An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

Conducted by:
iHUB Divya Sampark, IIT Roorkee
and
Ritvij Bharat Private Limited (RBPL)

Choosing a K Value

Presented by:
Shreyas Shukla

fig ref: pierian data

Recall our previous considerations:

- How do we choose a reasonable K value?
- Is there any way we can evaluate how good our current K value is at determining clusters?

3 clusters here, how to measure “goodness of fit”? We could measure the sum of the distances from points to cluster centers.

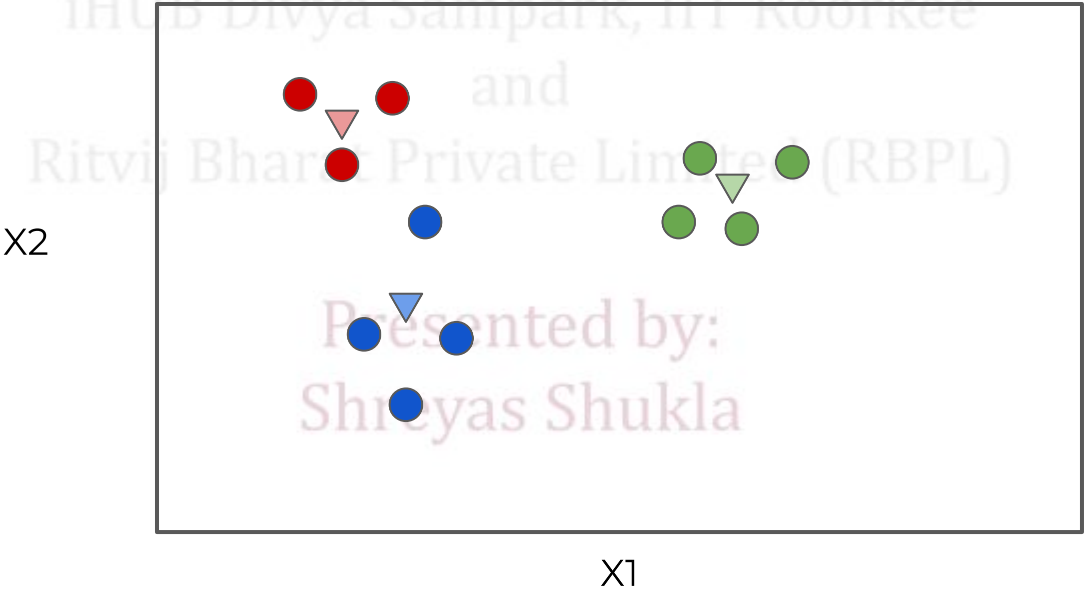


fig ref: pierian data

Imagine a simple example starting with K=2.
We measure the sum of the squared distances from points to the cluster center
Then we fit an entirely new KMeans model with K+1

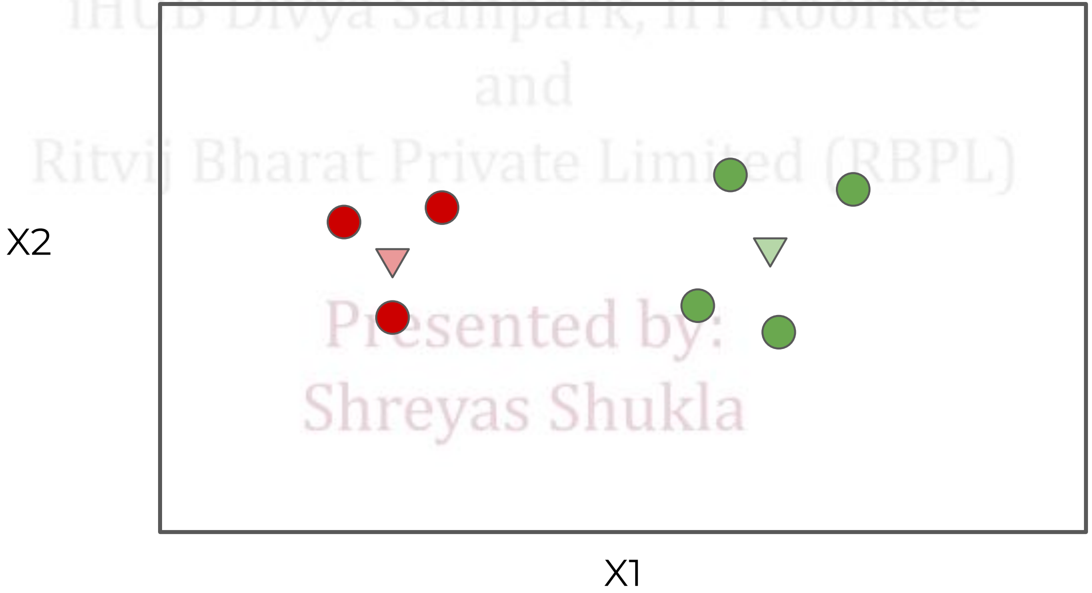


fig ref: pierian data

- Then we fit an entirely new KMeans model with K+1
- Then measure again the sum of the squared distance (SSD) to center.
- In theory this SSD would go to zero once K is equal to the number of points.

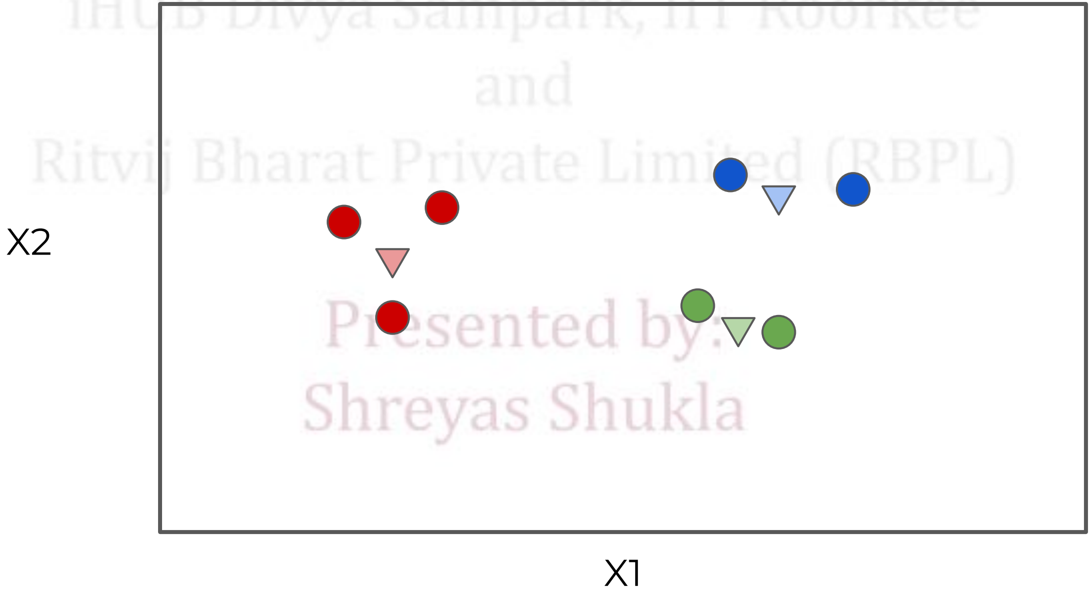


fig ref: pierian data

You would have a cluster for each point! SSD would be perfect at 0!

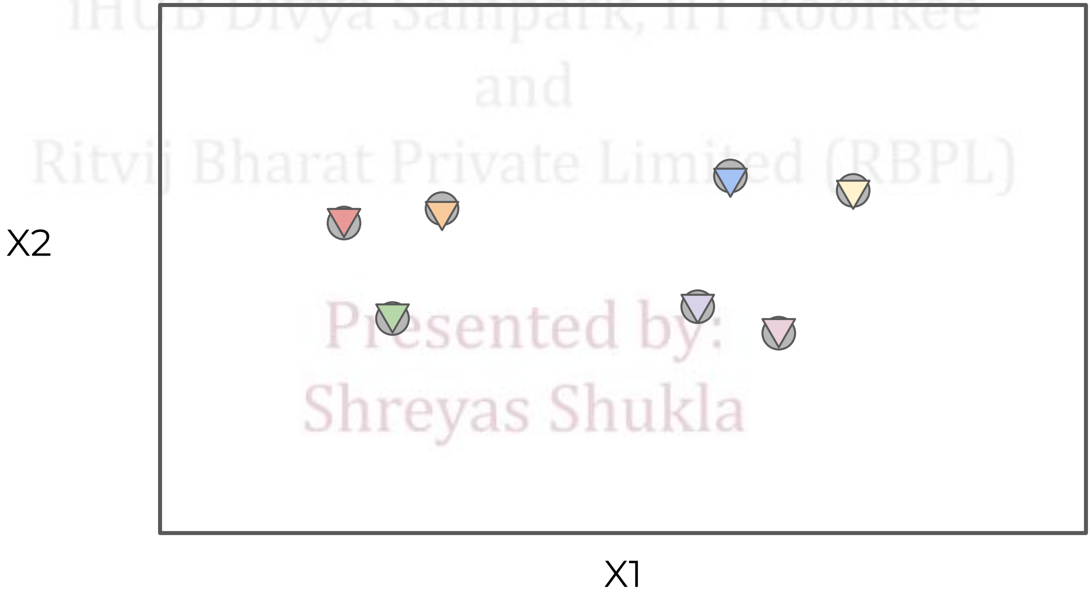
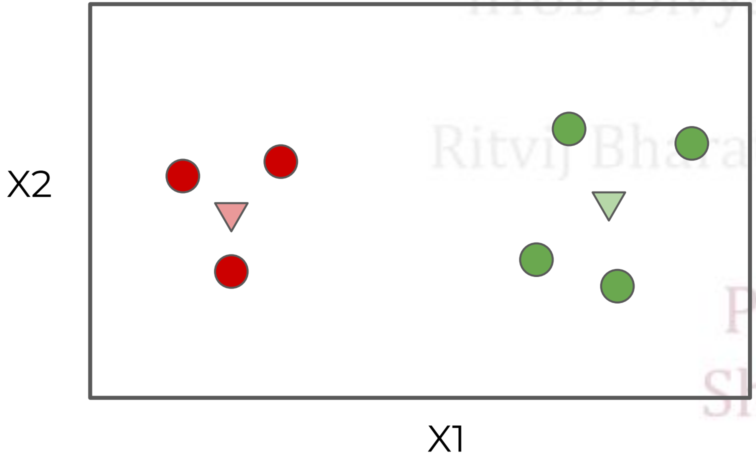


fig ref: pierian data

- Keep track of this SSD value for a range of different K values.
- Then, look for a K value where **rate of reduction in SSD** begins to decline.
- This signifies that adding an extra cluster is **not** obtaining enough clarity of cluster separation to justify increasing K.
- This is known as the “elbow” method since we will track where decrease in SSD begins to flatten out compared to increasing K values.

fig ref: pierian data

Start with K=2:



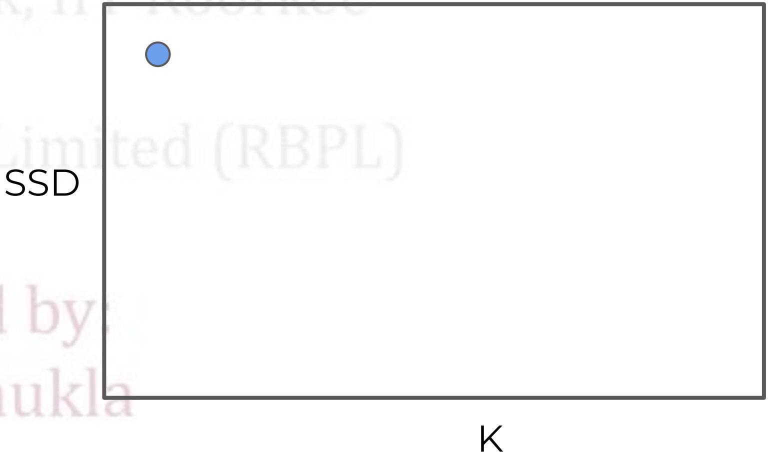


fig ref: pierian data

Increase K and measure SSD:

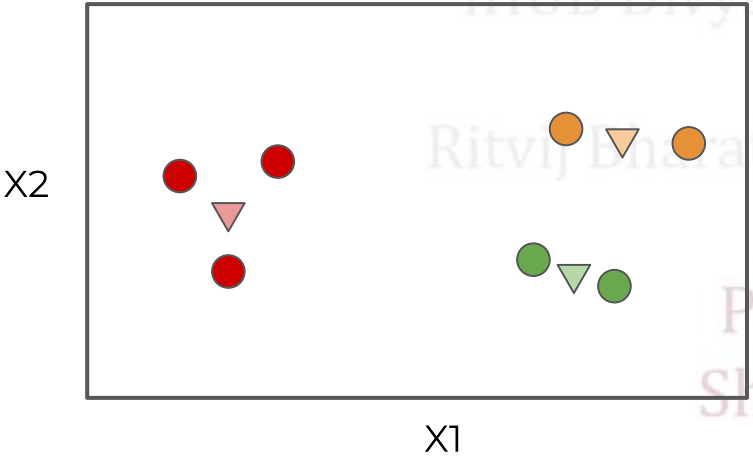
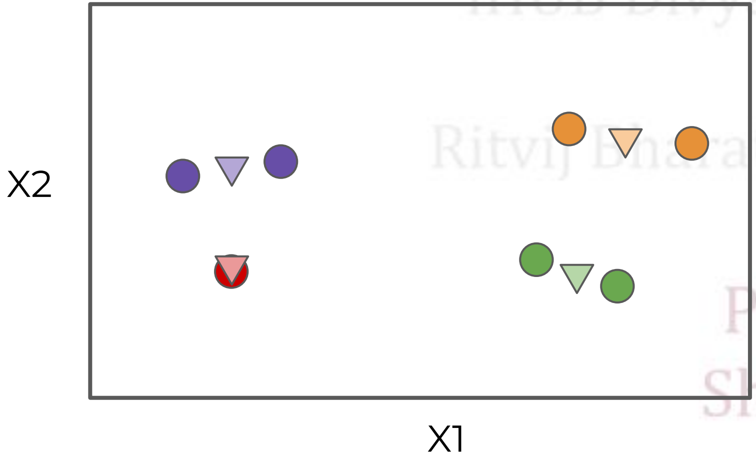


fig ref: pierian data

Increase K and measure SSD:



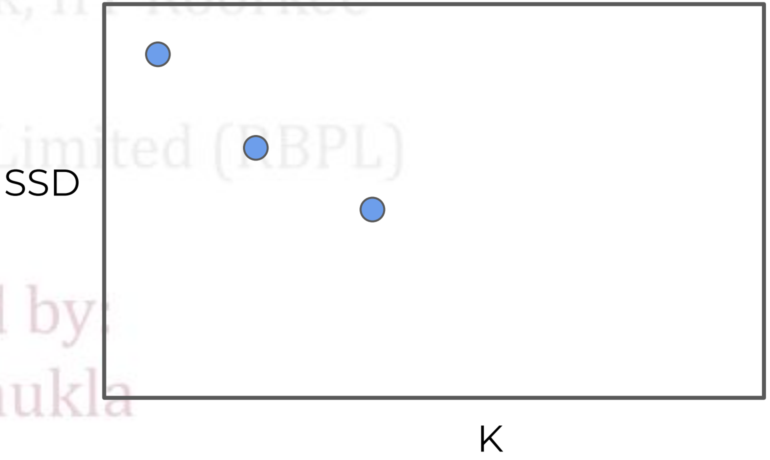
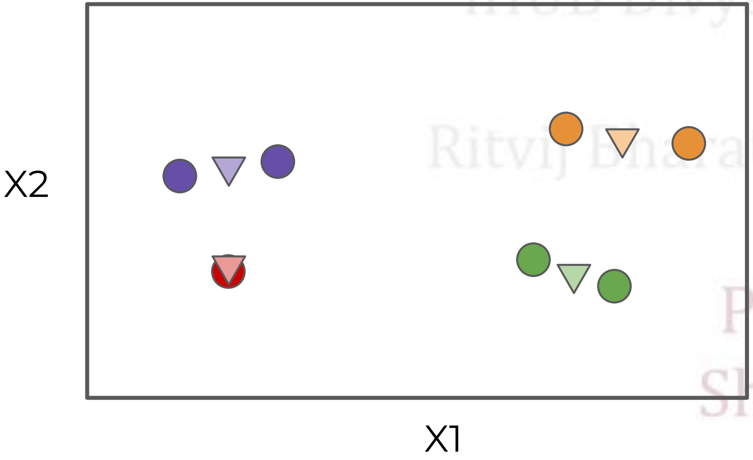


fig ref: pierian data

Repeat for some set number of K values:



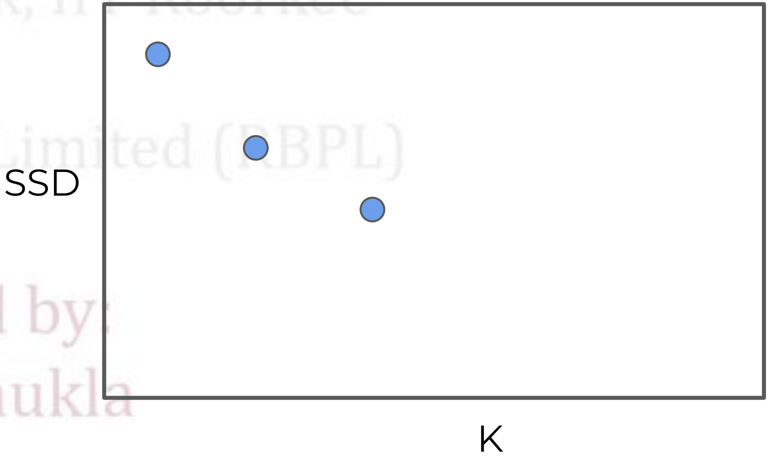


fig ref: pierian data

Notice continuous decline.

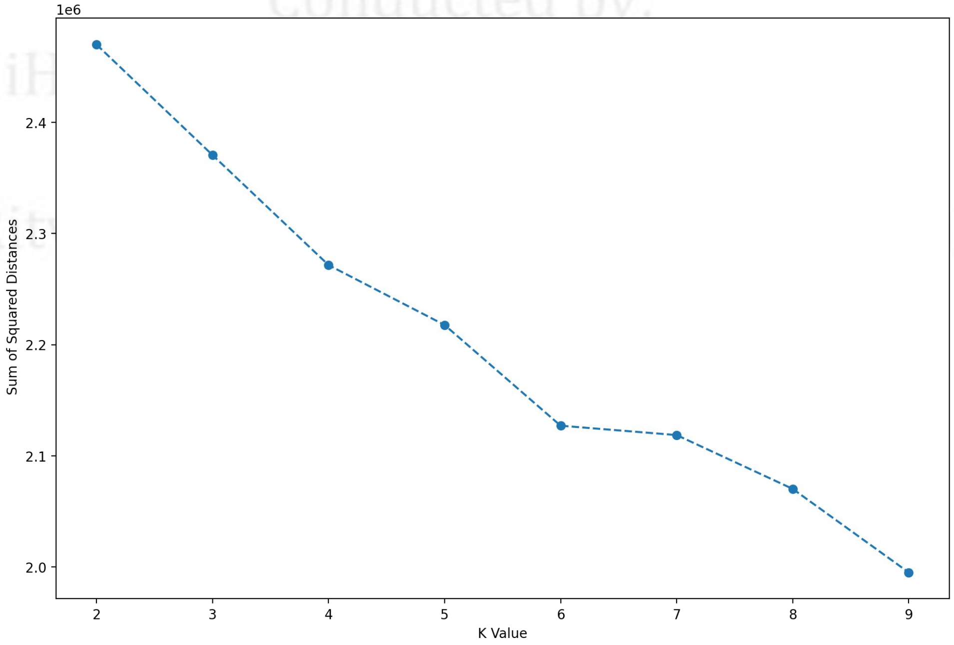


fig ref: pierian data

Eventually you will see “elbow” points:

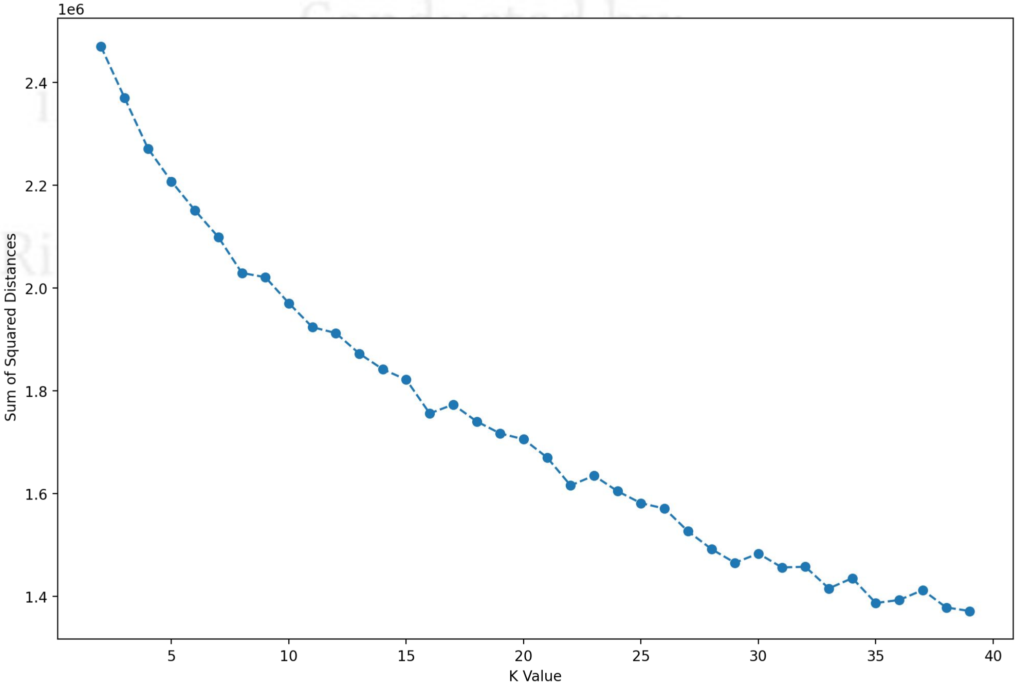


fig ref: pierian data

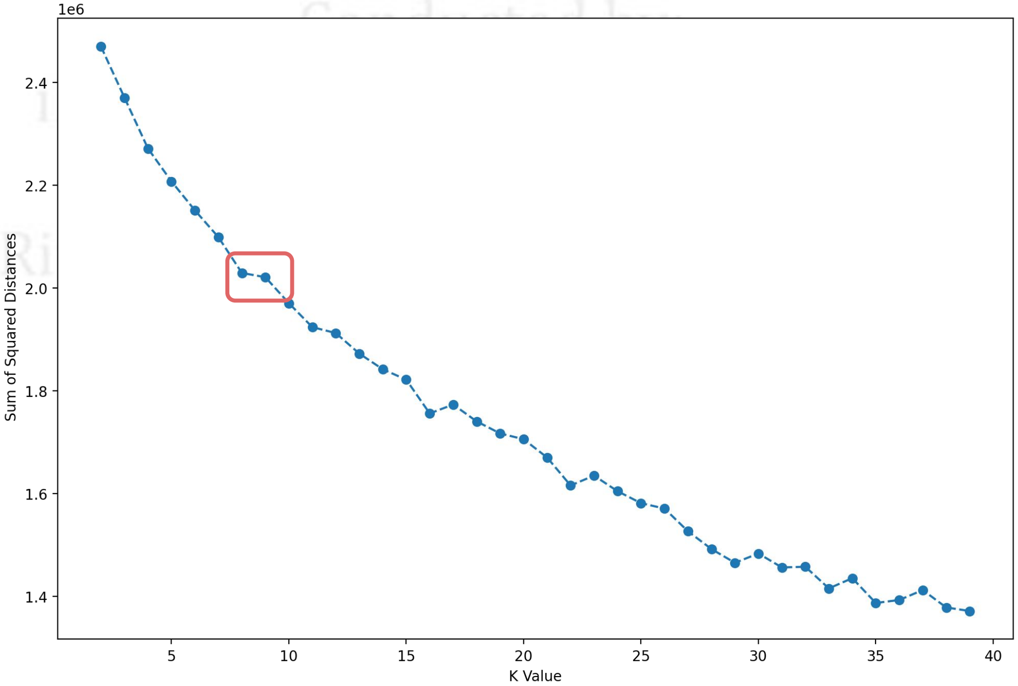


fig ref: pierian data

An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

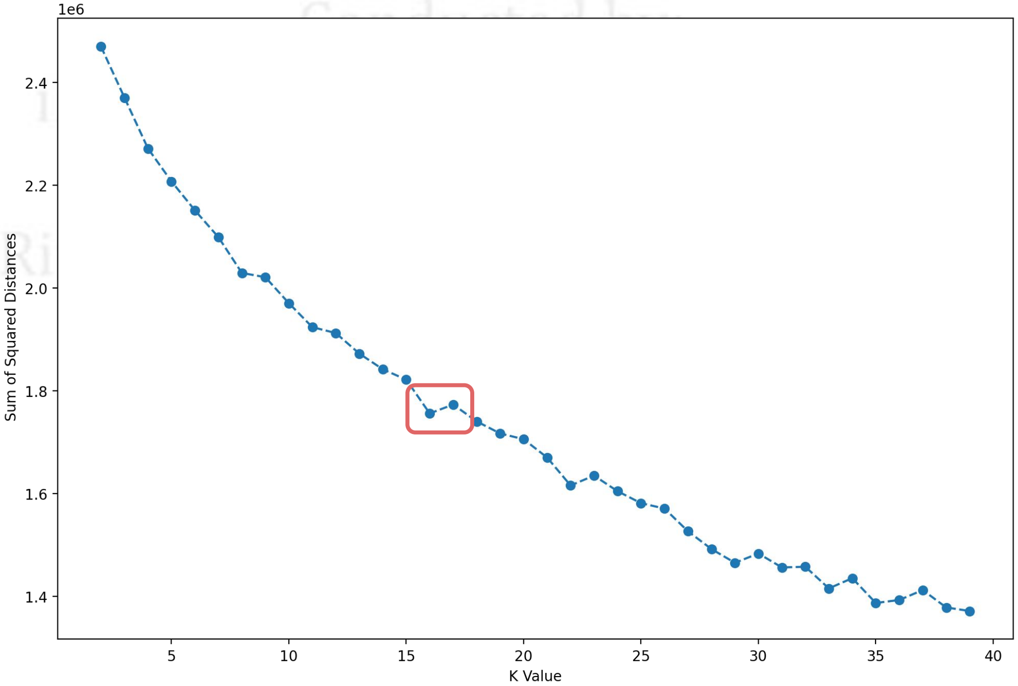


fig ref: pierian data

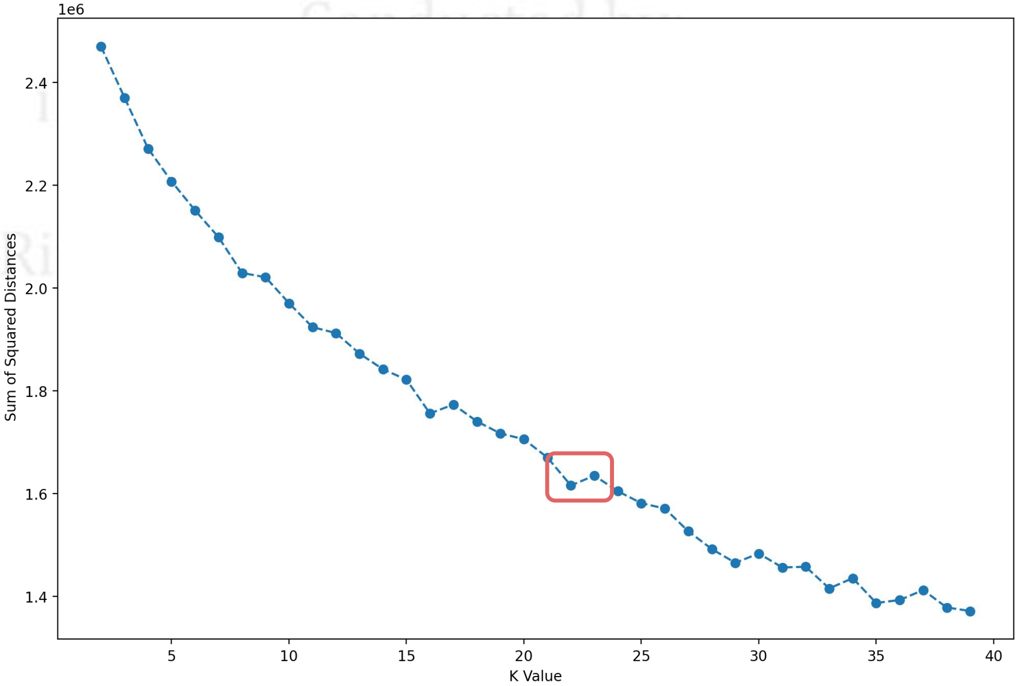


fig ref: pierian data

These points are strong indicators that increasing K further is no longer justified as it is not revealing more “signal”.

Let’s explore this further with code

fig ref: pierian data

An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

Hierarchical Clustering

Conducted by:

and

Presented by:
Shreyas Shukla

fig ref: pierian data

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.

fig ref: pierian data

Overview

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

fig ref: pierian data

An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

Conducted by:
iHUB Divya Sampark, IIT Roorkee
and
Ritvij Bharat Private Limited (RBPL)

Theory and Intuition

Presented by:
Shreyas Shukla

fig ref: pierian data

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.

fig ref: pierian data

So why use Hierarchical Clustering?

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

fig ref: pierian data

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.

fig ref: pierian data

Agglomerative:

fig ref: pierian data

Agglomerative:

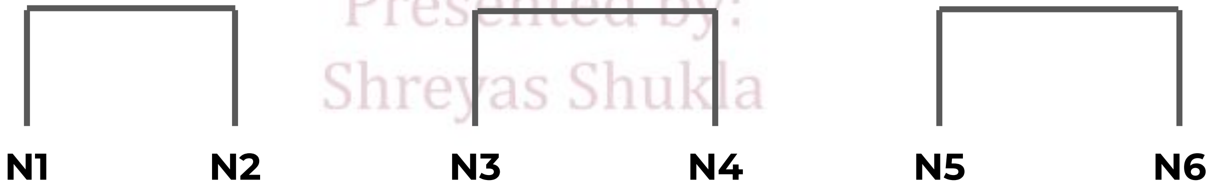


fig ref: pierian data

Agglomerative:

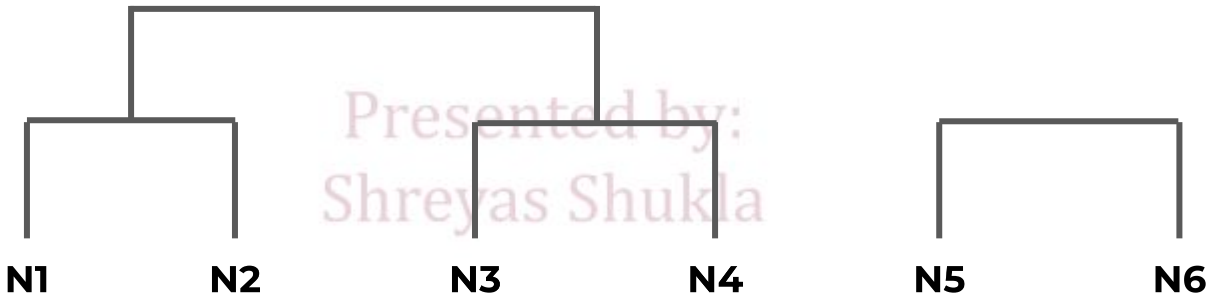


fig ref: pierian data

Agglomerative:

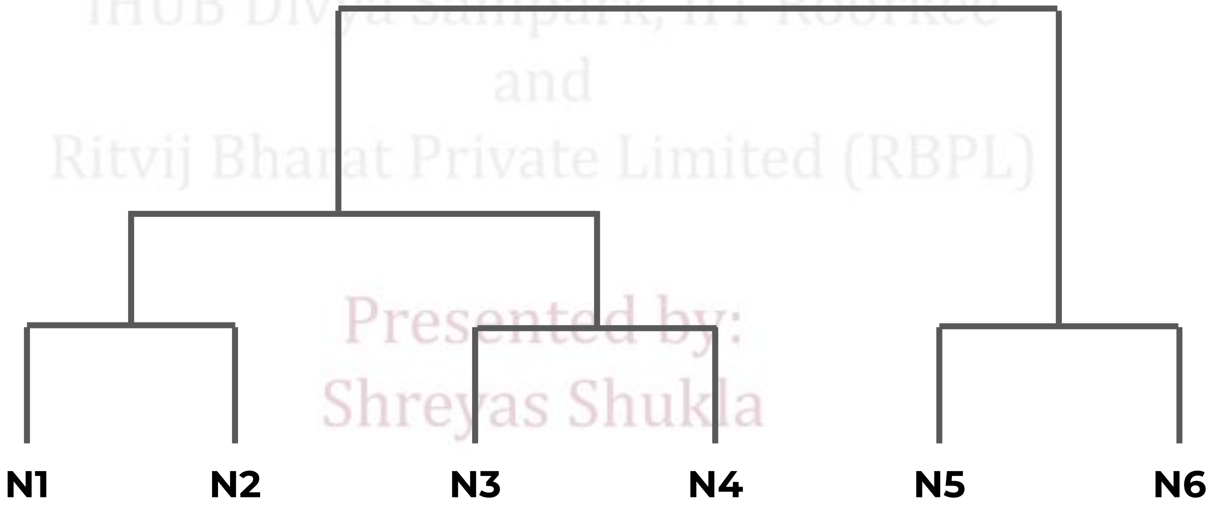


fig ref: pierian data

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.

fig ref: pierian data

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.

fig ref: pierian data

Hierarchical Clustering Process

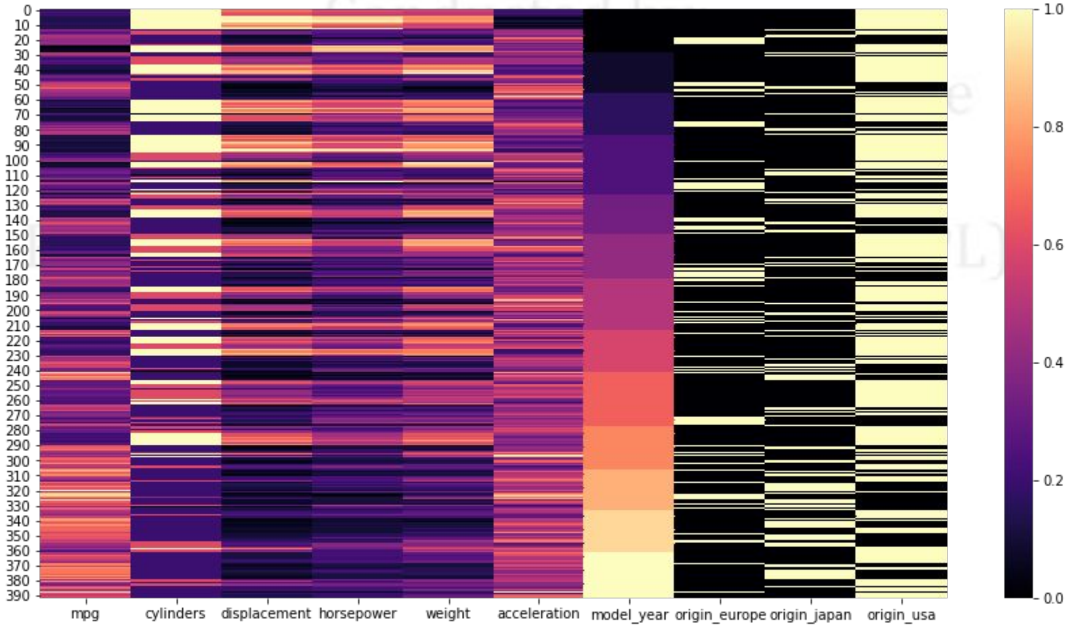


fig ref: pierian data

Hierarchical Clustering Process

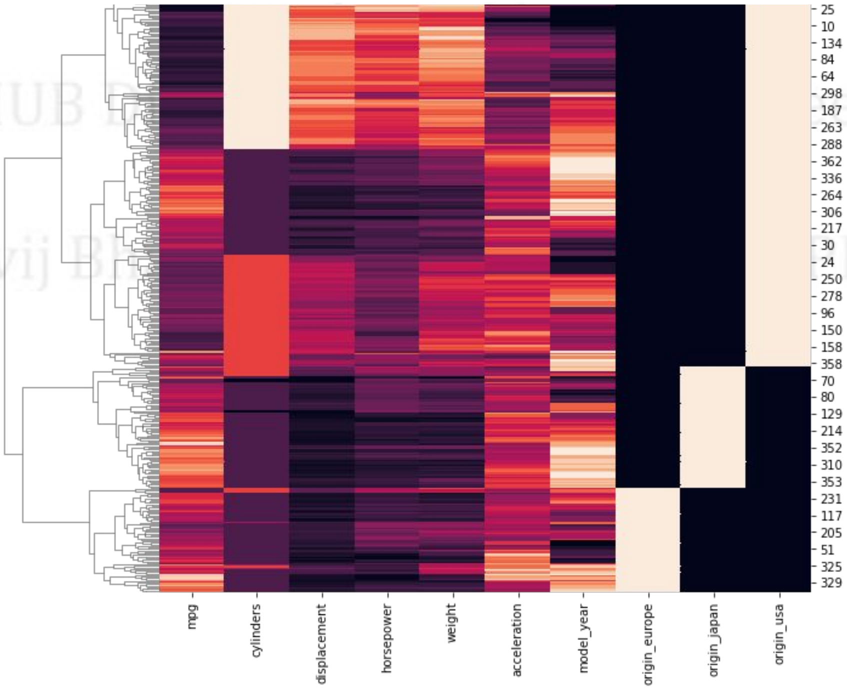


fig ref: pierian data

Topics which we still need to understand for Hierarchical Clustering:

- Similarity Metric
- Dendrogram
- Linkage Matrix

fig ref: pierian data

Similarity Metric

Measures distance between two points.

Many types:

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

fig ref: pierian data

Similarity Metric

Default choice is Euclidean

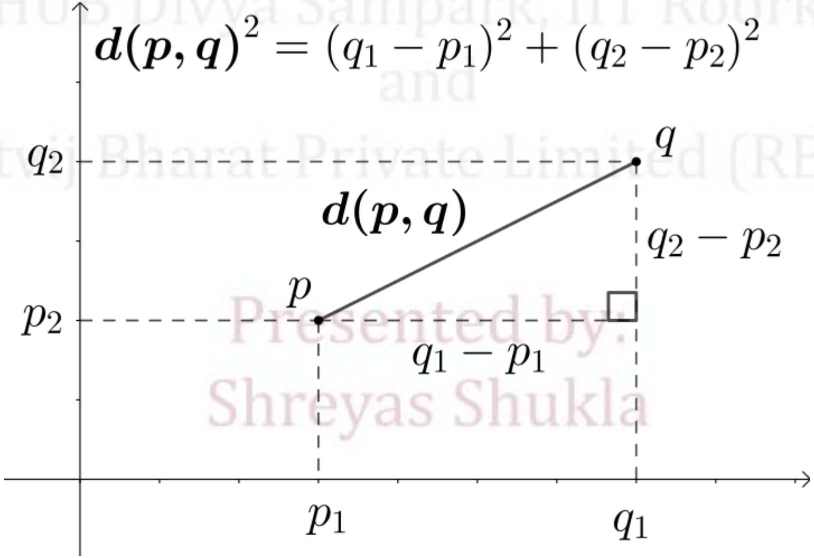


fig ref: pierian data

Similarity Metric

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

fig ref: pierian data

Similarity Metric

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

fig ref: pierian data

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.

fig ref: pierian data

Dendrogram:

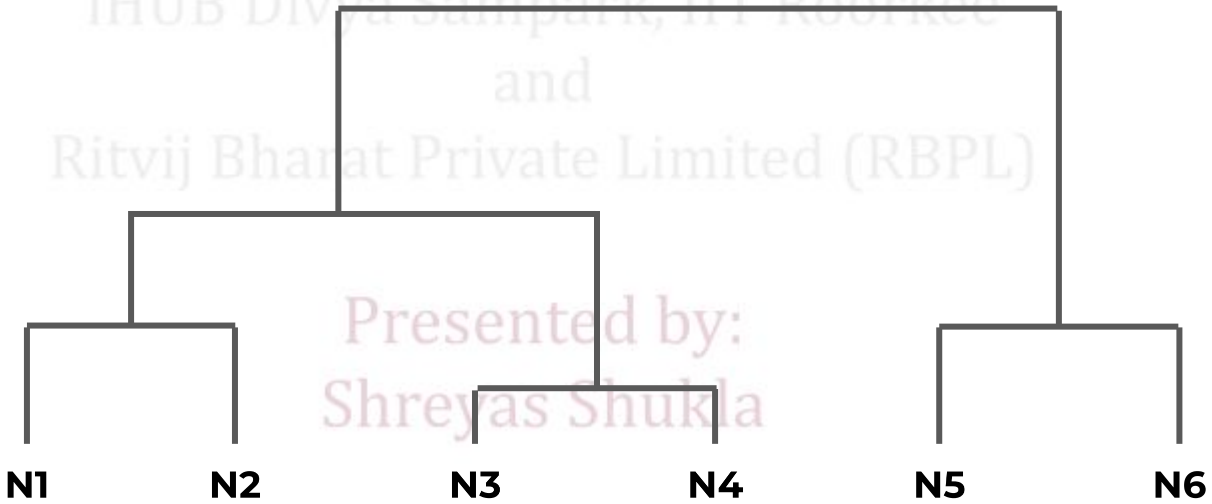


fig ref: pierian data

Dendrogram:

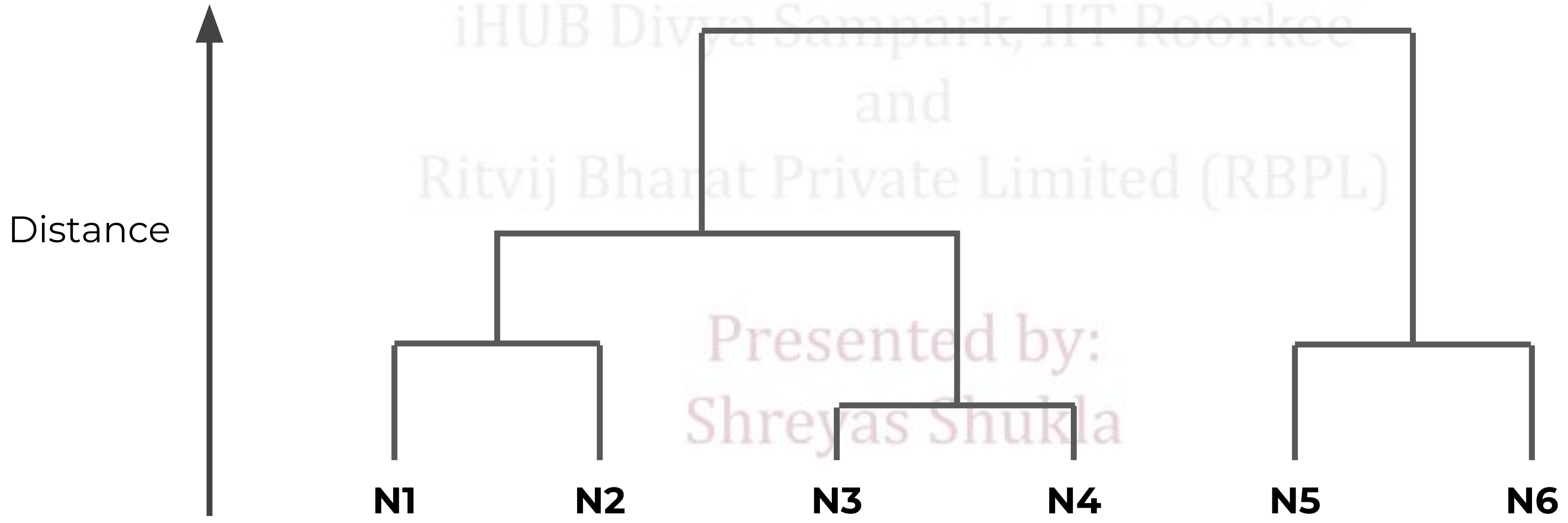


fig ref: pierian data

Dendrogram:

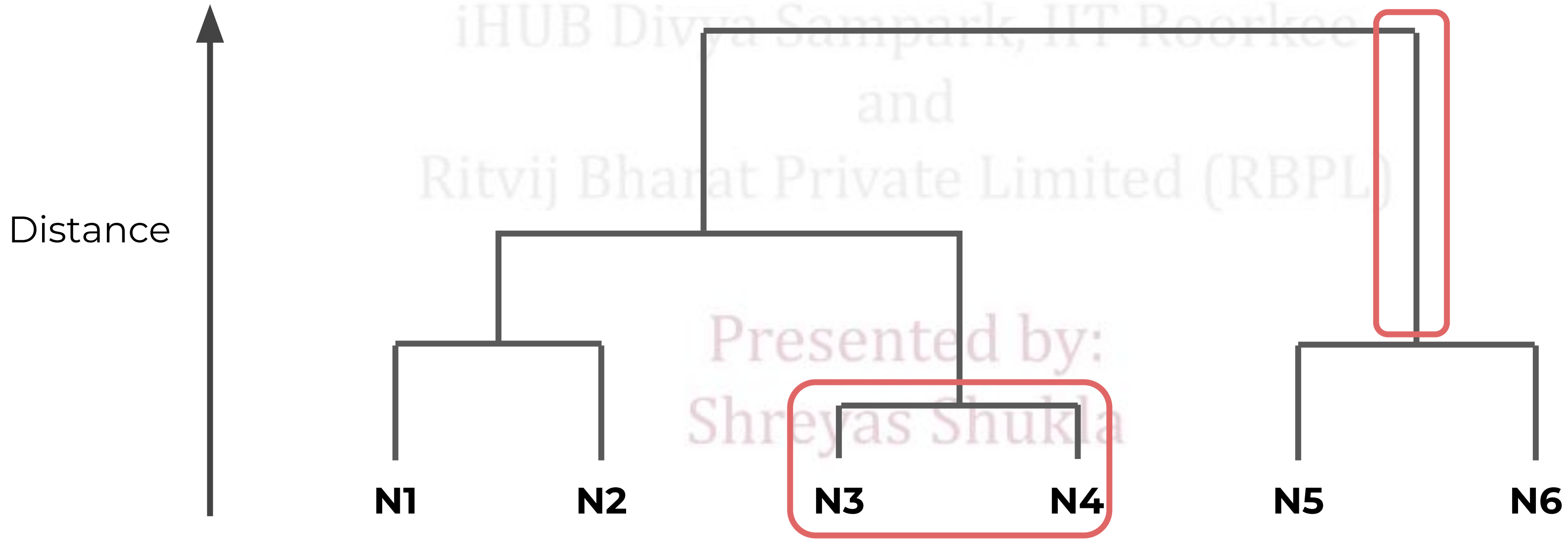


fig ref: pierian data

Dendrogram:

“Slice” to decide cluster count

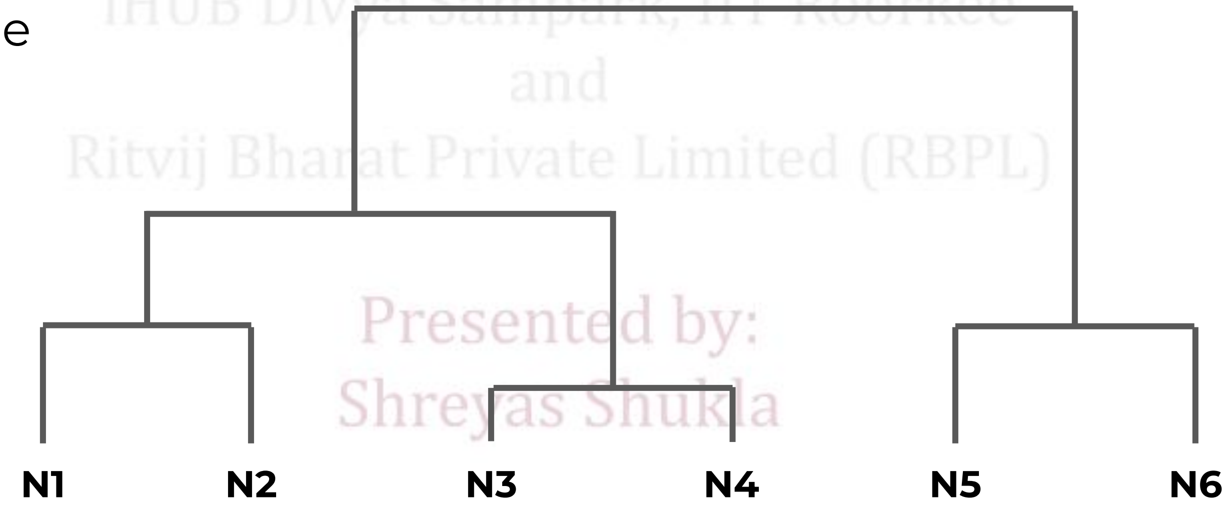


fig ref: pierian data

Dendrogram:

“Slice” to decide cluster count

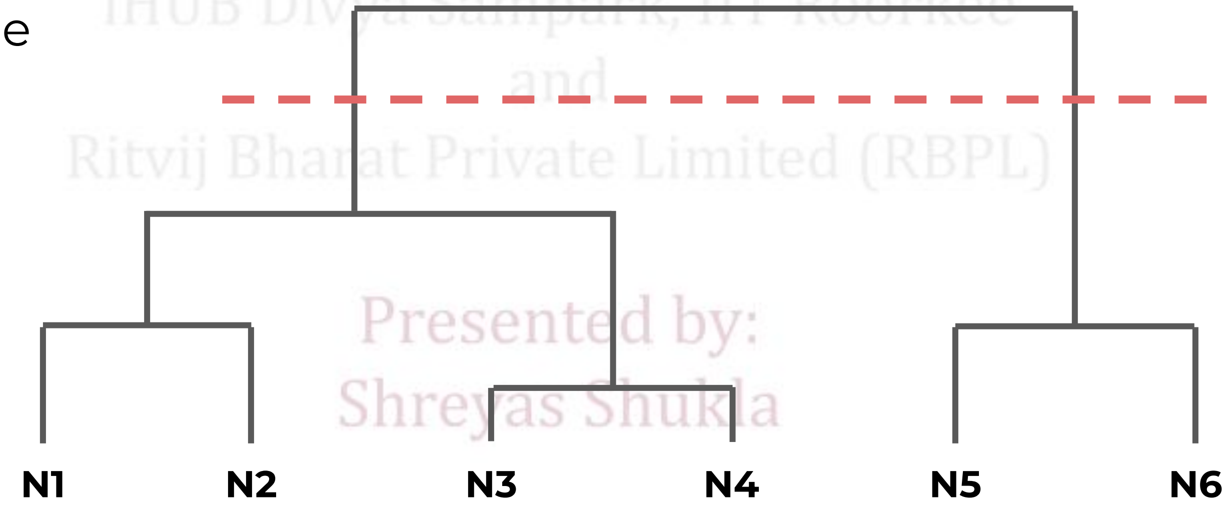
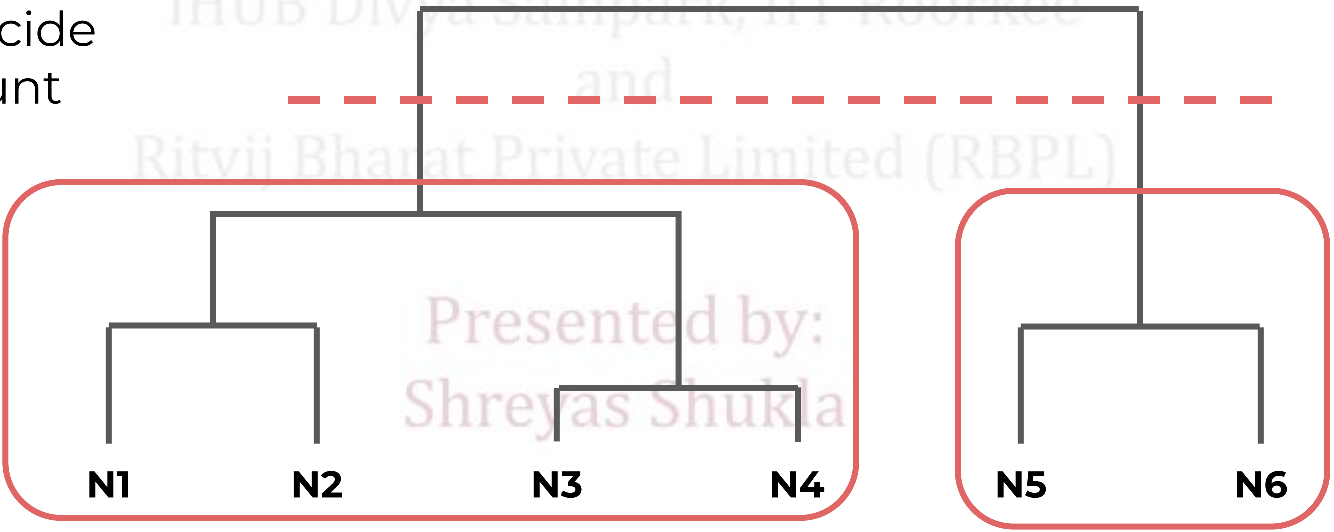


fig ref: pierian data

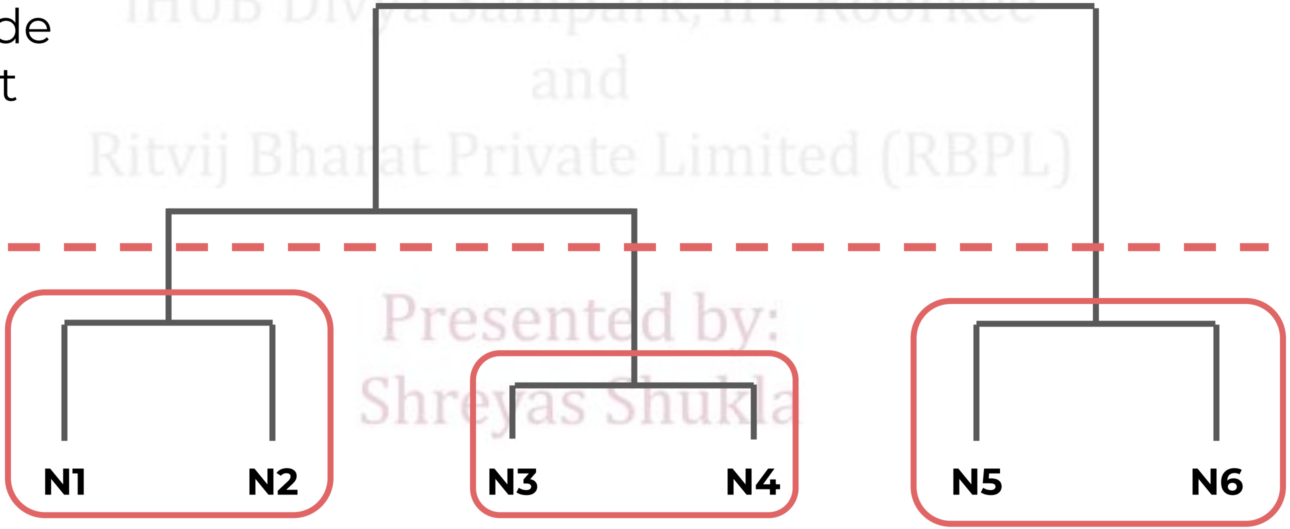
Dendrogram:

“Slice” to decide cluster count



Dendrogram:

“Slice” to decide cluster count



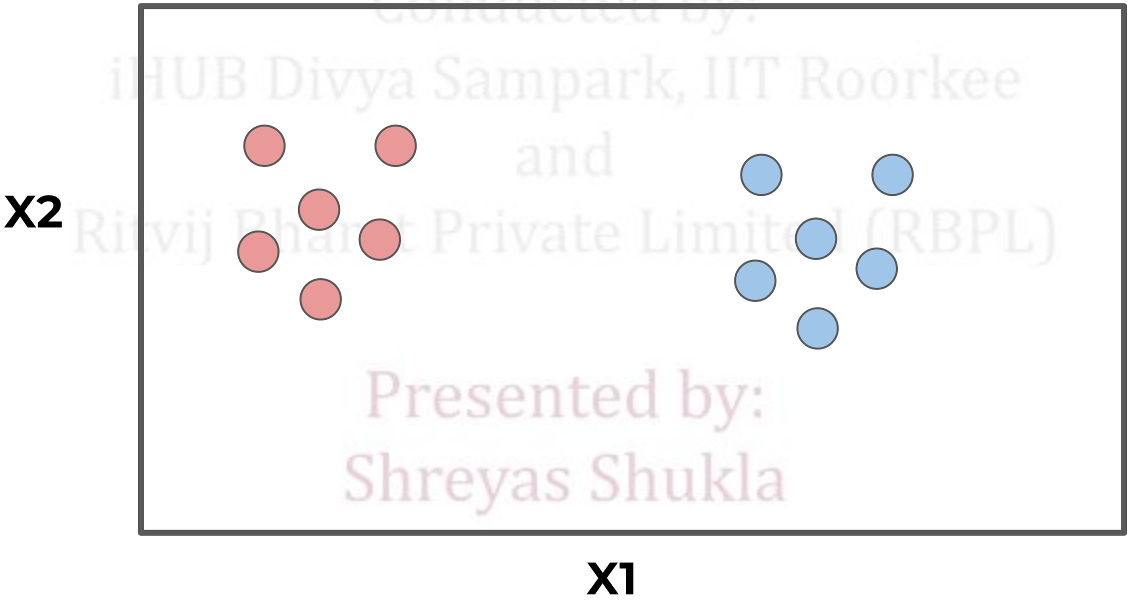
Linkage

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

fig ref: pierian data

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

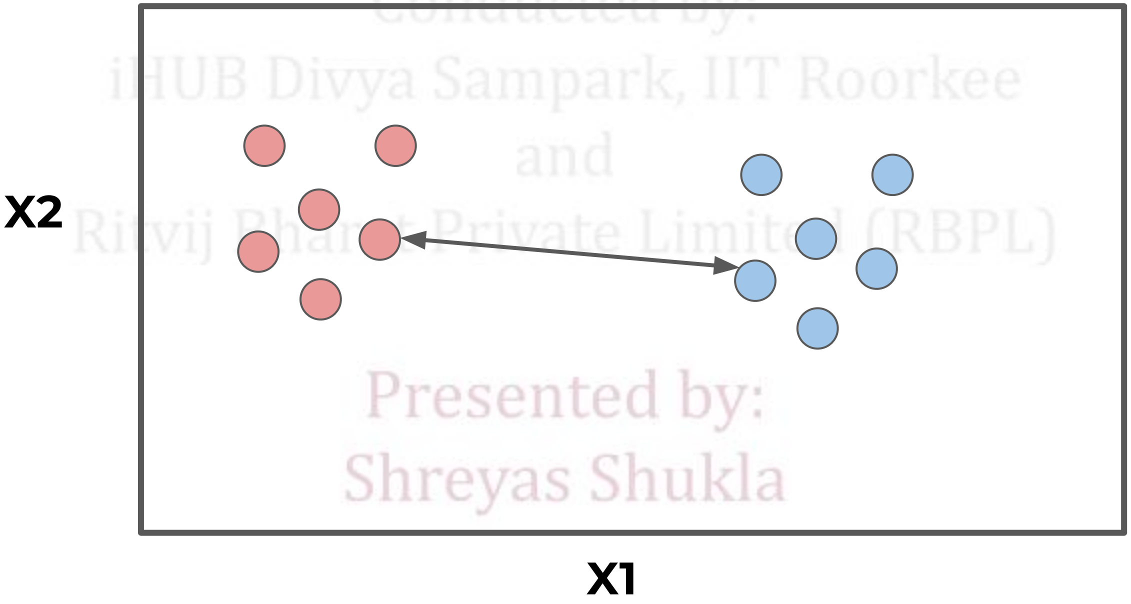


fig ref: pierian data

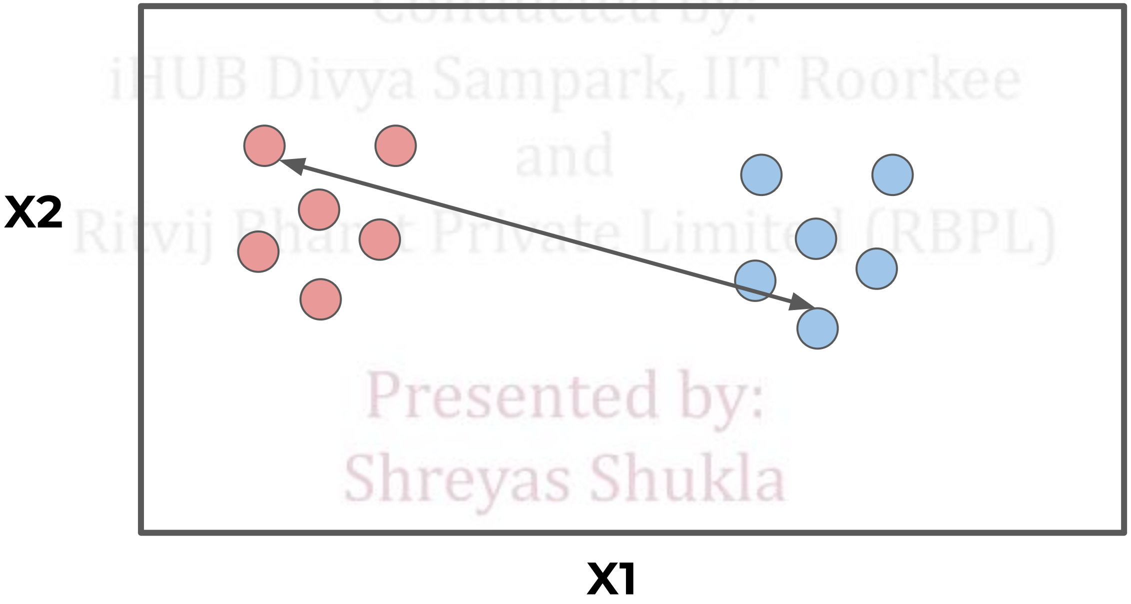


fig ref: pierian data

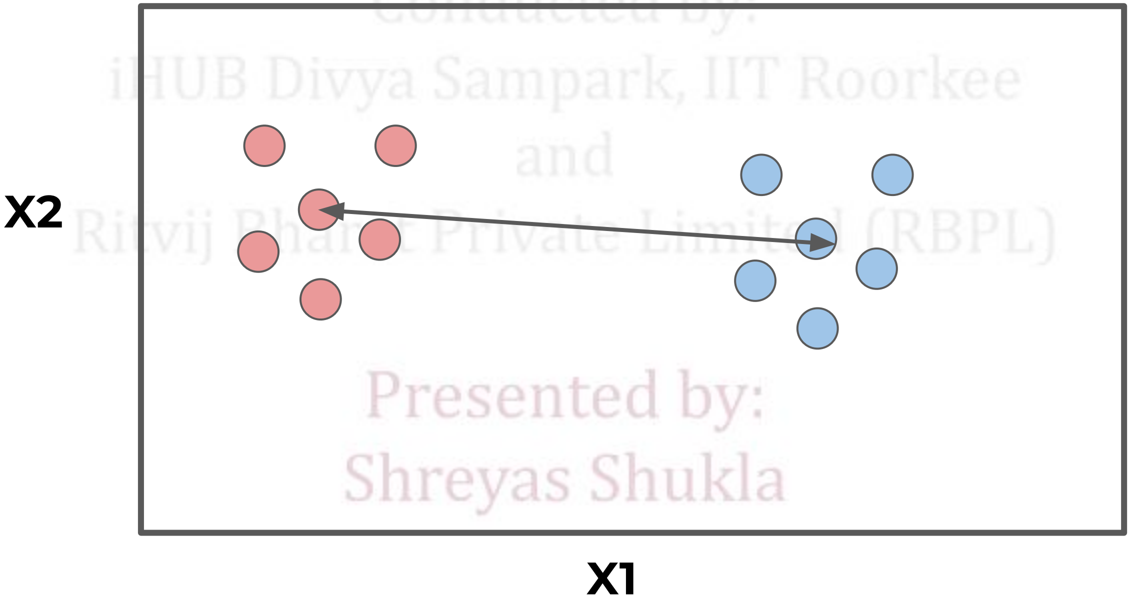


fig ref: pierian data

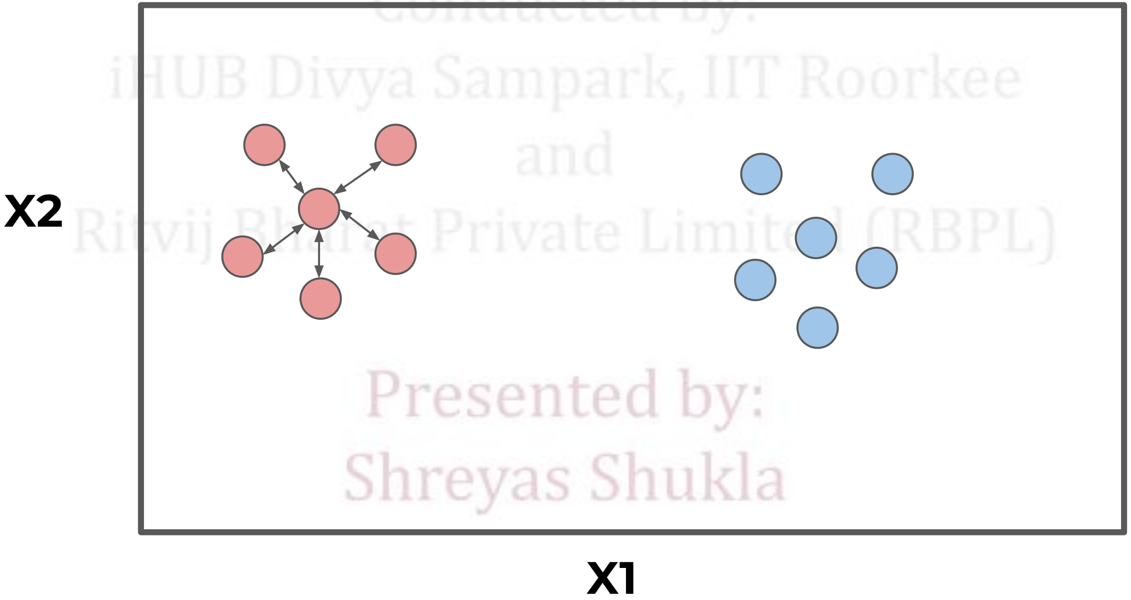


fig ref: pierian data

Linkage

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

fig ref: pierian data

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.

fig ref: pierian data

An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

Conducted by:
iHUB Divya Sampark, IIT Roorkee

Let’s code!!

and
Ritvij Bharat Private Limited (RBPL)

Presented by:
Shreyas Shukla

fig ref: pierian data