Hierarchical Clustering in Machine Learning

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as hierarchical cluster analysis or HCA. This method works via grouping data into a tree of clusters. Hierarchical clustering begins by treating every data point as a separate cluster. Then, it repeatedly executes the subsequent steps.

In Hierarchical Clustering, the aim is to produce a hierarchical series of nested clusters. A diagram called Dendrogram (A Dendrogram is a tree-like diagram that statistics the sequences of merges or splits) graphically represents this hierarchy and is an inverted tree that describes the order in which factors are merged (bottom-up view) or clusters are broken up (top-down view). In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram**.

The dendrogram from hierarchical clustering reveals the hierarchy of clusters at different levels, highlighting natural groupings in the data. It provides a visual representation of the relationships between clusters, helping to identify patterns and outliers, making it a useful tool for exploratory data analysis.

Sometimes the results of K-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the number of clusters as we did in the K-Means algorithm.

Why Hierarchical Clustering?

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.

This is a gap hierarchical clustering bridges with aplomb. It takes away the problem of having to pre-define the number of clusters.

Types of Hierarchical Clustering

There are mainly two types of hierarchical clustering:

- Agglomerative hierarchical clustering : Agglomerative is a **bottom-up approach**, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
- Divisive Hierarchical clustering : Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach**.

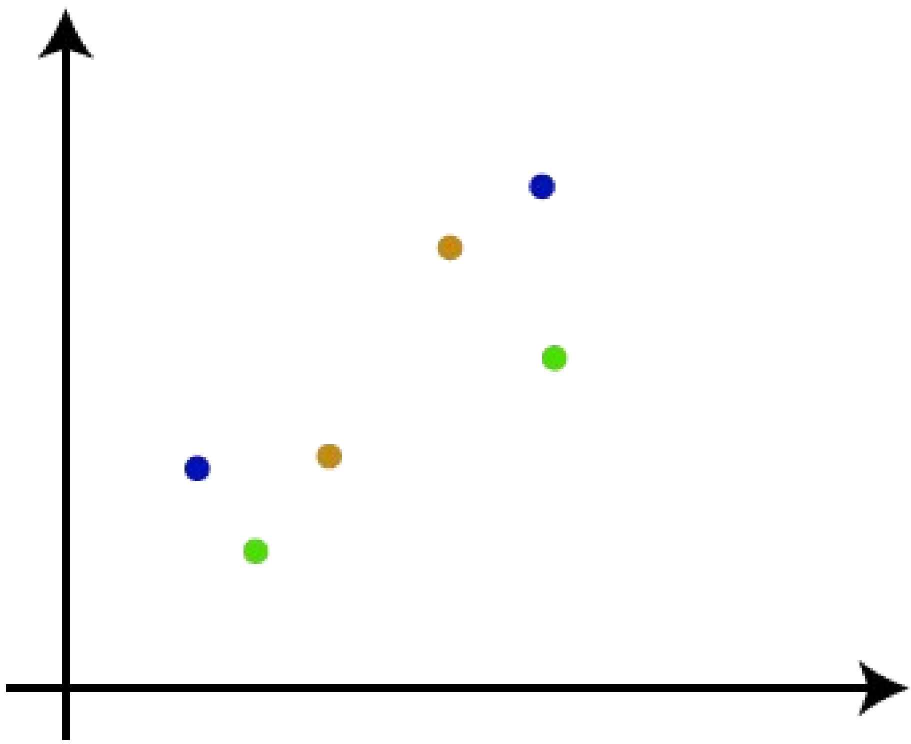
Agglomerative hierarchical clustering

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.

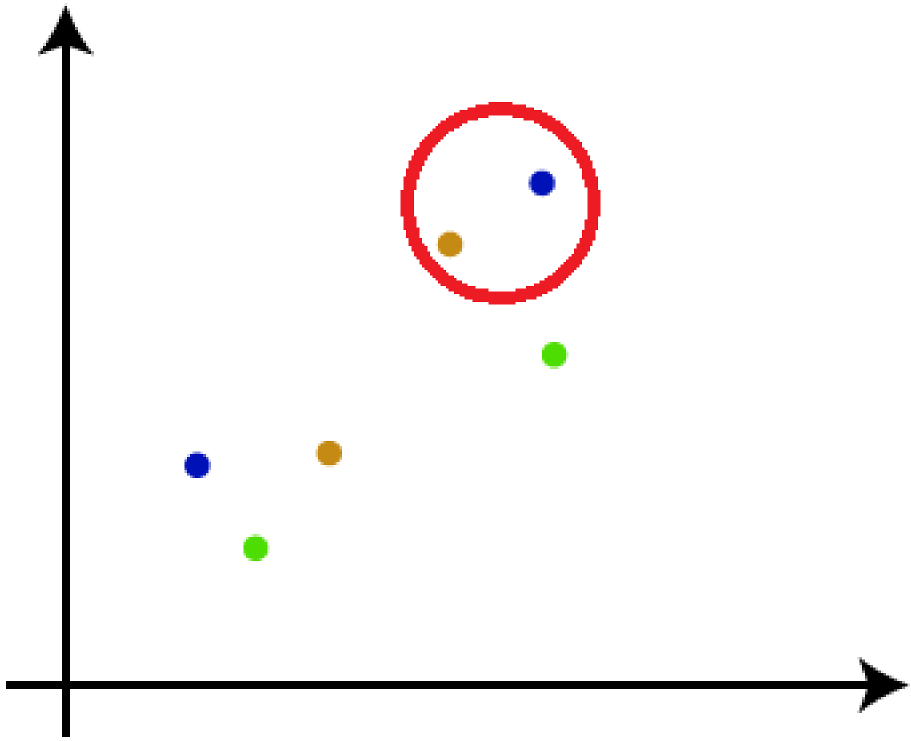
How the Agglomerative Hierarchical clustering Work?

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

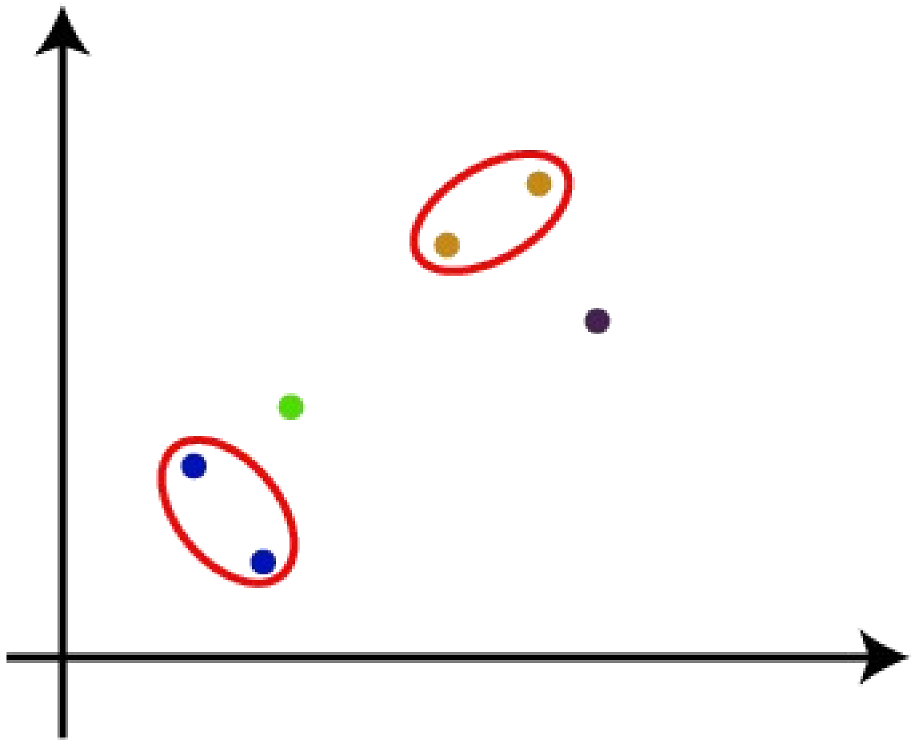
- Step-1: Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.



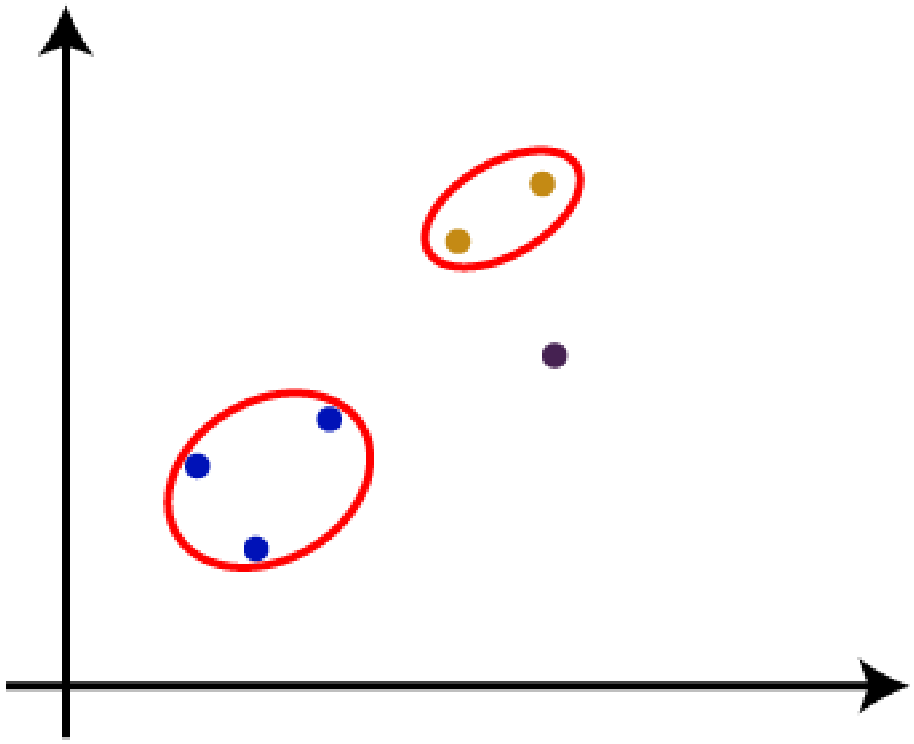
- Step-2: Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.

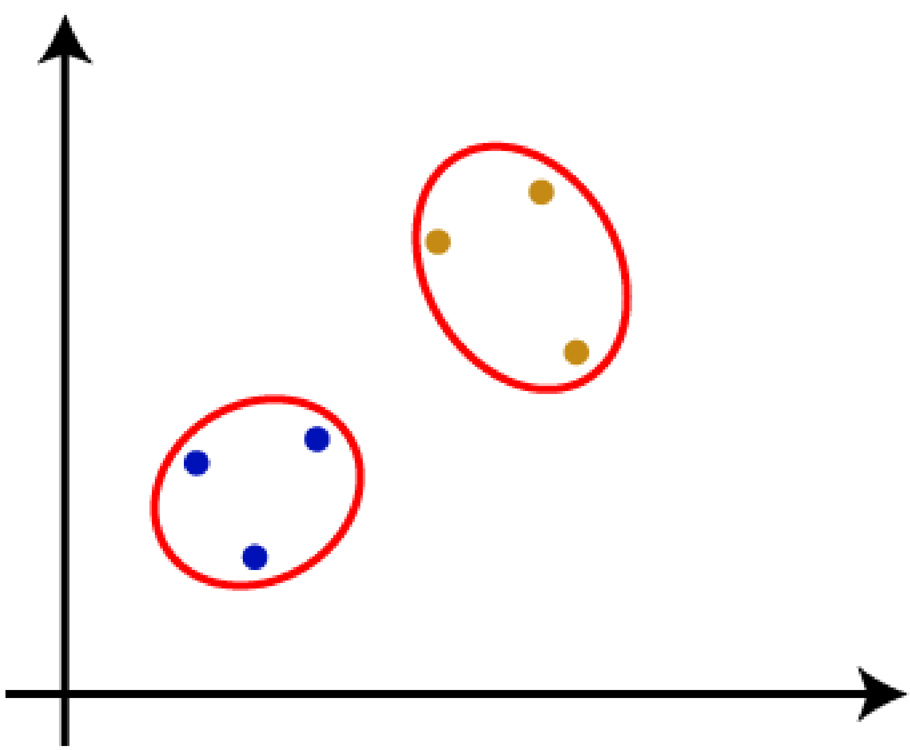


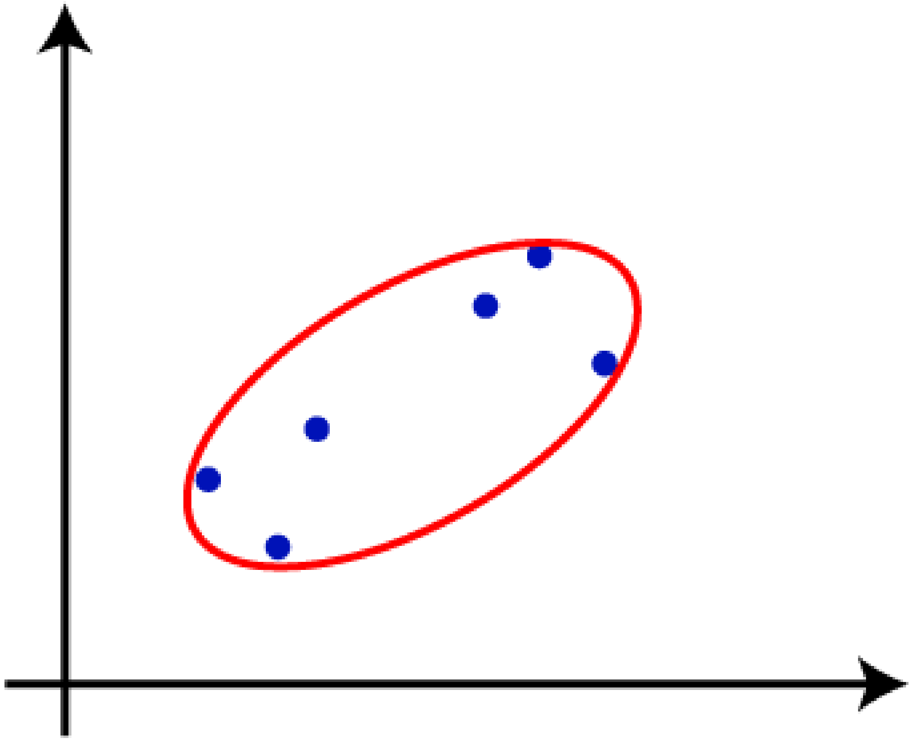
- Step-3: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



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

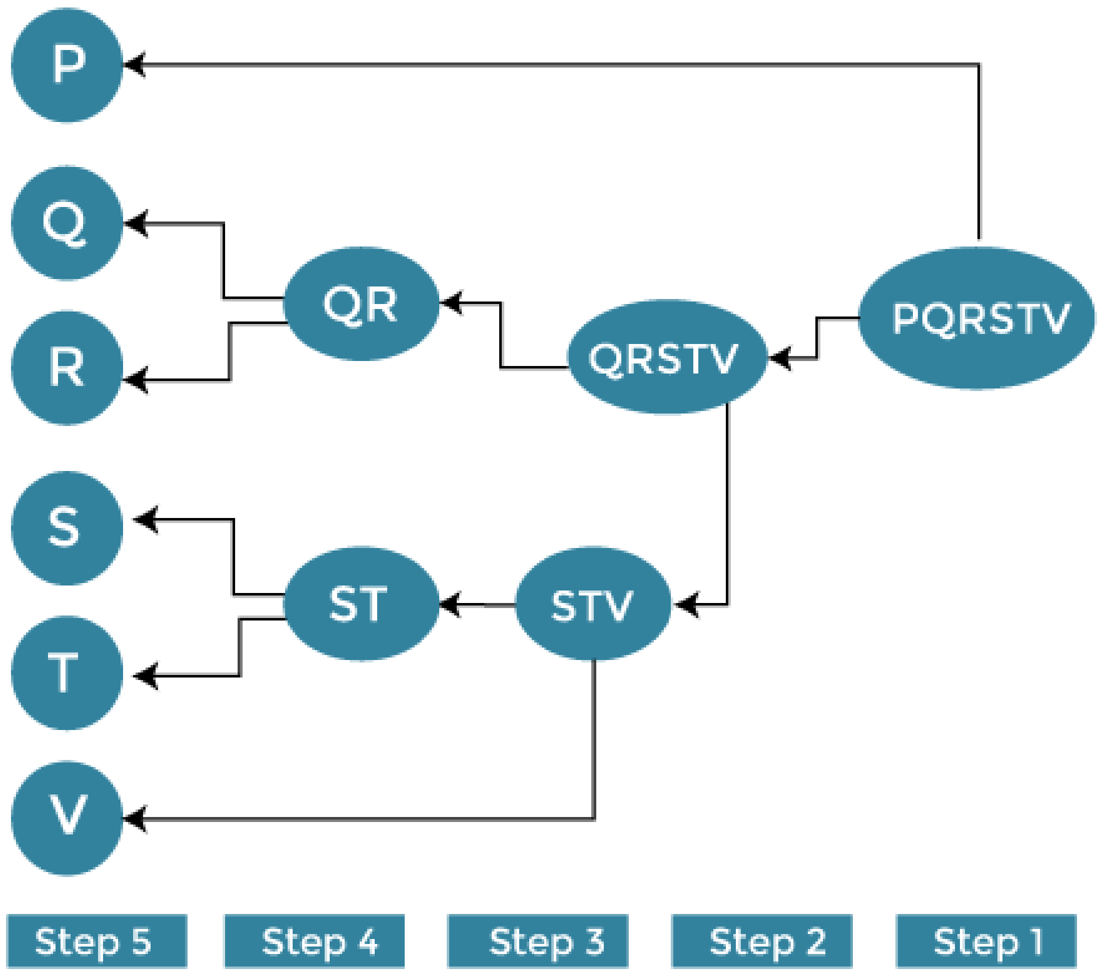






Divisive Hierarchical Clustering

Divisive hierarchical clustering is exactly the opposite of Agglomerative Hierarchical clustering. In Divisive Hierarchical clustering, all the data points are considered an individual cluster, and in every iteration, the data points that are not similar are separated from the cluster. The separated data points are treated as an individual cluster. Finally, we are left with N clusters.



- Step 1: Consider alphabet (PQRSTV) as an single cluster.
- Step 2: Choose a cluster to split based on some criterion (such as diameter, variance, etc.). Let's say cluster QRSTV and Cluster P are different to each other so that we can divide them in the second step. Finally, we get the clusters [P, QRSTV].
- Step 3: Divide the chosen cluster into two subclusters using a flat clustering method (such as K-means, K-medoids, etc.). Here, we divide the cluster [P, QRSTV] into [P, QR, STV].
- Step 4: Repeat the same process. The cluster [P, QR, STV] divide to form [P, Q, R, ST, V].
- Step 5: Finally, the remaining two clusters split to form a n cluster [P, Q, R, S, T, V].