Unsupervised Learning

Zeham Management Technologies BootCamp

by SDAIA

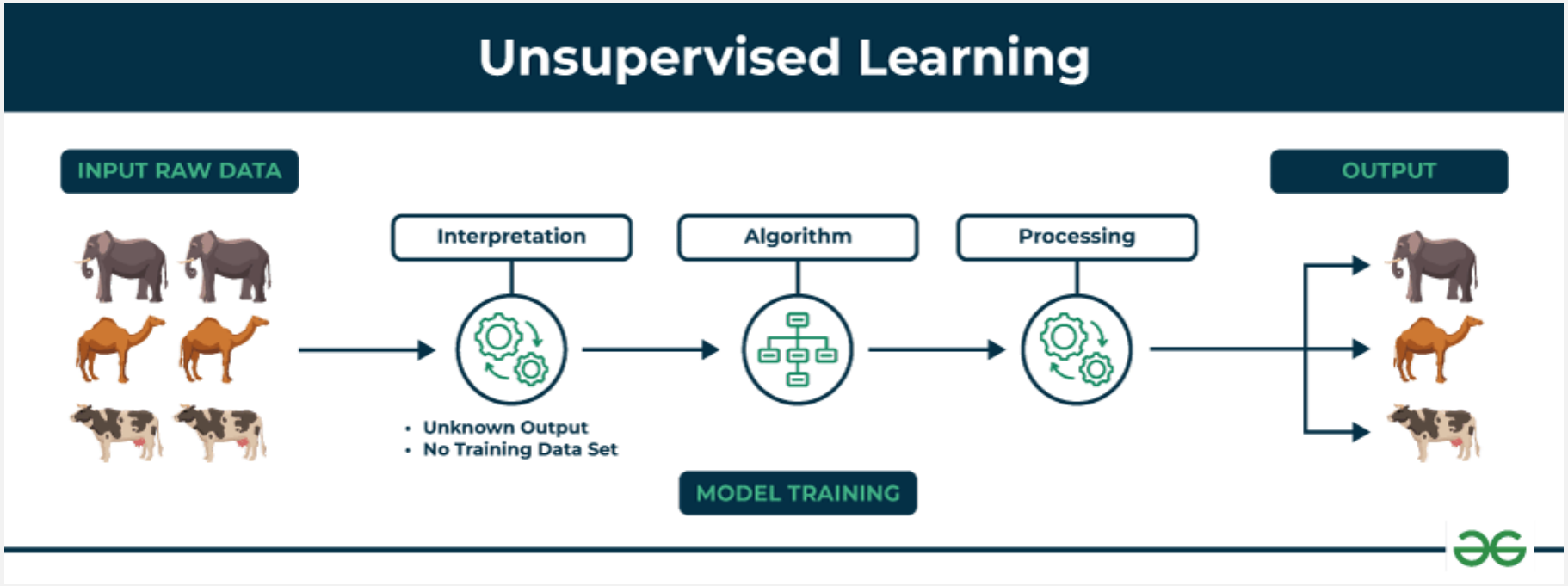
August 6th, 2024

Agenda

- Unsupervised Learning.
- Clustering Methods.
- Clustering Algorithms.

Introduction to unsupervised learning

Introduction to unsupervised learning



Source

Supervised vs. Unsupervised

	Supervised Learning	Unsupervised learning
Objective	To approximate a function that maps inputs to outputs based out example input-output pairs.	To build a concise representation of the data and generate imaginative content from it.
Accuracy	Highly accurate and reliable.	Less accurate and reliable.
Complexity	Simpler method.	Computationally complex.
Classes	Number of classes is known.	Number of classes is unknown.

Source

Introduction to unsupervised learning

- Unsupervised learning in artificial intelligence is a type of machine learning where models learn from data without human supervision. Unlike supervised learning, unsupervised machine learning models are provided with unlabeled data and allowed to discover patterns and insights without any explicit guidance or instruction.

How does unsupervised learning work?

- As the name suggests, unsupervised learning uses self-learning algorithms that learn without any labels or prior training. Instead, the model is given raw, unlabeled data and must infer its own rules and structure the information based on similarities, differences, and patterns without explicit instructions on how to handle each piece of data.

- Unsupervised learning algorithms are better suited for complex processing tasks, such as organizing large datasets into clusters. They are useful for identifying previously undetected patterns in data and can help pinpoint features useful for categorizing data.

How does unsupervised learning work?

Imagine you have a large dataset about weather. An unsupervised learning algorithm will analyze the data and identify patterns among the data points. For instance, it might group data by temperature or similar weather patterns.

While the algorithm itself does not understand these patterns based on any previous information you provided, you can review the data groupings and classify them based on your understanding of the dataset. For example, you might recognize that different temperature groups correspond to the four seasons, or that the weather patterns are categorized into types such as rain, sleet, or snow.

How does unsupervised learning work?

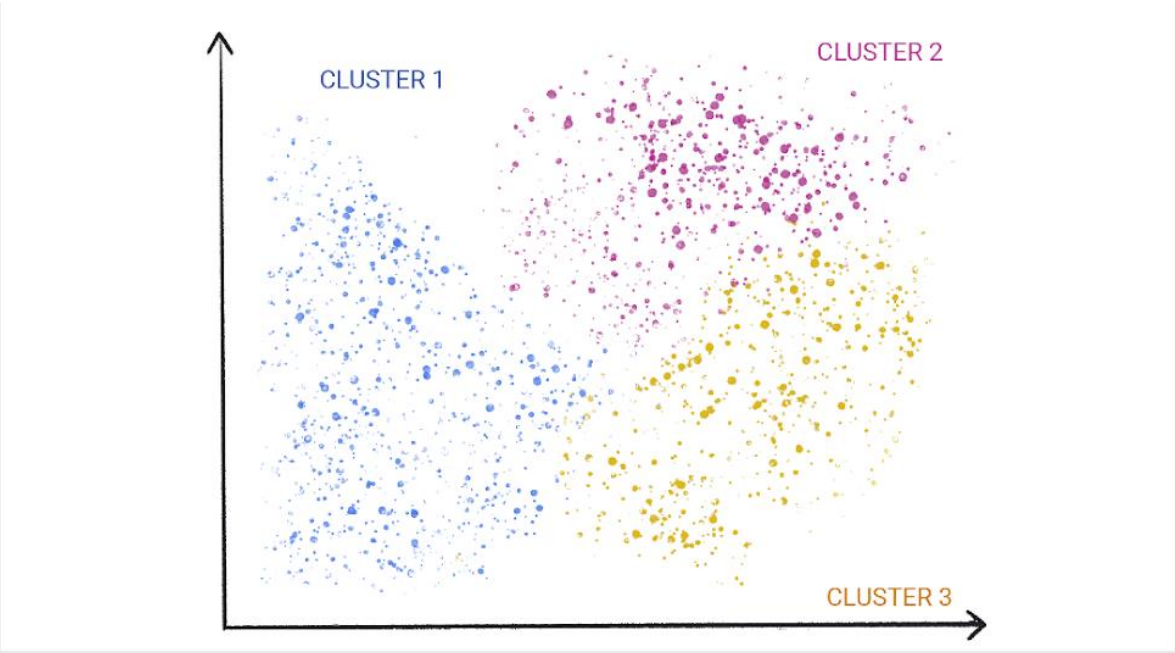


Figure 1. An ML model clustering similar data points.

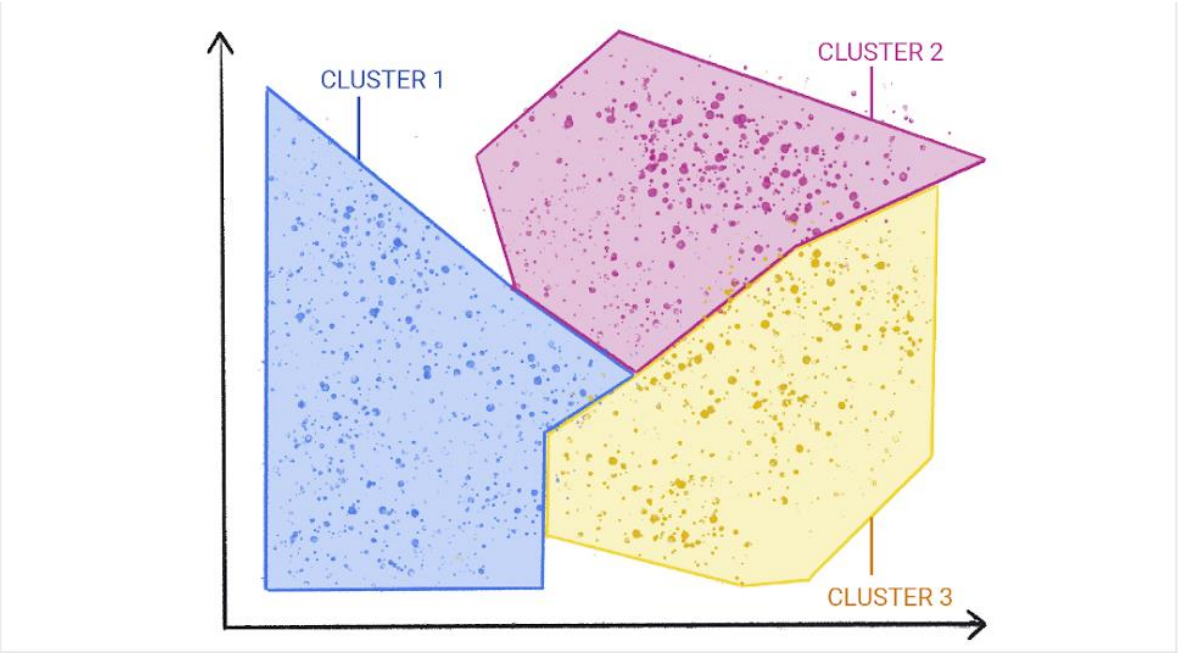
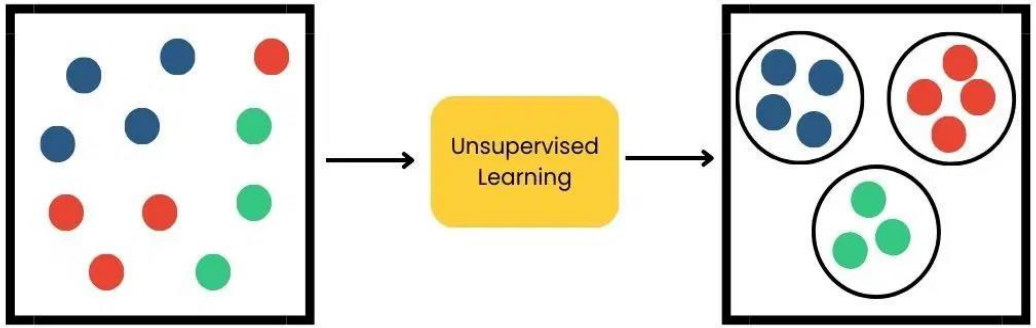


Figure 2. Groups of clusters with natural demarcations.

In short,

- Unsupervised learning deals with unlabeled data.
- The goal is to find hidden patterns or structures in the data.
- Common techniques include clustering, dimensionality reduction, and anomaly detection.



Source

Unsupervised Machine Learning Methods

In general, there are three types of unsupervised learning tasks: clustering, association rules, and dimensionality reduction.

Below, we'll delve a little deeper into each type of unsupervised learning technique. Which are :

1. **Clustering** involves grouping a set of objects in such a way that objects in the same group (or cluster) are more like each other than to those in other groups. Common clustering algorithms include:
    - K-Means: Partitions the dataset into K clusters by minimizing the variance within each cluster.
    - Hierarchical Clustering: Builds a tree of clusters by either iteratively merging smaller clusters or splitting larger ones.
    - DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Forms clusters based on the density of data points, making it effective for datasets with noise and outliers.

Unsupervised Machine Learning Methods

In general, there are three types of unsupervised learning tasks: clustering, association rules, and dimensionality reduction.

Below, we'll delve a little deeper into each type of unsupervised learning technique. Which are :

2. **Association Rules:** Association rules are used to find relationships between variables in large datasets. This technique is commonly used in market basket analysis to identify sets of products frequently bought together. Key concepts include:
    - Support: The frequency of a rule appearing in the dataset.
    - Confidence: The likelihood of the rule being true in the dataset.
    - Lift: The ratio of the observed support to that expected if the two items were independent.

Unsupervised Machine Learning Methods

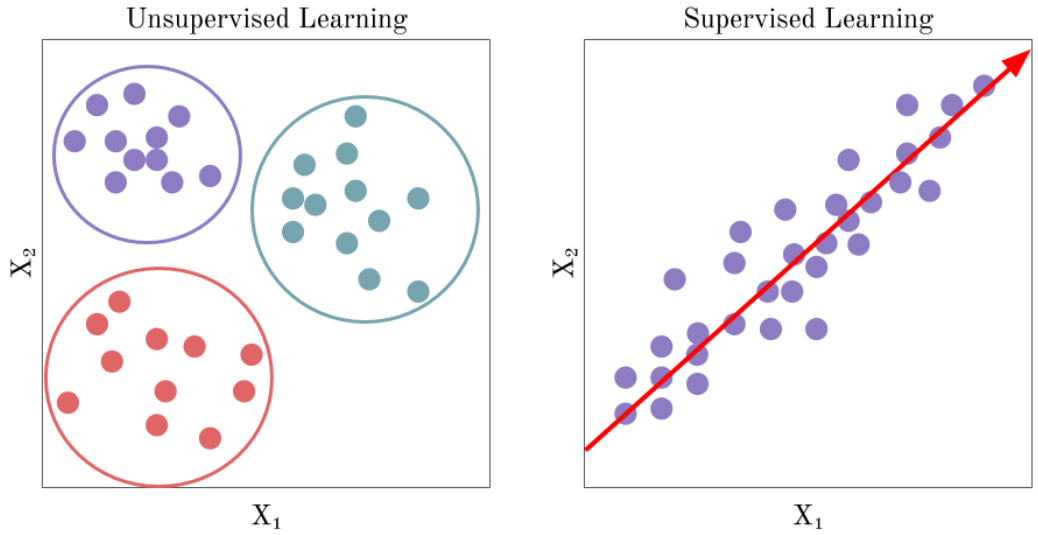
In general, there are three types of unsupervised learning tasks: clustering, association rules, and dimensionality reduction.

Below, we'll delve a little deeper into each type of unsupervised learning technique. Which are :

3. **Dimensionality Reduction:** Dimensionality reduction techniques are used to reduce the number of random variables under consideration, simplifying the dataset while preserving its structure and relationships. Common methods include:
    - Principal Component Analysis (PCA): Reduces the dimensionality of data by transforming it into a new set of variables (principal components) that are uncorrelated and that capture the maximum variance in the data.
    - t-Distributed Stochastic Neighbor Embedding (t-SNE): Reduces high-dimensional data to two or three dimensions for visualization, focusing on preserving local relationships between data points.
    - Autoencoders: Neural networks designed to learn a compressed representation of the input data, often used for noise reduction and feature extraction.

Why use unsupervised learning ?

- Useful when labeled data is not available.
- Helps discover the underlying structure of the data.
- Can be used for preprocessing, such as feature extraction or noise reduction.

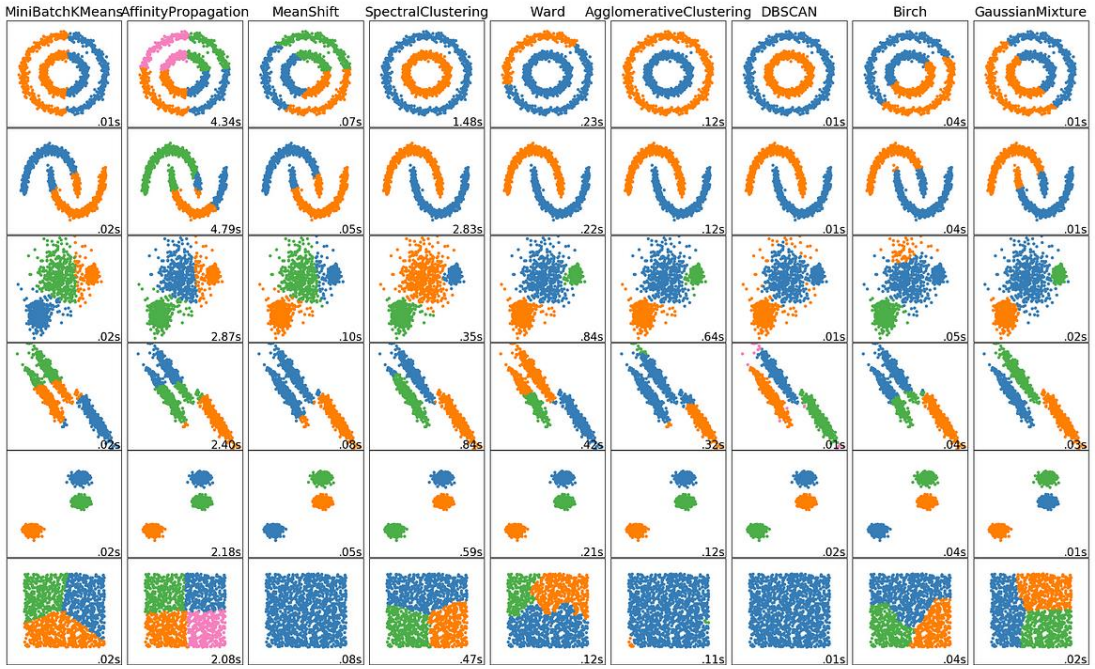


Source

Clustering Models

What is clustering?

- Clustering is a key technique in unsupervised learning.
- Groups similar data points together based on certain criteria.
- Excludes K-means and DBSCAN for this discussion.



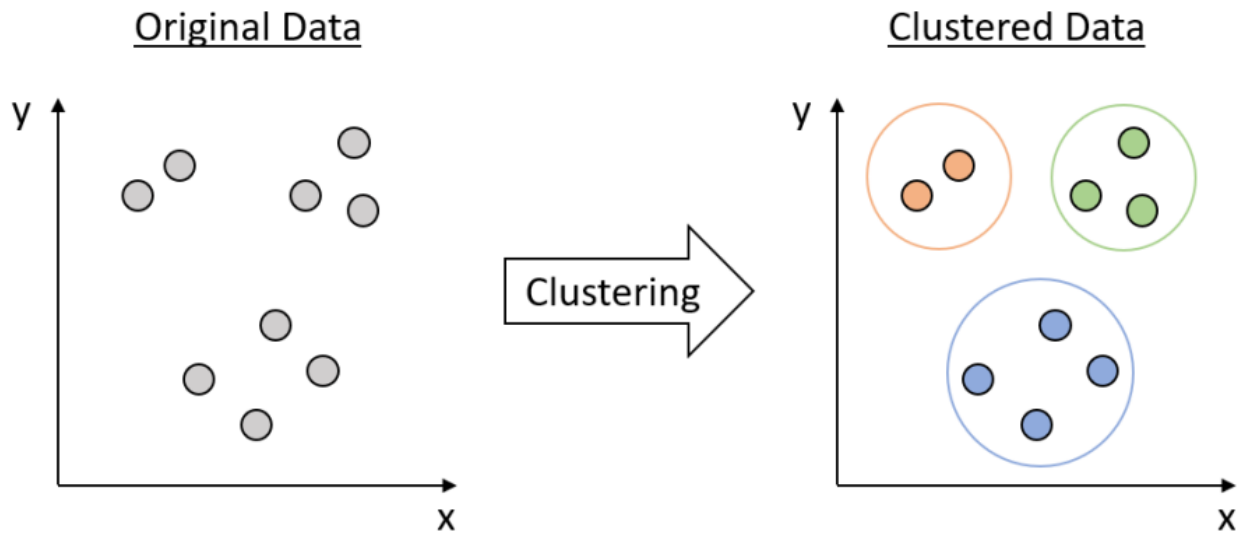
Source

What is clustering?

- Suppose you possess a dataset containing patient details from a healthcare system, featuring both categorical and numerical data, making it complex. You wish to discover patterns and similarities within this dataset. What approach would you take for this task?

What is clustering?

- Clustering is a machine learning technique that groups unlabeled examples based on their similarities. If the examples have labels, this type of grouping is known as classification.

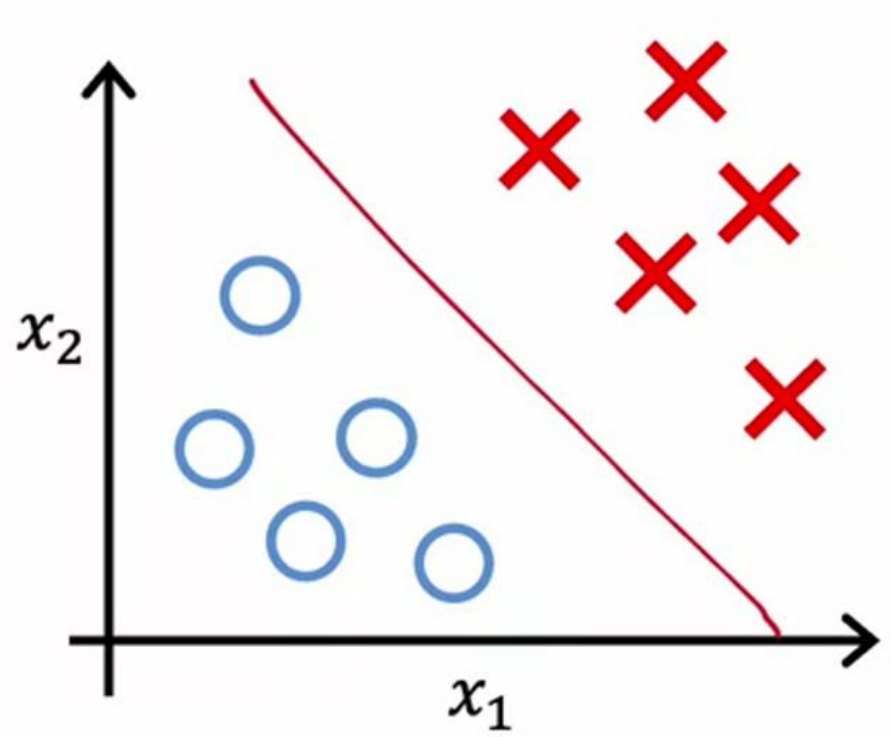


Supervised Learning vs Unsupervised Learning

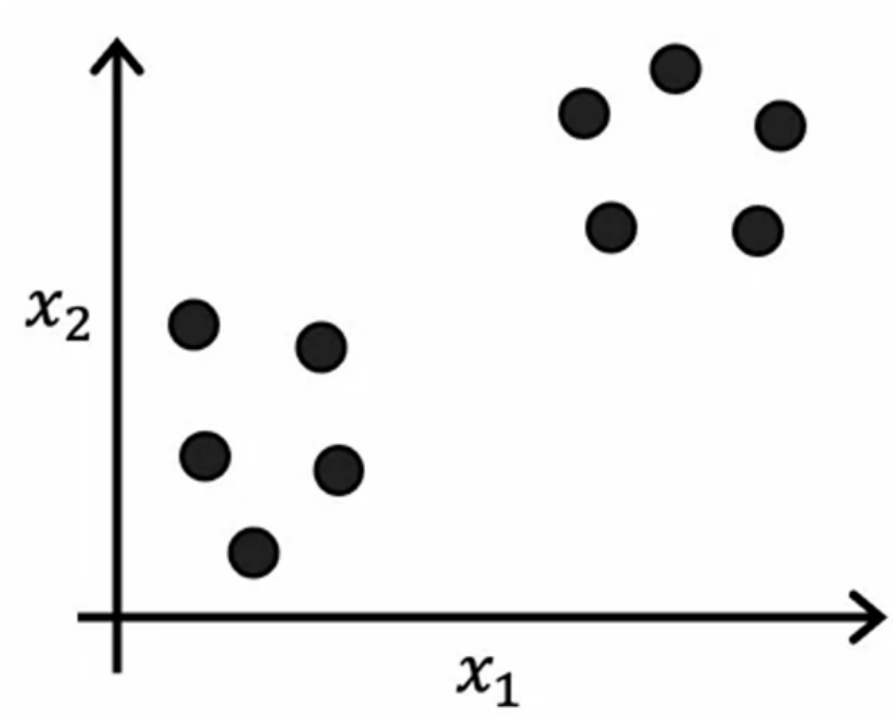
$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$$

$$\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$$

Supervised: labels guide the learning in training phase.



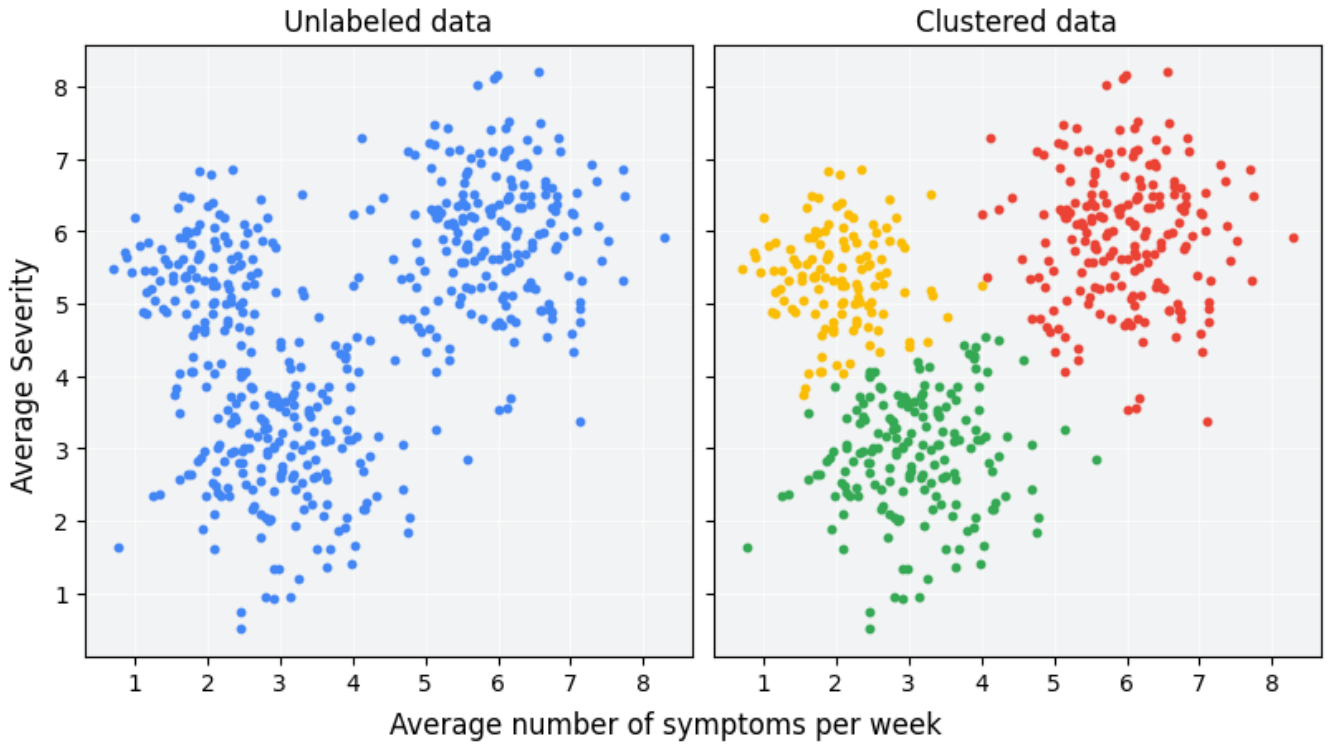
Unsupervised: no labels to guide the model during training.



Clustering Use Cases

Clustering has many use cases and applications including:

- Market segmentation
- Social network analysis
- Search result grouping
- Medical imaging
- Image segmentation
- Anomaly detection



Clustering Methods

Density models: for example, DBSCAN and OPTICS defines clusters as connected dense regions in the data space.

Centroid models(Partitioning methods): for example, the k-means algorithm represents each cluster by a single mean vector.

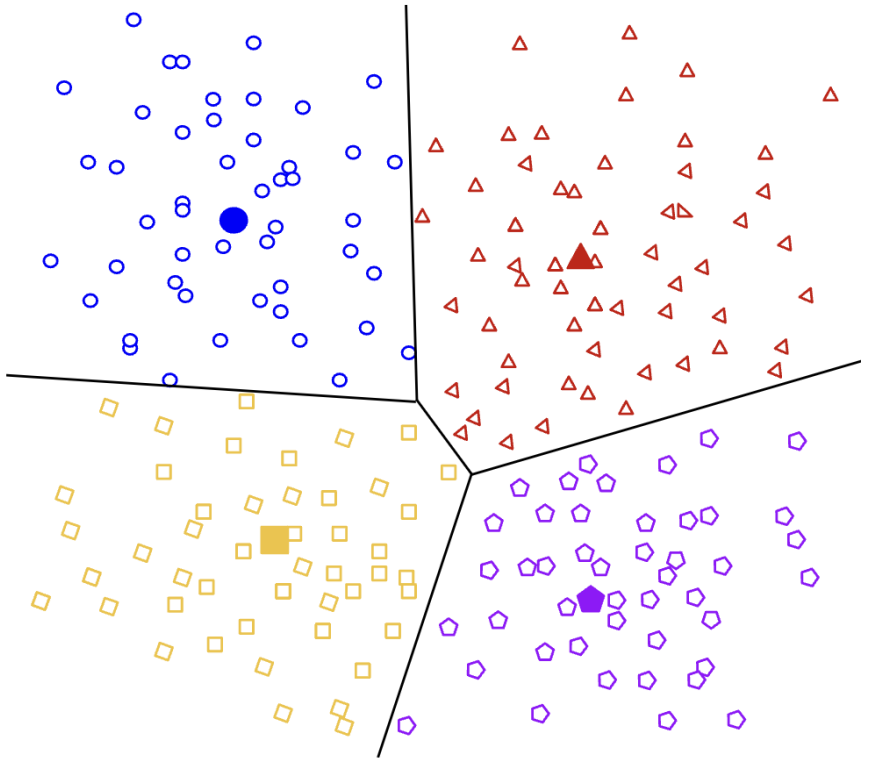
Connectivity models(Hierarchical clustering): for example, hierarchical clustering builds models based on distance connectivity.

Distribution models: for example, multivariate normal distributions used by the expectation-maximization algorithm.

Clustering Algorithms

Centroid-based clustering

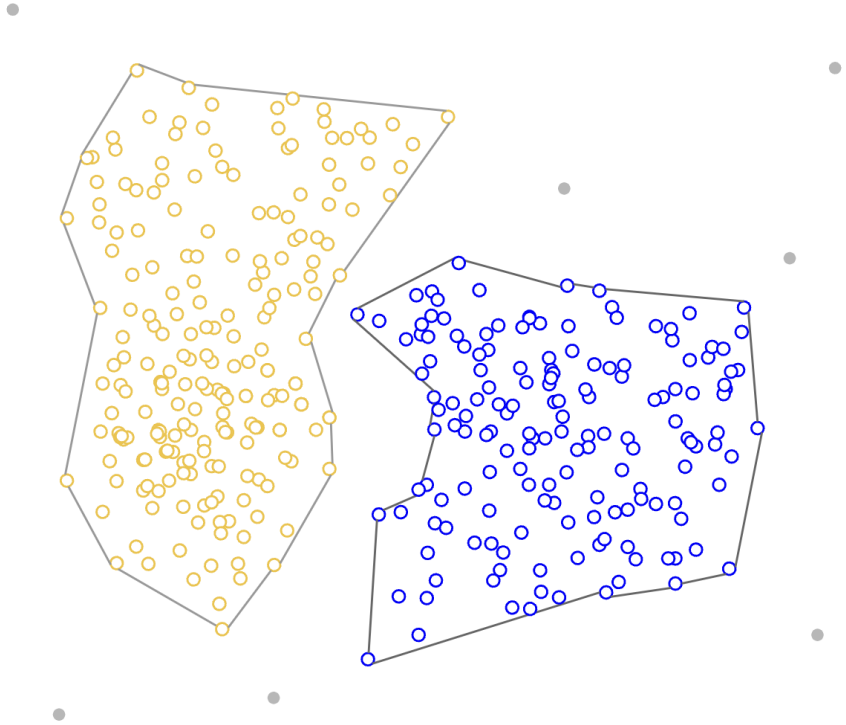
A cluster's centroid is the average of all points within the cluster. Centroid-based clustering arranges data into non-hierarchical clusters. These algorithms are efficient but can be affected by initial conditions and outliers. The most popular of these is k-means, which requires users to set the number of centroids, k, and is effective with clusters of similar size.



Source

Density-based clustering

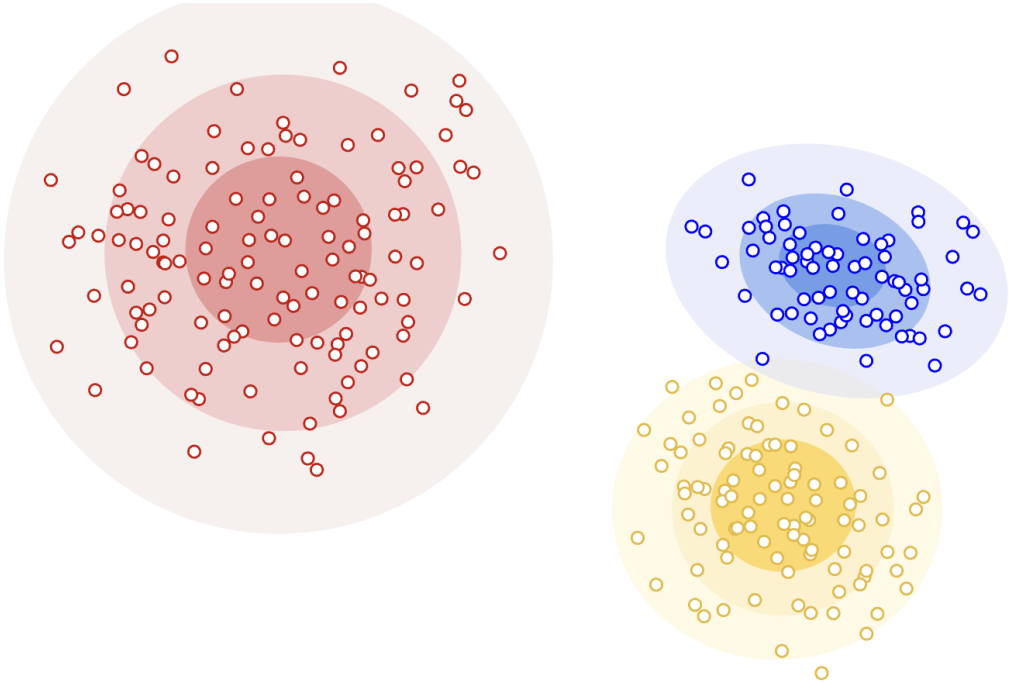
Density-based clustering links continuous regions of high example density into clusters, enabling the detection of clusters in any number and shape. Outliers remain unassigned to clusters. These algorithms struggle with clusters of varying density and high-dimensional data.



Source

Distribution-based clustering

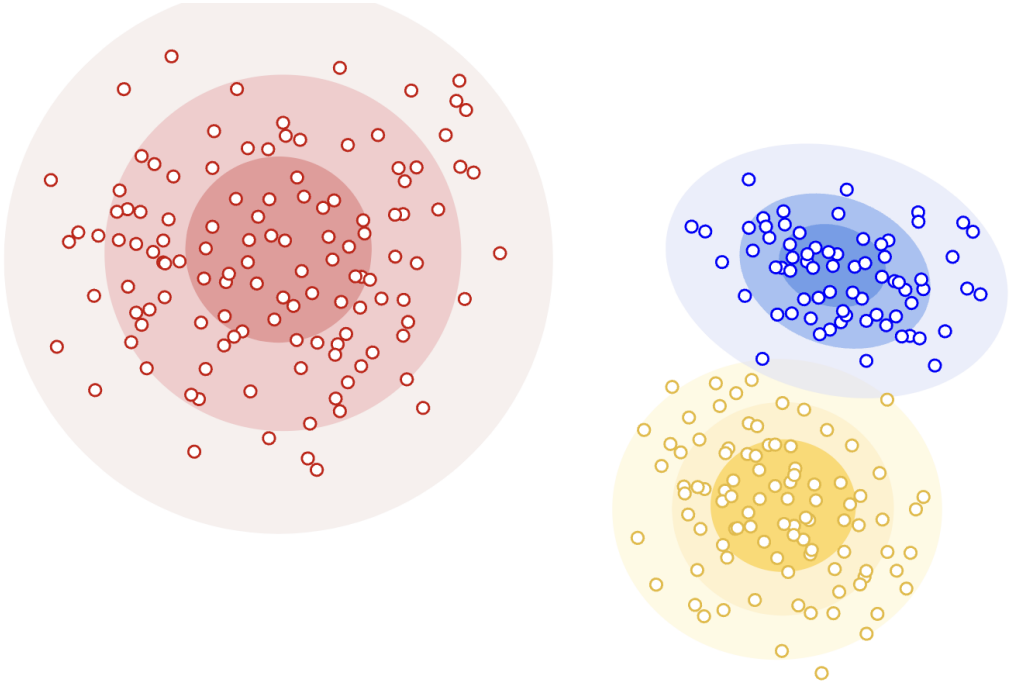
This clustering approach assumes that data consists of probabilistic distributions, like Gaussian distributions. In the figure , the algorithm organizes data into three Gaussian distributions.



Source

Distribution-based clustering Cont.

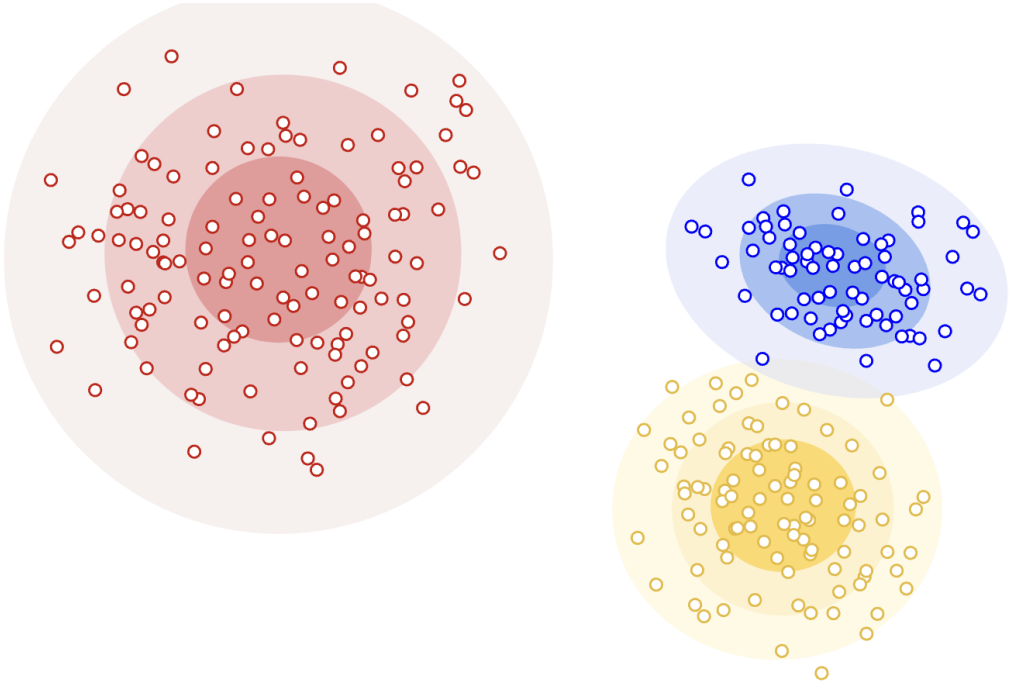
The likelihood of a point belonging to a distribution diminishes as the distance from the center increases, with bands illustrating this decrease in probability.



Source

Distribution-based clustering Cont.

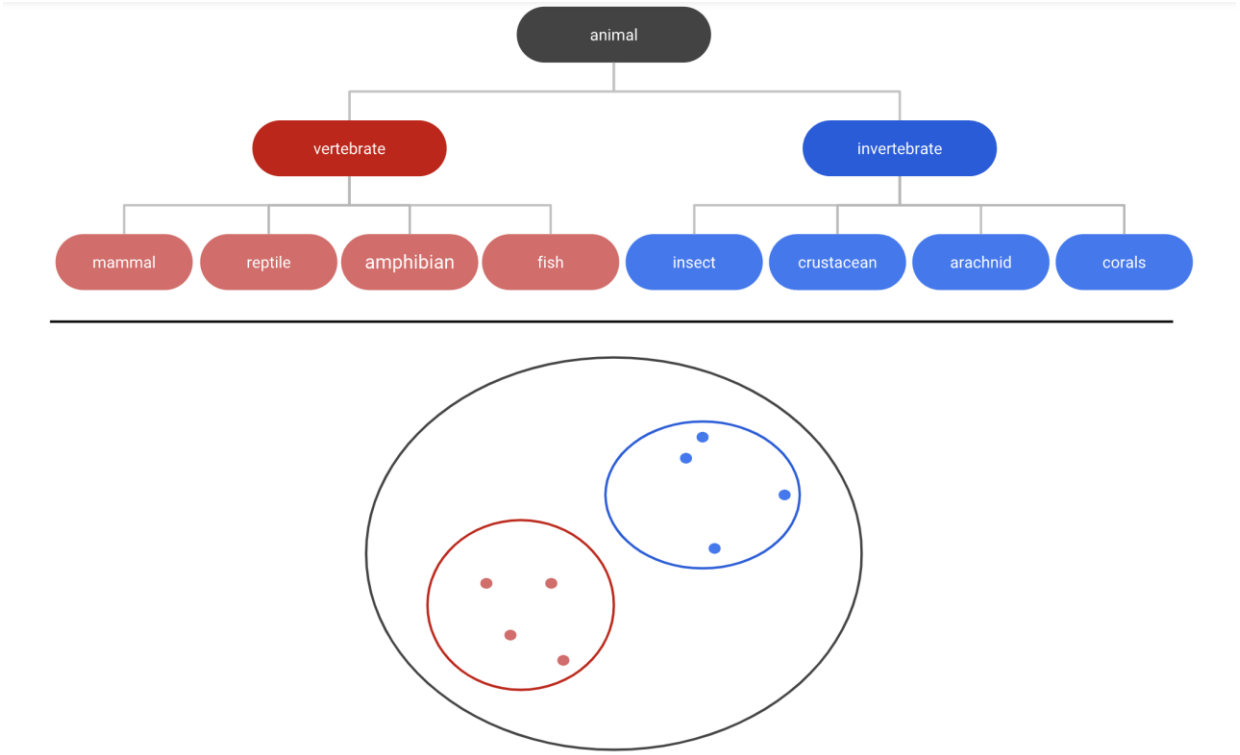
If you're not comfortable assuming a specific distribution for the data, you should opt for a different algorithm.



Source

Hierarchical clustering

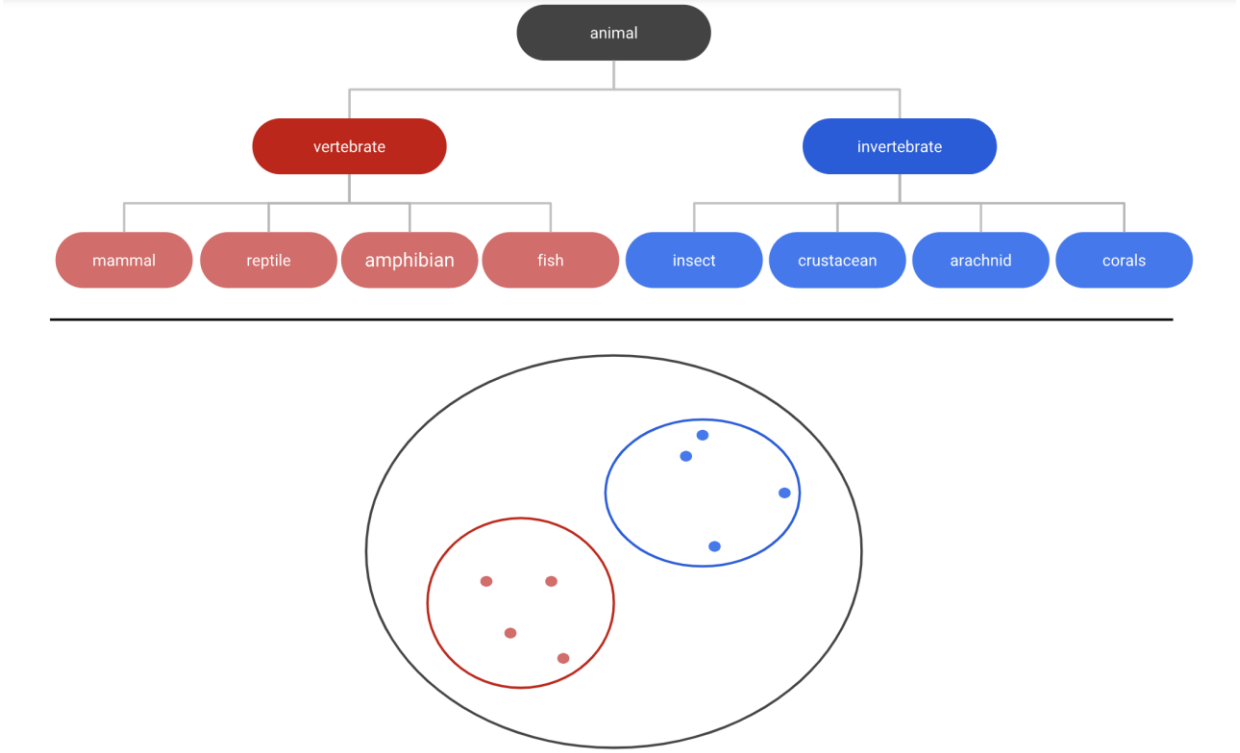
Hierarchical clustering forms a tree of clusters and is particularly suited for hierarchical data, like taxonomies.



Source

Hierarchical clustering Cont.

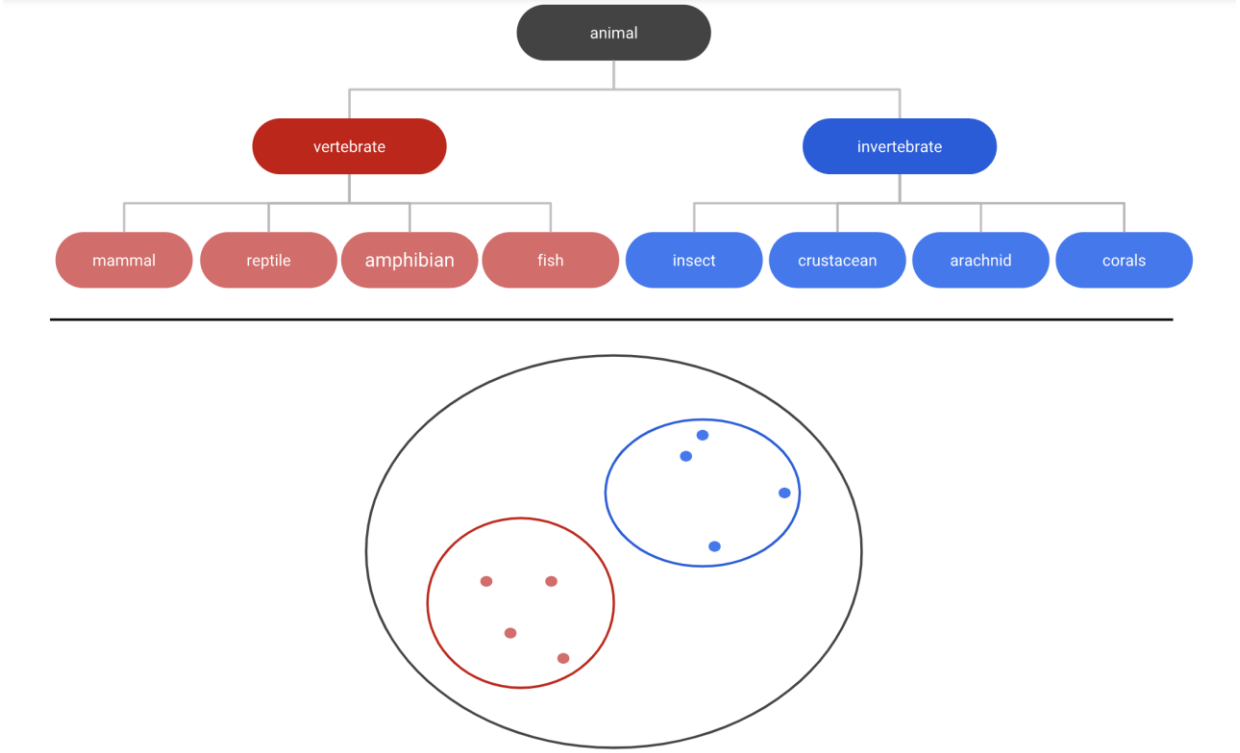
For example, see the study "Comparison of 61 Sequenced Escherichia coli Genomes" by Oksana Lukjancenko, Trudy Wassenaar, and Dave Ussery.



Source

Hierarchical clustering Cont.

You can select any number of clusters by cutting the tree at the appropriate level.

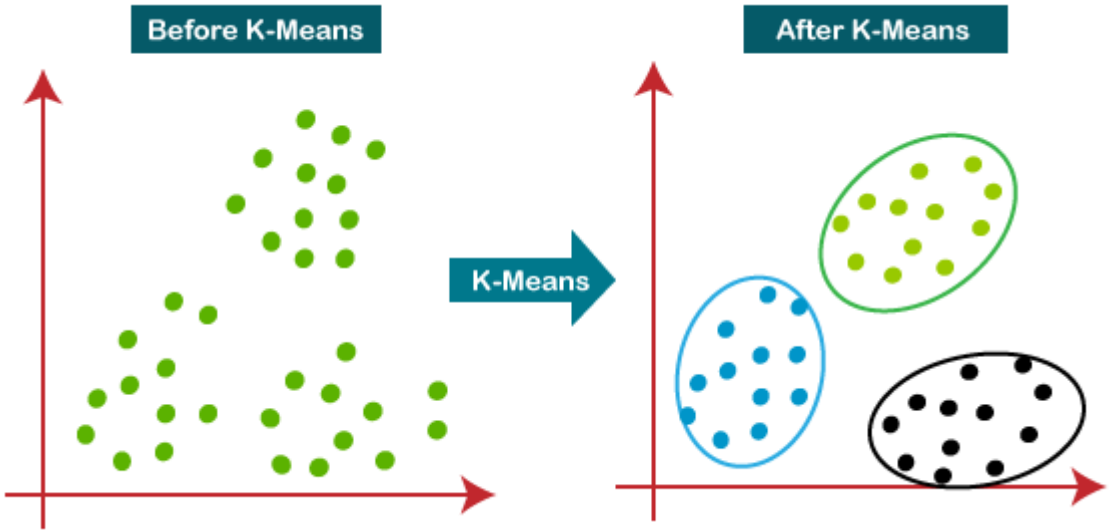


Source

K-means Clustering

What is K-means Clustering

K-means clustering assigns data points to one of the K clusters based on their distance from the cluster centers.



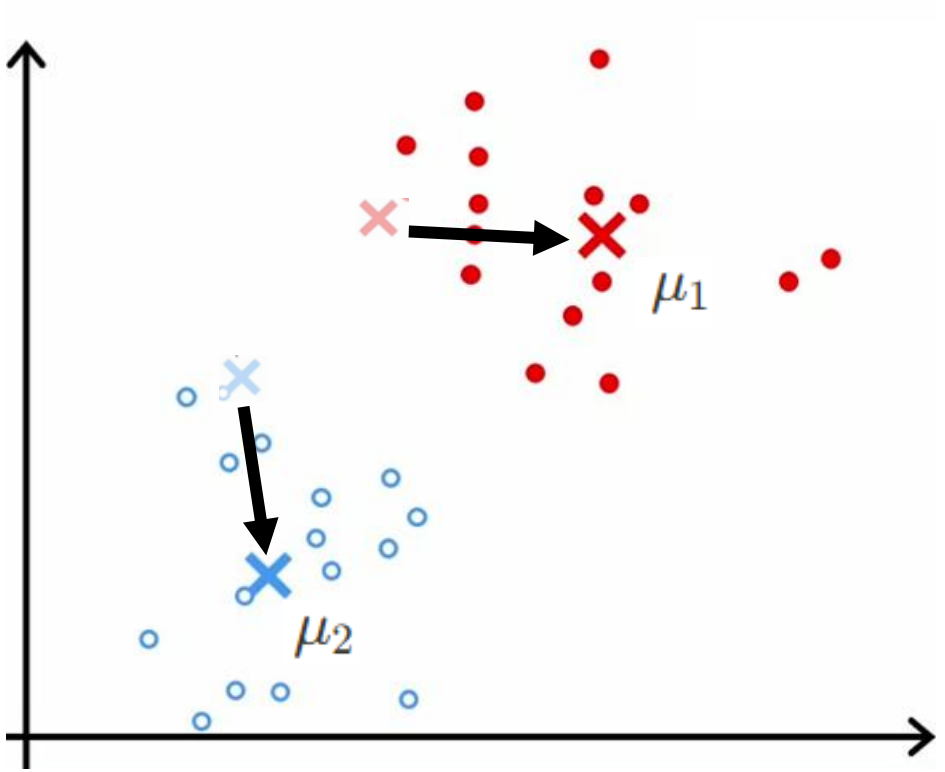
Source

K-means Objective

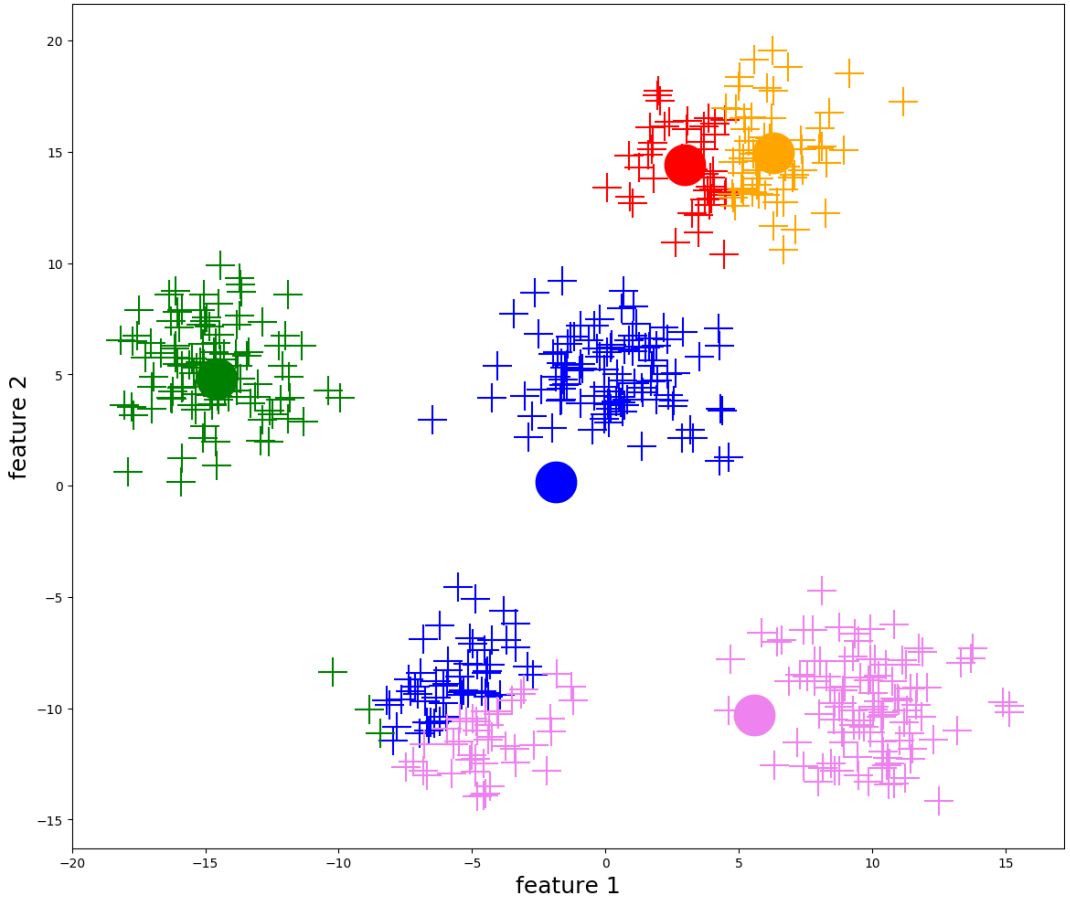
The goal of clustering is to split a population or set of data points into groups where the data points in each group are more like each other and different from those in other groups. It is essentially organizing items based on their similarities and differences.

K-means Clustering Algorithm

1. Specify K: the number of clusters
2. Randomly select K data points as centroids
3. Repeat until convergence:
    1. **Assign** each point in the training dataset to its closest centroid
    2. **Recompute centroids**: center the centroid to newly assigned points
4. Repeat the whole thing 50-1000 times *(or until convergence)*



K-means with K=5 Clusters



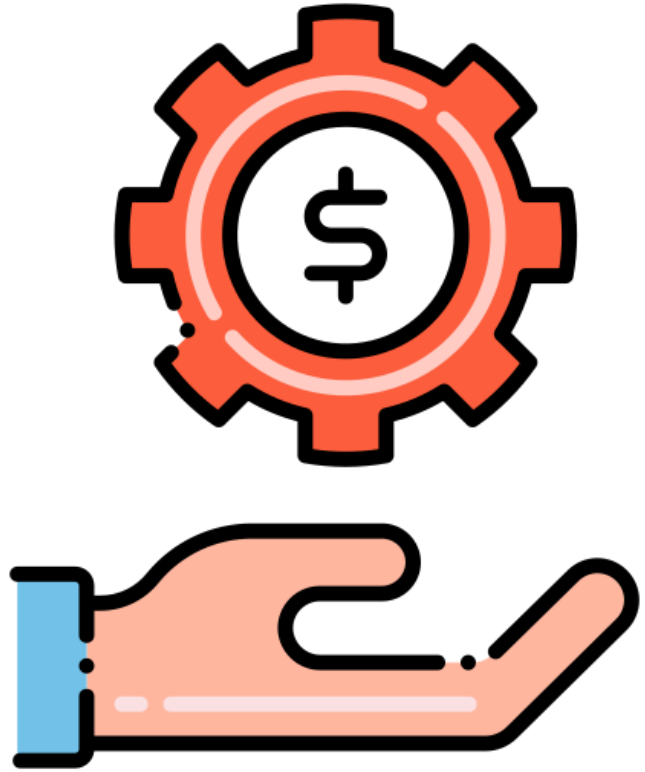
Source: K-means Clustering And Real World Use-Cases.. | by Pritee Dharme . | Medium

K-means Optimization Objective

Cost Function:

Minimize the squared distance between a training example $\chi^{(i)}$ and the location of the centroid $\mu_c{}^{(i)}$ to which has been assigned.

$$J = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

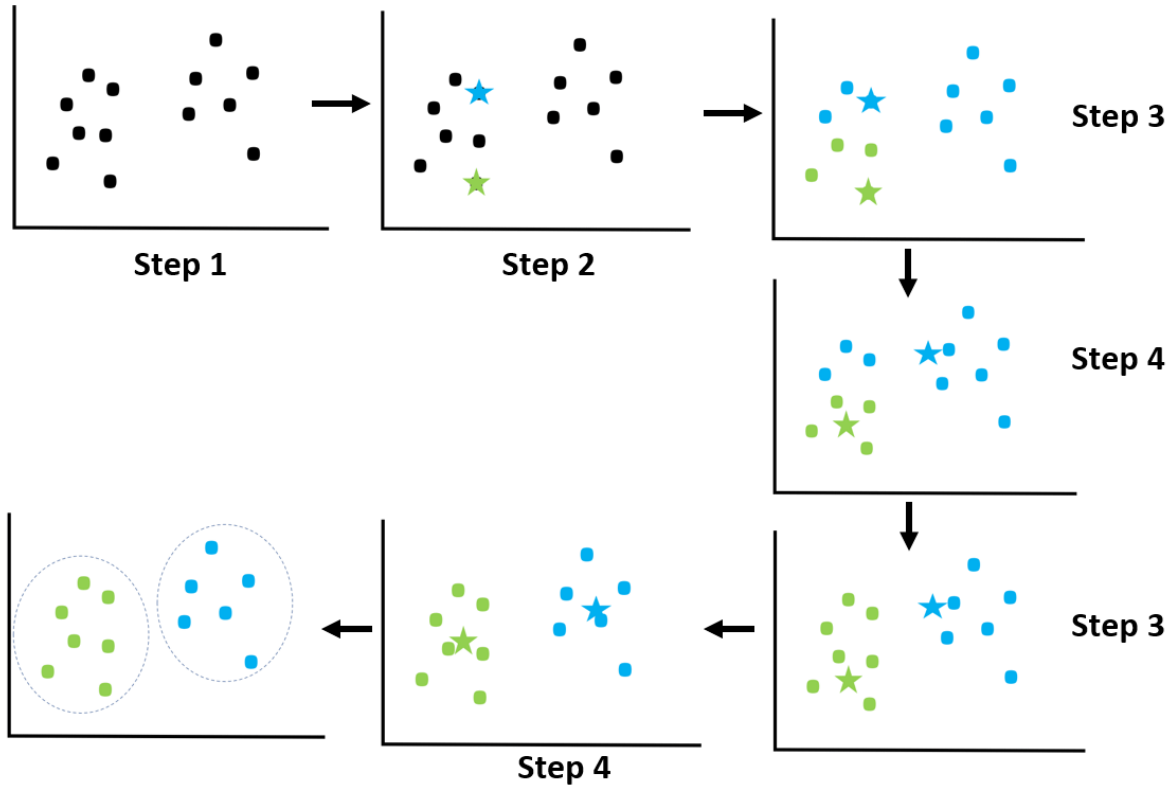


Convergence

Convergence: signifies that the centroids (cluster centers) no longer significantly change between iterations, indicating the clustering has reached a (local) optimum.

True Convergence: The algorithm might have genuinely converged on an optimal (or at least good) clustering solution. Stopping here saves computational resources.

- There's often a trade-off between achieving the absolute best and computational efficiency.
- If the cost doesn't budge in an iteration, it suggests convergence (or a "good enough" state).



Random Initialization

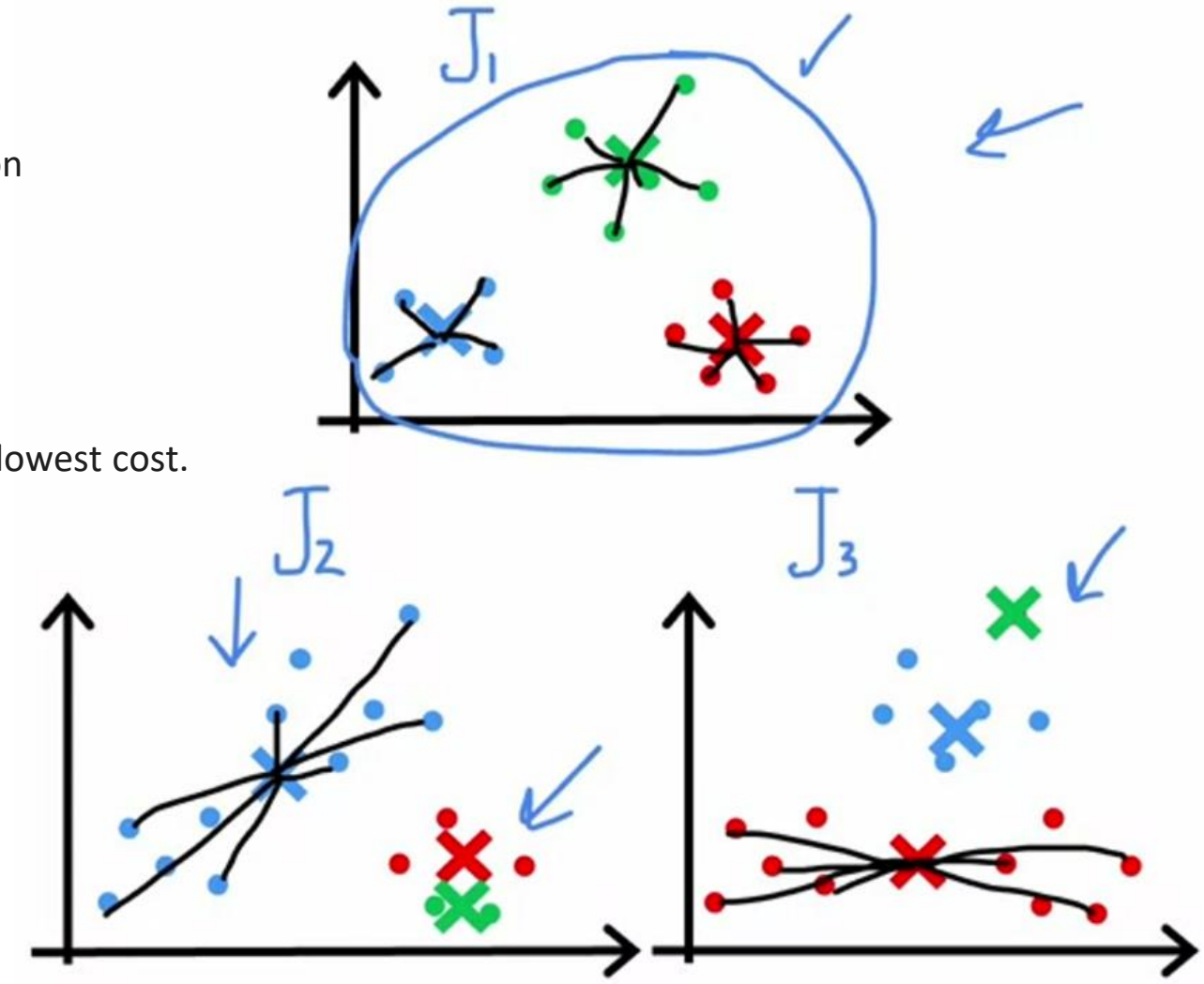
Problem:

- K-means converge to different clusters depending on initialization.

Solution:

- Run K-means multiple times with different random initializations. Then, choose the clustering with the lowest cost.

$$J = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$



Notebooks

Let's practice on collab:

Tutorial:

Advanced Machine Learning/ 3- Unsupervised Learning Clustering/LAB/K-means.ipynb

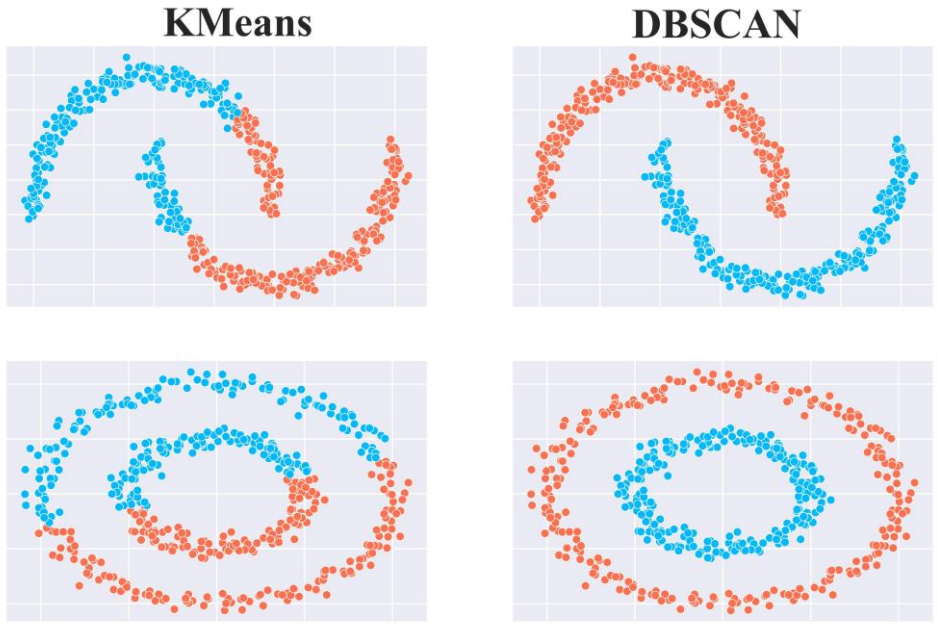
Exercise:

Advanced Machine Learning/ 3- Unsupervised Learning Clustering/LAB/K-means_Exercise.ipynb

Density-Based Spatial Clustering Of Applications With Noise (DBSCAN)

DBSCAN

Clusters are dense areas in the data space, separated by regions with fewer points. The DBSCAN algorithm uses this idea of "clusters" and "noise." It works by requiring that each point in a cluster must have at least a minimum number of points within a given radius around it.

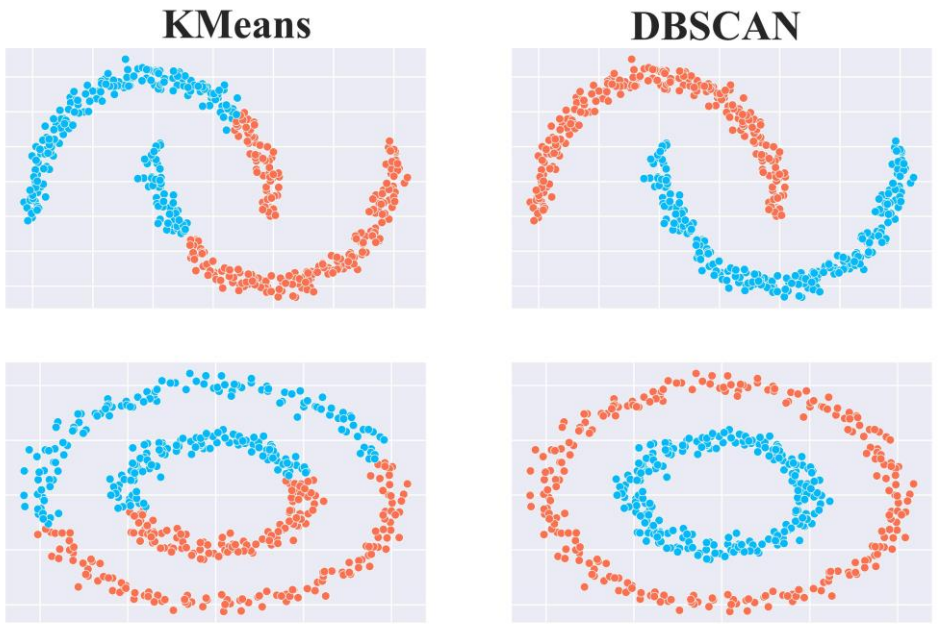


Source

DBSCAN

Real-life data may have irregularities like:

- Clusters of any shape, not just regular forms.
- Noise in the data.



Source

DBSCAN Algorithm

Parameters Required For DBSCAN Algorithm:

- **eps:** This defines the neighborhood around a data point. If the distance between two points is less than or equal to eps, they are considered neighbors. If eps is too small, many data points may be labeled as outliers. If eps is too large, clusters may merge, and most data points will end up in the same cluster. To find a good eps value, you can use a k-distance graph.

- **MinPts:** This is the minimum number of neighbors (data points) required within the eps radius. For larger datasets, you should use a larger MinPts value. A general rule is to set MinPts to at least the number of dimensions plus one (MinPts ≥ D+1). The minimum value of MinPts should be at least 3.

Data Points in DBSCAN:

- **Core Point:** A point that has more than MinPts points within the eps radius.

- **Border Point:** A point that has fewer than MinPts points within the eps radius but is in the neighborhood of a core point.

- **Noise or Outlier:** A point that is neither a core point nor a border point.

Steps Used In DBSCAN Algorithm

1. Find all the neighboring points within the eps radius and identify core points, which have more than MinPts neighbors.

2. For each core point that is not already assigned to a cluster, create a new cluster.

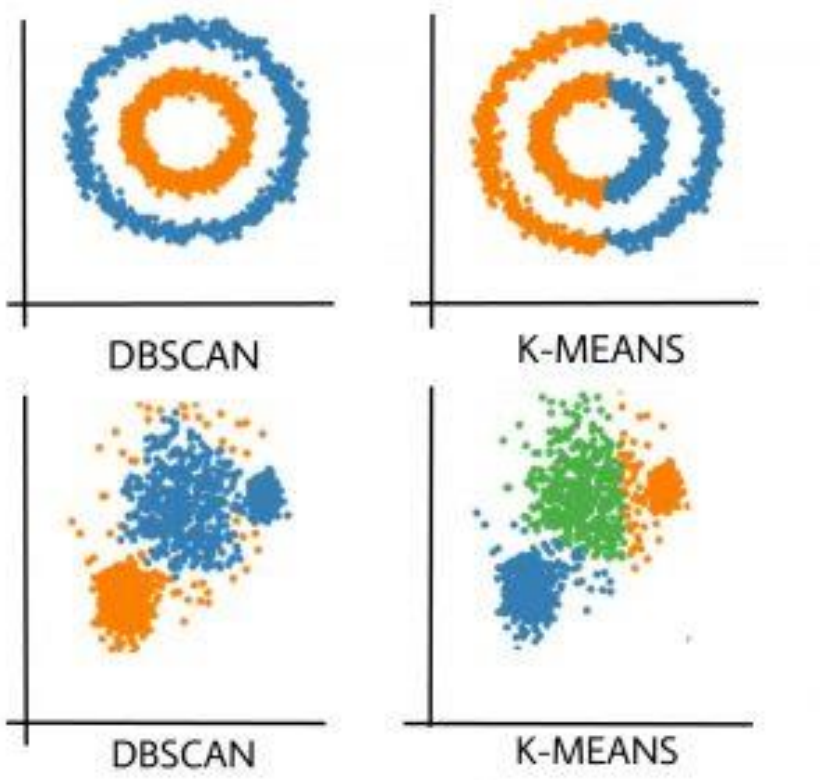
3. Recursively find all density-connected points for each core point and assign them to the same cluster. Points a and b are density-connected if there is a point c with enough neighbors where both a and b are within eps distance from c. This forms a chaining process: if b is a neighbor of c, c is a neighbor of d, and d is a neighbor of e, which is also a neighbor of a, then b is a neighbor of a.

4. After processing all core points, check the remaining unvisited points. Those that are not part of any cluster are considered noise.

DBSCAN vs K-Means

DBSCAN	K-Means
In DBSCAN, you do not need to specify the number of clusters.	K-Means is very sensitive to the number of clusters, so it must be specified in advance.
Clusters formed in DBSCAN can be of any arbitrary shape.	Clusters formed in K-Means are spherical or convex in shape.
DBSCAN can handle datasets with noise and outliers effectively.	K-Means does not work well with outliers. Outliers can significantly skew the clusters.
In DBSCAN, two parameters are required	In K-Means, only one parameter is required for training the model: the number of clusters (k).

DBSCAN vs K-Means



Source

Notebooks

Let's practice on collab:

Tutorial:

Advanced Machine Learning/ 3- Unsupervised Learning Clustering/LAB/DBSCAN.ipynb

Exercise:

Advanced Machine Learning/ 3- Unsupervised Learning Clustering/LAB/DBSCAN_Exercise.ipynb

Hierarchical Clustering

What is Hierarchical Clustering?

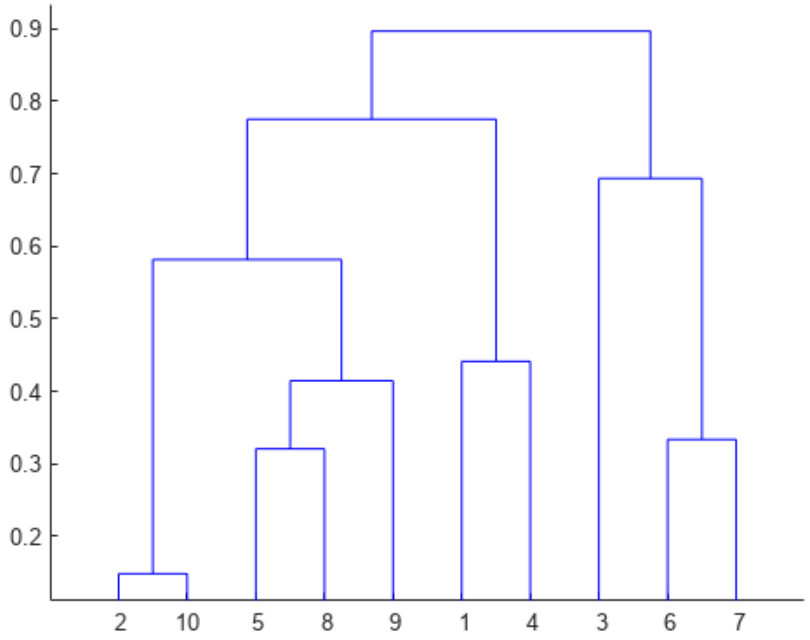
Hierarchical clustering in **Machine Learning** is a connectivity-based clustering method that groups data points that are close to each other based on similarity or distance. It assumes that data points that are nearer to each other are more similar or related than those that are farther apart.

What is Dendrogram?

- A dendrogram is a tree-like diagram created by hierarchical clustering that shows the hierarchical relationships between groups. Individual data points are positioned at the bottom, and the largest clusters, which encompass all the data points, are at the top. To obtain different numbers of clusters, you can cut the dendrogram at various heights.

- The dendrogram is created by iteratively merging or splitting clusters based on a measure of similarity or distance between data points. Clusters are divided or merged repeatedly until all data points are contained within a single cluster, or until the predetermined number of clusters is attained.

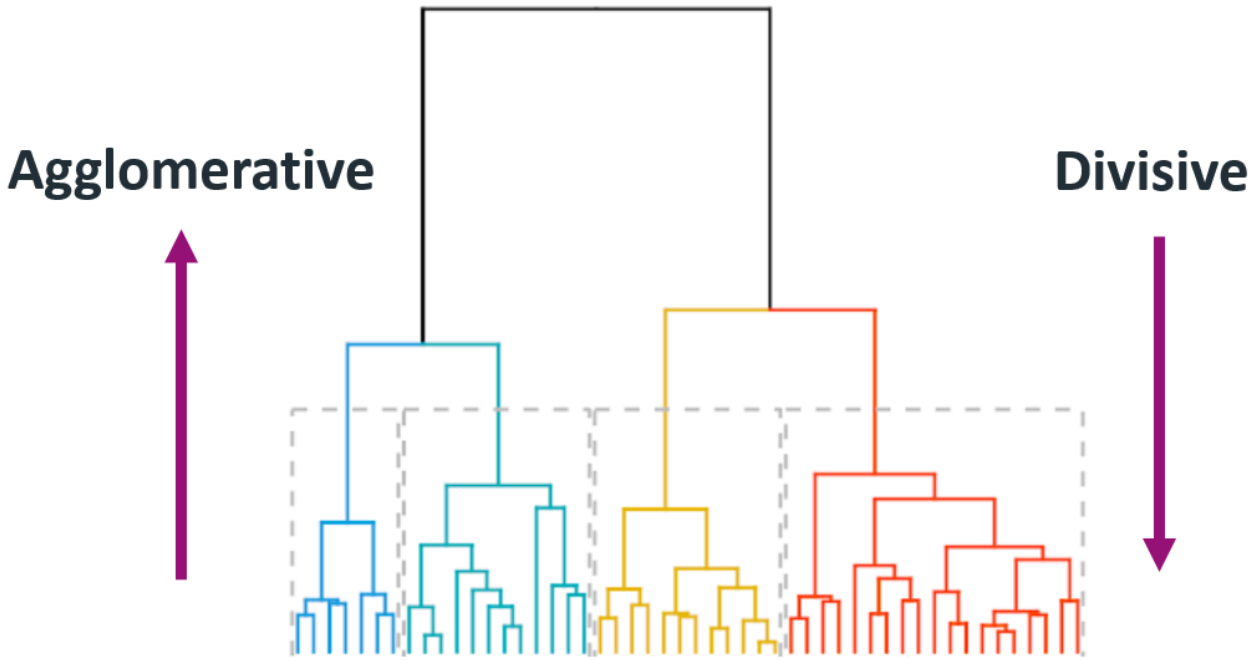
Dendogram



Types of Hierarchical Clustering

Imagine a hierarchy like a family tree categorized into:

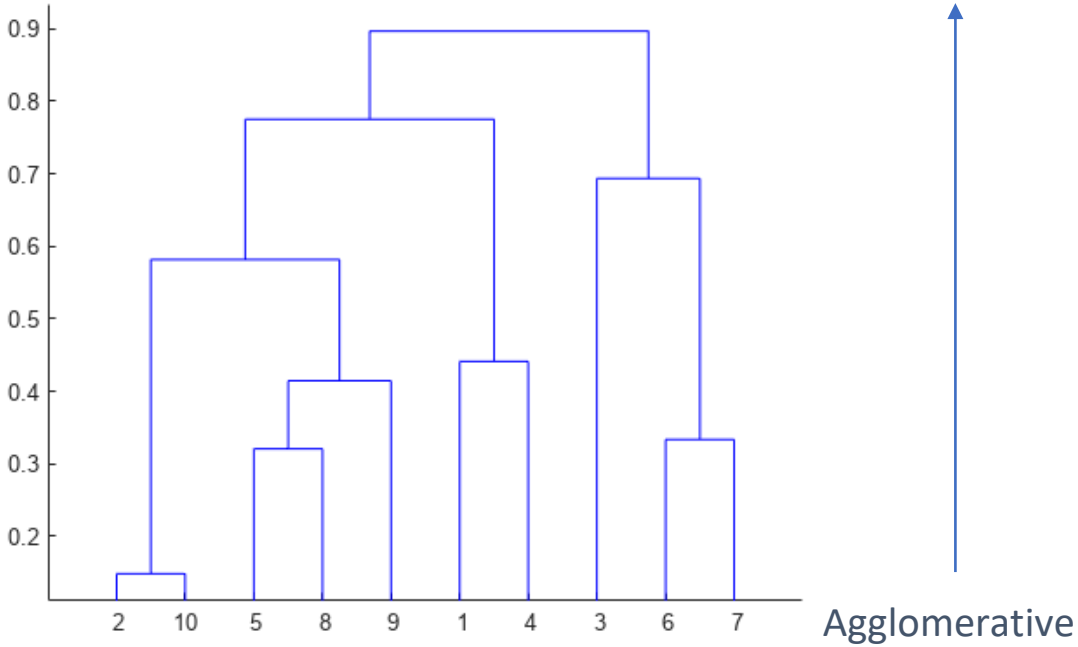
- **Agglomerative:** Starts with individual points and iteratively merges the closest ones, forming a hierarchy of clusters.
- **Divisive:** Starts with all data points in one cluster and iteratively splits them based on decreasing similarity, resulting in a tree-like structure



Agglomerative Clustering

This method is also referred to as the bottom-up approach or hierarchical agglomerative clustering (HAC). It provides a more detailed structure compared to the unstructured clusters produced by flat clustering.

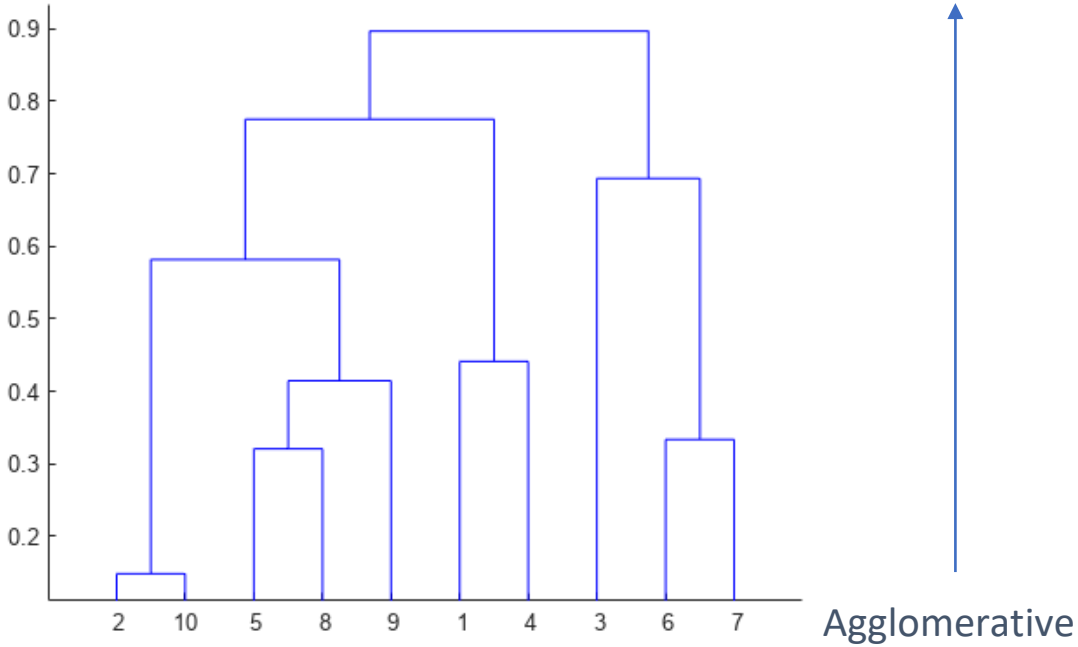
Dendogram



Agglomerative Clustering Cont.

Unlike other methods, this algorithm does not require us to specify the number of clusters in advance. Bottom-up algorithms start with each data point as its own singleton cluster and then progressively merge pairs of clusters until all clusters are combined into a single cluster containing all the data.

Dendogram



Agglomerative Clustering Graphical Representation

1. Start by treating each letter as an individual cluster and calculate the distance between each cluster and all other clusters.

2. Merge the most similar clusters. For example, if clusters (B) and (C) are very similar, merge them, and do the same for clusters (D) and (E), resulting in clusters [(A), (BC), (DE), (F)].

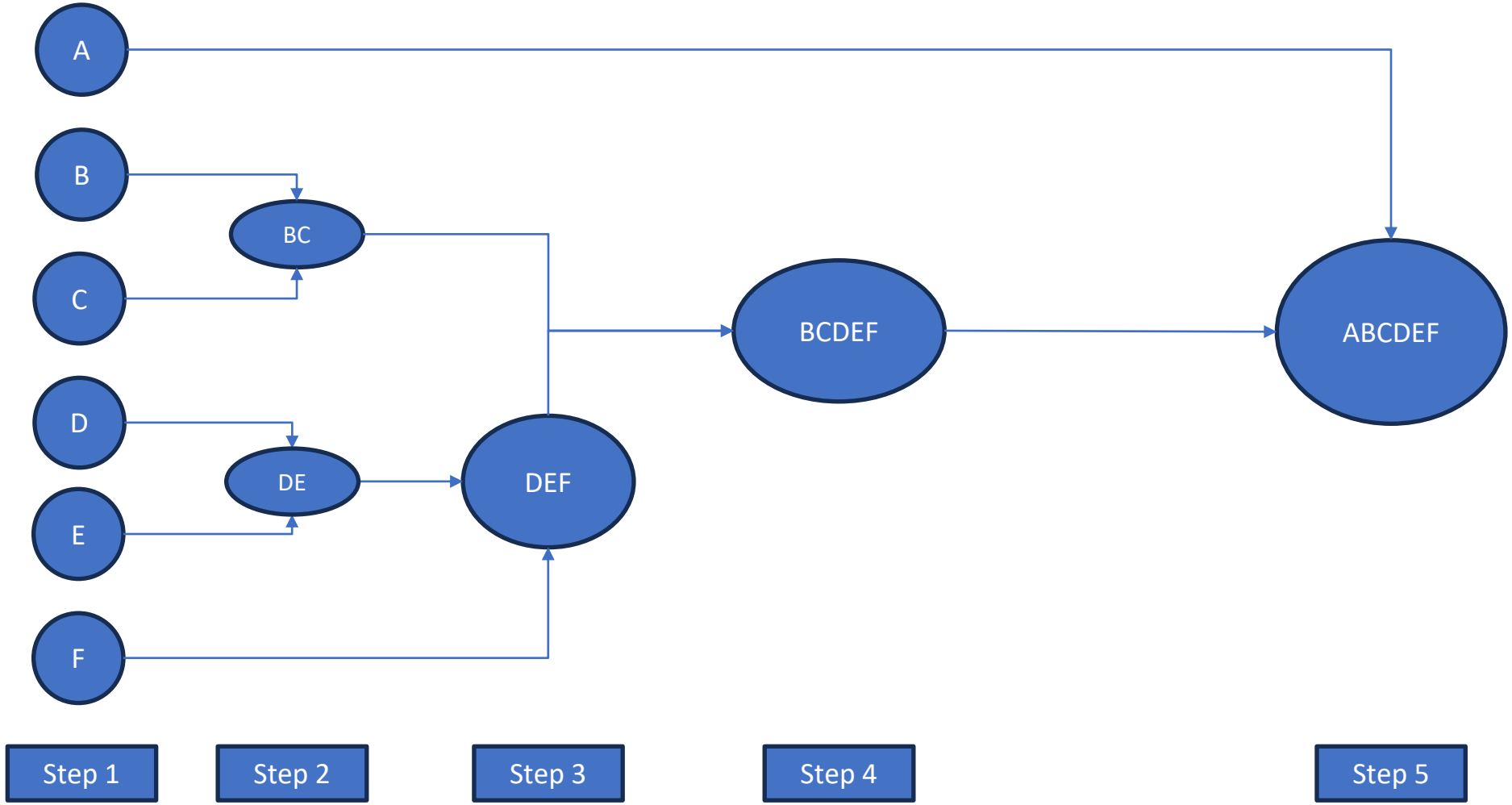
3. Recalculate the proximity of the updated clusters and merge the closest ones, such as clusters [(DE) and (F)], to form new clusters [(A), (BC), (DEF)].

4. Continue this process; merge clusters DEF and BC to create a new cluster, resulting in [(A), (BCDEF)].

5. Finally, merge the remaining clusters into one single cluster [(ABCDEF)].

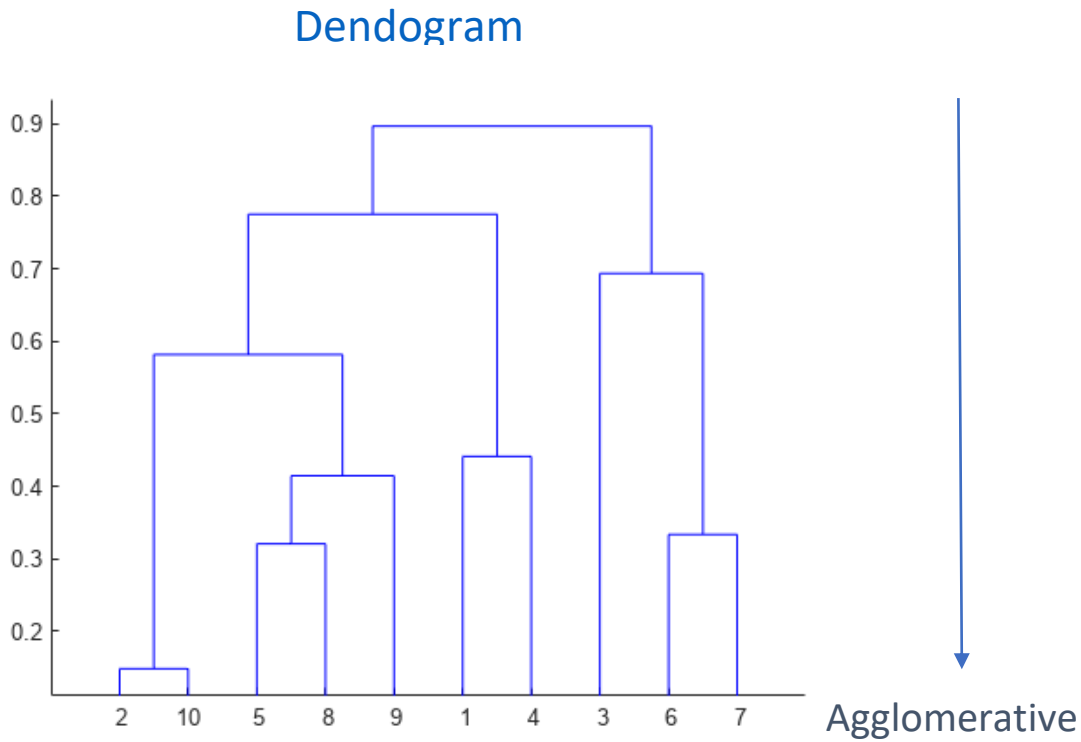
Agglomerative Clustering Graphical Representation

Let’s assume we have 6 data points A, B, C, D, E, and F.



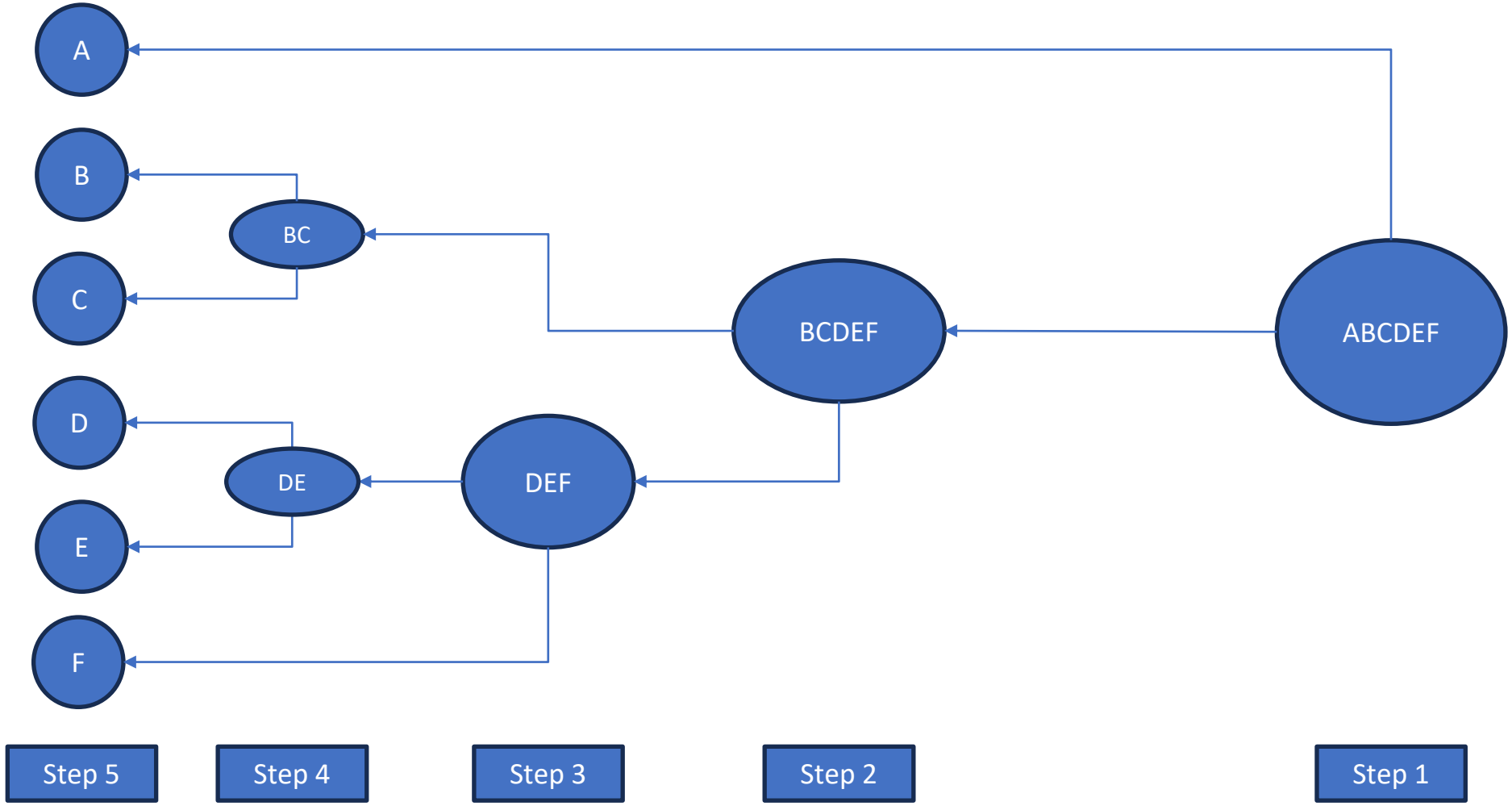
Divisive Hierarchical Clustering

This method is also called the top-down approach. Like hierarchical agglomerative clustering, it does not require specifying the number of clusters in advance. Top-down clustering starts with a single cluster containing all the data and recursively splits this cluster into smaller clusters until each data point is isolated into its own singleton cluster.



Divisive Hierarchical Clustering Graphical Representation

Let’s assume we have 6 data points A, B, C, D, E, and F.



Computing Distance Matrix

When merging two clusters, the distance between them can be determined in various ways. Here are some methods:

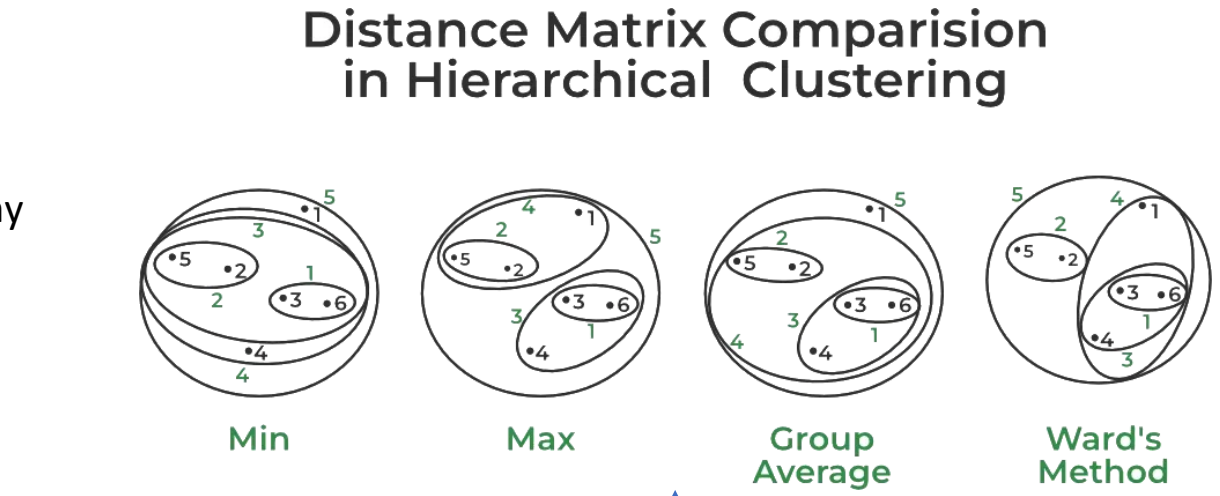
1. **Min Distance:** Calculate the minimum distance between any two points from the different clusters.

2. **Max Distance:** Determine the maximum distance between any two points from the different clusters.

3. **Group Average:** Compute the average distance between all pairs of points from the two clusters.

4. **Ward's Method:** Measure the similarity based on the increase in squared error when merging the two clusters.

For example, using different methods to group data can lead to different results:



Models Summary

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters, inductive	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry, inductive	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry, inductive	Distances between points
Spectral clustering	number of clusters	Medium `n samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry, transductive	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints, transductive	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances, transductive	Any pairwise distance
DBSCAN	neighborhood size	Very large `n samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes, outlier removal, transductive	Distances between nearest points
HDBSCAN	minimum cluster membership, minimum point neighbors	large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes, outlier removal, transductive, hierarchical, variable cluster density	Distances between nearest points
OPTICS	minimum cluster membership	Very large `n samples`, large `n_clusters`	Non-flat geometry, uneven cluster sizes, variable cluster density, outlier removal, transductive	Distances between points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation, inductive	Mahalanobis distances to centers
BIRCH	branching factor, threshold, optional global clusterer.	Large `n clusters` and `n_samples`	Large dataset, outlier removal, data reduction, inductive	Euclidean distance between points

https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

Notebooks

Let's practice on collab:

Tutorial:

Advanced Machine Learning/ 3- Unsupervised Learning Clustering/LAB/Hierarchical_Clustering.ipynb

Exercise:

Advanced Machine Learning/ 3- Unsupervised Learning Clustering/LAB/Hierarchical_Clustering_Exercise.ipynb

Thank you!

SDAIA
الهيئة السعودية للبيانات والذكاء الاصطناعي
Saudi Data & AI Authority