**Clustering in Machine Learning**

Clustering or cluster analysis is a machine learning technique, which groups the unlabelled dataset. It can be defined as ***"A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."***

It does it by finding some similar patterns in the unlabelled dataset such as shape, size, color, behavior, etc., and divides them as per the presence and absence of those similar patterns.

It is an unsupervised learning method, hence no supervision is provided to the algorithm, and it deals with the unlabeled dataset.

After applying this clustering technique, each cluster or group is provided with a cluster-ID. ML system can use this id to simplify the processing of large and complex datasets.

The clustering technique is commonly used for **statistical data analysis.**

#### Note: Clustering is somewhere similar to the [classification algorithm](https://www.javatpoint.com/classification-algorithm-in-machine-learning), but the difference is the type of dataset that we are using. In classification, we work with the labeled data set, whereas in clustering, we work with the unlabelled dataset.

**Example**: Let's understand the clustering technique with the real-world example of Mall: When we visit any shopping mall, we can observe that the things with similar usage are grouped together. Such as the t-shirts are grouped in one section, and trousers are at other sections, similarly, at vegetable sections, apples, bananas, Mangoes, etc., are grouped in separate sections, so that we can easily find out the things. The clustering technique also works in the same way.

Here are 10 simple points about clustering in Machine Learning:

1. Unsupervised Learning: Clustering is a type of unsupervised learning, meaning there are no labeled data points, and the goal is to find natural groupings in the data.

2. Grouping Data: The main aim of clustering is to group similar data points together based on their features.

3. Types of Clustering: Popular clustering methods include:

- K-Means : Divides data into a predefined number of clusters.

- Hierarchical Clustering: Builds a hierarchy of clusters.

- DBSCAN: Groups points that are close to each other based on density.

4. Distance Measures: Clustering algorithms often use distance measures (like Euclidean distance) to find how similar or different the data points are.

5. No Labeled Output : Since clustering is unsupervised, there's no right or wrong answer; it’s about discovering patterns.

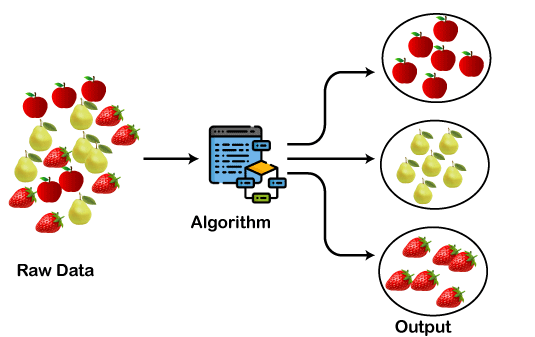
6. Applications: Clustering is used in customer segmentation, document classification, image compression, and more.

7. Centroids in K-Means: K-Means clustering uses "centroids," which represent the center of each cluster.

8. Hierarchical Approach: In hierarchical clustering, data is split into clusters either from the bottom up (agglomerative) or the top down (divisive).

9. Scalability: K-Means is highly scalable and can handle large datasets, while other methods like hierarchical clustering may not be as scalable.

10. Evaluation: Clustering can be evaluated using metrics like the Silhouette Score or visual methods like Elbow Method for K-Means to determine the optimal number of clusters.



**Soft and hard clustering differences**

1. Definition:

- Hard Clustering: Each data point belongs exclusively to one cluster. There is no overlap between clusters.

- Soft Clustering: A data point can belong to multiple clusters, with each point assigned a probability or degree of belonging to each cluster.

2. Assignment:

- Hard Clustering: Data points are assigned to a single cluster with certainty. Example: K-Means.

- Soft Clustering: Data points are assigned to different clusters with varying degrees of membership. Example: Fuzzy C-Means.

3. Cluster Boundaries:

- Hard Clustering: The boundaries between clusters are strict and clear-cut.

- Soft Clustering: The boundaries between clusters are flexible, allowing data points to be part of multiple clusters.

4. Uncertainty:

- Hard Clustering: No uncertainty; each point is definitively assigned to one cluster.

- Soft Clustering: Accounts for uncertainty, with each point having a membership value for different clusters.

5. Membership Representation:

- Hard Clustering: Membership is binary (0 or 1), where a point either belongs to a cluster or not.

- Soft Clustering: Membership is represented as a probability or fractional degree (ranging from 0 to 1) for each cluster.

6. Examples:

- Hard Clustering: K-Means, DBSCAN, Hierarchical Clustering.

- Soft Clustering: Fuzzy C-Means, Gaussian Mixture Models (GMM).

7. Use Cases:

- Hard Clustering: Suitable when strict and distinct groupings are needed.

- Soft Clustering: Useful when data points may naturally belong to multiple clusters or when clusters overlap.

8. Complexity:

- Hard Clustering: Simpler and faster, since each point is assigned to only one cluster.

- Soft Clustering: More computationally complex as it requires calculating the membership values for multiple clusters.

9. Interpretability:

- Hard Clustering: Easier to interpret because of the definitive cluster assignment.

- Soft Clustering: More nuanced interpretation, as points can belong to multiple clusters with varying degrees.

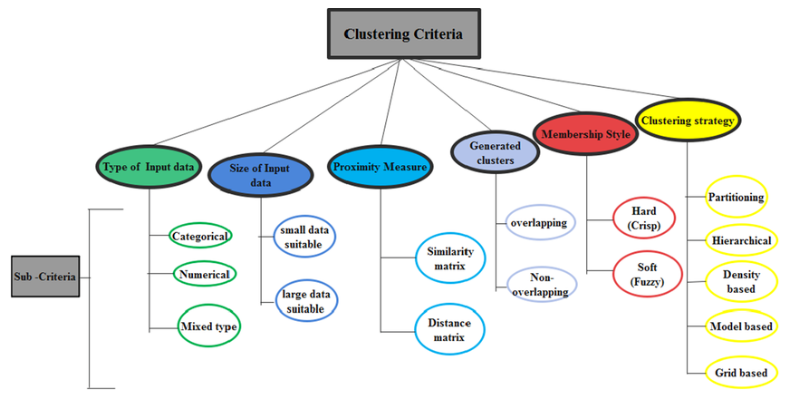
10. Real-World Applications:

- Hard Clustering: Document classification, customer segmentation.

-Soft Clustering: Image segmentation, bioinformatics (genes belonging to multiple functional groups).

Both methods are useful depending on the nature of the data and the desired outcome.

**Clustering Criteria**



## **Types of Clustering Methods**

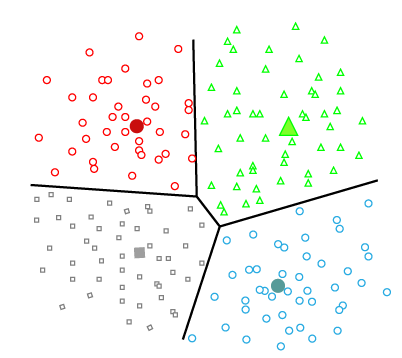
The clustering methods are broadly divided into **Hard clustering** (datapoint belongs to only one group) and **Soft Clustering** (data points can belong to another group also). But there are also other various approaches of Clustering exist.

1. **Partitioning Clustering**
2. **Density-Based Clustering**
3. **Distribution Model-Based Clustering**
4. **Hierarchical Clustering**
5. **Fuzzy Clustering**

### **Partitioning Clustering**

It is a type of clustering that divides the data into non-hierarchical groups. It is also known as the **centroid-based method**. The most common example of partitioning clustering is the [**K-Means Clustering algorithm**](https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning).

In this type, the dataset is divided into a set of k groups, where K is used to define the number of pre-defined groups. The cluster center is created in such a way that the distance between the data points of one cluster is minimum as compared to another cluster centroid.



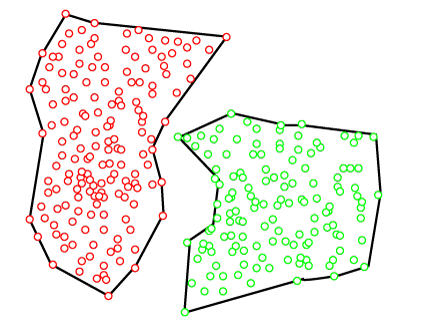
1. **Definition**: Partitioning clustering is a method where the dataset is divided into a fixed number of clusters, with each data point belonging to exactly one cluster.
2. **Predefined Clusters**: The user specifies the number of clusters (k) beforehand, and the algorithm assigns data points into these clusters.
3. **Centroid-Based**: Most partitioning methods, like **K-Means**, assign data points to clusters based on the distance to a central point (centroid).
4. **Iterative Process**: Partitioning clustering algorithms iteratively update the assignment of points to clusters to improve clustering quality.
5. **K-Means**: The most common partitioning method, K-Means, groups data points into k clusters by minimizing the distance between points and their cluster centroids.
6. **Distance Measure**: These algorithms typically use distance metrics like **Euclidean distance** to measure the similarity between data points and cluster centroids.
7. **Non-Hierarchical**: Unlike hierarchical clustering, partitioning methods do not create nested clusters but directly divide the dataset into disjoint clusters.
8. **Fixed Membership**: In partitioning clustering, each data point belongs exclusively to one cluster (hard clustering), although soft partitioning methods also exist.
9. **Efficiency**: Partitioning clustering is relatively fast and works well with large datasets, making it scalable.
10. **Limitations**: The number of clusters (k) must be chosen before running the algorithm, and the results can be sensitive to the choice of initial cluster centers.

Partitioning clustering is widely used for tasks like customer segmentation, image compression, and market research.

**Density-Based Clustering**

The density-based clustering method connects the highly-dense areas into clusters, and the arbitrarily shaped distributions are formed as long as the dense region can be connected. This algorithm does it by identifying different clusters in the dataset and connects the areas of high densities into clusters. The dense areas in data space are divided from each other by sparser areas.

These algorithms can face difficulty in clustering the data points if the dataset has varying densities and high dimensions.



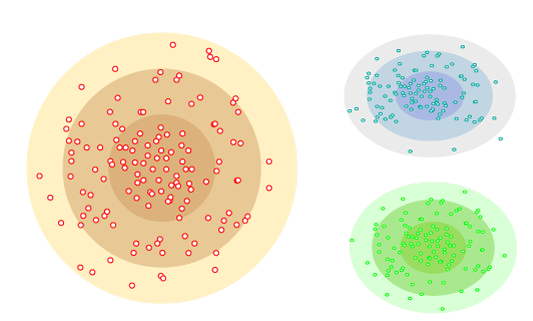
1. **Definition**: Density-based clustering groups data points that are closely packed together (dense regions) and separates points in sparse regions as noise or outliers.
2. **No Predefined Clusters**: Unlike K-Means, the number of clusters is not predefined. Clusters form naturally based on the density of the data points.
3. **DBSCAN**: **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) is the most popular density-based clustering algorithm.
4. **Core, Border, and Noise Points**:
   * **Core Points**: Points in dense regions, with many neighbors within a specified distance.
   * **Border Points**: Points near a core point but not dense enough to be a core point.
   * **Noise Points**: Points in sparse regions, considered as outliers.
5. **Parameters**: DBSCAN requires two main parameters:
   * **Epsilon (ε)**: The maximum distance between two points to be considered neighbors.
   * **MinPts**: The minimum number of neighboring points to form a dense region.
6. **Cluster Shape**: Density-based clustering can find clusters of arbitrary shape, unlike K-Means which forms spherical clusters.
7. **Handles Outliers**: One of the strengths of density-based clustering is its ability to handle noise and outliers effectively by marking points in low-density regions as outliers.
8. **Scalability**: Density-based algorithms can be computationally intensive and may not scale well for very large datasets.
9. **No Need for Centroids**: Unlike partitioning clustering, density-based clustering does not rely on centroids but instead focuses on the local density of points.
10. **Applications**: It is used in image processing, spatial data analysis, and identifying geographical clusters, where data points are naturally grouped together in dense regions.

Overall, density-based clustering is ideal for datasets with varying densities and clusters of irregular shapes, where traditional methods like K-Means might fail.

### **Distribution Model-Based Clustering**

In the distribution model-based clustering method, the data is divided based on the probability of how a dataset belongs to a particular distribution. The grouping is done by assuming some distributions commonly **Gaussian Distribution**.

The example of this type is the **Expectation-Maximization Clustering algorithm** that uses Gaussian Mixture Models (GMM).

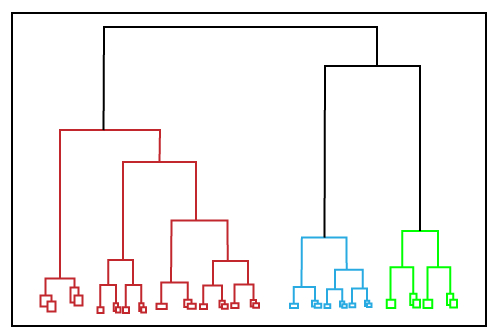


1. **Definition**: Distribution model-based clustering assumes that data points are generated by a mixture of underlying probability distributions, where each cluster corresponds to one distribution.
2. **Probabilistic Approach**: Each data point is assigned to a cluster based on the probability that it belongs to a specific distribution, usually using methods like **Expectation-Maximization (EM)**.
3. **Gaussian Mixture Models (GMM)**: The most popular model-based clustering algorithm, which assumes that clusters are Gaussian (normal) distributions with parameters like mean and covariance.
4. **Soft Clustering**: Unlike hard clustering methods, model-based clustering is often a form of **soft clustering**, where each data point can belong to multiple clusters with different probabilities.
5. **Cluster Shape Flexibility**: Distribution-based models can model clusters with different shapes, sizes, and orientations, making them more flexible than methods like K-Means.
6. **Parameter Estimation**: The algorithm estimates the parameters (like mean, variance) of the underlying distributions for each cluster, often using the EM algorithm.
7. **Generative Model**: These models treat the data as generated by a mixture of distributions, which helps in identifying how the data fits to the model's assumptions.
8. **Scalability**: Model-based clustering can handle moderately large datasets but can be slower than simpler methods like K-Means, especially if the number of clusters or data points is large.
9. **Number of Clusters**: Often, the number of clusters must be defined beforehand, though some techniques can automatically determine this based on model fitting.
10. **Applications**: Commonly used in areas like **image segmentation**, **bioinformatics**, and **financial data analysis**, where data follows an assumed distribution.

Model-based clustering works well when you expect the data to follow a known distribution and need a flexible approach for handling clusters with complex shapes and properties.

### **Hierarchical Clustering**

Hierarchical clustering can be used as an alternative for the partitioned clustering as there is no requirement of pre-specifying the number of clusters to be created. In this technique, the dataset is divided into clusters to create a tree-like structure, which is also called a **dendrogram**. The observations or any number of clusters can be selected by cutting the tree at the correct level. The most common example of this method is the **Agglomerative Hierarchical algorithm**.



1. **Definition**: Hierarchical clustering is a method that builds a hierarchy of clusters by either merging smaller clusters into larger ones (agglomerative) or splitting larger clusters into smaller ones (divisive).
2. **Agglomerative vs. Divisive**:
   * **Agglomerative**: Starts with each data point as its own cluster and successively merges the closest clusters.
   * **Divisive**: Starts with all data points in one cluster and successively splits it into smaller clusters.
3. **Dendrogram**: The output is often visualized using a **dendrogram**, a tree-like diagram that shows the sequence of cluster merges or splits.
4. **No Need to Predefine Clusters**: Unlike K-Means, you don’t need to specify the number of clusters in advance. You can decide based on the dendrogram or a distance threshold.
5. **Linkage Criteria**:
   * **Single Linkage**: Clusters are merged based on the shortest distance between points.
   * **Complete Linkage**: Merging happens based on the largest distance between points.
   * **Average Linkage**: Uses the average distance between points in different clusters.
6. **Distance Metrics**: Hierarchical clustering can use different distance metrics, such as **Euclidean distance** or **Manhattan distance**, to determine how similar or different data points are.
7. **Works for Arbitrary Shapes**: It can capture clusters of various shapes and sizes, as it doesn't assume clusters are spherical.
8. **Computational Complexity**: Hierarchical clustering is more computationally intensive than methods like K-Means, especially for large datasets, because of repeated distance calculations.
9. **No Reassignment**: Once a merge or split is made, data points cannot move between clusters. This can sometimes lead to suboptimal clusters.
10. **Applications**: Hierarchical clustering is used in **gene expression analysis**, **document clustering**, **social network analysis**, and any scenario where a hierarchical structure or relationships among clusters is important.

Hierarchical clustering provides flexibility in the clustering process and is particularly useful when you want a hierarchical representation of the data.

### **Fuzzy Clustering**

[Fuzzy](https://www.javatpoint.com/fuzzy-logic) clustering is a type of soft method in which a data object may belong to more than one group or cluster. Each dataset has a set of membership coefficients, which depend on the degree of membership to be in a cluster. **Fuzzy C-means algorithm** is the example of this type of clustering; it is sometimes also known as the Fuzzy k-means algorithm.

1. **Definition**: Fuzzy clustering (or soft clustering) allows each data point to belong to multiple clusters with varying degrees of membership, unlike traditional clustering where each point belongs to exactly one cluster.
2. **Fuzzy C-Means (FCM)**: The most common fuzzy clustering algorithm, which generalizes K-Means by assigning membership levels to data points for each cluster.
3. **Membership Values**: Instead of assigning a data point to a single cluster, fuzzy clustering assigns a membership value (between 0 and 1) to each cluster, indicating the degree to which a point belongs to that cluster.
4. **Centroid Calculation**: Similar to K-Means, fuzzy clustering calculates centroids, but the centroids are updated by considering the membership degrees of points in each cluster.
5. **Flexibility**: Fuzzy clustering is more flexible than hard clustering methods, as it allows points to belong partially to several clusters, which can reflect real-world situations better.
6. **Fuzziness Parameter**: A **fuzziness parameter (m)** controls how much overlap between clusters is allowed. A higher value leads to more overlap, while a lower value makes the clusters more distinct.
7. **Iterative Process**: Fuzzy clustering is an iterative algorithm that updates both cluster centroids and membership values until convergence.
8. **Use Cases**: Fuzzy clustering is useful in scenarios where boundaries between clusters are not clearly defined, such as **image segmentation**, **market segmentation**, and **pattern recognition**.
9. **Interpretability**: It provides a more nuanced interpretation of data, as you can understand how strongly or weakly a data point belongs to different clusters.
10. **Limitations**: Fuzzy clustering can be computationally expensive, especially for large datasets, and can be sensitive to the choice of parameters, like the number of clusters and fuzziness degree.

Fuzzy clustering is ideal when data points naturally belong to multiple groups or when uncertainty is inherent in cluster assignments.