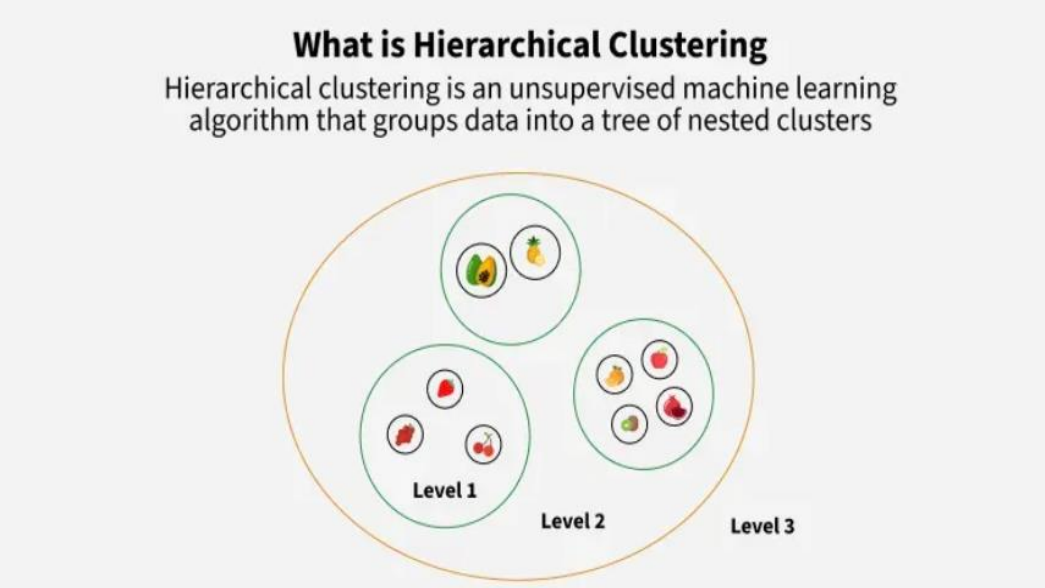
HIERARCHICAL CLUSTERING IN MACHINE LEARNING

- Hierarchical Clustering is an unsupervised learning technique that groups data into a **hierarchy of clusters** based on similarity.
- It builds a tree-like structure (dendrogram) that helps ***visualize relationships and decide the optimal number of clusters***.
- Does not require ***pre-selecting the number of clusters***.
- Uses **agglomerative or divisive approaches**.
- Commonly applied in data exploration and pattern discovery.
- It is commonly used in pattern recognition, customer segmentation and image grouping

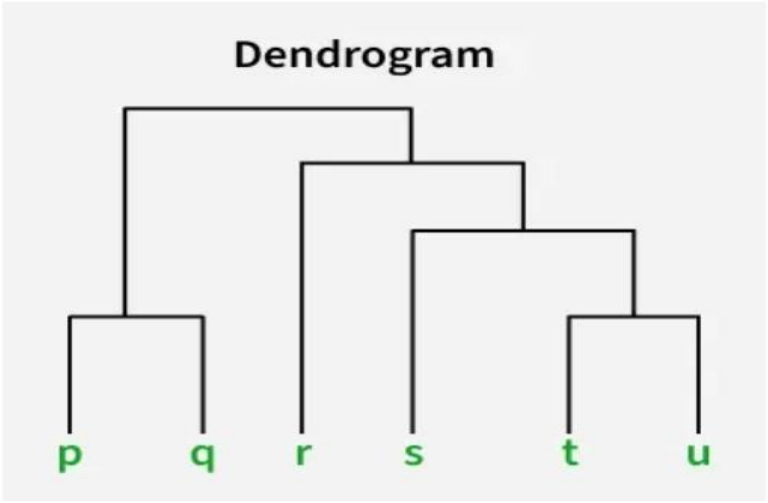


- Imagine we have four fruits with different weights: an apple (100g), a banana (120g), a cherry (50g) and a grape (30g).
- Hierarchical clustering starts by treating each fruit as its own group.
- Start with each fruit as its own cluster.
- Merge the closest items: grape (30g) and cherry (50g) are grouped first.
- Next, apple (100g) and banana (120g) are grouped.
- Finally, these two clusters merge into one.
- Finally all the fruits are merged into one large group, showing how hierarchical clustering progressively combines the most similar data points.

DENDROGRAM

- A dendrogram is like a family tree for clusters. It shows how individual data points or groups of data merge together.
- The bottom shows each data point as its **own group** and as we move up, similar groups are combined.

- The lower the merge point, the more similar the groups are. It helps us see how things are grouped step by step.

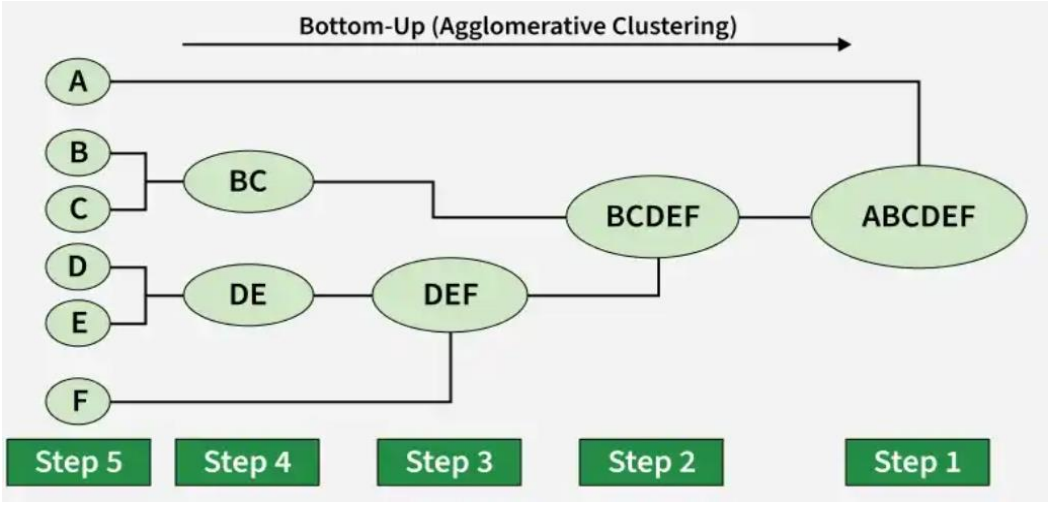


Types of Hierarchical Clustering

- There are two main types of hierarchical clustering.
- Agglomerative Clustering
- Divisive clustering

1. Hierarchical Agglomerative Clustering

- It is also known as the bottom-up approach or hierarchical agglomerative clustering (HAC).
- Bottom-up algorithms treat each data as a **singleton cluster** at the outset and then successively agglomerate pairs of clusters until all clusters have been merged into a single cluster that contains all data.



Workflow for Hierarchical Agglomerative clustering

- **Start with individual points:** Each data point is its own cluster. For example if we have 5 data points we start with 5 clusters each containing just one data point.
- **Calculate distances between clusters**: Calculate the distance between every pair of clusters. Initially since each cluster has one point this is the distance between the two data points.
- **Merge the closest clusters**: Identify the two clusters with the smallest distance and merge them into a single cluster.
- **Update distance matrix**: After merging we now have one less cluster. Recalculate the distances between the new cluster and the remaining clusters.
- **Repeat steps 3 and 4**: Keep merging the closest clusters and updating the distance matrix until we have only one cluster left.
- **Create a dendrogram**: As the process continues we can visualize the merging of clusters using a tree-like diagram called a dendrogram. It shows the hierarchy of how clusters are merged.

Proximity Matrix and Linkage

- The proximity matrix is a matrix consisting of the distance between each pair of data points. The distance is computed by a distance function.
- Euclidean distance is one of the most commonly used distance functions.

	x_1	x_2	x_3	...	x_n
x_1	$d(x_1,x_1)$	$d(x_1,x_2)$	$d(x_1,x_3)$	...	$d(x_1,x_n)$
x_2	$d(x_2,x_1)$	$d(x_2,x_2)$	$d(x_2,x_3)$	...	$d(x_2,x_n)$
x_3	$d(x_3,x_1)$	$d(x_3,x_2)$	$d(x_3,x_3)$	...	$d(x_3,x_n)$
...	...	...	...	...	...
x_n	$d(x_n,x_1)$	$d(x_n,x_2)$	$d(x_n,x_3)$		$d(x_n,x_n)$

- The above proximity matrix consists of ***n* points named *x***, and the ***d(xi,xj)*** represents the distance between the points.
- In order to group the data points in a cluster, a linkage function is used where the values in the proximity matrix are taken and the data points are grouped based on similarity. The newly formed clusters are linked to each other until they form a single cluster containing all the data points.
- The most common linkage methods are:
- **Complete linkage:** The maximum of all pairwise distance between elements in each pair of clusters is used to measure the distance between two clusters.

- **Single linkage:** The minimum of all pairwise distance between elements in each pair of clusters is used to measure the distance between two clusters.
- **Average linkage:** The average of all pairwise distances between elements in each pair of clusters is used to measure the distance between two clusters.
- **Centroid linkage:** Before merging, the distance between the two clusters' centroids are considered.
- **Ward's Method:** It uses squared error to compute the similarity of the two clusters for merging.

Benefits of Agglomerative Clustering

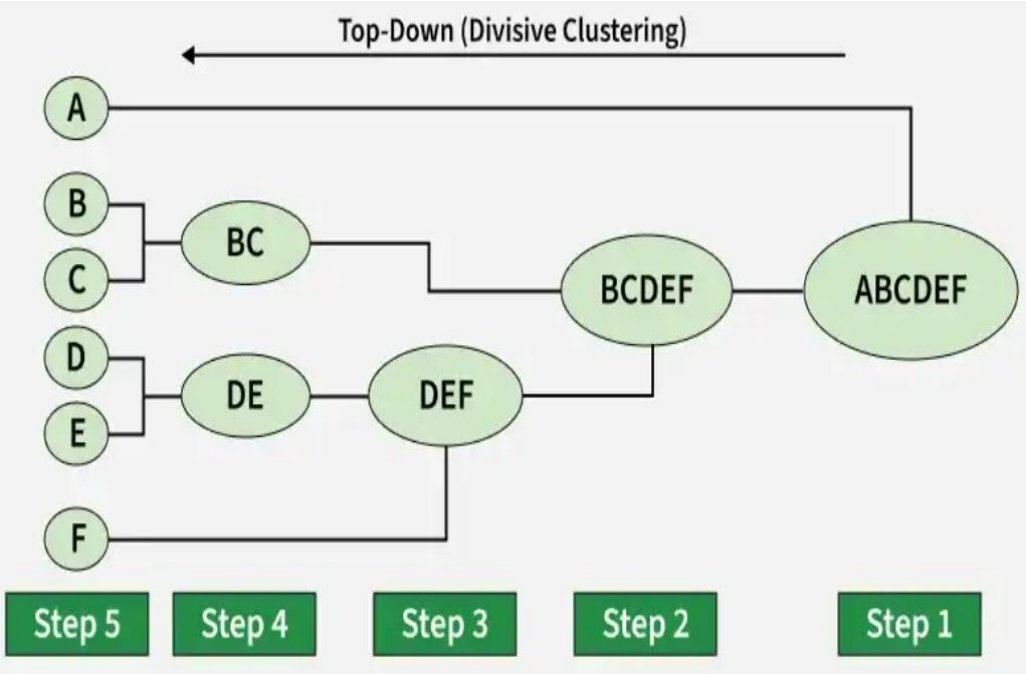
- It is widely applicable.
- Here, we form dendrogram, we know that which point is closest to which point through the hierarchy.

Limitations of Agglomerative Clustering

- It is not for big datasets because we have to make a proximity matrix and if we have large number of columns then we have to make n x n columns.

2. Hierarchical Divisive clustering

- Divisive clustering is also known as a top-down approach.
- Top-down clustering requires a method for splitting a cluster that contains the whole data and proceeds by splitting clusters recursively until individual data have been split into singleton clusters.



Workflow for Hierarchical Divisive clustering

- **Start with all data points in one cluster:** Treat the entire dataset as a single large cluster.
- **Split the cluster**: Divide the cluster into two smaller clusters. The division is typically done by finding the two most dissimilar points in the cluster and using them to separate the data into two parts.
- **Repeat the process**: For each of the new clusters, repeat the splitting process: Choose the cluster with the most dissimilar points and split it again into two smaller clusters.
- **Stop when each data point is in its own cluster**: Continue this process until every data point is its own cluster or the stopping condition (such as a predefined number of clusters) is met.

Use cases for hierarchical clustering

- Hierarchical clustering provides data scientists with insights into the structure and relationships within datasets and it can be applied in various use cases.
- **Business**
- Hierarchical clustering can help analyze trends and segment customer data—-for example, grouping by product choice, demographic, purchasing behavior, risk profile or interactions with social media.
- **Clinical research and bioinformatics**
- Patient cohorts for clinical research can run into the thousands. Hierarchical clustering helps categorize mixed populations into more homogeneous groups[20] by using, for example, diagnostic criteria, physiological responses or DNA mutations. It can also be applied to group species by biological features to understand evolutionary development.
- **Image and information processing**
- Hierarchical clustering is used in image-based text recognition applications to group handwritten characters by their shape. It is also used to store and retrieve information by using certain criteria or to categorize search results.