DSA IDEATHON

Project Title : **Single Linkage Clustering Of Dynamic Data**

Course Details : **DSA/CSL2020**

Instructor : **Suchetana Chakraborty**

Mentor : **Akanksha Dwivedi**

Team Members :

Jatavath Naveen Kumar(B22ME027)
Banoth Manohar(B22MT009)
Dadabada Sai Prasad(B22MT012)
Dhanavath Pavani(B22MT013)

INTRODUCTION

Single linkage clustering is a hierarchical clustering method that is often used to cluster data points based on their similarity or distance. In single linkage clustering, the distance between two clusters is defined as the shortest distance between any two points in the two clusters. This method is also known as minimum linkage clustering.

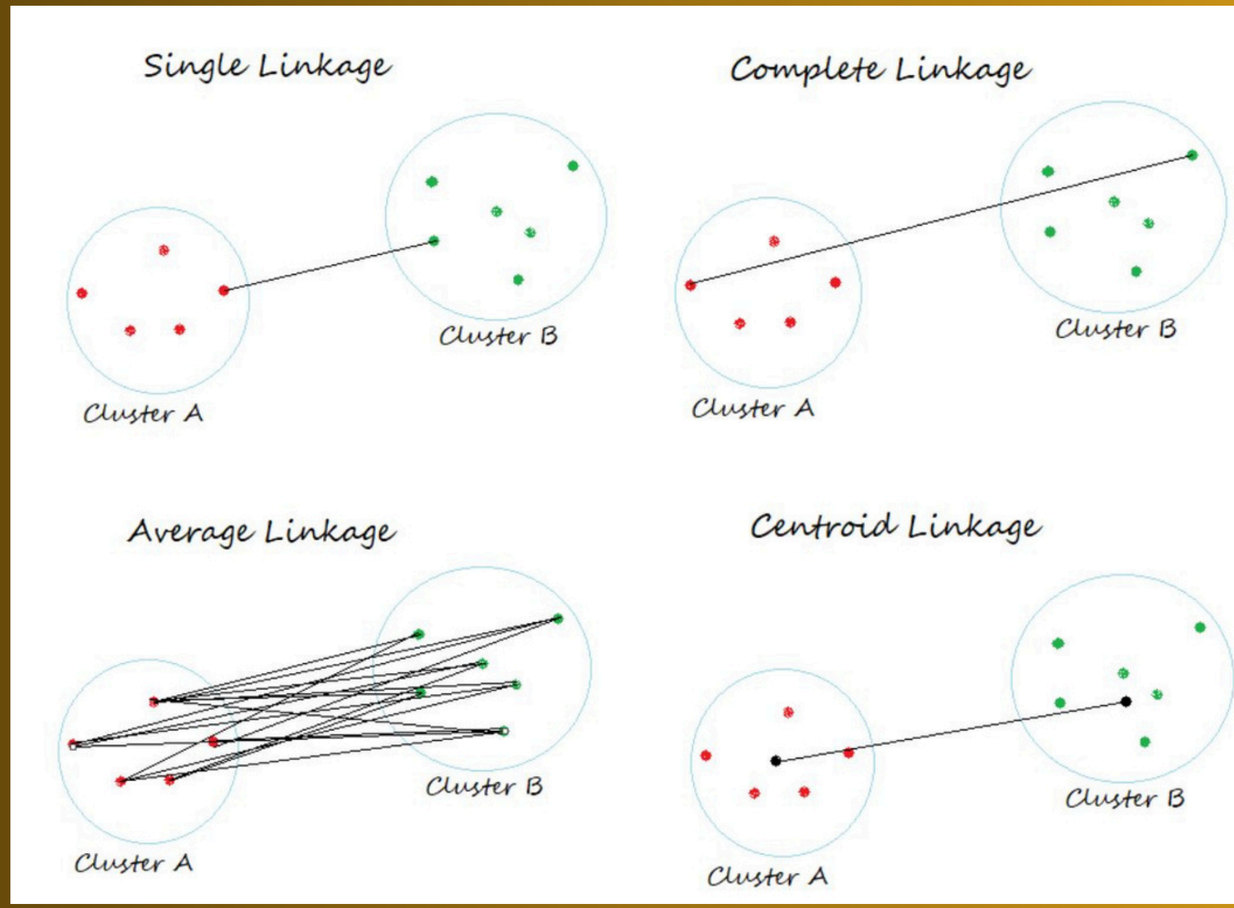
Main objectives

Implement Hierarchical Clustering: The project aims to implement the hierarchical clustering algorithm, which is a method of cluster analysis that builds a hierarchy of clusters. It starts by considering each data point as a separate cluster and then iteratively merges clusters based on their similarity until a predefined number of clusters or a stopping criterion is reached.

Clustering Algorithm: The project implements the core logic of the hierarchical clustering algorithm, including finding the nearest neighbors, merging clusters, and updating the cluster structure accordingly.

Why single linkage clustering?

- Simple Concept
- Computational Efficiency
- Flexibility in Cluster Shape
- No Predefined Number of Clusters



Handling Dynamic Data

- Dynamic data poses challenges as it evolves over time, requiring clustering algorithms to adapt to changing patterns.
- In dynamic data clustering, clusters may appear, disappear, split, or merge over time.
- Traditional clustering algorithms may not be directly applicable to dynamic data without modification.

Algorithms used for clustering

- Hierarchical Agglomerative Clustering (HAC) is a method used for hierarchical clustering, where clusters are formed iteratively by merging or agglomerating individual data points or existing clusters based on their similarity or distance.

Challenges and future direction

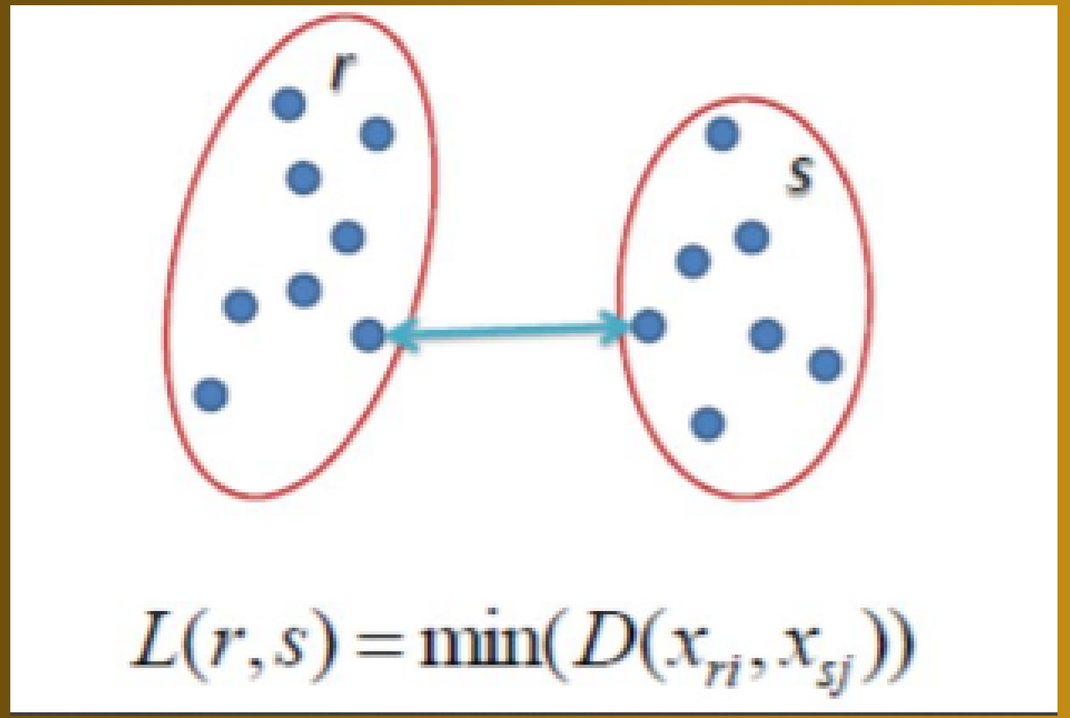
- Computational complexity: As the dataset grows and evolves, the computational cost of updating clusters and distance matrices may become prohibitive.
- Handling noise and outliers: Dynamic data often contains noise and outliers that can affect clustering results. Robust methods need to be developed to handle these challenges.

Agglomerative Method:

Given:

A set of objects {x1,.....xn }
A distance function dist(c1,c2)
for i = 1 to n
    ci ={xi}
end for
C = {c1,.....,cn}
l = n +1

While C.size >1 **do**
    __ (cmin1,cmin2) = minimum dist(ci,cj) for ci,cj in C
    __ remove cmin1 and c min2 from C
    __ add {cmin1,cmin2} to C
    __ l = l +1
end while



Mathematical Approach:

Finding Clusters using single Linkage technique

Step 1: Compute the distance matrix by:

$$d[(x,y)(a,b)] = \sqrt{(x-a)^2 + (y-b)^2}$$

The distance Matrix will be:

$$\begin{pmatrix} & P1 & P2 & P3 & P4 & P5 & P6 \\ P1 & 0 & & & & & \\ P2 & 0.23 & 0 & & & & \\ P3 & & & 0 & & & \\ P4 & & & & 0 & & \\ P5 & & & & & 0 & \\ P6 & & & & & & 0 \end{pmatrix}$$

Sample No.	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Similarly, finding the Euclidean distance for every point.

Updated Distance matrix will be:

$$\begin{pmatrix} & P1 & P2 & P3 & P4 & P5 & P6 \\ P1 & 0 & & & & & \\ P2 & 0.23 & 0 & & & & \\ P3 & 0.22 & 0.14 & 0 & & & \\ P4 & 0.37 & 0.19 & 0.13 & 0 & & \\ P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 & \\ P6 & 0.24 & 0.24 & 0.10 & 0.22 & 0.39 & 0 \end{pmatrix}$$

Step 2: Merging the two closest members of the two clusters and finding the minimum element in distance matrix.

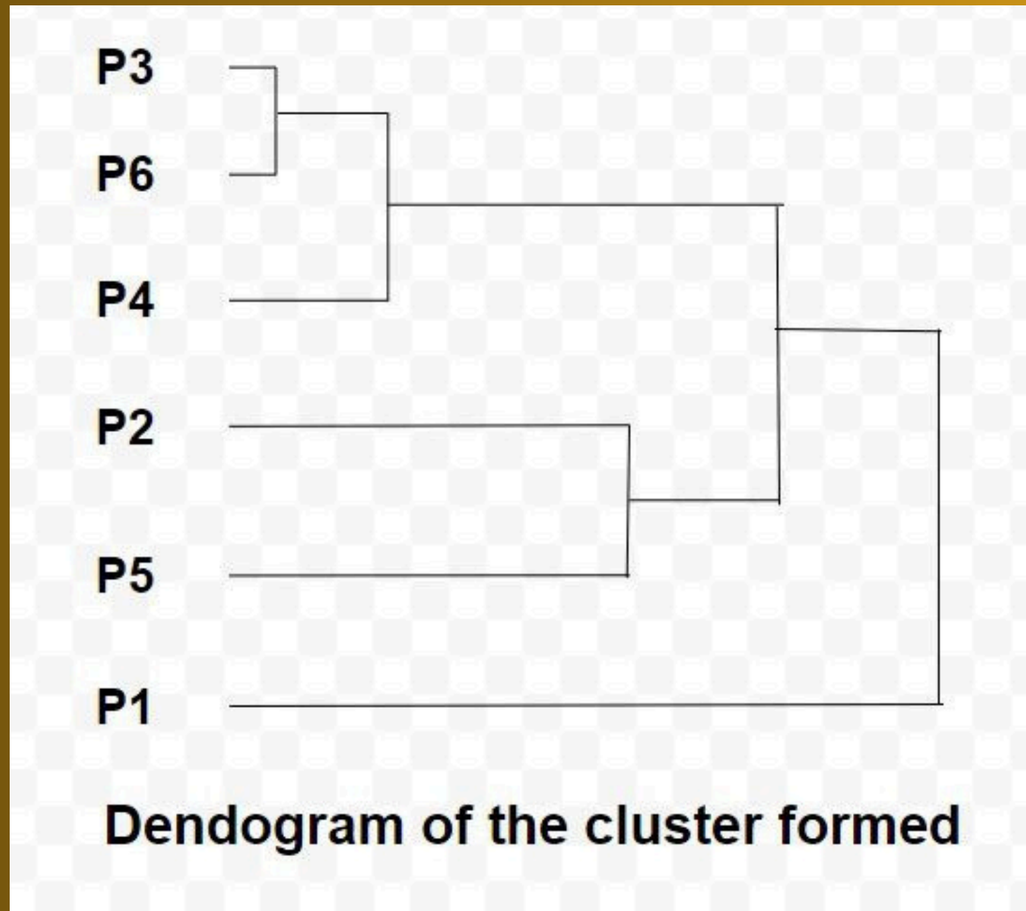
The updated distance matrix will be:

$$\begin{pmatrix} & P1 & P2 & P3, P6 & P4 & P5 \\ P1 & 0 & & & & \\ P2 & 0.23 & 0 & & & \\ P3, P6 & 0.22 & 0.14 & 0 & & \\ P4 & 0.37 & 0.19 & 0.13 & 0 & \\ P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} & P1 & P2 & P3, P6, P4 & P5 \\ P1 & 0 & & & \\ P2 & 0.23 & 0 & & \\ P3, P6, P4 & 0.22 & 0.14 & 0 & \\ P5 & 0.34 & 0.14 & 0.23 & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} & P1 & P2, P5 & P3, P6, P4 \\ P1 & 0 & & \\ P2, P5 & 0.23 & 0 & \\ P3, P6, P4 & 0.22 & 0.14 & 0 \end{pmatrix}$$

Final Matrix:

$$\begin{pmatrix} & P1 & P2, P5, P3, P6, P4 \\ P1 & 0 & \\ P2, P5, P3, P6, P4 & 0.22 & 0 \end{pmatrix}$$

We have reached the final solution.The dendogram related to that is:



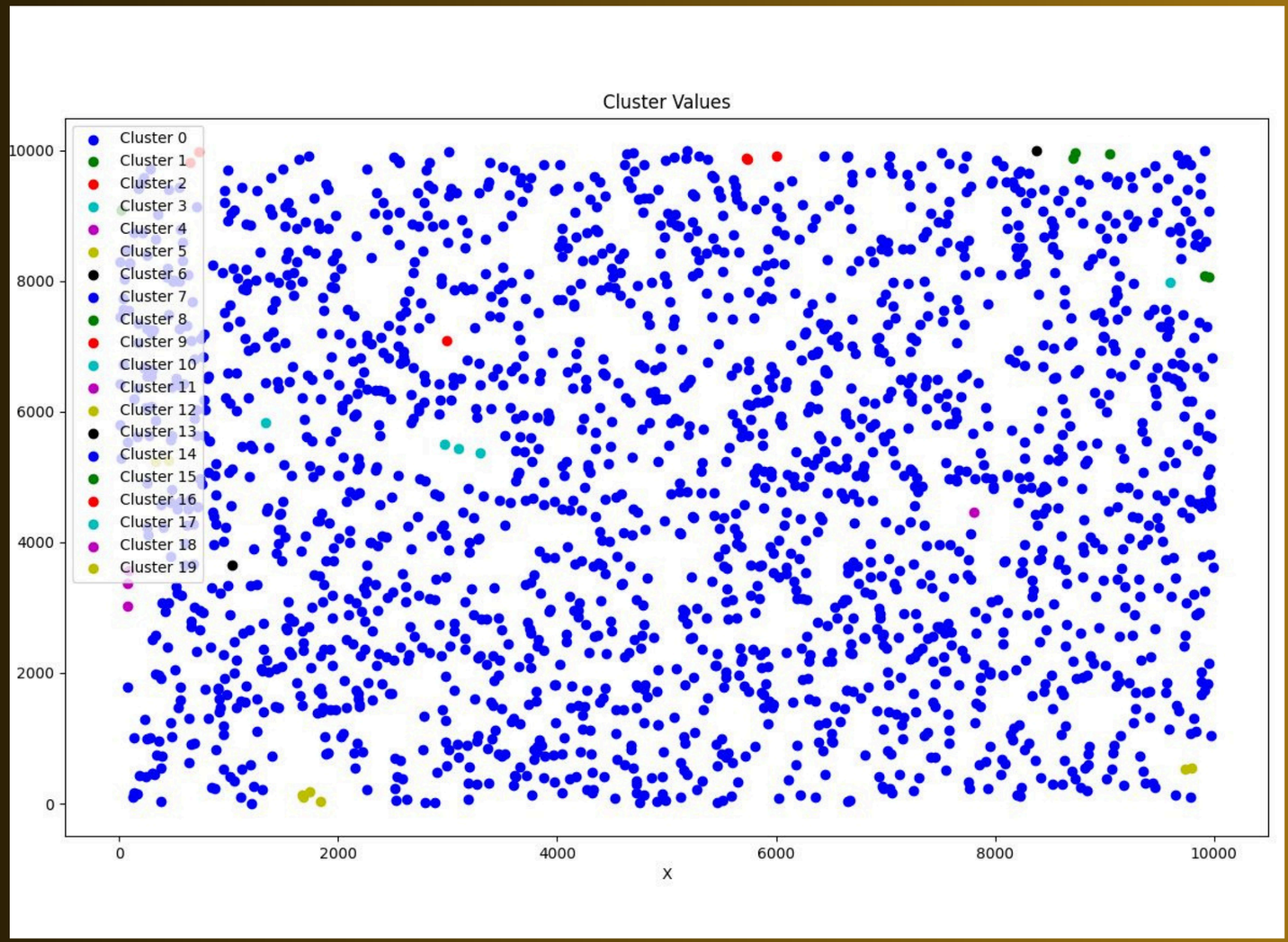
Dendogram of the cluster formed

What we have done:

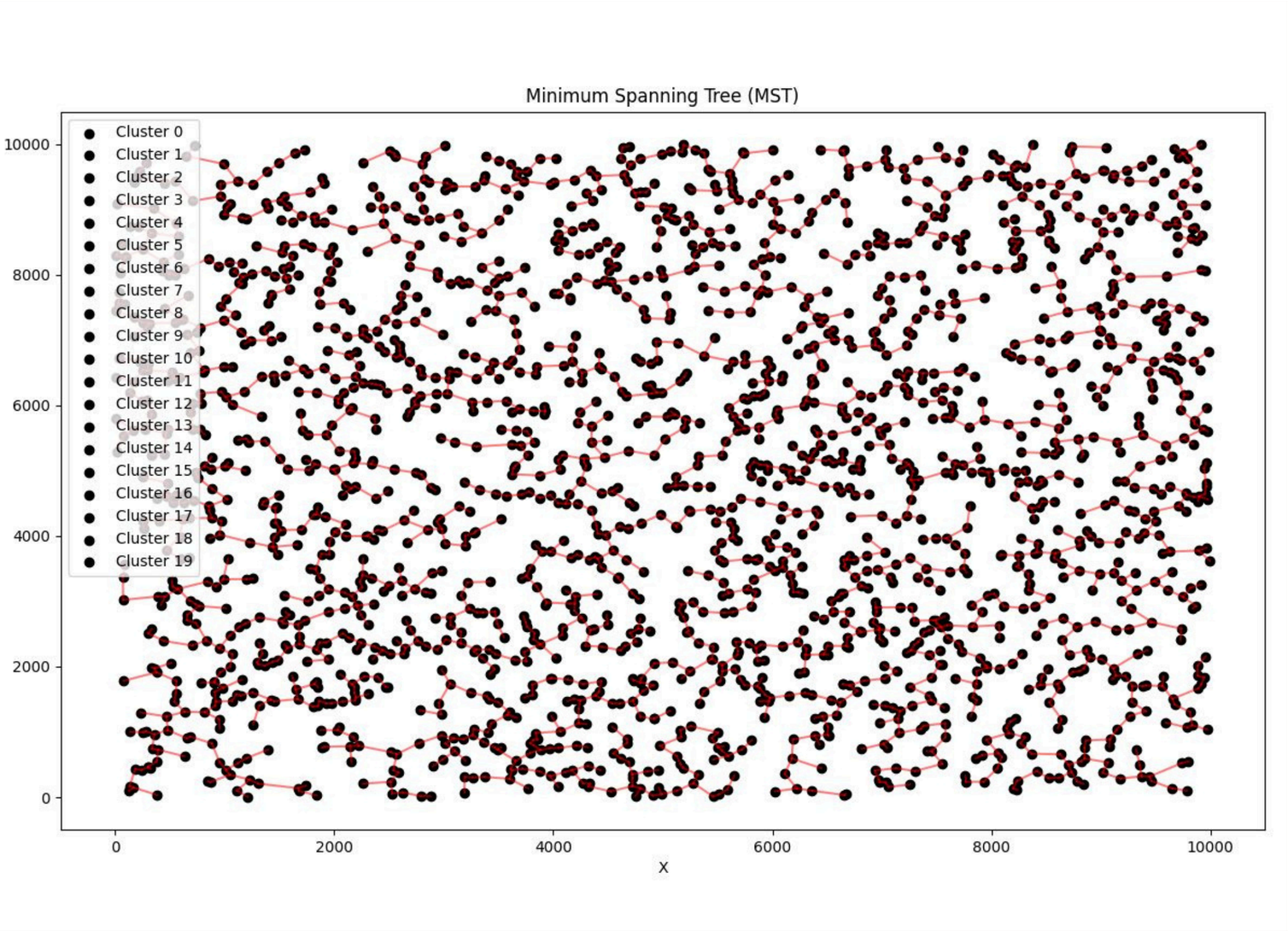
- Initialization and Loading Data.
- Defined Structures and Constants.
- Defined the obj_t and cluster_t structures.
- Defined the CLUSTER_CHUNK constant.
- Implemented functions for initializing clusters (init_cluster), resizing clusters. (resize_cluster), and clearing clusters (clear_cluster).
- File I/O and Data Loading.
- Implemented a function to parse command-line arguments (parse_args).
- Defined Clustering Logic

We implemented our code on Faker Dataset.(https://faker.readthedocs.io/en/master/)

The number of samples we have used is 2000. And the Clusters to be formed is 20.Also we have visvualized the plot.



Minimum Spanning Tree:



Contribution:

In the project, Naveen (B22ME027) and Sai Prasad (B22MT012) excelled in coding and debugging, ensuring algorithm precision. Pavani (B22MT012) and Manohar (B22MT009) handled presentation and report duties, crafting a compelling narrative. Together, our diverse strengths showcased effective teamwork, ensuring project success and comprehensive documentation for future reference.

References:

https://github.com/vsaltxx/single-linkage-clustering.git

https://www.slideserve.com/genemiller/efficient-algorithms-for-non-parametric-clustering-with-clutter-powerpoint-ppt-presentation

https://www.geeksforgeeks.org/agglomerative-methods-in-machine-learning/

Conclusion:

Our project offers a functional implementation of hierarchical clustering with error handling and memory management. It serves as a basis for further enhancements or customization according to specific clustering requirements or application needs. With improvements and additional features, it could be extended for a wide range of clustering tasks.

THANK YOU