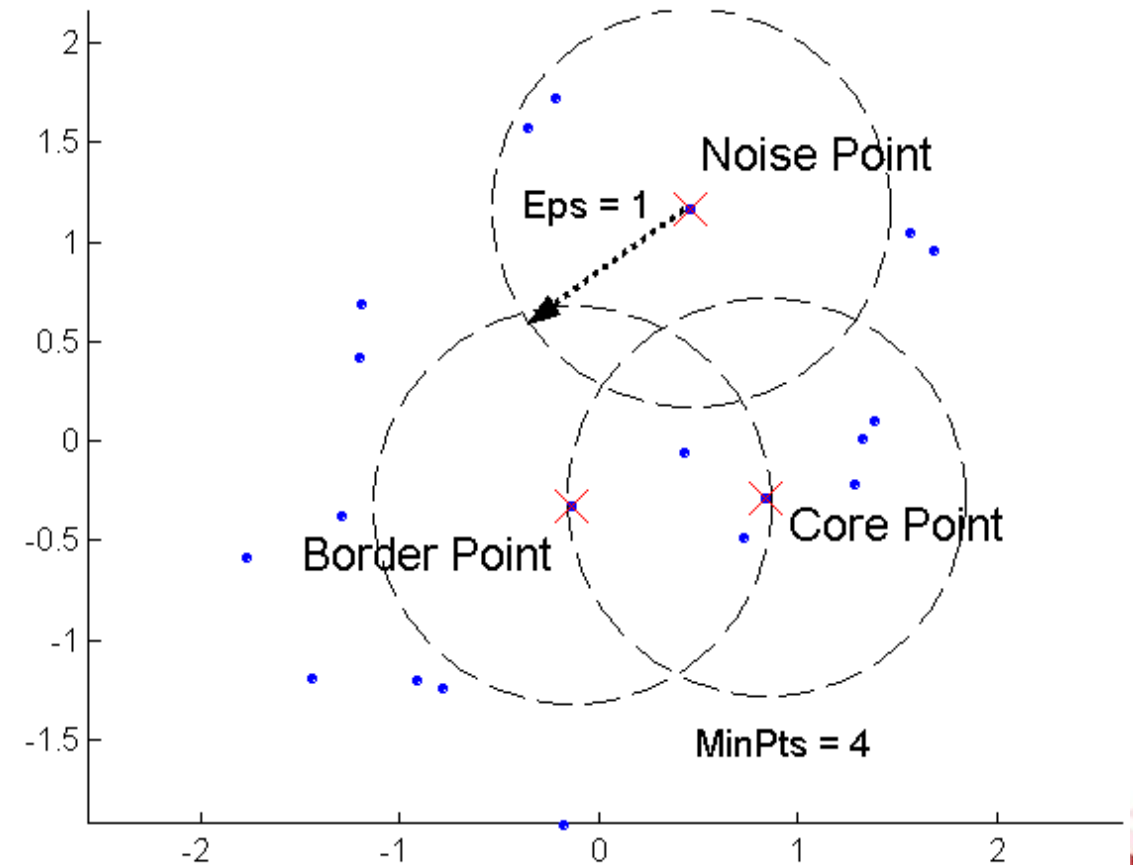


# Density-Based Partitioning

## DBSCAN Algorithm

### Core, Border, and Noise Points

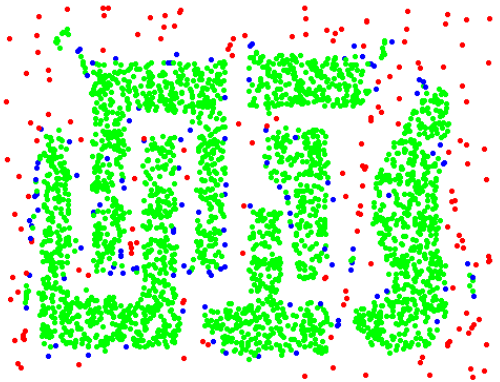
- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius (Eps)
  - A point is a core point if it has more than a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
  - A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
  - A noise point is any point that is not a core point or a border point.



# DBSCAN Scenario



Original Points



Point types: **core**, **border** and **noise**

Eps = 10, MinPts = 4

## When DBSCAN Works Well



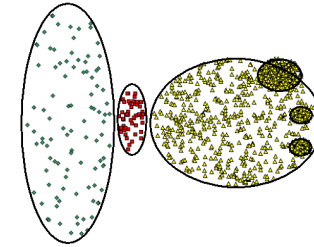
Original Points



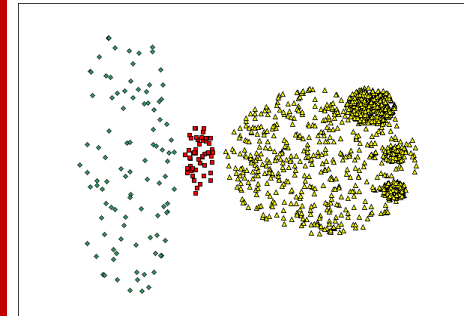
Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

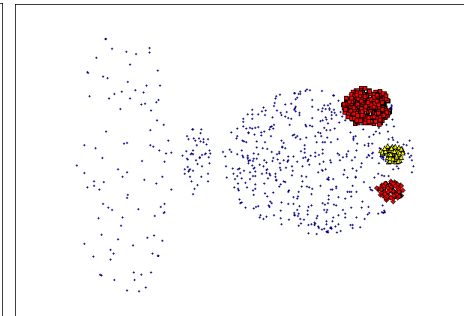
## When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

- Varying densities
- High-dimensional data







# Cluster Validity

- “Clusters are in the eye of the beholder”!
- Then why do we want to evaluate them?
  - To avoid finding patterns in noise : Determining the **clustering tendency** of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
  - To compare clustering algorithms : Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels
  - To compare some of clusters : Evaluating how well the results of a cluster analysis fit the data *without* reference to external information (use only the data)
  - Reduction Information

we have the option to evaluate the entire clustering or just individual clusters.



*Thank  
you*

