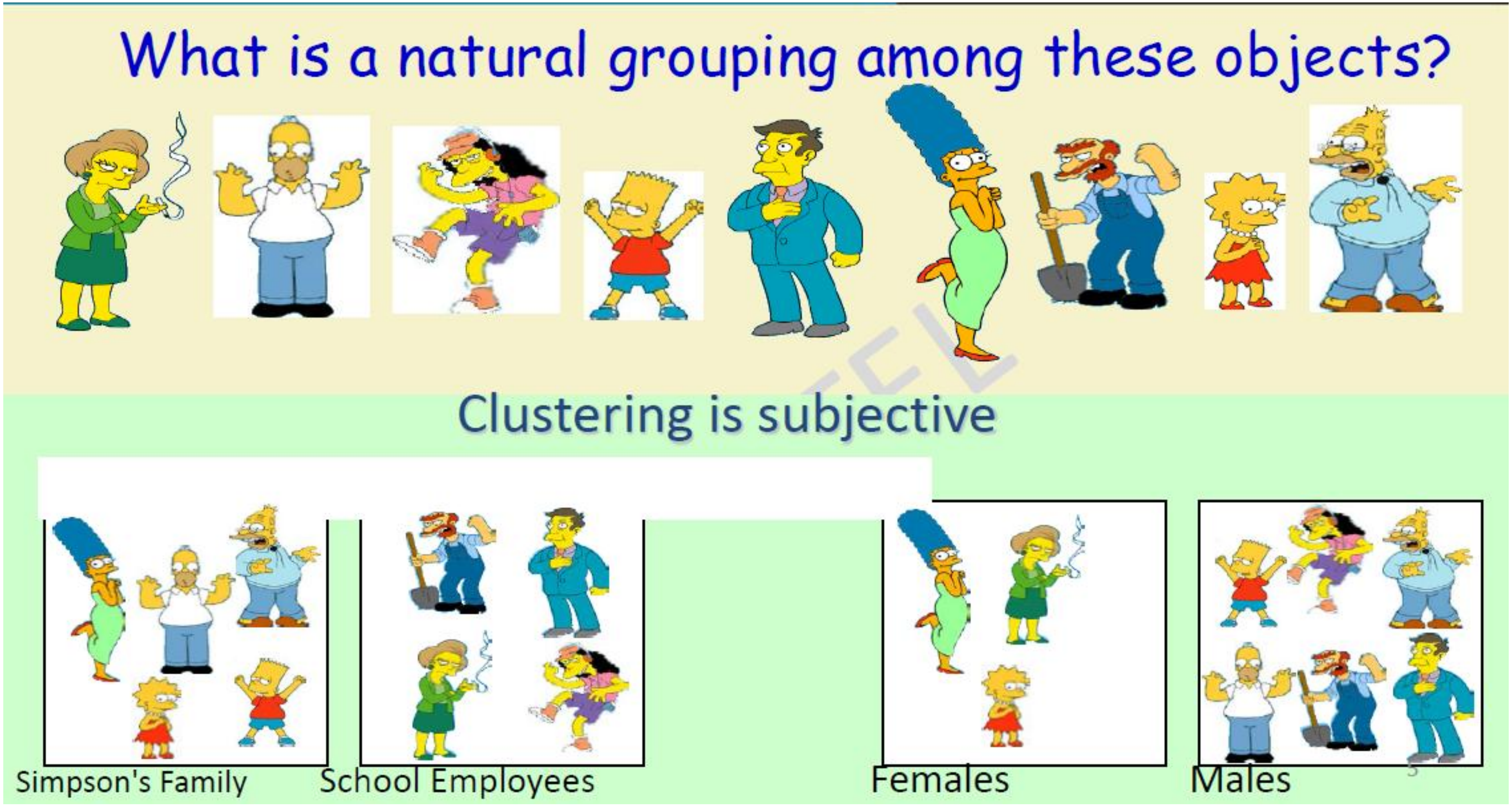
Clustering Techniques

By: Sapna Yadav

What is Cluster Analysis?

- **Cluster: A collection of data objects**
  - **similar (or related) to one another within the same group**
  - **dissimilar (or unrelated) to the objects in other groups**
- **Cluster analysis (or *clustering*, *data segmentation, …*)**
  - **Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters**
- **Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)**
- **Typical applications**
  - **As a stand-alone tool to get insight into data distribution**
  - **As a preprocessing step for other algorithms**

Clustering



What is a natural grouping among these objects?

Clustering is subjective

What is Clustering?

- Clustering is dividing data points into homogeneous classes or clusters:
- Points in the same group are as similar as possible
- Points in different group are as dissimilar as possible
- Organizing data into classes such that there is
  - high intra-class similarity
  - low inter-class similarity
- Finding the class labels and the number of classes directly from the data (in contrast to classification).

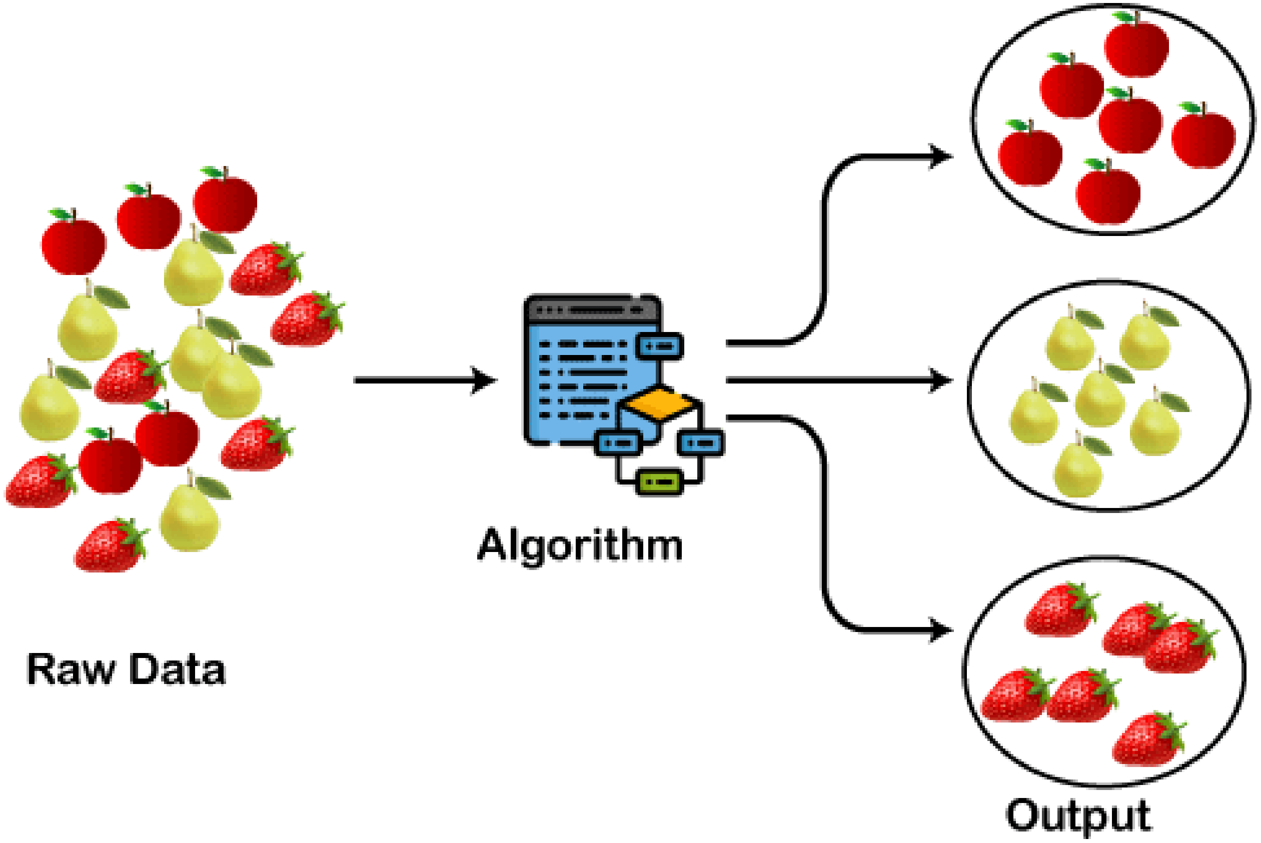
Clustering Example

The clustering technique can be widely used in various tasks. Some most common uses of this technique are:

- Market Segmentation
- Statistical data analysis
- Social network analysis
- Image segmentation
- Anomaly detection, etc.

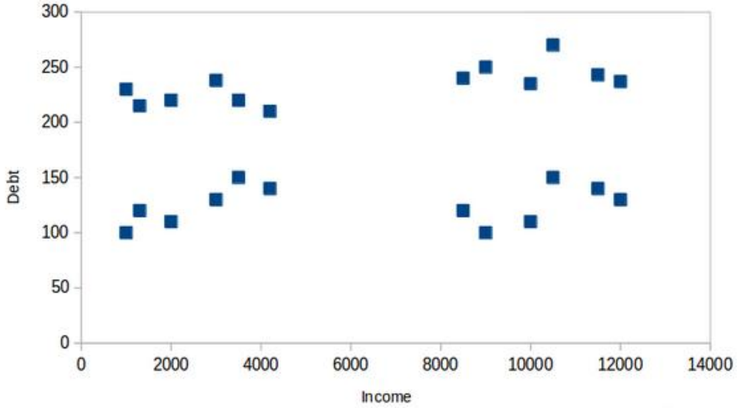
Apart from these general usages, it is used by the **Amazon** in its recommendation system to provide the recommendations as per the past search of products. **Netflix** also uses this technique to recommend the movies and web-series to its users as per the watch history.

Clustering Example

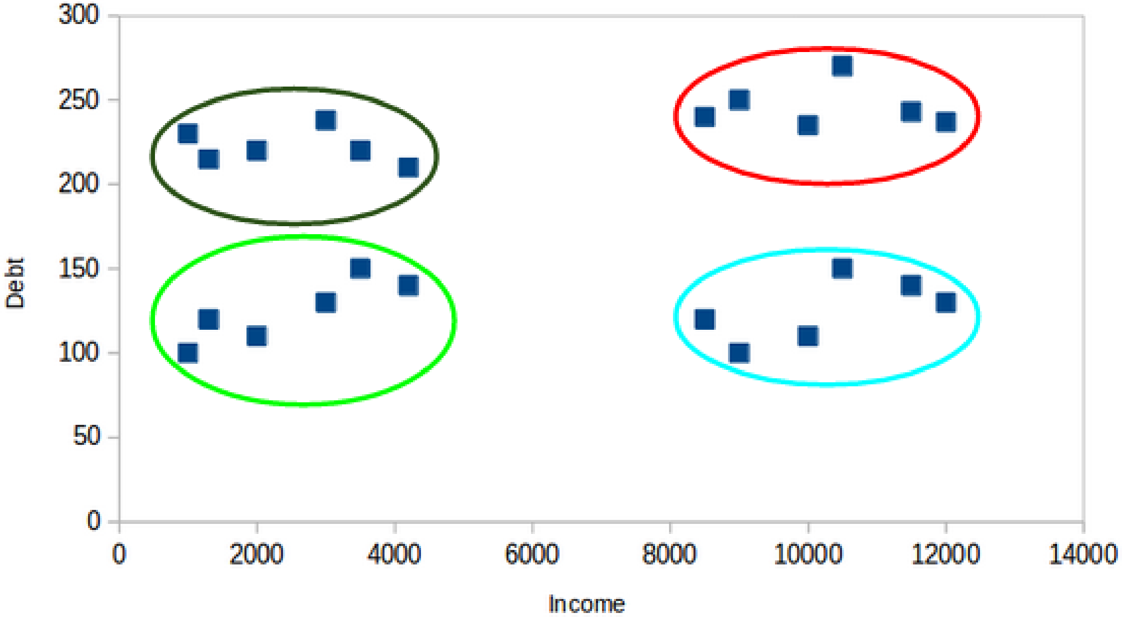


A Case Study-Customer segmentation

- let's say the bank only wants to use the income and debt to make the segmentation. They collected the customer data and used a scatter plot to visualize it:



On the X-axis, we have the income of the customer and the y-axis represents the amount of debt. Here, we can clearly visualize that these customers can be segmented into 4 different clusters as shown below:



This is how clustering helps to create segments (clusters) from the data. The bank can further use these clusters to make strategies and offer discounts to its customers.

A Case Study-Customer segmentation

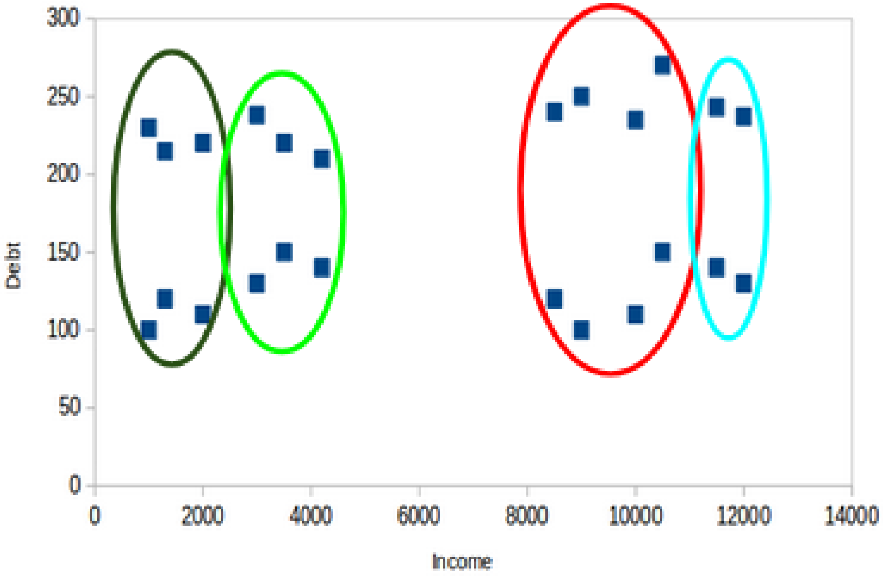
Property 1

- **All the data points in a cluster should be similar to each other.** Let me illustrate it using the above example:
- If the customers in a particular cluster are not similar to each other, then their requirements might vary, right? If the bank gives them the same offer, they might not like it and their interest in the bank might reduce. Not ideal.
- Having similar data points within the same cluster helps the bank to use targeted marketing. You can think of similar examples from your everyday life and think about how clustering will (or already does) impact the business strategy.

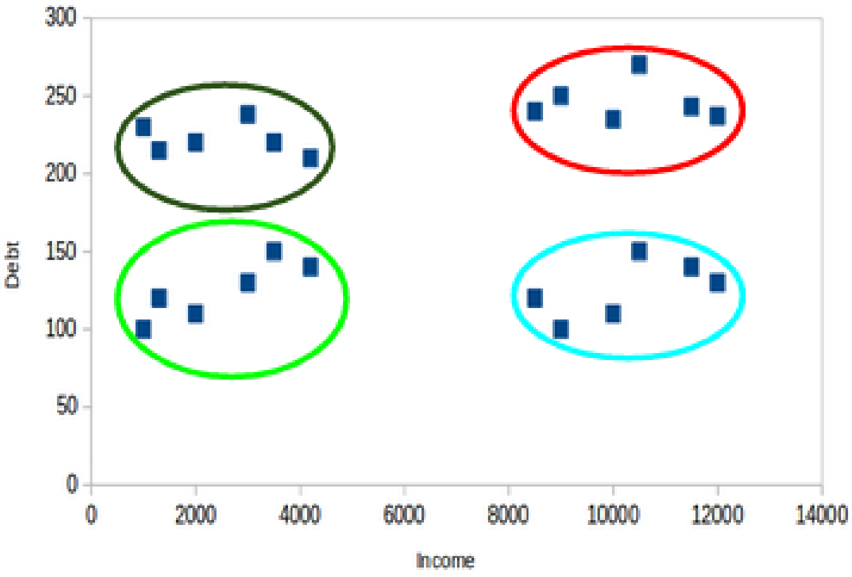
Case Study-Customer segmentation

Property 2

The data points from different clusters should be as different as possible. Let's again take the same example to understand this property:



Case - I



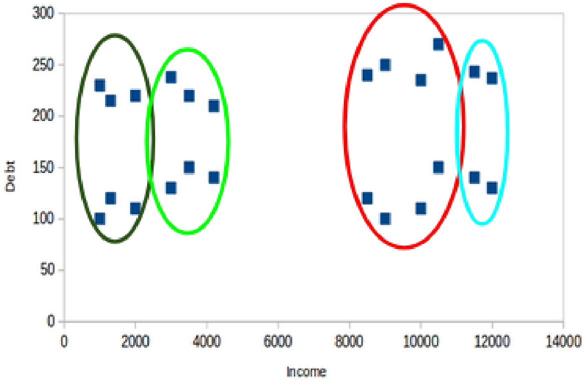
Case - II

Major Clustering Approaches

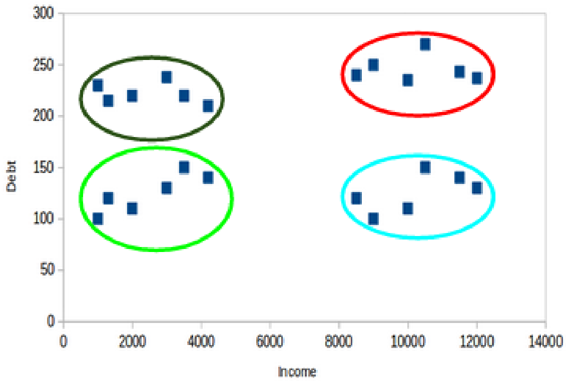
- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue

Evaluation Metrics for Clustering

- The primary aim of clustering is not just to make clusters, but to make good and meaningful ones



Case - I



Case - II

Here, we used only two features and hence it was easy for us to visualize and decide which of these clusters is better.

Unfortunately, that's not how real-world scenarios work. We will have a ton of features to work with. In customer segmentation example – we will have features like customer's income, occupation, gender, age, and many more. Visualizing all these features together and deciding better and meaningful clusters would not be possible for us.

This is where we can make use of evaluation metrics.

Evaluation Metrics for Clustering

Inertia

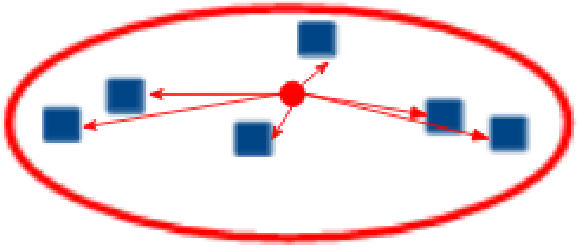
- It tells us how far the points within a cluster are. So, **inertia actually calculates the sum of distances of all the points within a cluster from the centroid of that cluster.**
- We calculate this for all the clusters and the final inertial value is the sum of all these distances. This distance within the clusters is known as **intracluster distance**. So, inertia gives us the sum of intracluster distances:

▪Now, what do you think should be the value of inertia for a good cluster?

▪Is a small inertial value good or do we need a larger value?

▪We want the points within the same cluster to be similar to each other

▪Hence, the distance between them should be as low as possible.

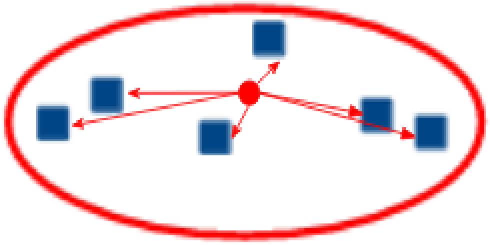


Intra cluster distance

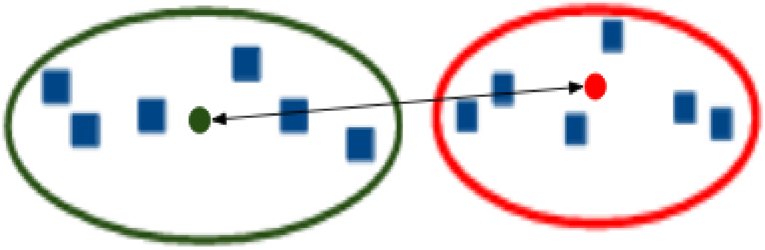
Evaluation Metrics for Clustering

Dunn Index

- We now know that inertia tries to minimize the intracluster distance. It is trying to make more compact clusters.
- if the distance between the centroid of a cluster and the points in that cluster is small, it means that the points are closer to each other.
- So, inertia makes sure that the first property of clusters is satisfied. But it does not care about the second property – that different clusters should be as different from each other as possible.



Intra cluster distance



Inter cluster distance

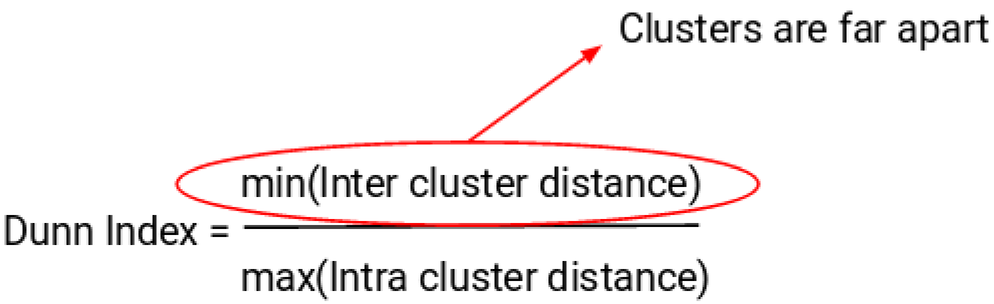
Evaluation Metrics for Clustering

- **the Dunn index also takes into account the distance between two clusters**. This distance between the centroids of two different clusters is known as **inter-cluster distance**.

Dunn index is the ratio of the minimum of inter-cluster distances and maximum of intracluster distances.

$$\text{Dunn Index} = \frac{\text{min(Inter cluster distance)}}{\text{max(Intra cluster distance)}}$$

We want to maximize the Dunn index. The more the value of the Dunn index, the better will be the clusters.



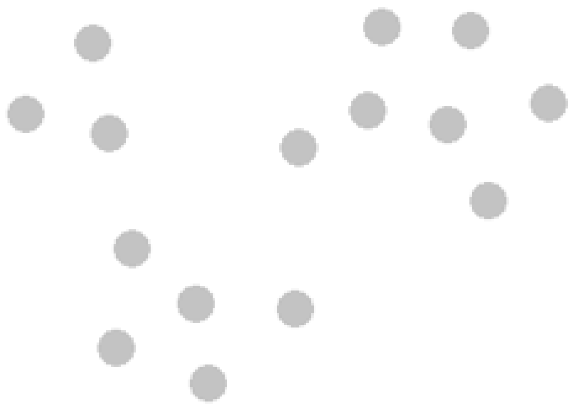
Clusters are far apart

$$\text{Dunn Index} = \frac{\text{min(Inter cluster distance)}}{\text{max(Intra cluster distance)}}$$

K-Means Clustering

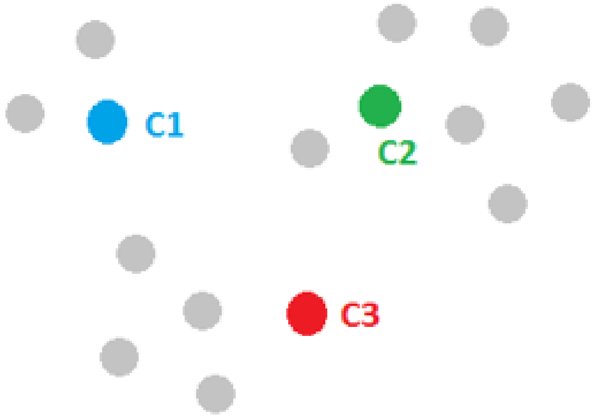
K-Means-Steps

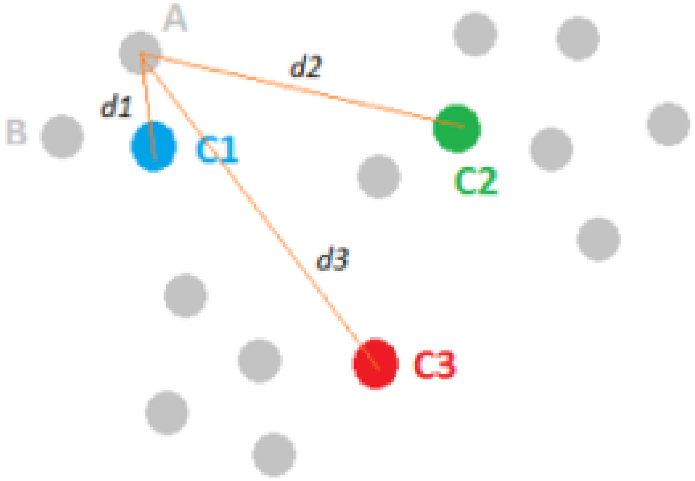
- k-means clustering aims to find the set of k clusters such that every data point is assigned to the closest center, and the sum of the distances of all such assignments is minimized.
- Imaging we have these gray points in the following figure and want to assign them into **three** clusters. K-means follows the four steps listed below.



K-Means-steps

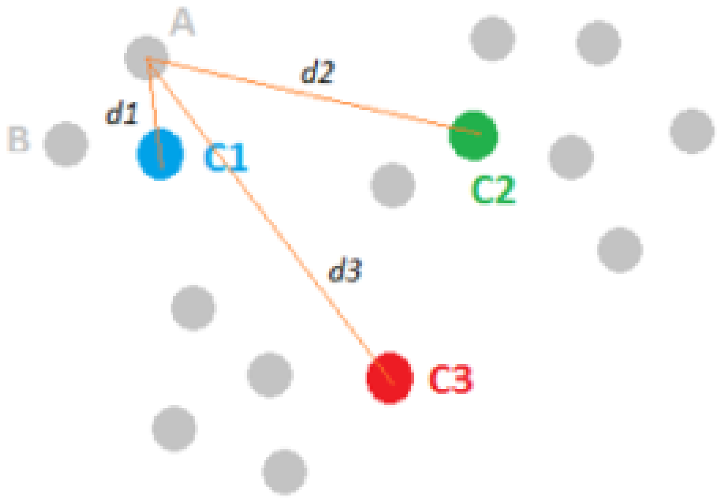
- ***Step one:*** *Initialize cluster centers*
- We randomly pick three points C1, C2 and C3, and label them with blue, green and red color separately to represent the cluster centers.





Step two: *Assign observations to the closest cluster center*

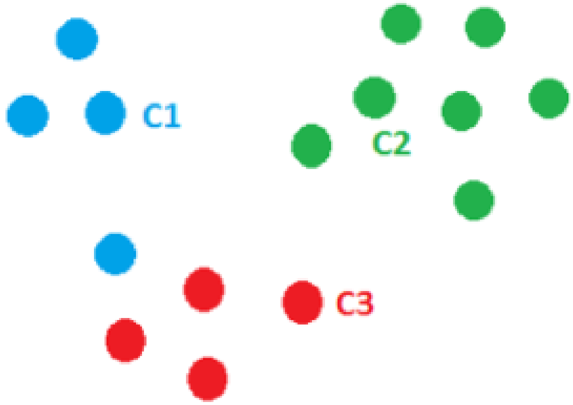
K-Means-steps



- Once we have these cluster centers, we can assign each point to the clusters based on the minimum distance to the cluster center.
- For the gray point A, compute its distance to C1, C2 and C3, respectively.
- And after comparing the lengths of *d1*, *d2* and *d3*, we figure out that *d1* is the smallest, therefore, we assign point A to the blue cluster and label it with blue.

K-Means-steps

- We then move to point B and follow the same procedure. This process can assign all the points and leads to the following figure.

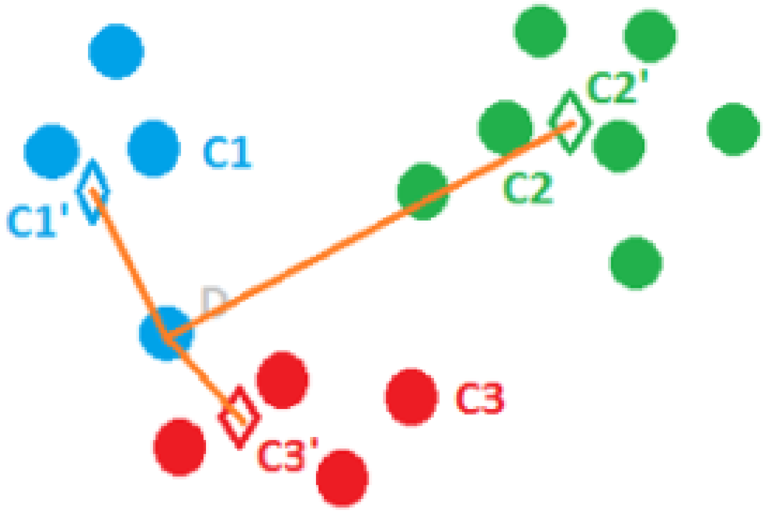


Step three: *Revise cluster centers as mean of assigned observations*

Now we've assigned all the points based on which cluster center they were closest to. Next, we need to update the cluster centers based on the points assigned to them.

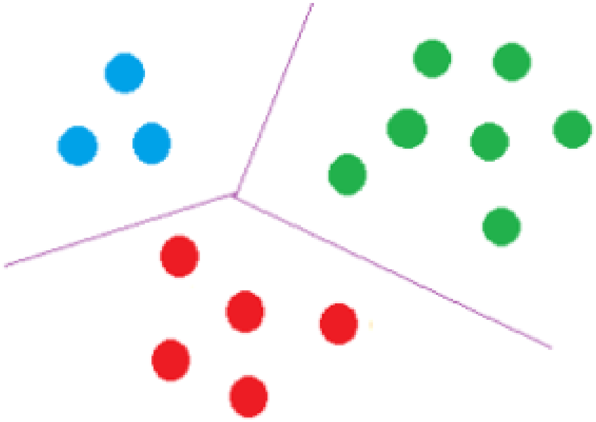
K-Means-steps

- For instance, we can find the center mass of the blue cluster by summing over all the blue points and dividing by the total number of points, which is four here.
- And the resulted center mass C1’, represented by a blue diamond, is our new center for the blue cluster. Similarly, we can find the new centers C2’ and C3’ for the green and red clusters.



K-Means-steps

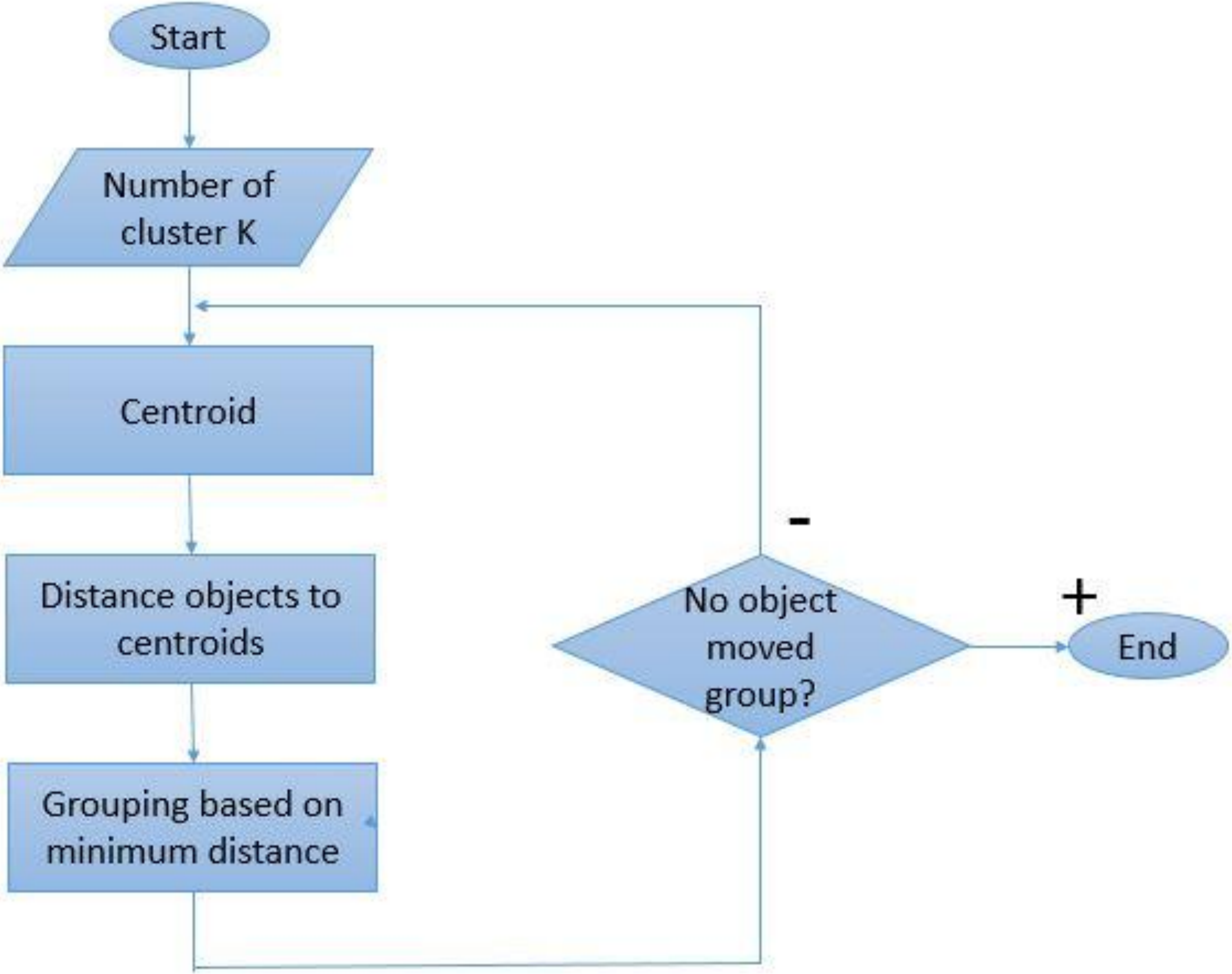
- ***Step four:* Repeat step 2 and step 3 until convergence**
- The last step of k-means is just to repeat the above two steps. For example, in this case, once C1’, C2’ and C3’ are assigned as the new cluster centers, point D becomes closer to C3’ and thus can be assigned to the red cluster. We keep on iterating between assigning points to cluster centers, and updating the cluster centers until convergence.
- Finally, we may get a solution like the following figure.



Some additional remarks about K-means

▪**The k-means algorithm converges to local optimum. Therefore, the result found by K-means is not necessarily the most optimal one.**

▪**The initialization of the centers is critical to the quality of the solution found.**


Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object moved group?
-
+
End

Comments on the *K-Means* Method

- **Strength:** ***Efficient*: *O*(*tkn*),** where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
- Often terminates at a *local optimal*.
- **Weakness**
  - Applicable only to objects in a continuous n-dimensional space
  - Need to specify *k,* the *number* of clusters, in advance
  - Sensitive to noisy data and *outliers*

Hierarchical Clustering

Hierarchical Clustering

- K-means clustering is an iterative process. It will keep on running until the centroids of newly formed clusters do not change or the maximum number of iterations are reached.

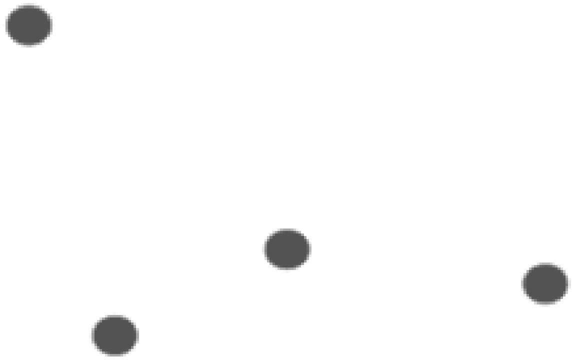
- But there are certain challenges with K-means. It always tries to make clusters of the same size.

- Also, we have to decide the number of clusters at the *beginning* of the algorithm. Ideally, we would not know how many clusters should we have, in the beginning of the algorithm and hence it a challenge with K-means.

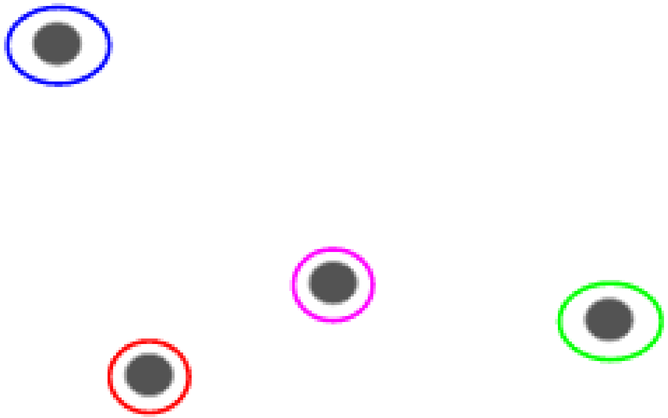
- To solve these two challenges, we can opt for the hierarchical clustering algorithm because, in this algorithm, we don't need to have knowledge about the predefined number of clusters.

Hierarchical Clustering

Let's say we have the below points and we want to cluster them into groups:

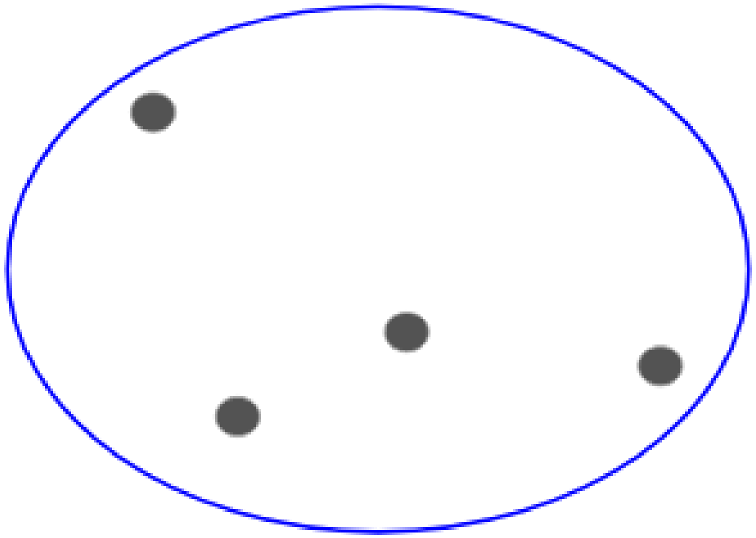


We can assign each of these points to a separate cluster:



Hierarchical Clustering

Now, based on the similarity of these clusters, we can combine the most similar clusters together and repeat this process until only a single cluster is left:



We are essentially building a hierarchy of clusters. That’s why this algorithm is called hierarchical clustering.

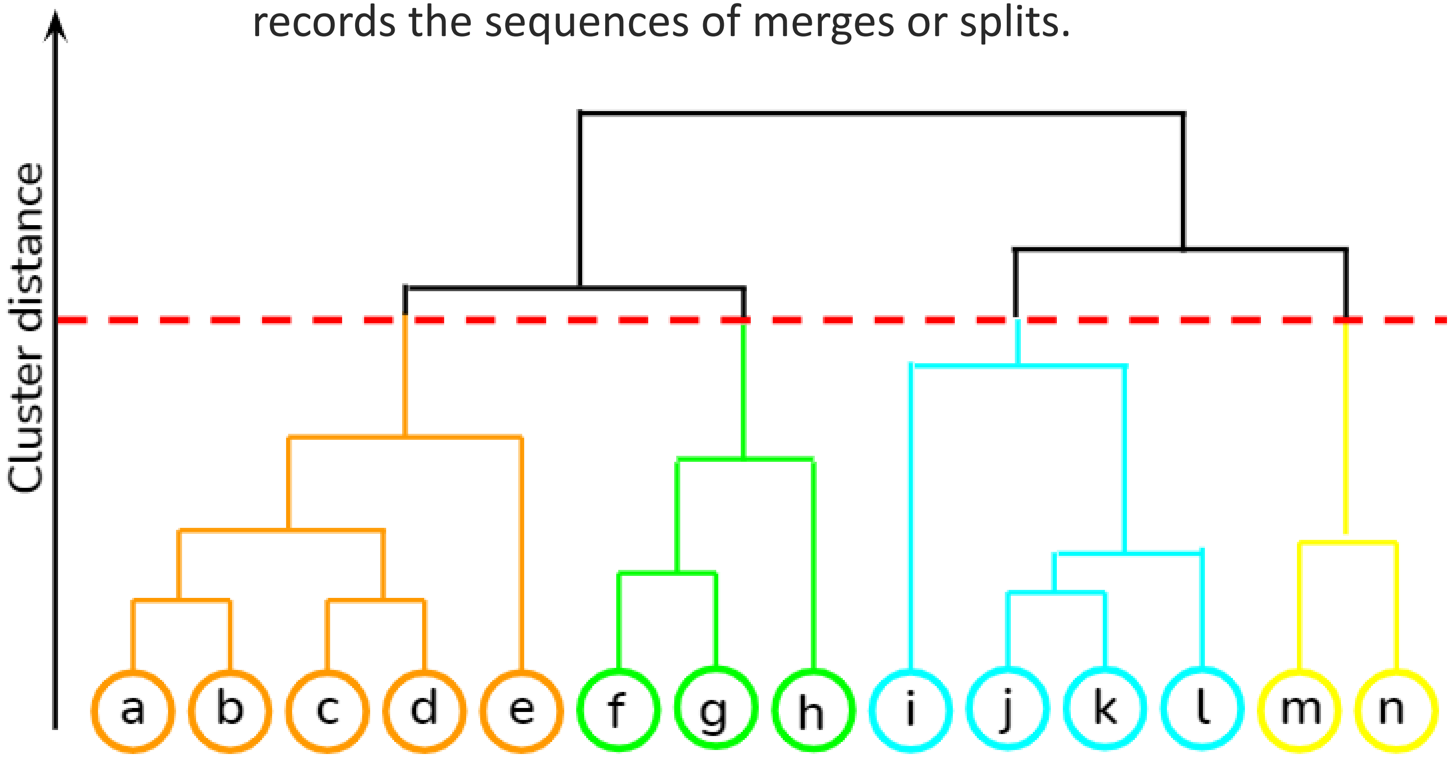
Types of Hierarchical Clustering

There are mainly two types of hierarchical clustering:

1. Agglomerative hierarchical clustering

2. Divisive Hierarchical clustering

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



Agglomerative Hierarchical Clustering

- The agglomerative hierarchical clustering algorithm is a popular example of HCA. To group the datasets into clusters, it follows the **bottom-up approach**.

- It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together.

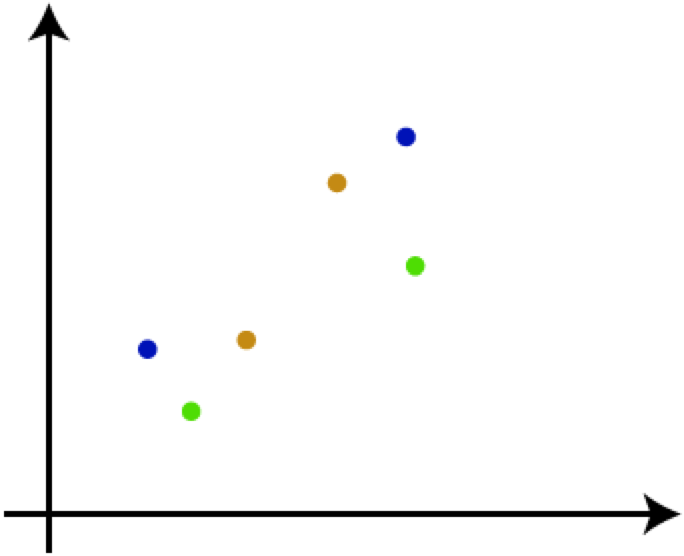
- It does this until all the clusters are merged into a single cluster that contains all the datasets.

- This hierarchy of clusters is represented in the form of the dendrogram.

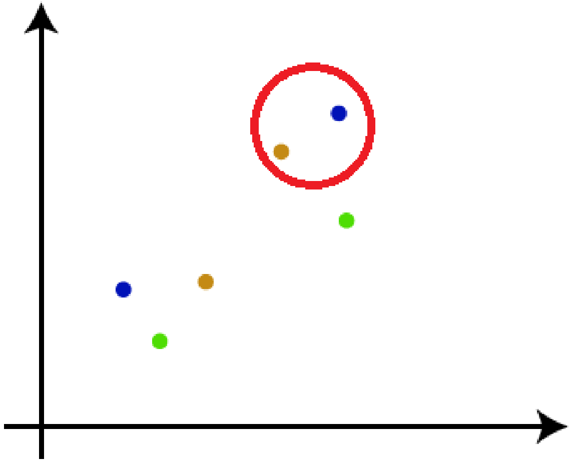
How the Agglomerative Hierarchical clustering Work?

The working of the AHC algorithm can be explained using the below steps:

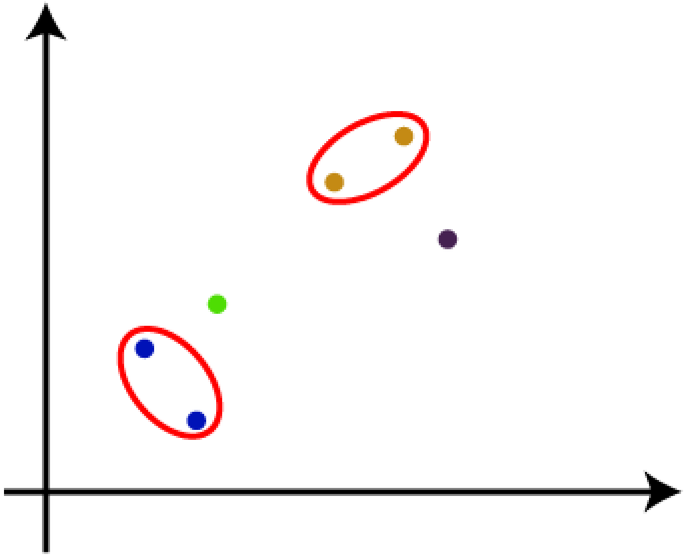
Step-1: Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.



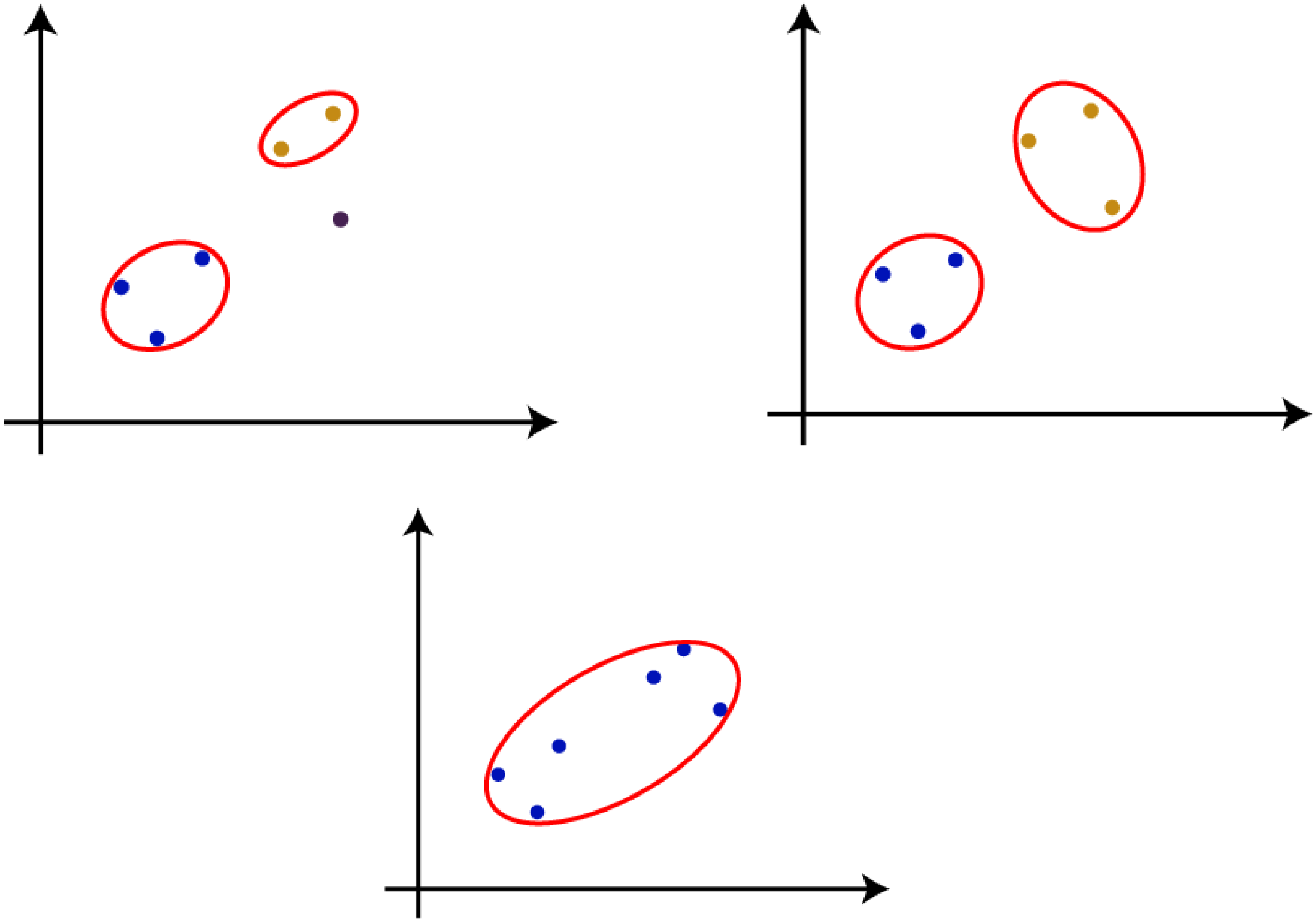
Step-2: Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.



Step-3: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



Step-4: Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images:

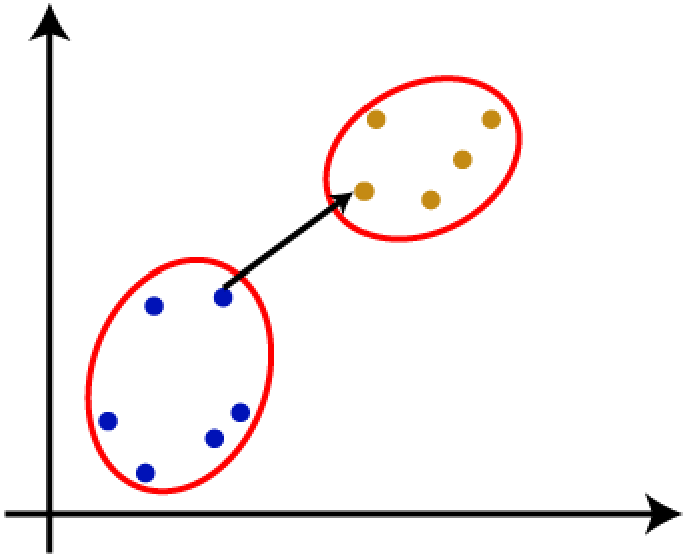


Step-5: Once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per the problem.

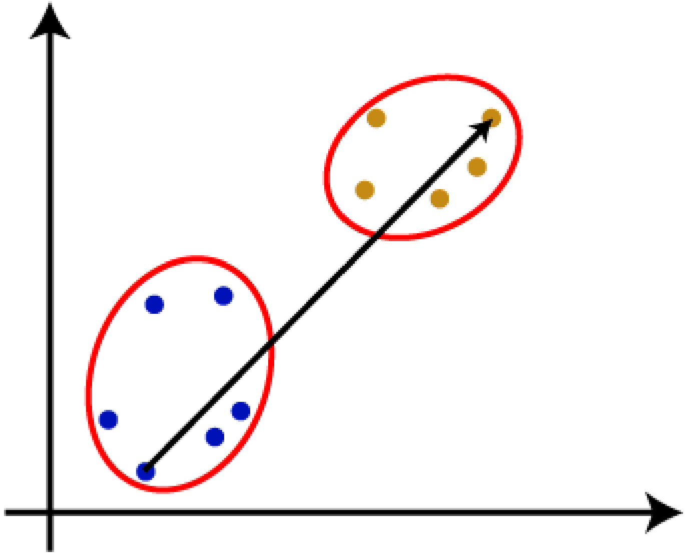
Measure for the distance between two clusters

As we have seen, the **closest distance** between the two clusters is crucial for the hierarchical clustering. There are various ways to calculate the distance between two clusters, and these ways decide the rule for clustering. These measures are called **Linkage methods**. Some of the popular linkage methods are given below:

Single Linkage: It is the Shortest Distance between the closest points of the clusters. Consider the below image:

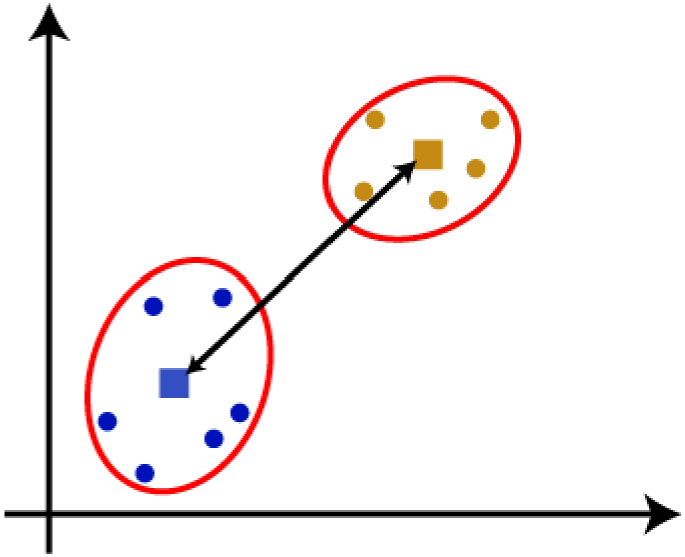


Complete Linkage: It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.

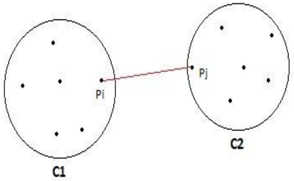


Average Linkage: It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. It is also one of the most popular linkage methods.

Centroid Linkage: It is the linkage method in which the distance between the centroid of the clusters is calculated. Consider the below image:

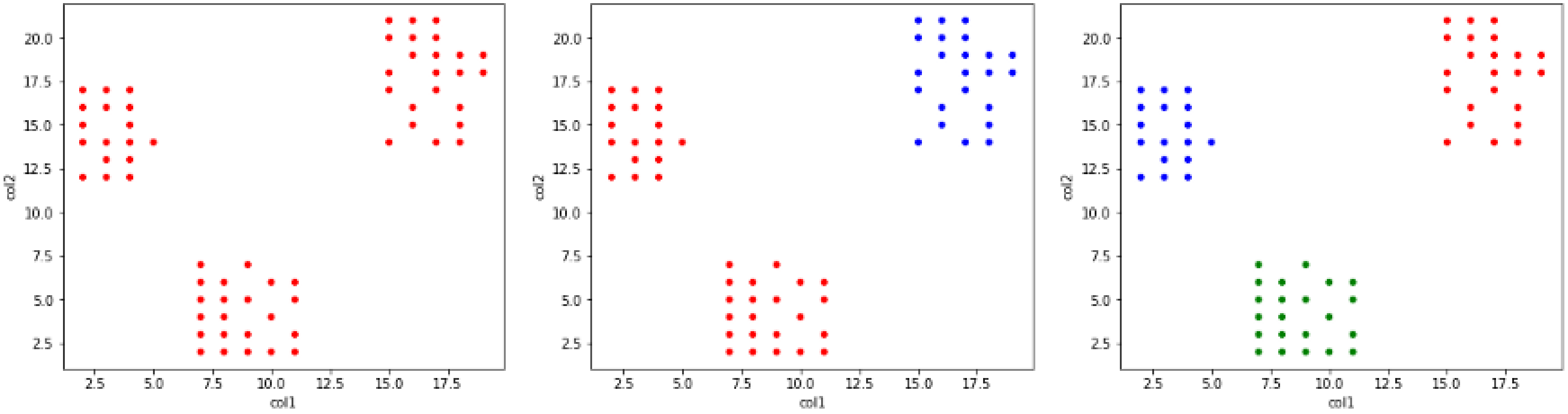


Ward Linkage: It is the linkage method that minimizes variance, measured by an index called E. This approach of calculating the similarity between two clusters is exactly the same as Group Average except that Ward's method calculates the sum of the square of the distances Pi and PJ.



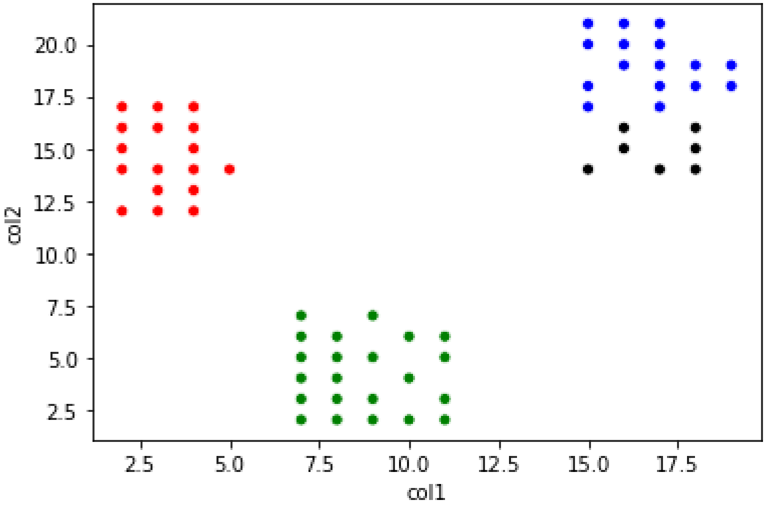
Divisive Clustering:
The divisive clustering algorithm is a top-down clustering approach, initially, all the points in the dataset belong to one cluster and split is performed recursively as one moves down the hierarchy.

Steps of Divisive Clustering:
Initially, all points in the dataset belong to one single cluster.
Partition the cluster into two least similar cluster
Proceed recursively to form new clusters until the desired number of clusters is obtained.



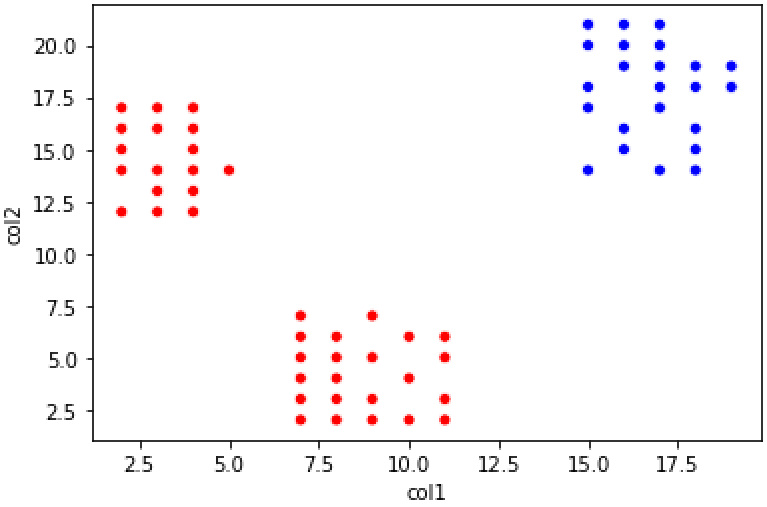
In this sample dataset, it is observed that there is 3 cluster that is far separated from each other. So we stopped after getting 3 clusters.

Even if start separating further more clusters, below is the obtained result.



How to choose which cluster to split?

Check the sum of squared errors of each cluster and choose the one with the largest value. In the below 2-dimension dataset, currently, the data points are separated into 2 clusters, for further separating it to form the 3rd cluster find the sum of squared errors (SSE) for each of the points in a red cluster and blue cluster.



The cluster with the largest SSE value is separated into 2 clusters, hence forming a new cluster. In the above image, it is observed red cluster has larger SSE so it is separated into 2 clusters forming 3 total clusters.