Hierarchical Clustering

Hierarchical clustering is very common in biology and lends itself nicely to visualizing clusters.

It can also help the user decide on an appropriate number of clusters.

Overview

1. *Theory and Intuition*
2. *Coding*

Theory and Intuition

Like most clustering algorithms, Hierarchical Clustering simply relies on measuring which data points are most “similar” to other data points.

“Similarity” is defined by choosing a distance metric.

Benefits of Hierarchical Clustering

- Easy to understand and visualize.
- Helps users decide how many clusters to choose.
- Not necessary to choose cluster amount **before** running the algorithm.

So why use Hierarchical Clustering?

- Divides points into ***potential*** clusters:

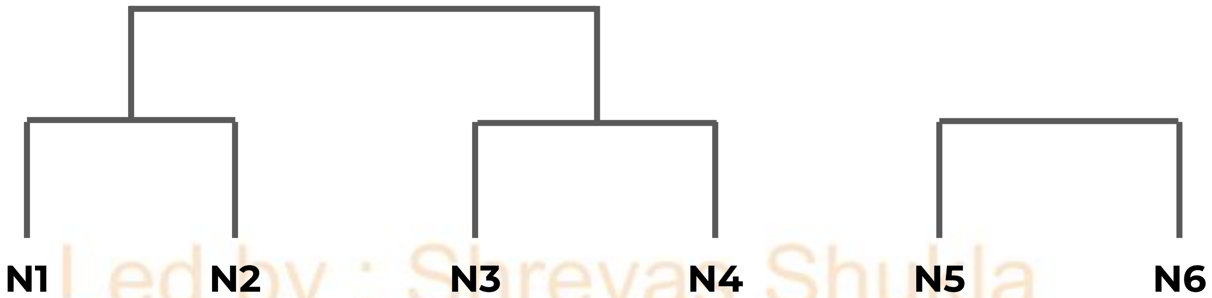
So why use Hierarchical Clustering?

- Divides points into ***potential*** clusters:
  - Agglomerative Approach:
    - Each point begins as its own cluster, then clusters are joined.
  - Divisive Approach:
    - All points begin in the same cluster, then clusters are split.

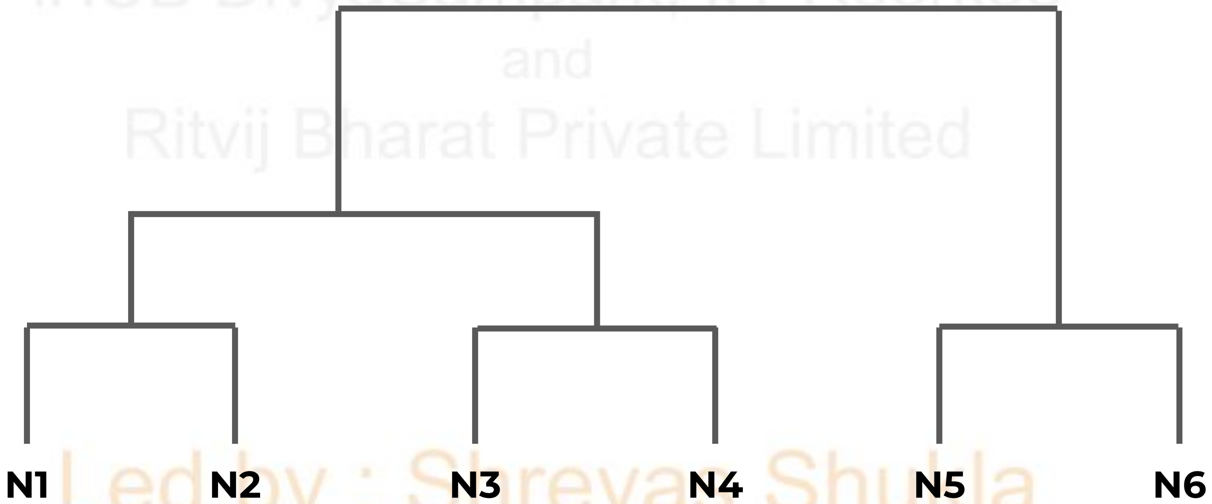
Agglomerative:

Agglomerative:

Agglomerative:



Agglomerative:

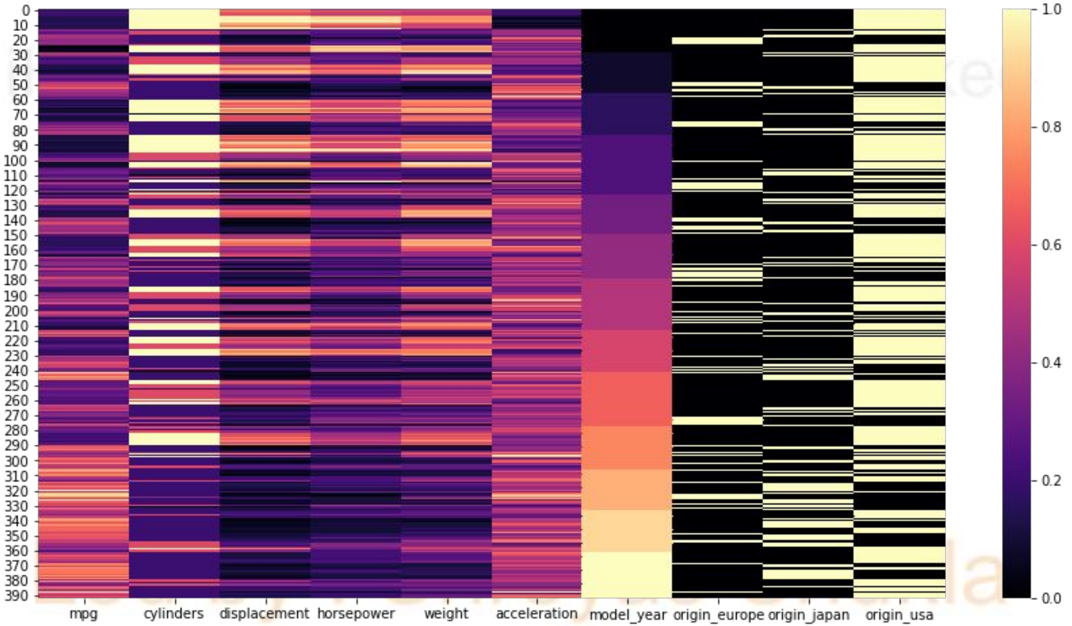


Opposite of the Agglomerative approach is a **Divisive** approach, which starts with all points belonging to the same cluster, and the begins divisions to separate out clusters.

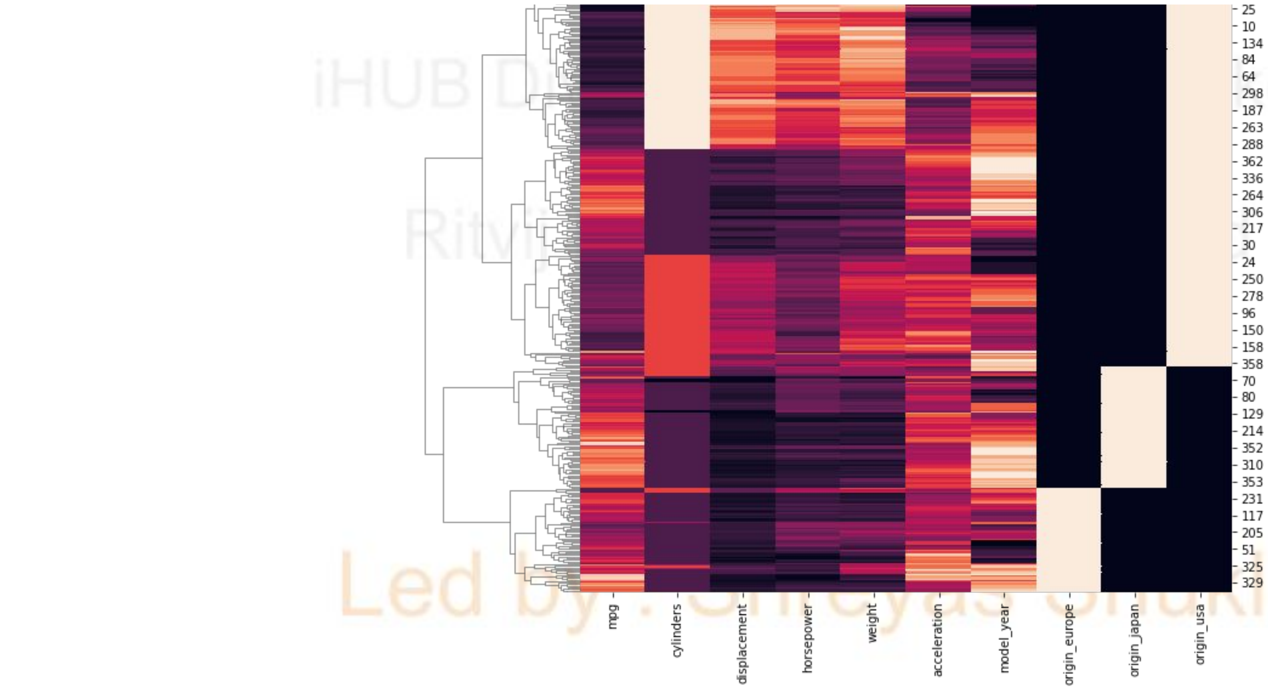
Hierarchical Clustering Process

- Compare data points to find most similar data points to each other.
- Merge these to create a cluster.
- Compare clusters to find most similar clusters and merge again.
- Repeat until all points in a single cluster.

Hierarchical Clustering Process



Hierarchical Clustering Process



Topics which we still need to understand for Hierarchical Clustering:

- Similarity Metric
- Dendrogram
- Linkage Matrix

Similarity Metric

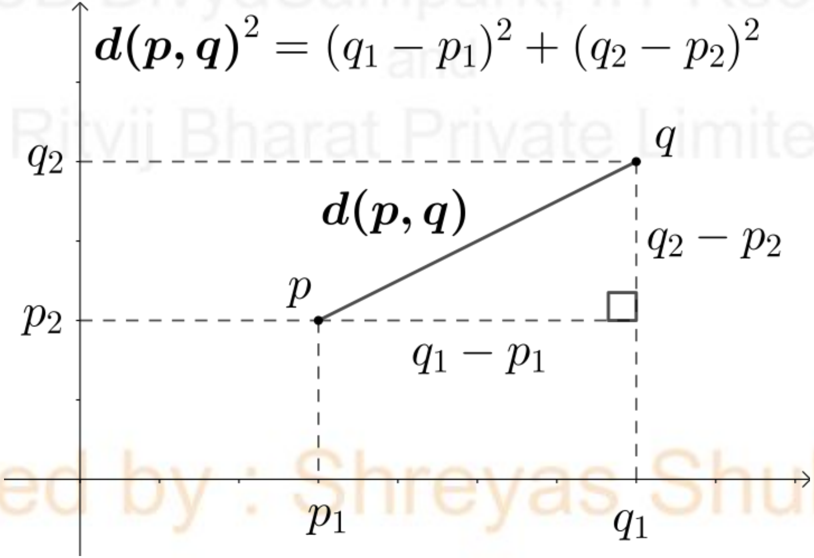
Measures distance between two points.

Many types:

- Euclidean Distance
- Manhattan
- Cosine
- and many more...

Similarity Metric

Default choice is Euclidean



Similarity Metric

- Each dimension would be a feature
- For **n** data points and **p** features:
  - $D^2 = (x_{11} - x_{12})^2 + \ldots + (x_{n\text{-}1p\text{-}1} - x_{np})^2$

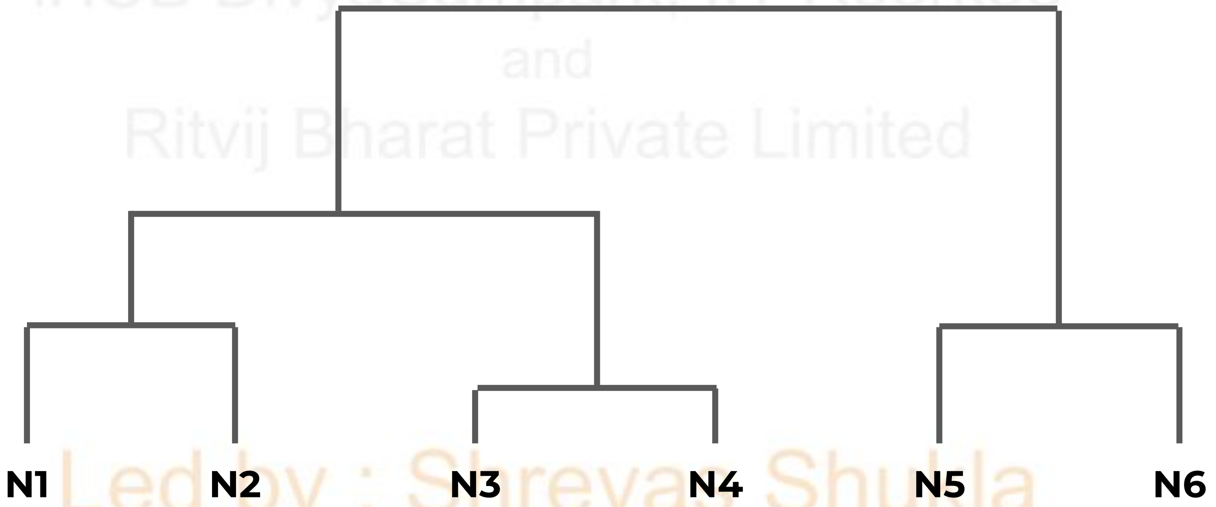
Similarity Metric

- Each dimension would be a feature
- For **n** data points and **p** features:
  - $D^2 = (x_{11} - x_{12})^2 + \ldots + (x_{n-1p-1} - x_{np})^2$
- Using MinMaxScaler we can scale all features to be between 0 and 1.
- This allows for maximum distance between a feature to be 1.

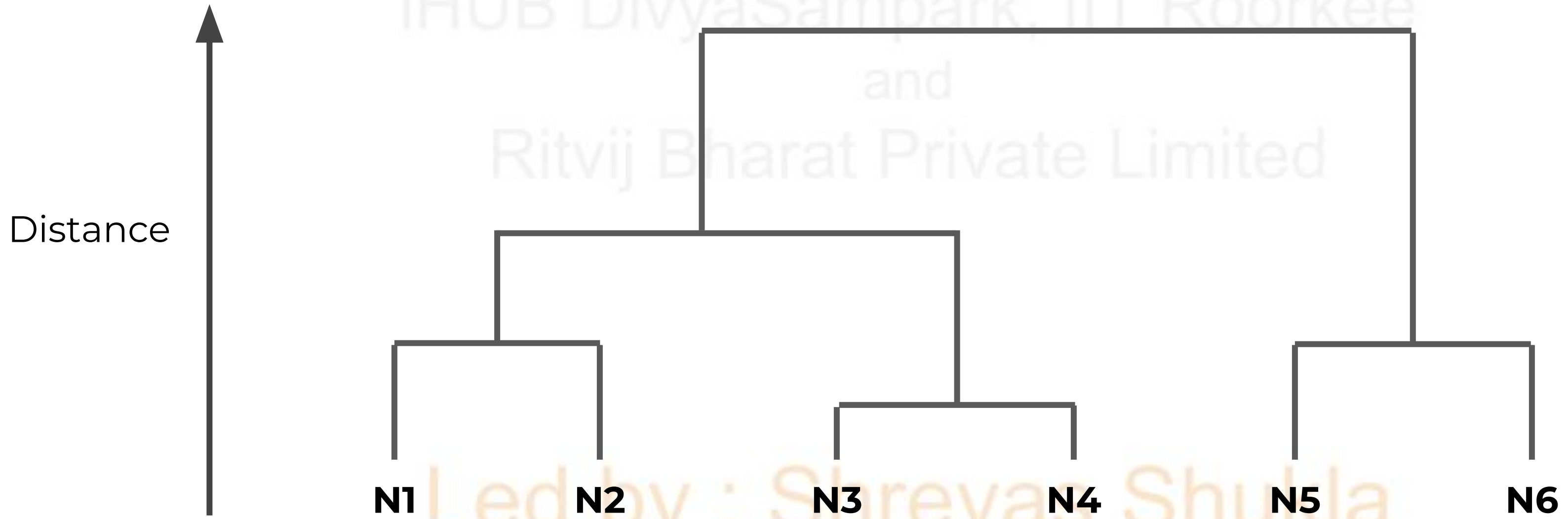
Dendrogram:

- Plot displaying all potential clusters.
- Very computationally expensive to compute and display for larger data sets.
- Very useful for deciding on number of clusters.

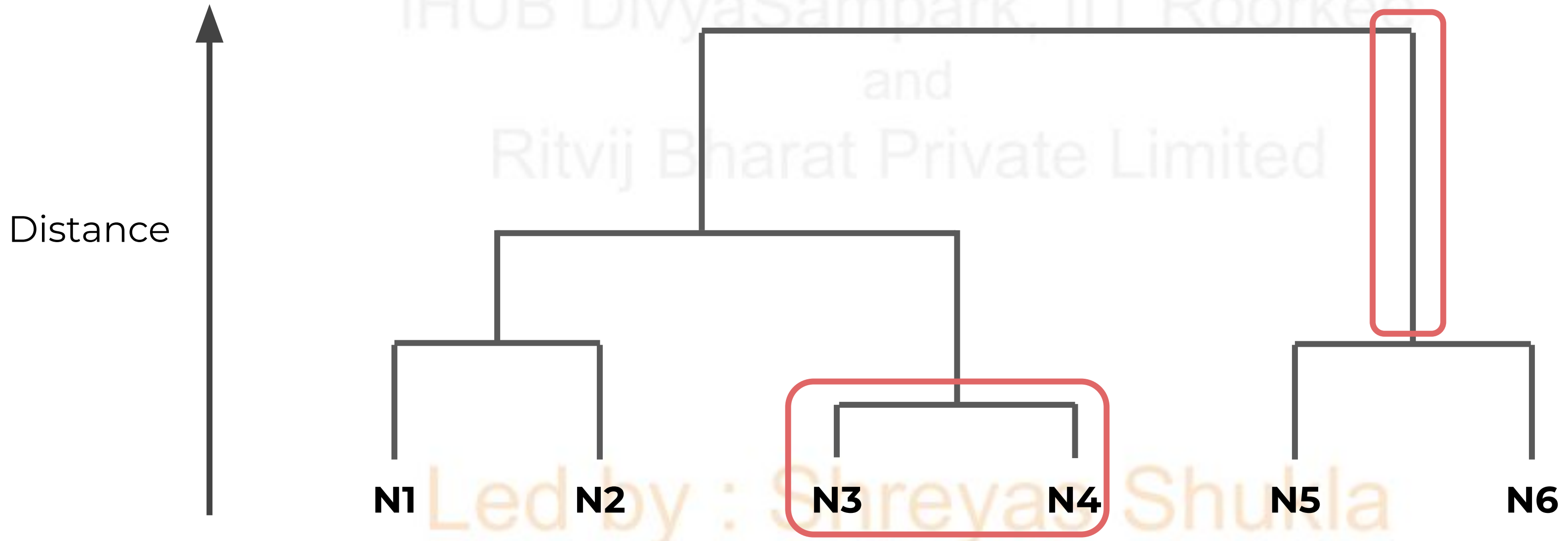
Dendrogram:



Dendrogram:

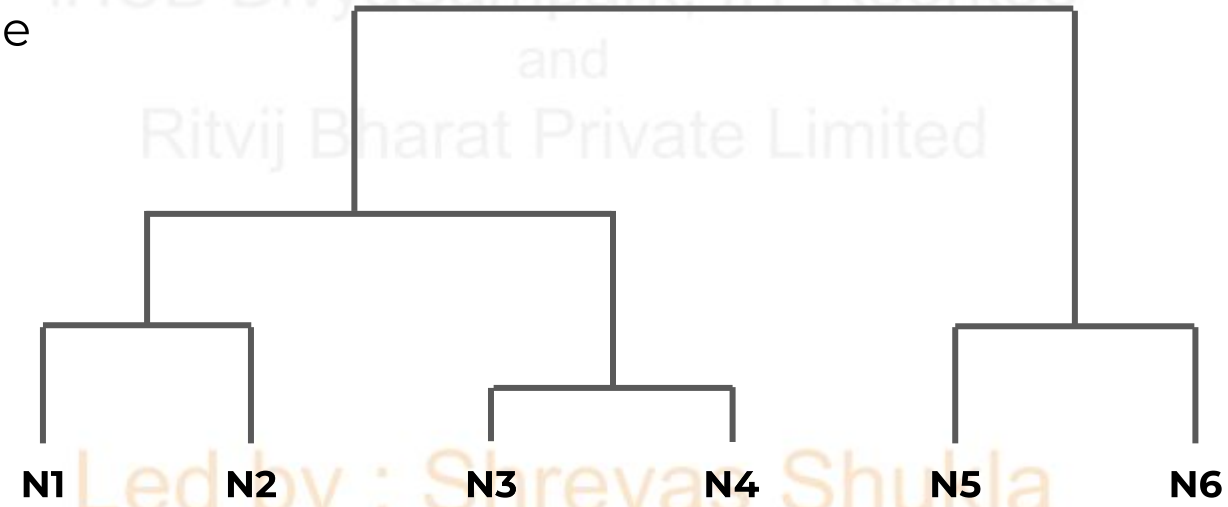


Dendrogram:



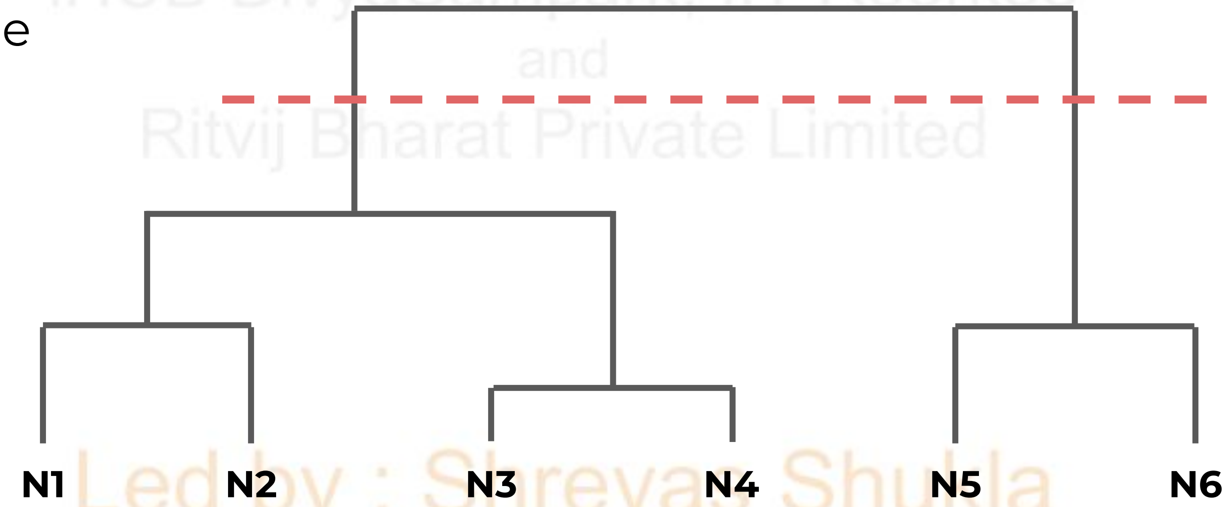
Dendrogram:

“Slice” to decide cluster count



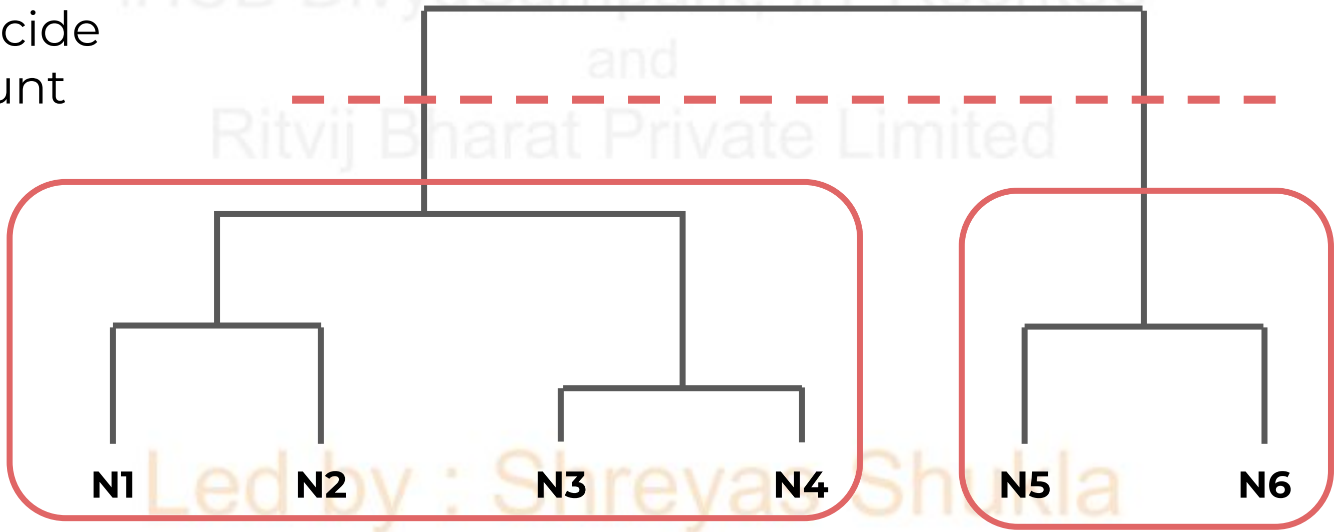
Dendrogram:

“Slice” to decide cluster count



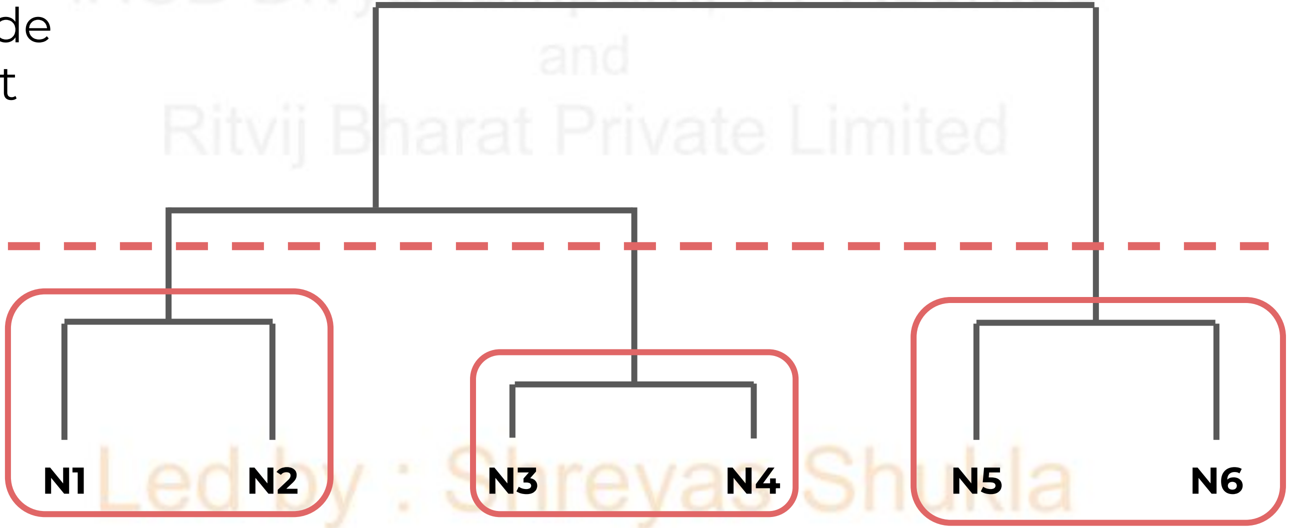
Dendrogram:

"Slice" to decide cluster count



Dendrogram:

“Slice” to decide
cluster count

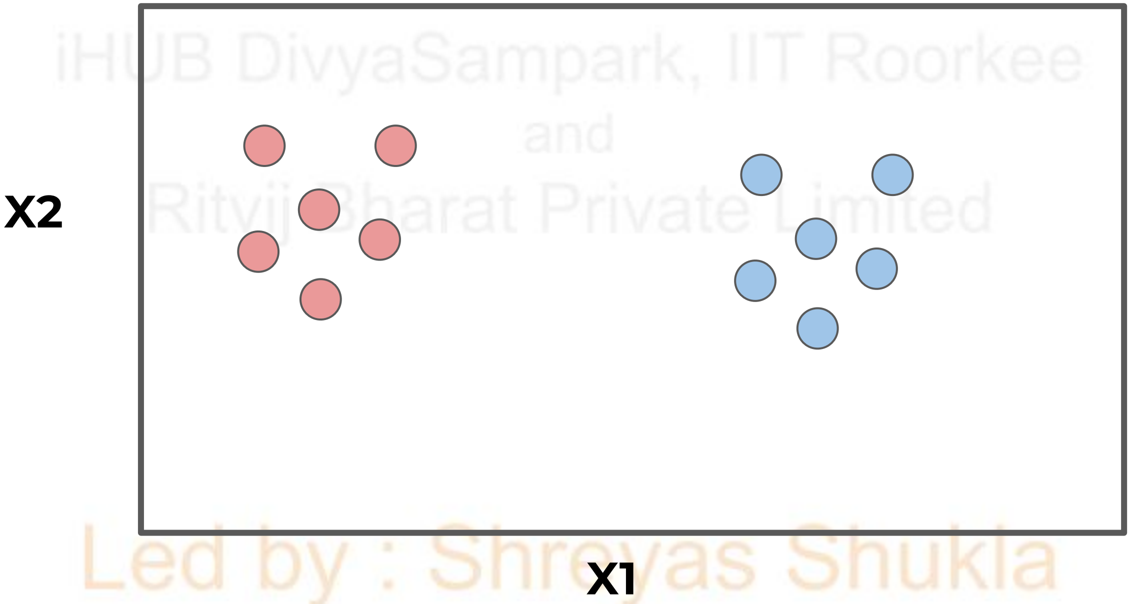


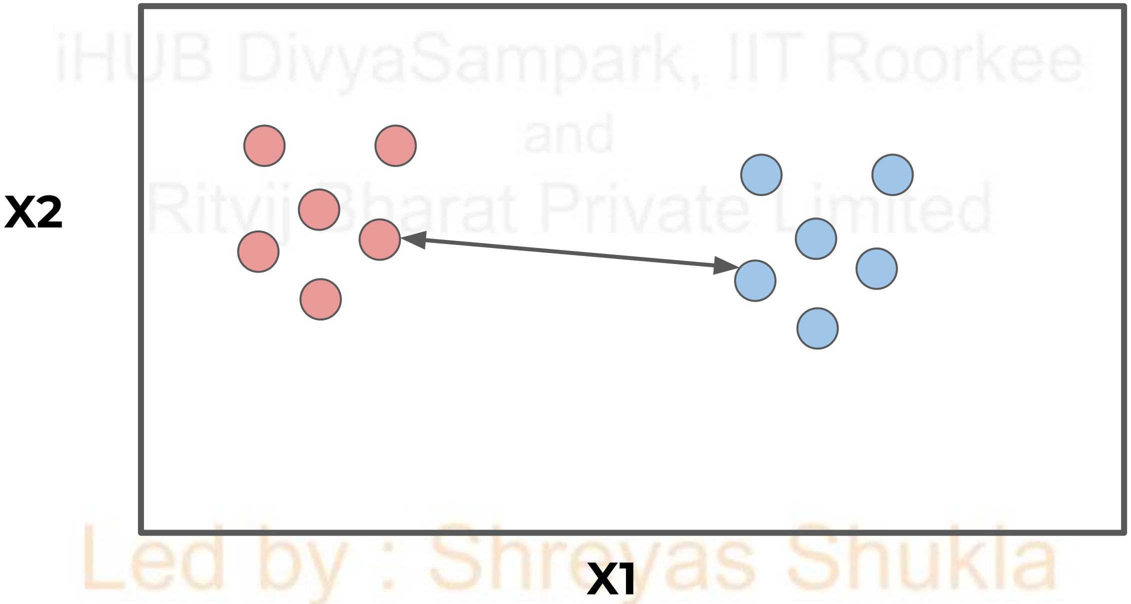
Linkage

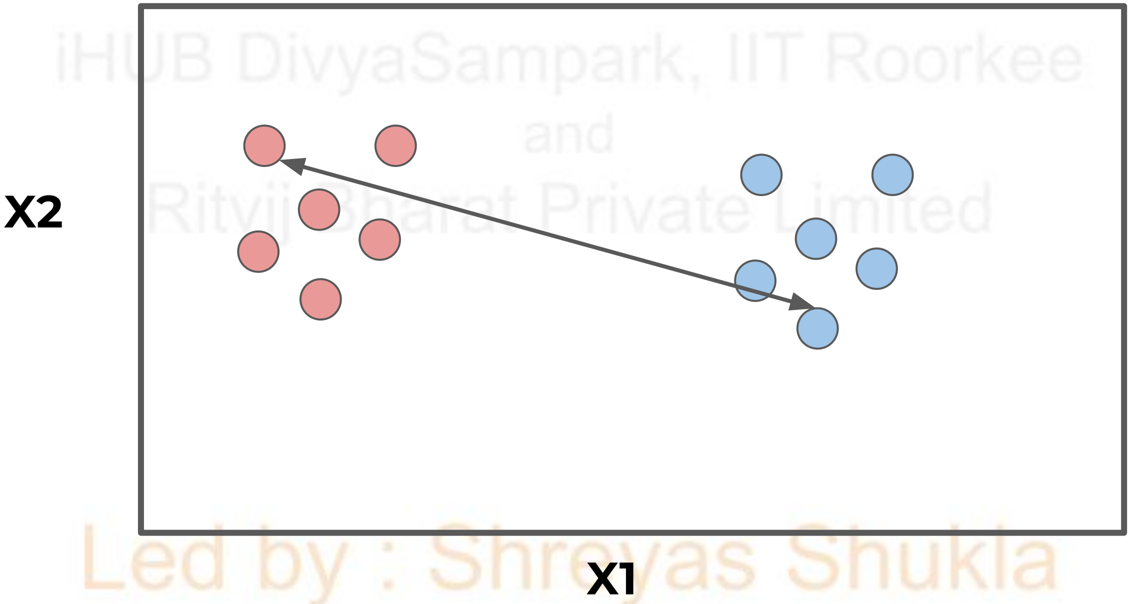
- How do we measure distance from a point to an entire cluster?
- How do we measure distance from a cluster to another cluster?

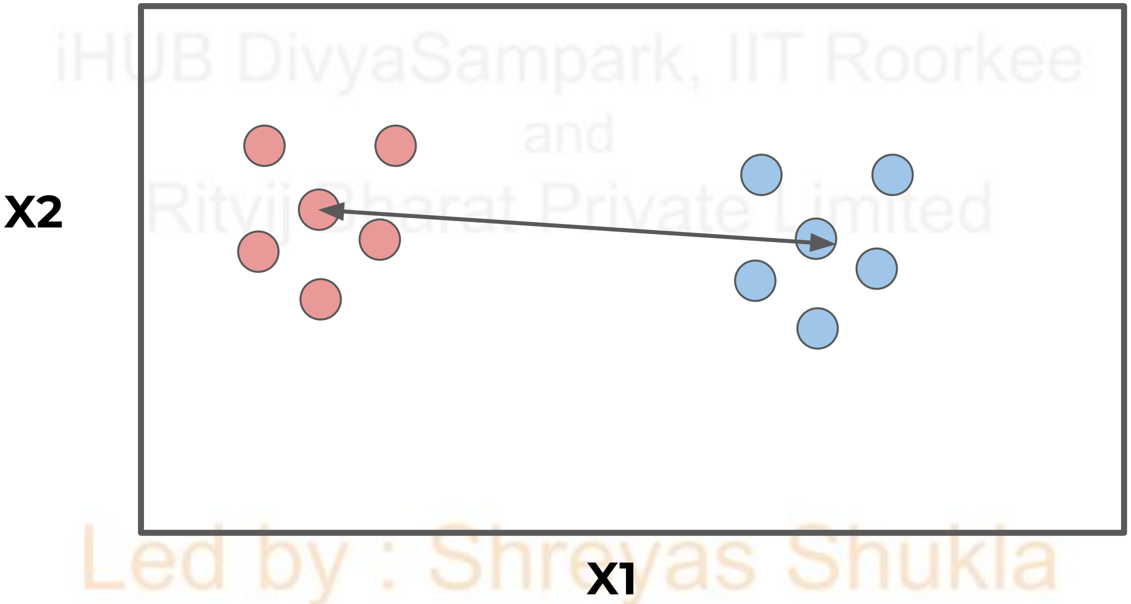
Linkage

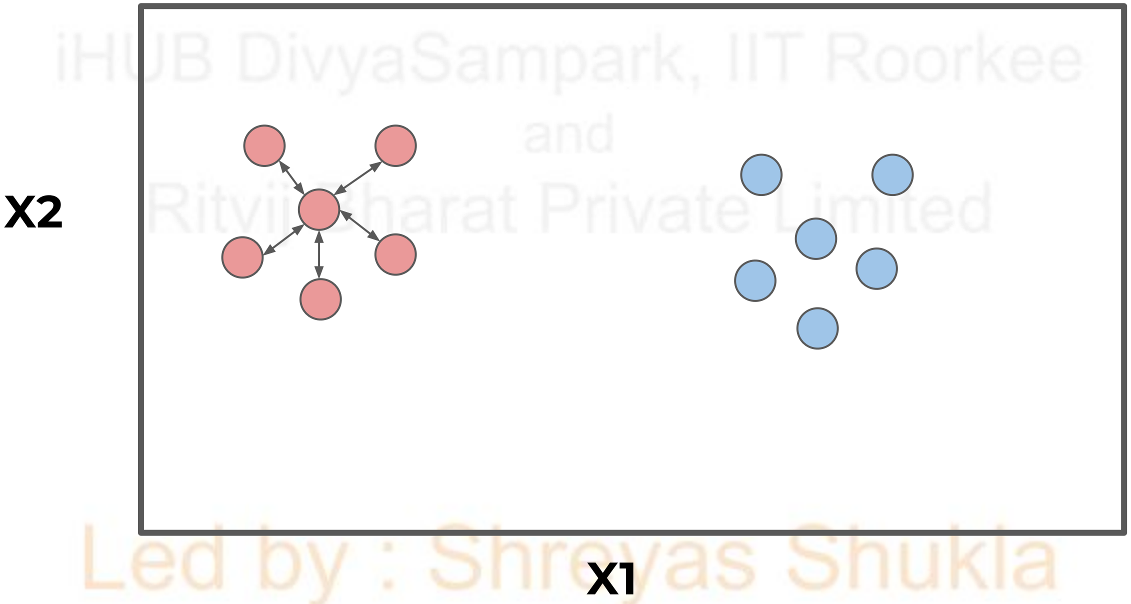
Once two or more points are together and we want to continue agglomerative clustering to join clusters, we need to decide on a **linkage** parameter.


X2
X1


X2
X1


X2
X1


X2
X1


X2
X1

Linkage

- Criterion determining which distance to use between sets of observation.
- Algorithm will merge pairs of clusters that minimizes the criterion.

Linkage:

- **Ward:** minimizes variance of clusters being merged.
- **Average:** uses average distances between two sets.
- **Minimum** or **Maximum** distances between all observations of the two sets.

Let’s code!!