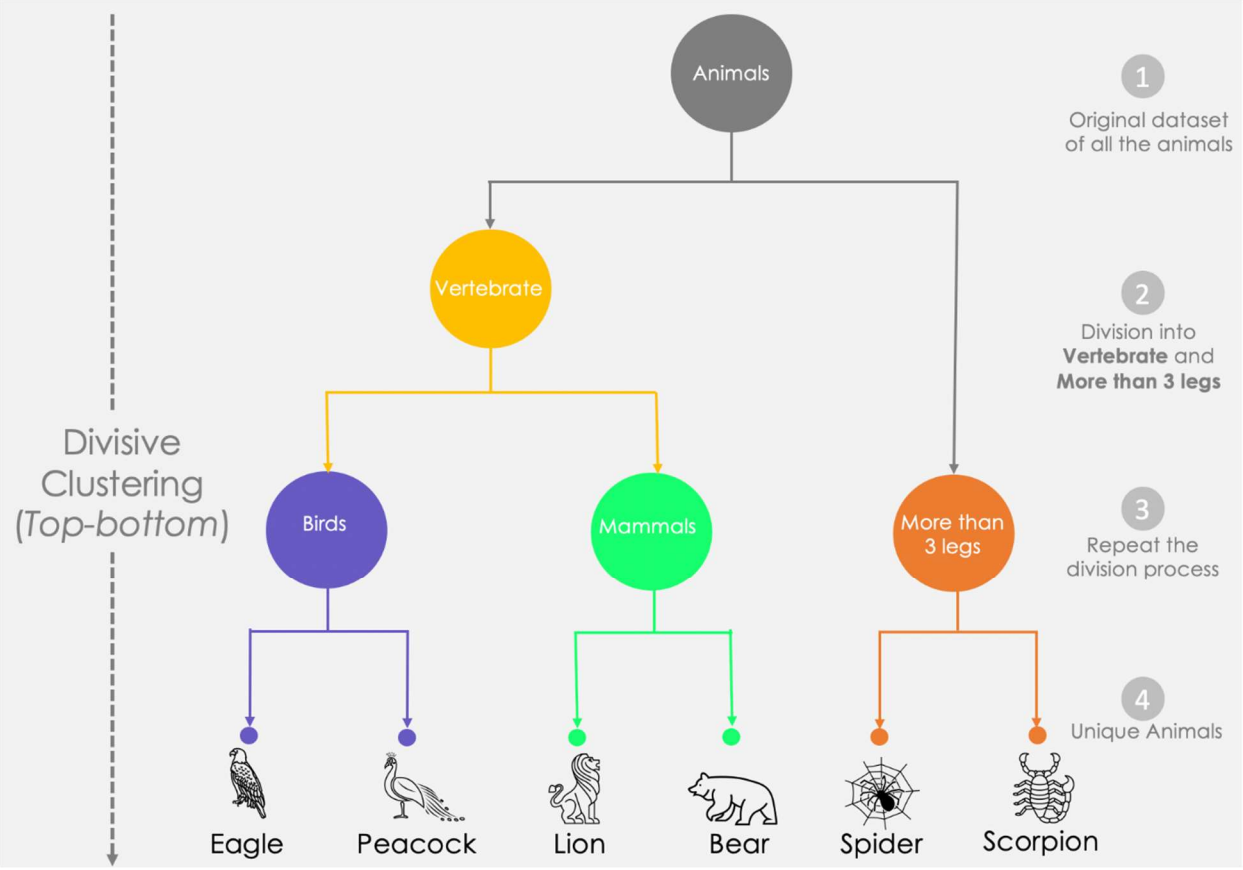
Divisive Hierarchical Clustering Algorithm

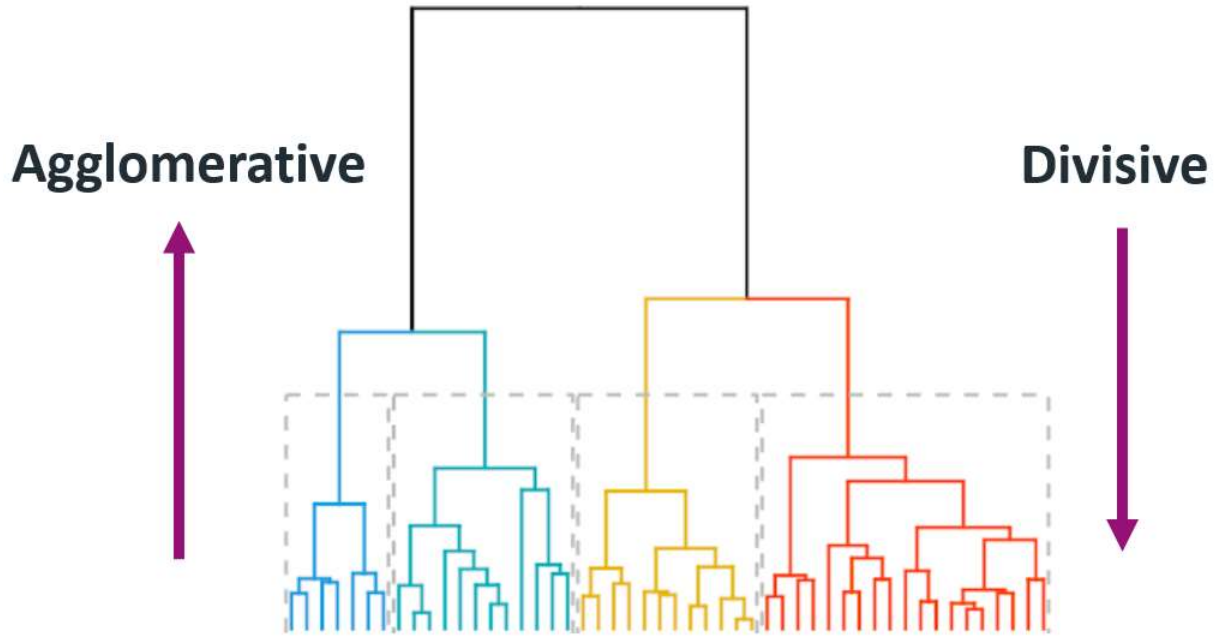
Clustering algorithms are vital tools in data analysis and machine learning, allowing us to find inherent structures within datasets and group data points with similar characteristics. Among various clustering techniques, hierarchical clustering plays a crucial role in understanding the relationships and patterns within the data. One of the fundamental approaches in hierarchical clustering is the divisive hierarchical clustering algorithm. In this article, we will delve into the intricacies of the divisive hierarchical clustering algorithm, exploring its working principles, advantages, and potential applications.



An Overview of Divisive Hierarchical Clustering

Divisive hierarchical clustering is a top-down approach where the entire dataset is treated as a single cluster, and then it is recursively divided into smaller clusters. The algorithm starts with all data points in one cluster and then splits the cluster into smaller subclusters until each data point is in its own individual cluster. This process creates a binary tree or dendrogram that represents

the hierarchy of clusters.



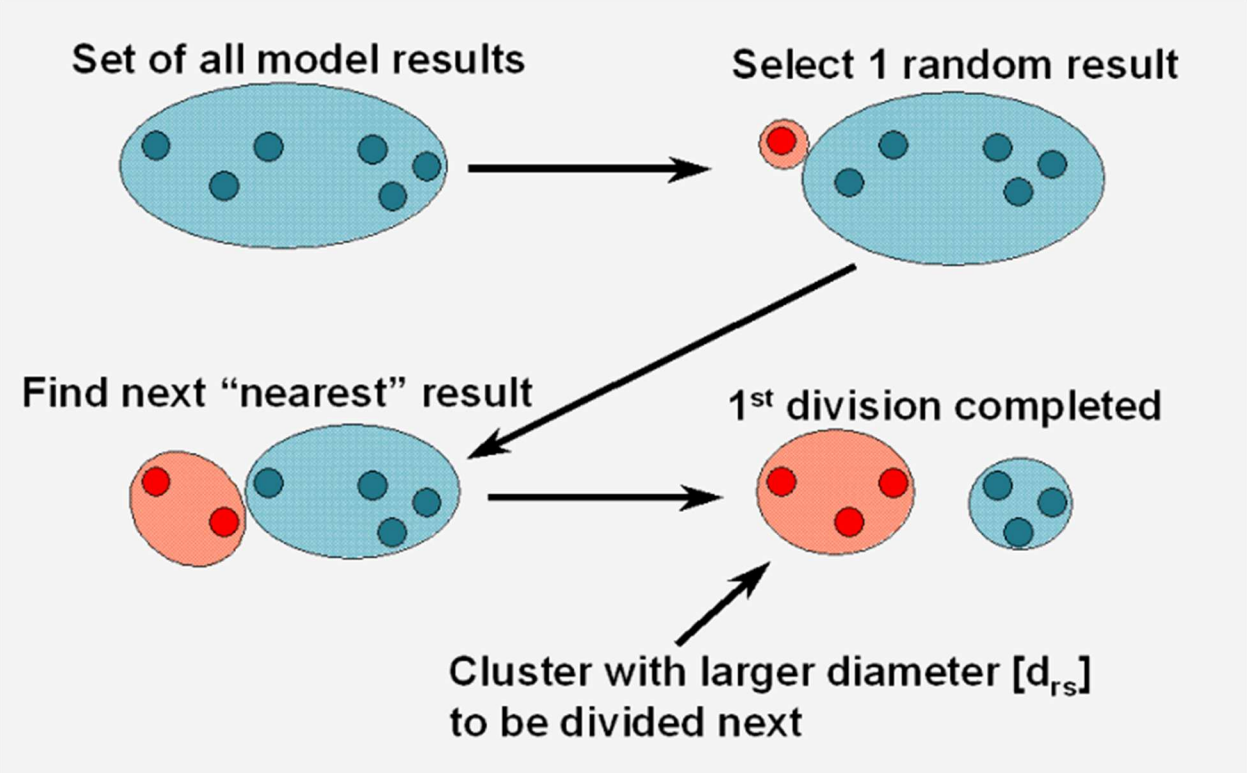
Algorithm Workflow

1. **Initialization**: Begin with the entire dataset as one cluster.
2. **Splitting Process**: Identify the clusters that can be separated to maximize the distance between the resulting clusters.
3. **Recursion**: Recursively apply the splitting process until each data point is in its own cluster.
4. **Dendrogram Construction**: Construct a dendrogram to represent the hierarchical relationships between clusters.

$t = 0$
Choose $R_0 = X$ as initial clustering
Repeat
    t=t+1
    For i = 1 to t,
        Among all possible pairs of clusters (C_r, C_s) that form a clustering $C_{t-1,i}$
        find the pair $(C^1_{t-1,i}, C^2_{t-1,i})$ with the largest dissimilarity.
    End
    From the t pairs of clusters defined at the previous stage, choose the one
    with the largest dissimilarity
    Let this pair be $C^1_{t-1,j}, C^2_{t-1,j}$
    Define the new clustering $R_t = R_{t-1} - C_{t-1,j} \cup (C^1_{t-1,j}, C^2_{t-1,j})$
    Relabel the clusters
Until each vector forms its own cluster

Calculating Cluster Separation

The divisive hierarchical clustering algorithm employs various methods to determine the optimal clusters to separate. One common approach is to measure the dissimilarity or distance between clusters. The choice of distance metric, such as Euclidean distance or Manhattan distance, significantly impacts the clustering results. The algorithm aims to find the clusters with the maximum dissimilarity to split them into distinct subclusters.



Complexity Analysis

The divisive hierarchical clustering algorithm's time complexity is often high, especially for large datasets. Since it involves a recursive process and the computation of distances between clusters, the algorithm's time complexity is typically O(n^3), making it less efficient for extensive datasets.

Algorithm	***Time Complexity***
Agglomerative hierarchical methods [16]	$O(n^2)$
Divisive Hierarchical methods [16]	$O(n^2)$
Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [17]	$O(n)$
Clustering Using REpresentatives (CURE) [18]	$O(n^2)$
RObust Clustering using (ROCK) [19]	$O(n^2)+nm$ $mma +n2logn)$
Performance guarantees for hierarchical clustering [20]	$O(n^2)$

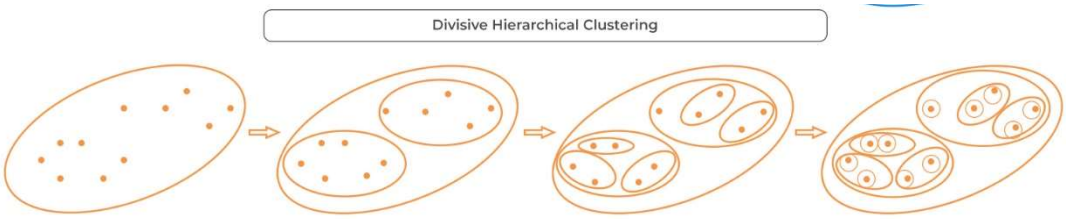
Advantages and Limitations

Advantages

1. **Hierarchical Structure**: Divisive hierarchical clustering provides a clear hierarchical structure, enabling users to analyze clusters at various levels of granularity.
2. **Dendrogram Visualization**: The algorithm generates dendrograms, which are valuable tools for understanding the hierarchical relationships between clusters.
3. **Ease of Interpretation**: The hierarchical nature of the algorithm makes it intuitive and easy to interpret, even for users without an in-depth understanding of clustering techniques.

Limitations

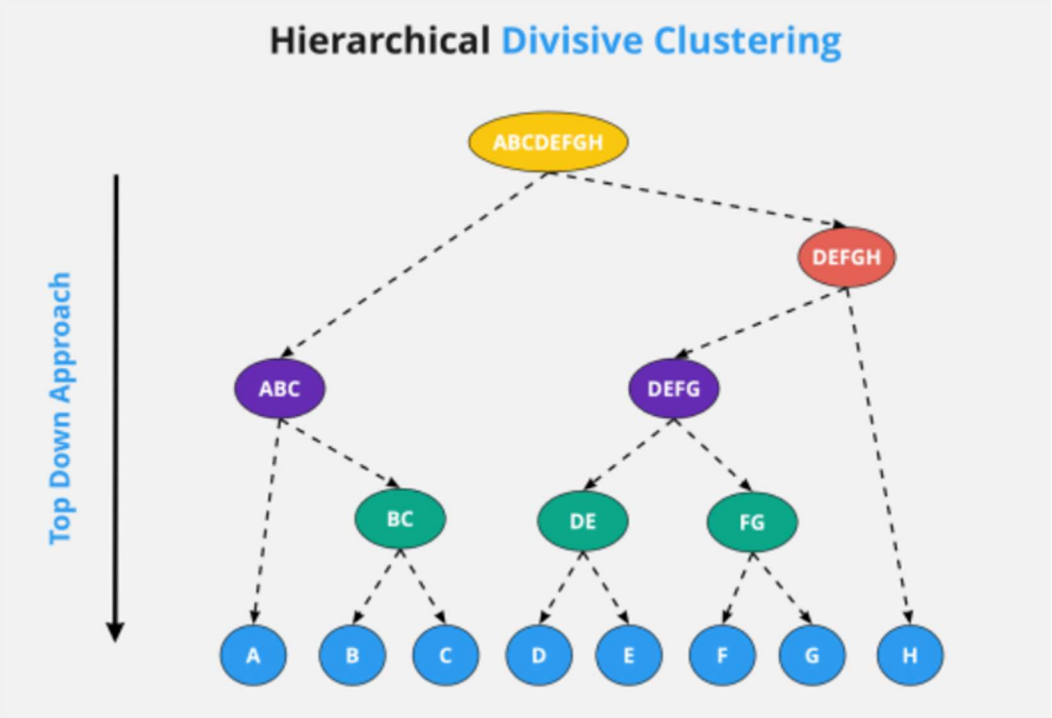
1. **High Computational Complexity**: Divisive hierarchical clustering can be computationally expensive, especially for large datasets, making it less practical for real-time analysis.
2. **Sensitivity to Noise**: The algorithm's performance may deteriorate in the presence of noisy data, leading to inaccurate cluster divisions.
3. **Difficulty in Handling Large Datasets**: As the algorithm's complexity grows with the dataset size, it becomes challenging to handle extensive datasets efficiently.



Applications of Divisive Hierarchical Clustering

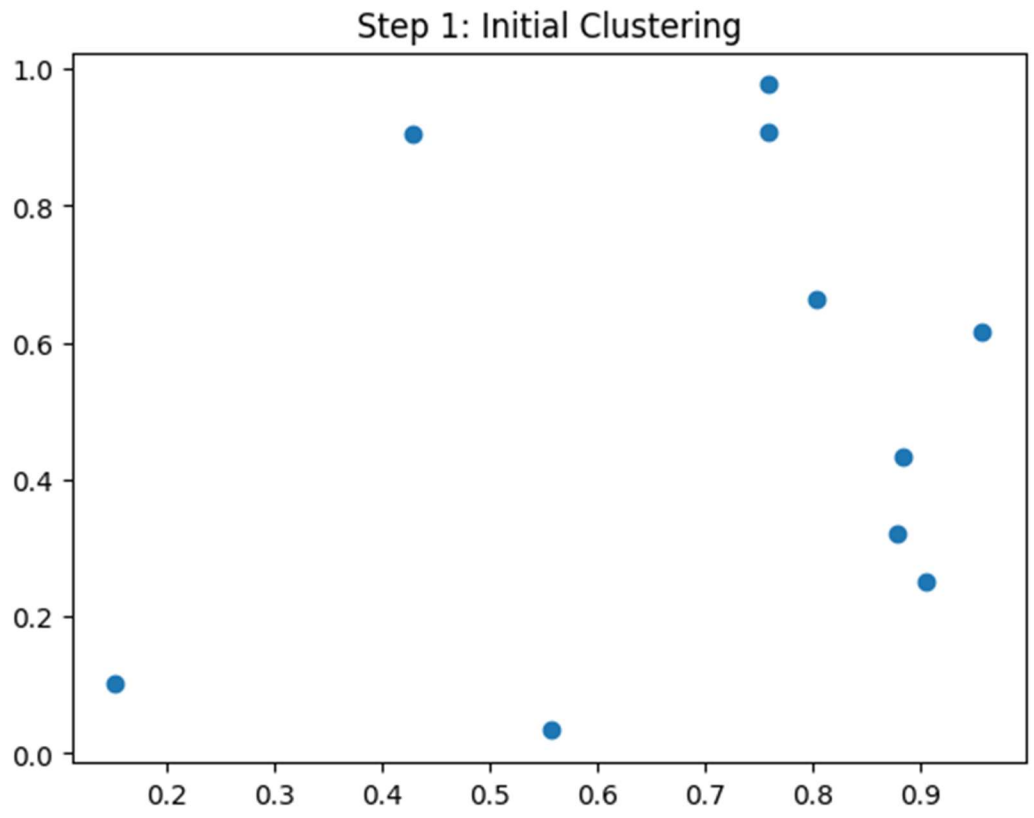
Divisive hierarchical clustering finds applications in various domains, including:

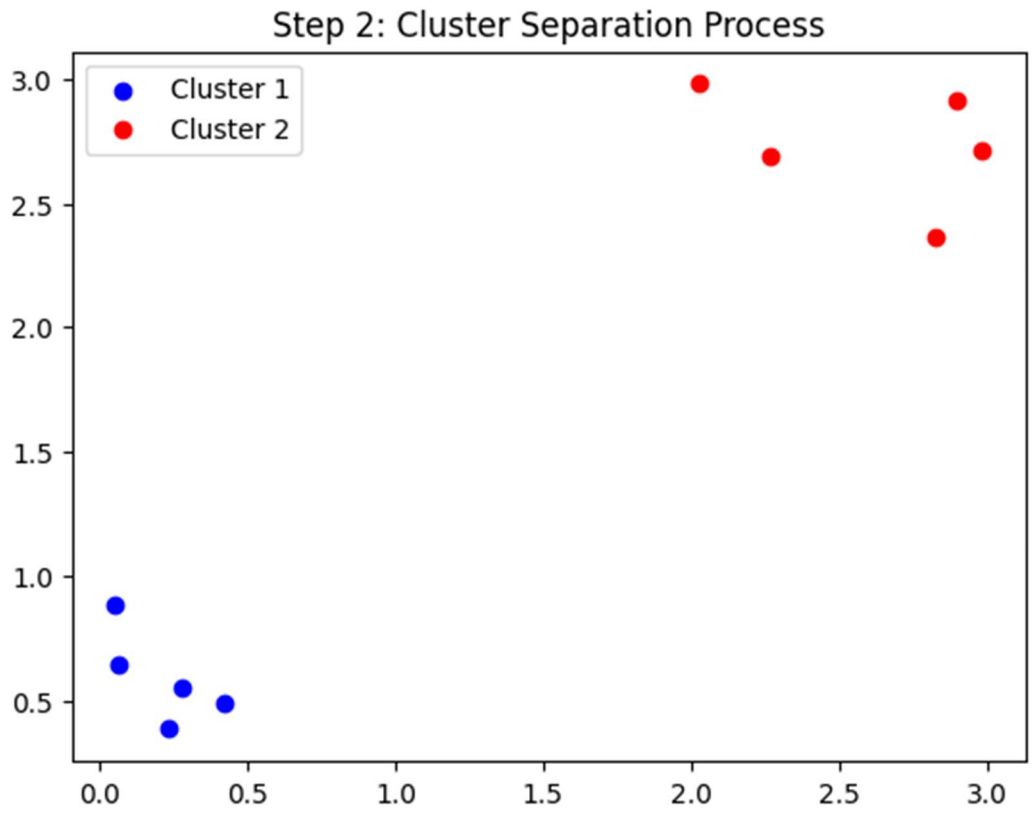
1. **Biology**: Analyzing gene expression data and identifying biological relationships among different species.
2. **Market Segmentation**: Grouping customers based on purchasing behavior for targeted marketing strategies.
3. **Image Segmentation**: Partitioning images into meaningful regions for image analysis and understanding.

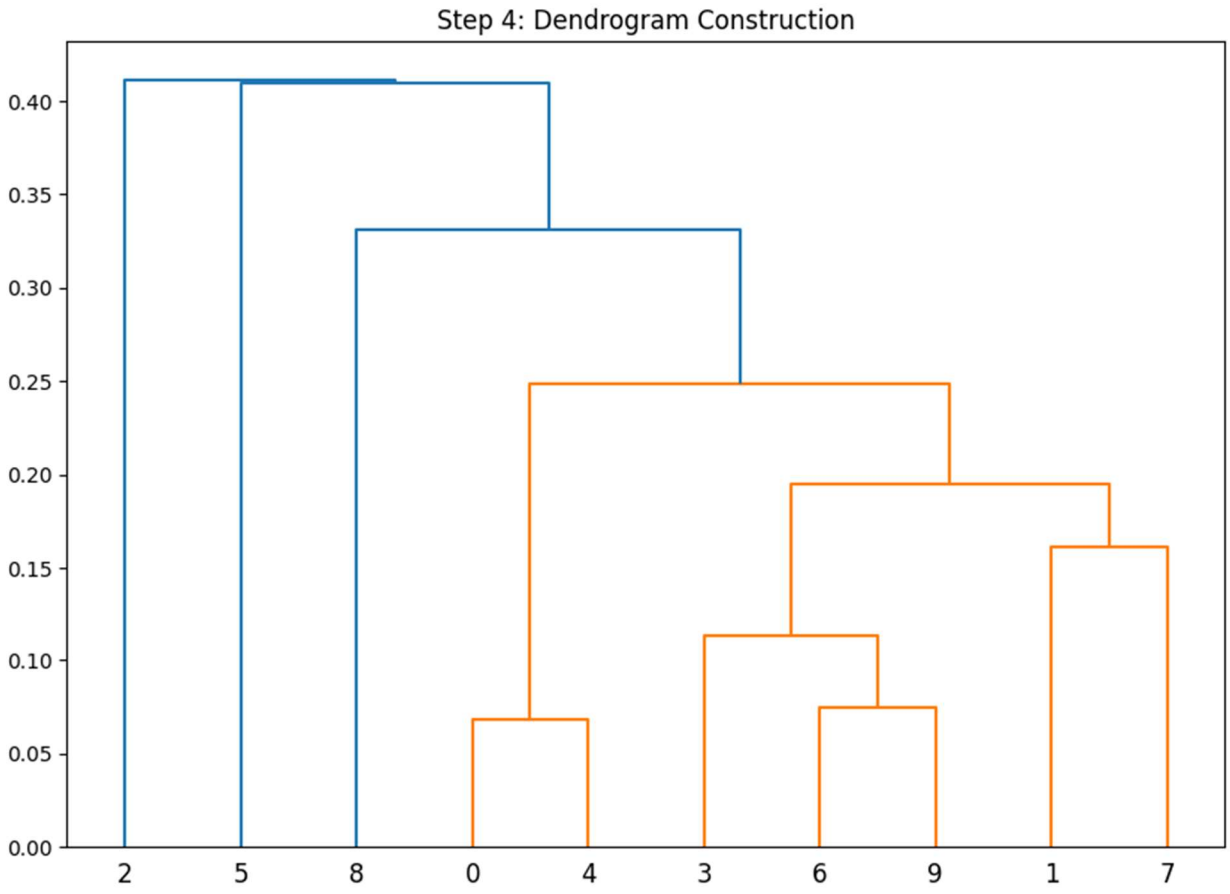


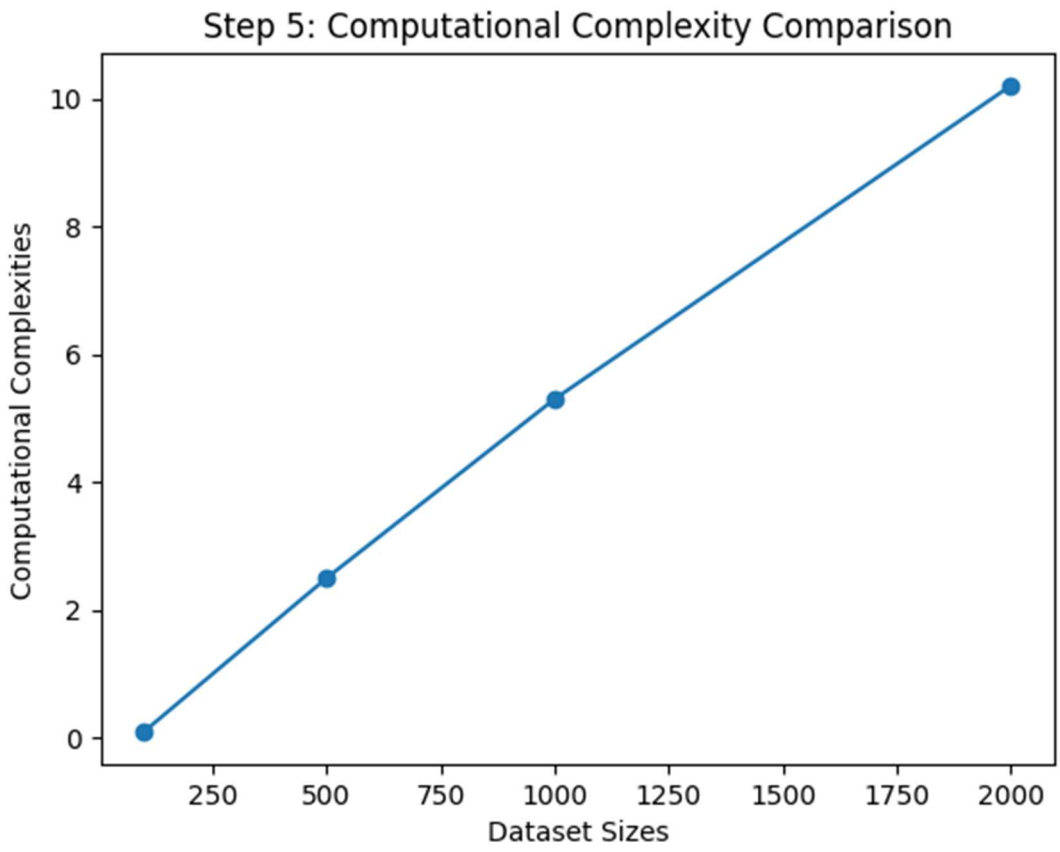
Code

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage
data_points = np.random.rand(10, 2)
plt.scatter(data_points[:, 0], data_points[:, 1])
plt.title("Step 1: Initial Clustering")
plt.show()
cluster1 = np.random.rand(5, 2)
cluster2 = np.random.rand(5, 2) + 2
plt.scatter(cluster1[:, 0], cluster1[:, 1], color='b', label='Cluster 1')
plt.scatter(cluster2[:, 0], cluster2[:, 1], color='r', label='Cluster 2')
plt.title('Step 2: Cluster Separation Process')
plt.legend()
```


Step 1: Initial Clustering
1.0
0.8
0.6
0.4
0.2
0.0
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

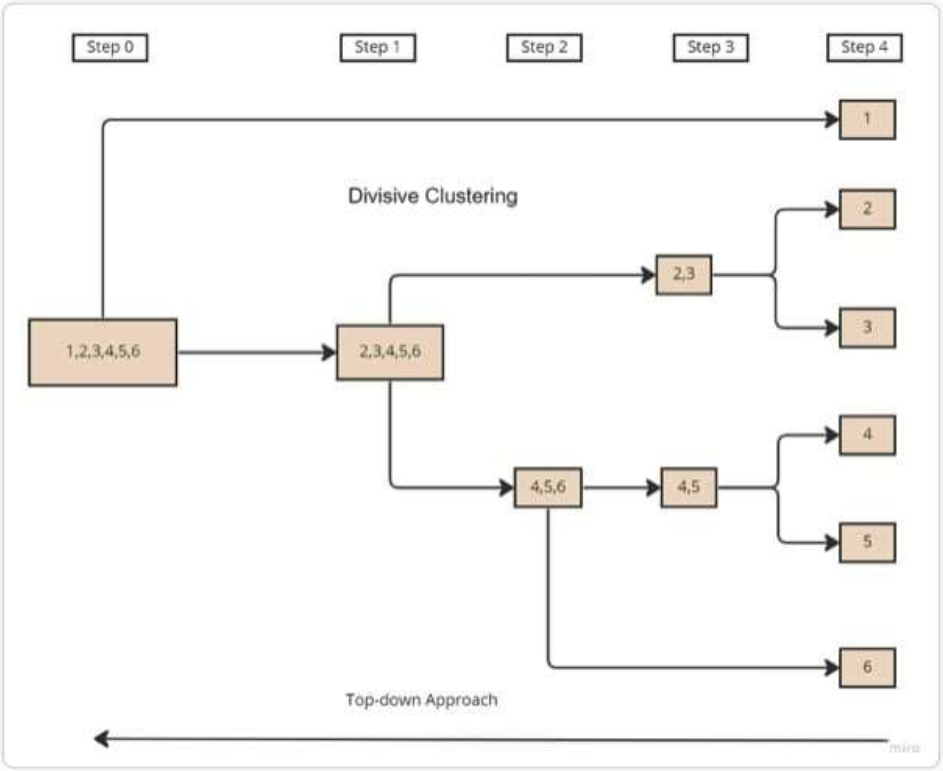

Step 2: Cluster Separation Process
Cluster 1
Cluster 2
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0


Step 4: Dendrogram Construction
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
2
5
8
0
4
3
6
9
1
7



Conclusion

Divisive hierarchical clustering serves as a valuable tool for understanding complex relationships within datasets. While it offers a hierarchical perspective and clear visualization through dendrograms, its high computational complexity and sensitivity to noise remain significant challenges. Despite its limitations, understanding the divisive hierarchical clustering algorithm is crucial for data analysts and researchers seeking to uncover intrinsic patterns within datasets and make informed decisions based on hierarchical cluster structures.


Step 0
Step 1
Step 2
Step 3
Step 4
1
Divisive Clustering
2
2,3
3
1,2,3,4,5,6
2,3,4,5,6
4
4,5,6
4,5
5
6
Top-down Approach