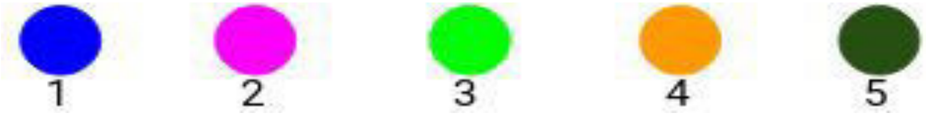
Steps to Perform Hierarchical Clustering

Step 1: First, we assign all the points to an individual cluster:

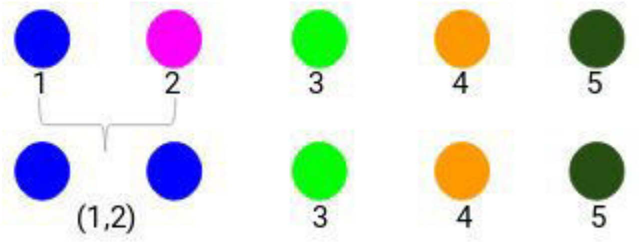
Different colors here represent different clusters. You can see that we have 5 different clusters for the 5 points in our data.



Step 2: Next, we will look at the smallest distance in the proximity matrix and merge the points with the smallest distance. We then update the proximity matrix:

ID	1	2	3	4	5
1	0	3	18	10	25
2	3	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0

Here, the smallest distance is 3 and hence we will merge point 1 and 2:



Let’s look at the updated clusters and accordingly update the proximity matrix:

Student_ID	Marks
(1,2)	10
3	28
4	20
5	35

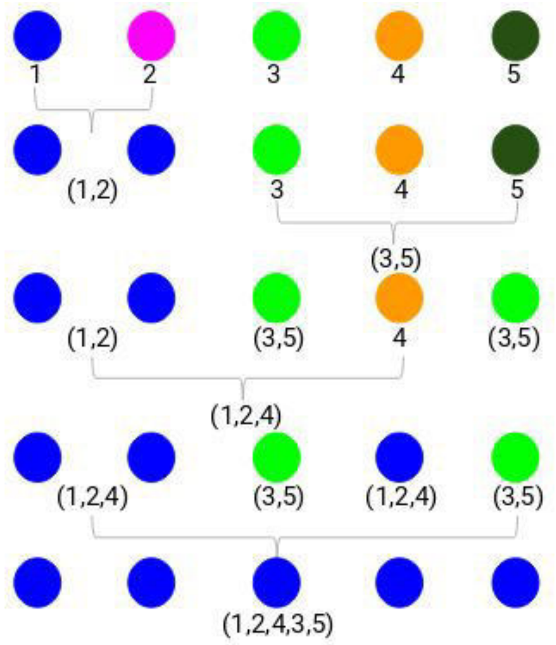
Here, we have taken the maximum of the two marks (7, 10) to replace the marks for this cluster. Instead of the maximum, we can also take the minimum value or the average values as well.

Now, we will again calculate the proximity matrix for these clusters:

ID	(1,2)	3	4	5
(1,2)	0	18	10	25
3	18	0	8	7
4	10	8	0	15
5	25	7	15	0

Step 3: We will repeat step 2 until only a single cluster is left.
So, we will first look at the minimum distance in the proximity matrix and then merge the closest pair of clusters.

We will get the merged clusters as shown below after repeating these steps:



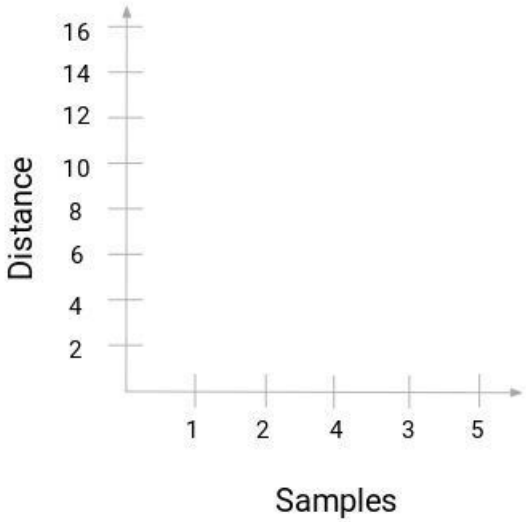
We started with 5 clusters and finally have a single cluster. This is how agglomerative hierarchical clustering works. But the burning question still remains – how do we decide the number of clusters? Let’s understand that in the next section.

How to Choose the Number of Clusters in Hierarchical Clustering?

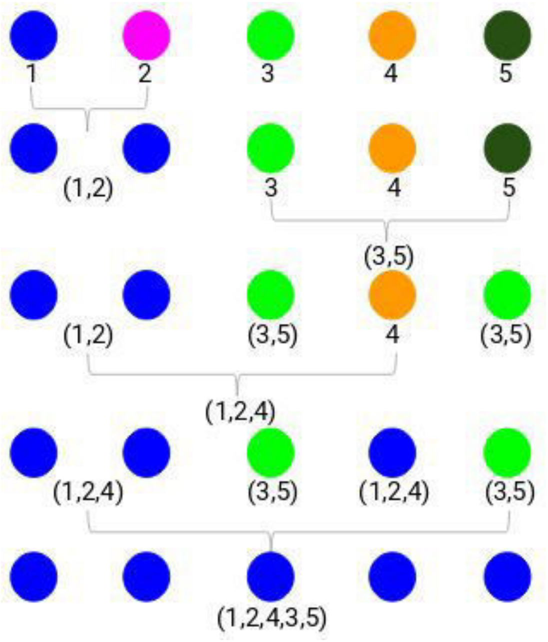
- To get the number of clusters for hierarchical clustering, we make use of an awesome concept called a Dendrogram.
- A dendrogram is a tree-like diagram that records the sequences of merges or splits.

Example

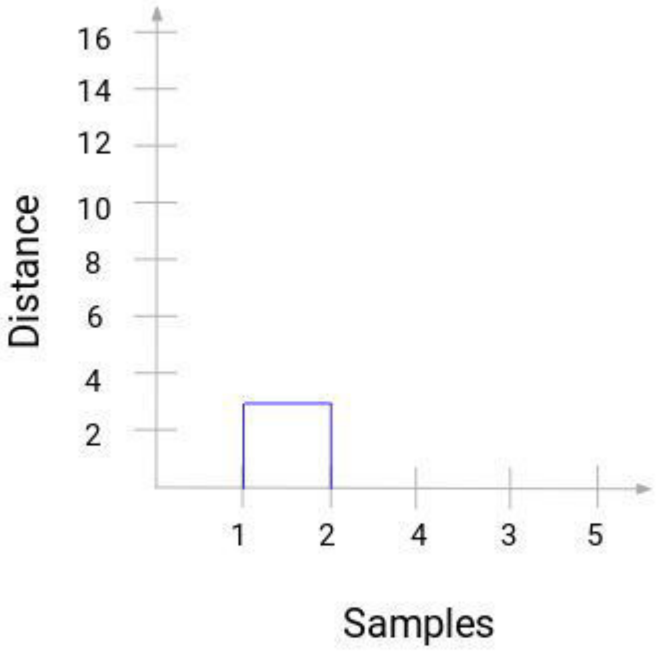
Let's get back to teacher-student example. Whenever we merge two clusters, a dendrogram will record the distance between these clusters and represent it in graph form. Let's see how a dendrogram looks:



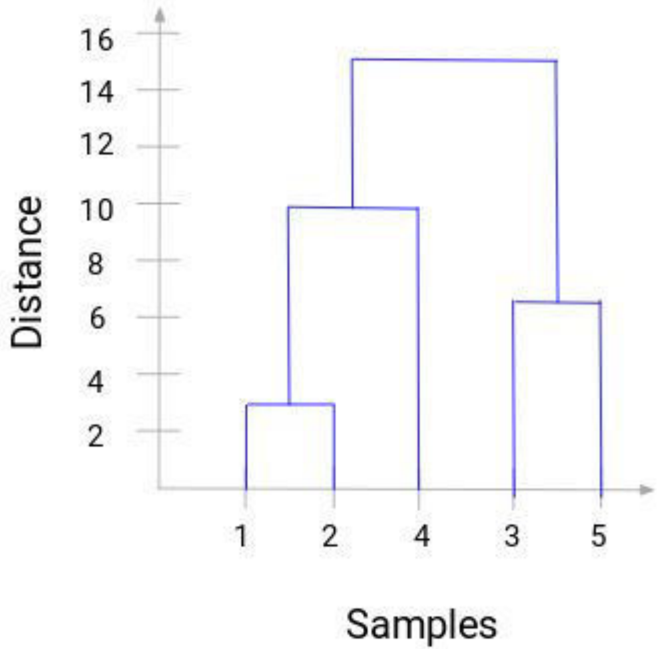
We have the samples of the dataset on the x-axis and the distance on the y-axis. Whenever two clusters are merged, we will join them in this dendrogram and the height of the join will be the distance between these points. Let's build the dendrogram for our example:



Take a moment to process the above image. We started by merging sample 1 and 2 and the distance between these two samples was 3 (refer to the first proximity matrix in the previous section). Let's plot this in the dendrogram:

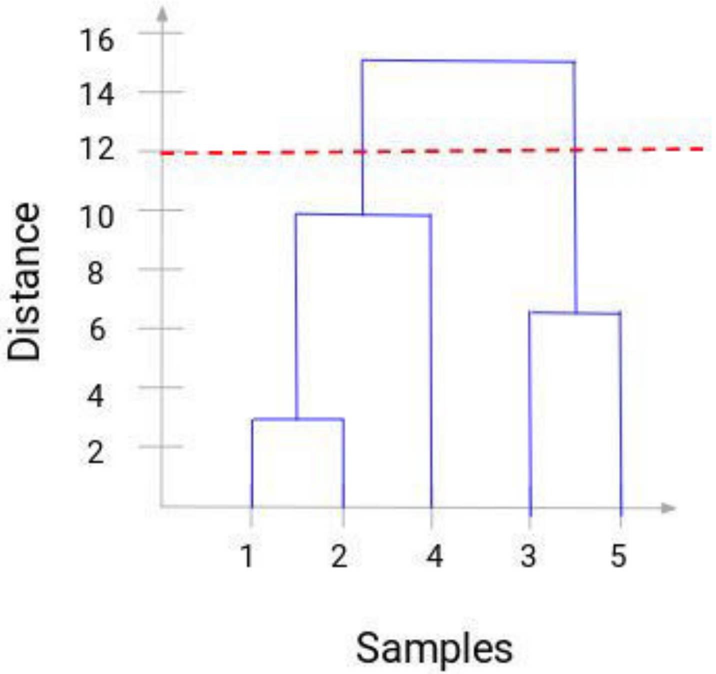


Here, we can see that we have merged sample 1 and 2. The vertical line represents the distance between these samples. Similarly, we plot all the steps where we merged the clusters and finally, we get a dendrogram like this:



We can visualize the steps of hierarchical clustering. More the distance of the vertical lines in the dendrogram, more the distance between those clusters.

Now, we can set a threshold distance and draw a horizontal line (*Generally, we try to set the threshold so that it cuts the tallest vertical line*). Let's set this threshold as 12 and draw a horizontal line:



The number of clusters will be the number of vertical lines intersected by the line drawn using the threshold. In the above example, since the red line intersects 2 vertical lines, we will have 2 clusters. One cluster will have a sample (1,2,4) and the other will have a sample (3,5).

This is how we can decide the number of clusters using a dendrogram in Hierarchical Clustering. In the next section, we will implement hierarchical clustering to help you understand all the concepts we have learned in this article.

Association Rule Mining

Association rule learning is a type of unsupervised learning technique that checks for the dependency of one data item on another data item and maps accordingly so that it can be more profitable.

It tries to find some interesting relations or associations among the variables of dataset.

It is employed in **Market Basket analysis, Web usage mining, continuous production, etc. Here market basket analysis is a** technique used by the various big retailer to discover the associations between items.

In a supermarket, all products that are purchased together are put together.

For example, if a customer buys bread, he most likely can also buy butter, eggs, or milk, so these products are stored within a shelf or mostly nearby. Consider the below diagram:

Customer 1

Customer 2

Customer 3

Customer n

Association rule learning can be divided into three types of algorithms:

1. **Apriori: This algorithm uses frequent datasets to generate association rules. It is designed to work on the** databases that contain transactions. This algorithm uses a breadth-first search and Hash Tree to calculate the itemset efficiently.

It is mainly used for market basket analysis and helps to understand the products that can be bought together. It can also be used in the healthcare field to find drug reactions for patients.

2. **Eclat: Eclat algorithm stands for Equivalence Class Transformation. This algorithm uses a depth-first search** technique to find frequent itemsets in a transaction database. It performs faster execution than Apriori Algorithm.
3. **F-P Growth Algorithm: The F-P growth algorithm stands for Frequent Pattern, and it is the improved version of the** Apriori Algorithm. It represents the database in the form of a tree structure that is known as a frequent pattern or tree. The purpose of this frequent tree is to extract the most frequent patterns.

Association Rule Learning working:

Association rule learning works on the concept of If and Else Statement, such as if A then B.

Here the If element is called **antecedent**, and then statement is called as **Consequent**. These types of relationships where we can find out some association or relation between two items is known *as single cardinality*.

To measure the associations between thousands of data items, there are several metrics. These metrics are given below:

- **Support**
- **Confidence**
- **Lift**

Support Support is defined as the frequency with which X and Y are purchased together over the total number of purchases or transactions.

$$Support = \frac{frq(X,Y)}{N}$$

Confidence

Confidence can be defined as the frequency with which X and Y are purchased together over the frequency with which X is purchased in isolation.

$$\text{Confidence} = \frac{Freq(X,Y)}{Freq(X)}$$

Lift

Lift is defined as the Support over the Support for X times the Support for Y.

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

- If **Lift= 1: The probability of occurrence of antecedent and consequent is independent of each other.**
- **Lift>1: It determines the degree to which the two item sets are dependent to each other.**
- **Lift<1: It tells us that one item is a substitute for other items, which means one item has a negative effect on** another.

Rule	Support	Confidence	Lift
$A \Rightarrow D$	2/5	2/3	10/9
$C \Rightarrow A$	2/5	2/4	5/6
$A \Rightarrow C$	2/5	2/3	5/6
$B \& C \Rightarrow D$	1/5	1/3	5/9

Applications of Association Rule Learning

It has various applications in machine learning and data mining. Below are some popular applications of association rule learning:

- **Market Basket Analysis: It is one of the popular examples and applications of association rule mining. This** technique is commonly used by big retailers to determine the association between items.
- **Medical Diagnosis: With the help of association rules, patients can be cured easily, as it helps in identifying the** probability of illness for a particular disease.
- **Protein Sequence: The association rules help in determining the synthesis of artificial Proteins.**
- It is also used for the **Catalog Design and Loss-leader Analysis and many more other applications.**

Introduction to reinforcement learning – Q learning

Reinforcement learning addresses the question of how an autonomous agent that senses and acts in its environment can learn to choose optimal actions to achieve its goals.

This very generic problem covers tasks such as learning to control a mobile robot, learning to optimize operations in factories, and learning to play board games.

Each time the agent performs an action in its environment, a trainer may provide a reward or penalty to indicate the desirability of the resulting state.

For example, when training an agent to play a game the trainer might provide a positive reward when the game is won, negative reward when it is lost, and zero reward in all other states.

The task of the agent is to learn from this indirect, delayed reward, to choose sequences of actions that produce the greatest cumulative reward.

Q learning that can acquire optimal control strategies from delayed rewards, even when the agent has no prior knowledge of the effects of its actions on the environment.