CAP 4630 – Clustering

Instructor: Aakash Kumar

University of Central Florida

What is Clustering?

- Clustering is an unsupervised learning technique used to group similar data points together based on specific criteria.
- It helps in identifying natural patterns and structures in data when labels are not available.
- Key Concept: Data points within a cluster are more similar to each other than to points in other clusters.
- Objective: Maximizing similarity within clusters and minimizing similarity between clusters.

Why Use Clustering?

- **Data Exploration:** Helps in finding hidden structures in data when you have no labels.
- Useful for Multiple Applications:
  - Market segmentation
  - Image segmentation
  - Customer behavior analysis
  - Anomaly detection
- Reduces the need for labeling in cases where labeling is expensive or impractical.

Real-Life Examples of Clustering

- Market Segmentation: Grouping customers by purchasing behavior.
- Document Clustering: Grouping articles by topics to produce hierarchies.
- Image Segmentation: Grouping pixels in images for object detection or medical imaging.

Types of Clustering

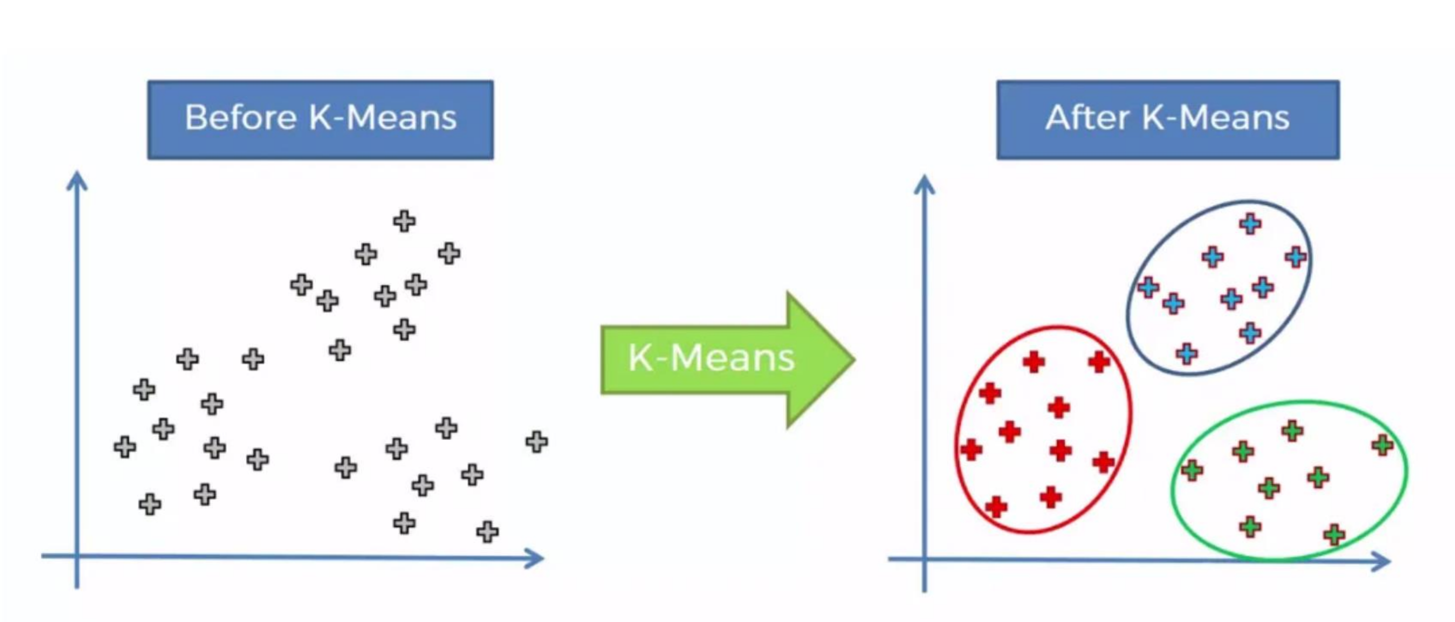
- Hierarchical Clustering: Builds a hierarchy of clusters.
- K-Means Clustering: Partitions data into a predefined number (K) of clusters.
- DBSCAN (Density-Based Clustering): Clusters points based on density and handles outliers well.

Goals of Clustering

- Maximizing Intra-Cluster Similarity: Data points in the same cluster should be as similar as possible.

- Minimizing Inter-Cluster Similarity: Different clusters should be distinct.

Example - 1

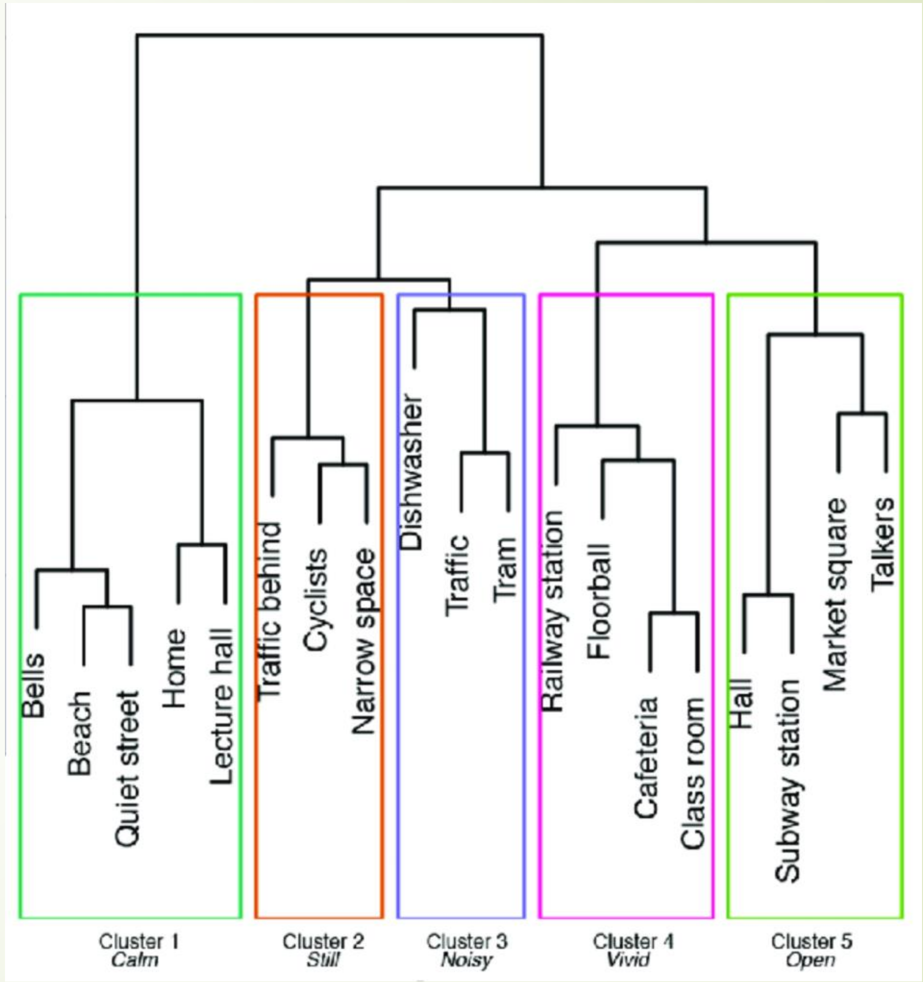


Ref: https://utsavdesai26.medium.com/the-beginners-guide-to-clustering-in-machine-learning-331987a7ceaf

Example - 2

Types of Clustering - Hierarchical Algorithms

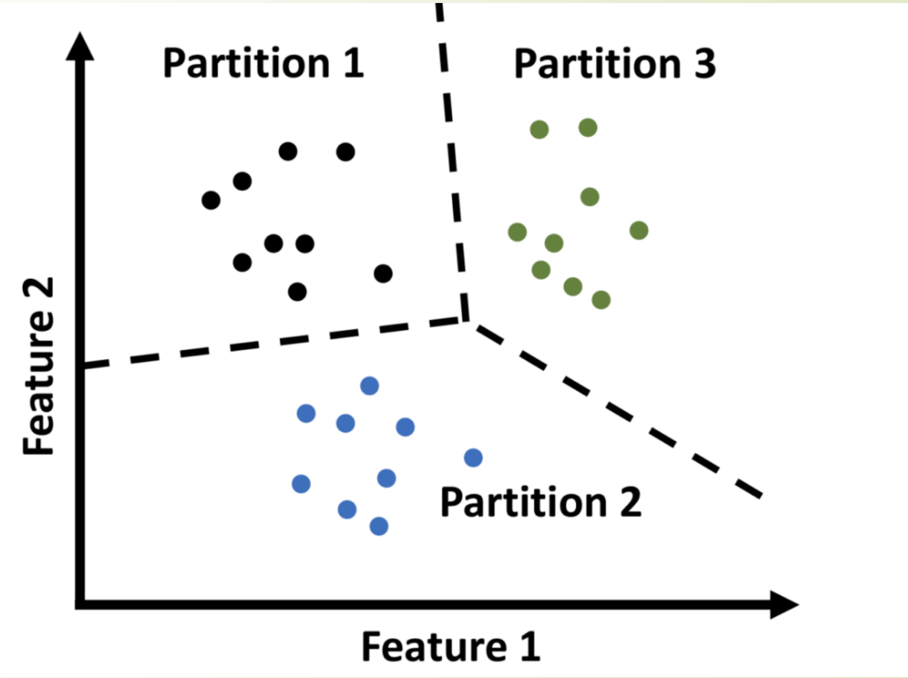
- Hierarchical algorithms create a hierarchical decomposition of objects based on similarity.
- Key Points:
  - Agglomerative (Bottom-Up): Start with individual points and merge them step by step.
  - Divisive (Top-Down): Start with all data points and split them step by step.



Ref: https://www.researchgate.net/figure/Dendrogram-showing-the-hierarchical-clustering-result-from-average-linkage-clustering_fig15_262025025

Types of Clustering - Partitional Algorithms

- These algorithms partition the dataset into K clusters, evaluating each partition based on a criterion.
- Key Examples:
  - K-Means Clustering
  - Gaussian Mixture Models
  - Spectral Clustering

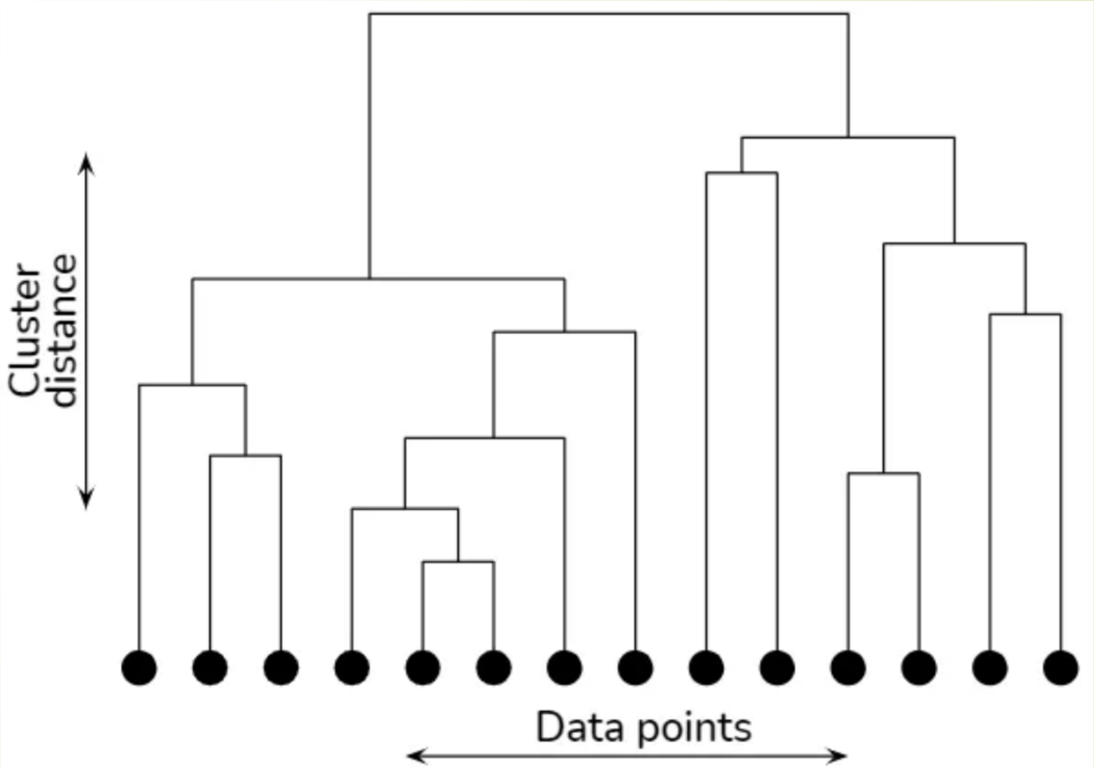


Ref: https://computing4all.com/courses/introductory-data-science/lessons/a-few-types-of-clustering-algorithms/

Hierarchical Clustering

Dendrogram

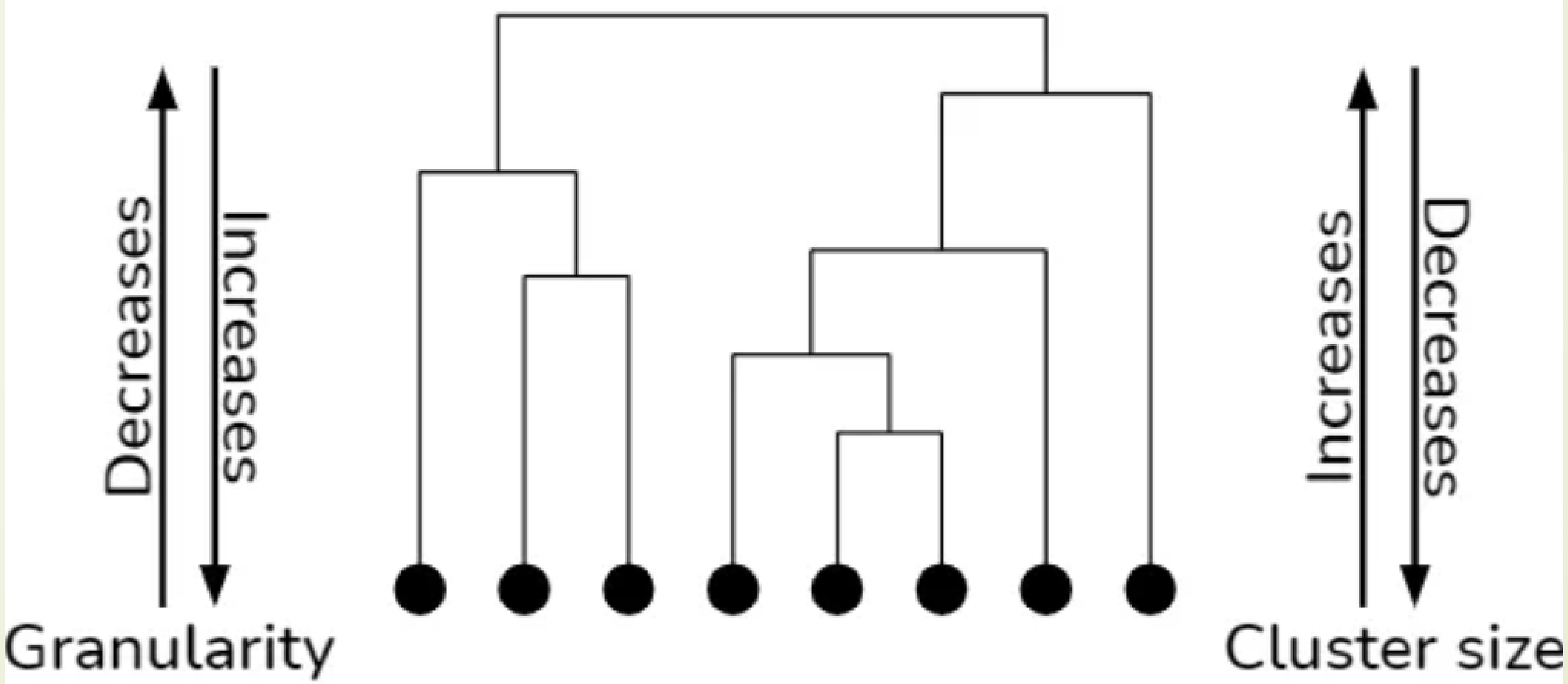
- A dendrogram is a tree-like diagram used to represent the relationships between data points in hierarchical clustering.
- It illustrates how data points are merged step-by-step into clusters based on similarity.
- Key Concept:
  - The height of the lowest internal node connecting two data points reflects their similarity. The shorter the height, the more similar the data points are.



Dendrogram

- Unlike a regular family tree, a dendrogram does not branch out at regular intervals. The branching structure is determined by the data.
- The Y-axis represents the distance or dissimilarity between clusters.
- As you move down the dendrogram, clusters split into smaller groups until each data point is isolated.
- As you move up the dendrogram, smaller clusters are merged into larger groups, reflecting hierarchical clustering.
- Hierarchical clustering is sometimes called "clustering of clusters" because it captures multiple levels of cluster formation.

Dendrogram



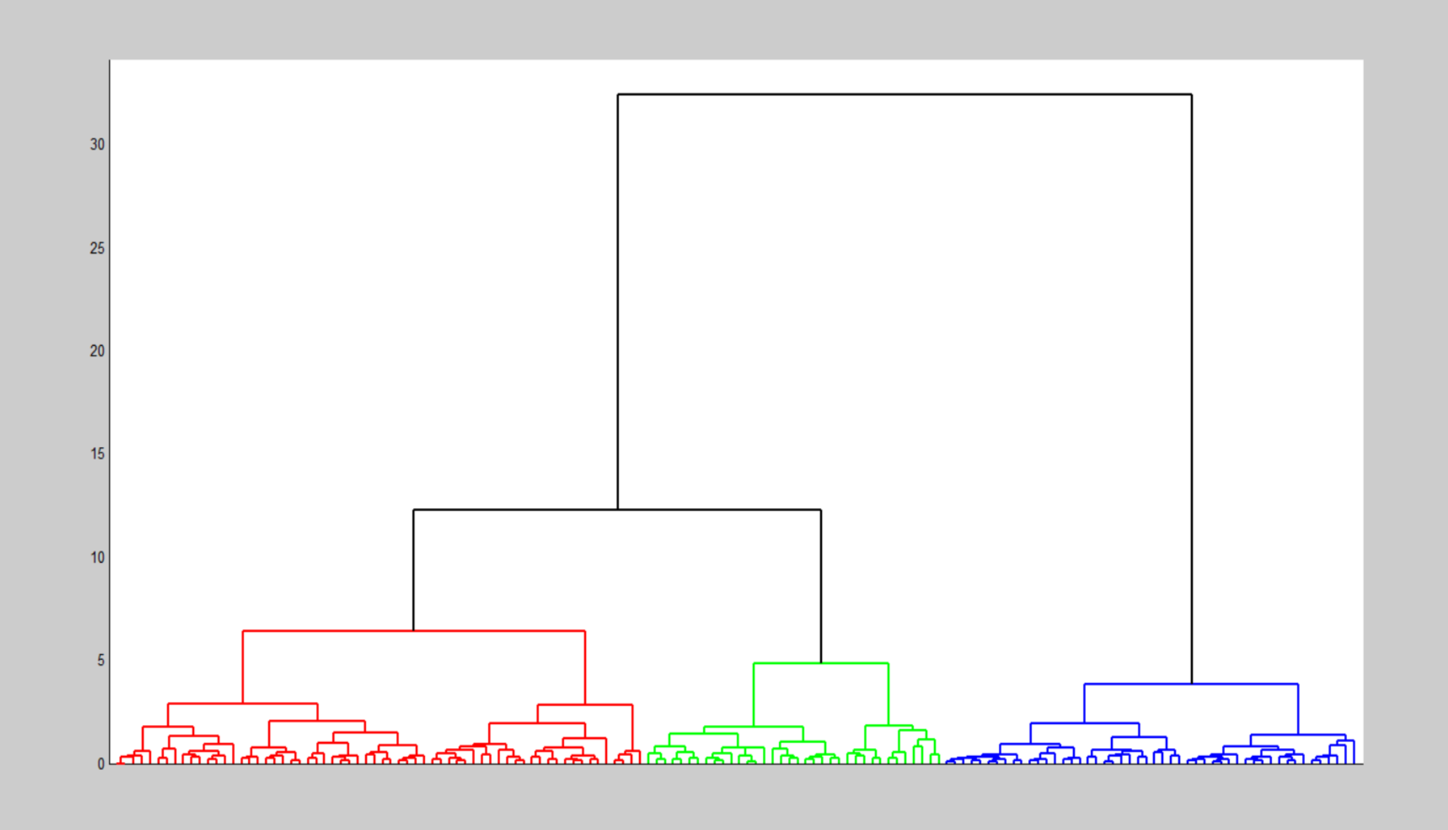
Effect of granularity and cluster size while traversing in the dendrogram (Image by Author)

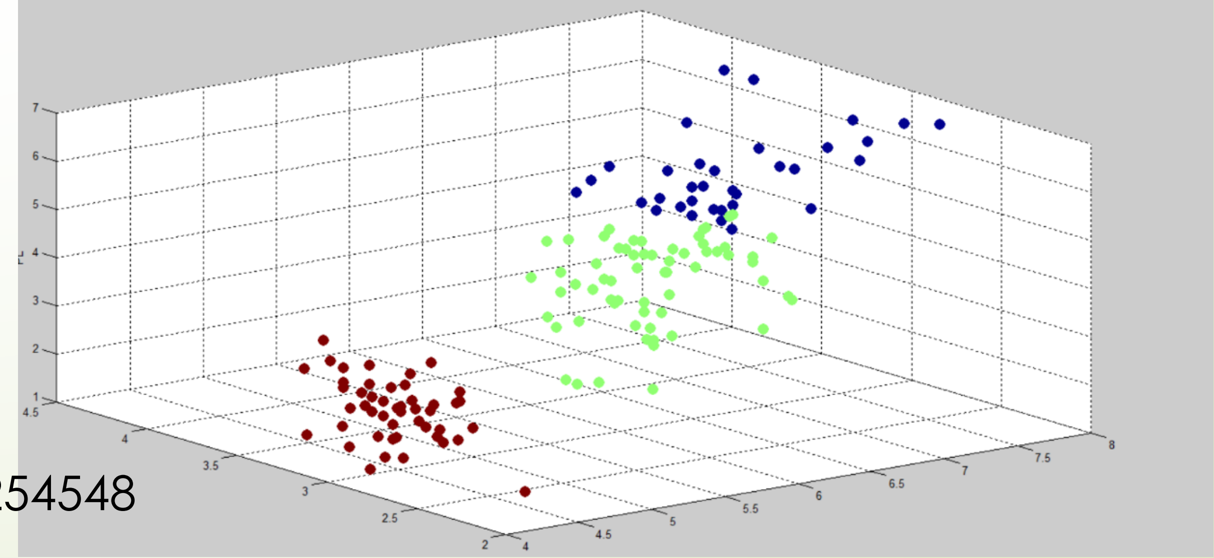
Determining the Number of Clusters

- No assumption is made about the number of clusters during dendrogram construction.
- After building the dendrogram, we determine the number of clusters by making a horizontal cut at the desired distance threshold.
- Each branch below the horizontal cut represents an individual cluster. This defines the cluster membership for each data point.

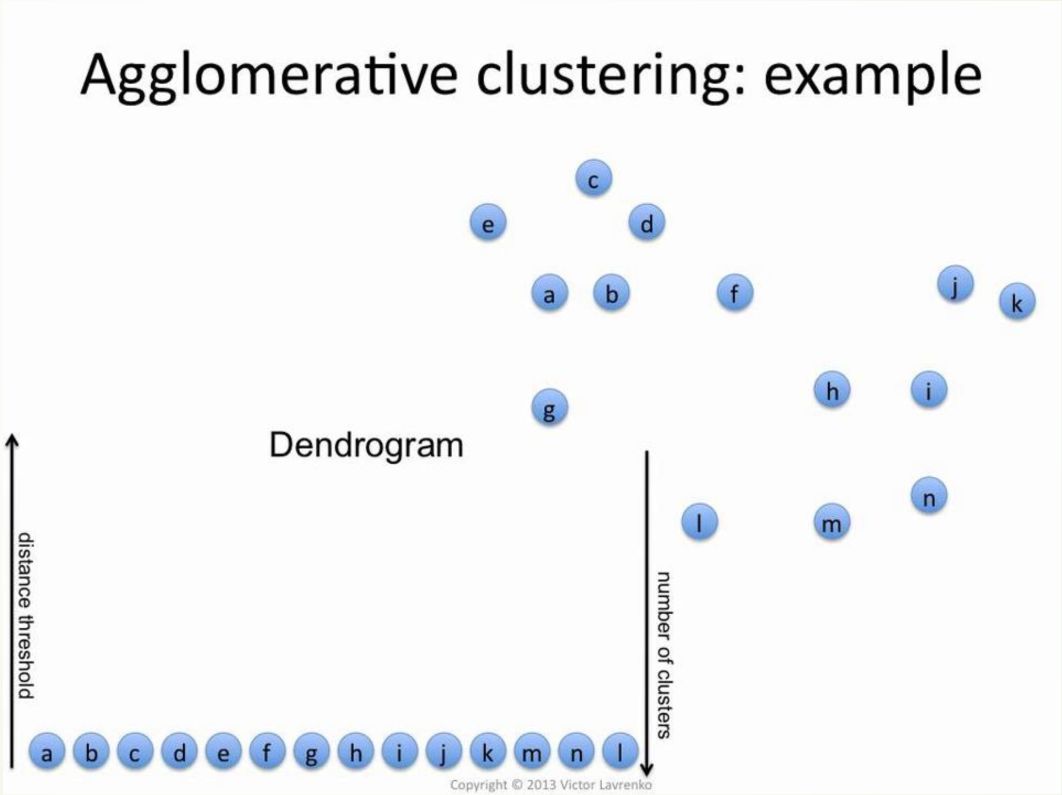
Hierarchical Clustering: Bottom-Up (Agglomerative)

- In Bottom-Up clustering, the process starts by treating each data point as its own cluster.
- At each step, the closest clusters are merged based on a distance metric, forming larger clusters.
- Single Linkage is a simple approach where the distance between clusters is defined as the shortest distance between any two points in the clusters.
- This process continues iteratively until all data points are merged into a single cluster.
- A distance metric (e.g., Euclidean distance) is used to determine the proximity between clusters at every step.

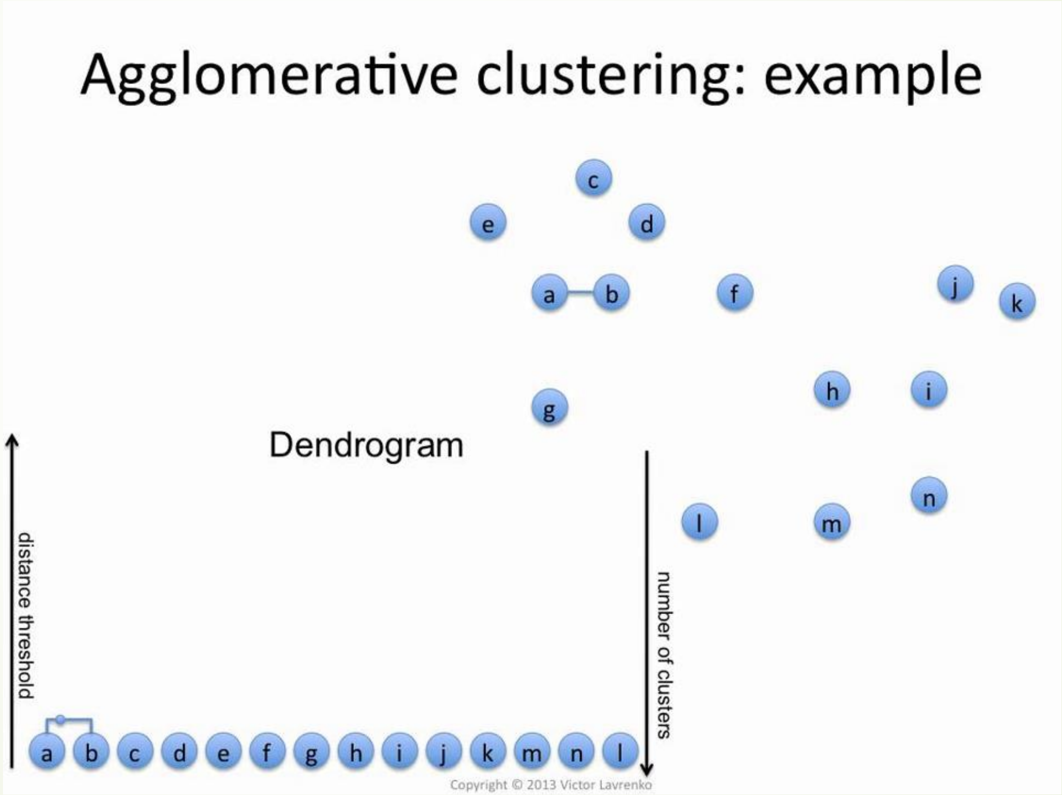




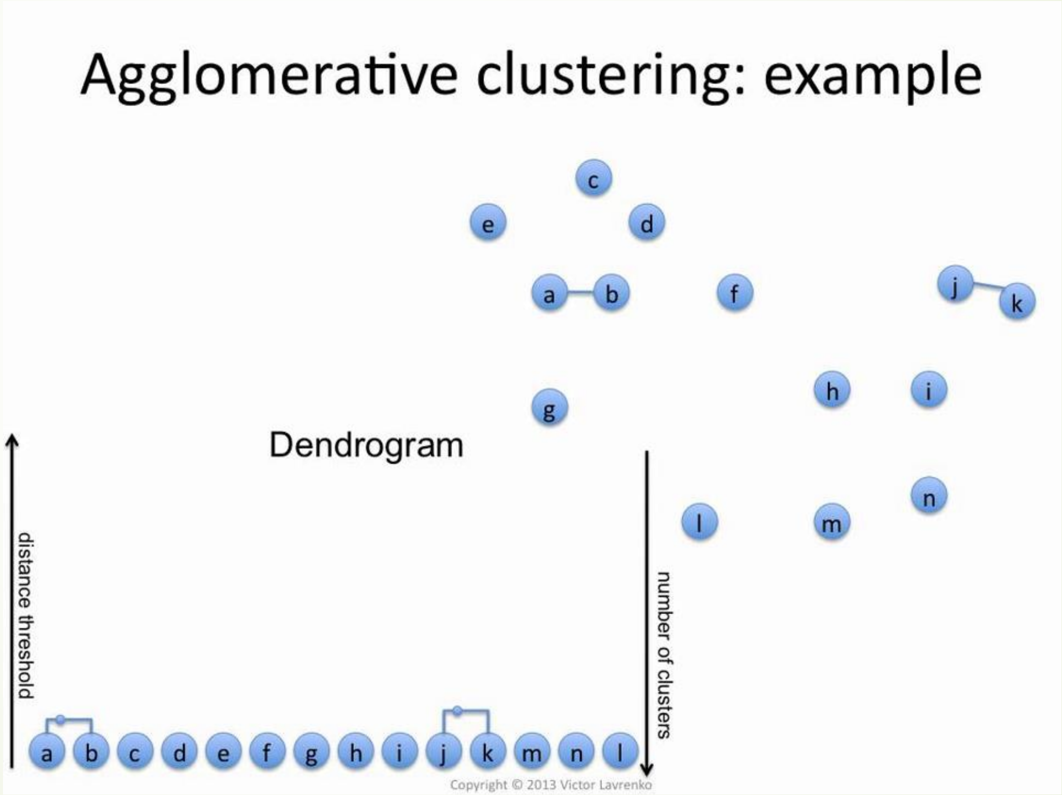
Ref: https://www.slideshare.net/slideshow/agglomarative/254548084



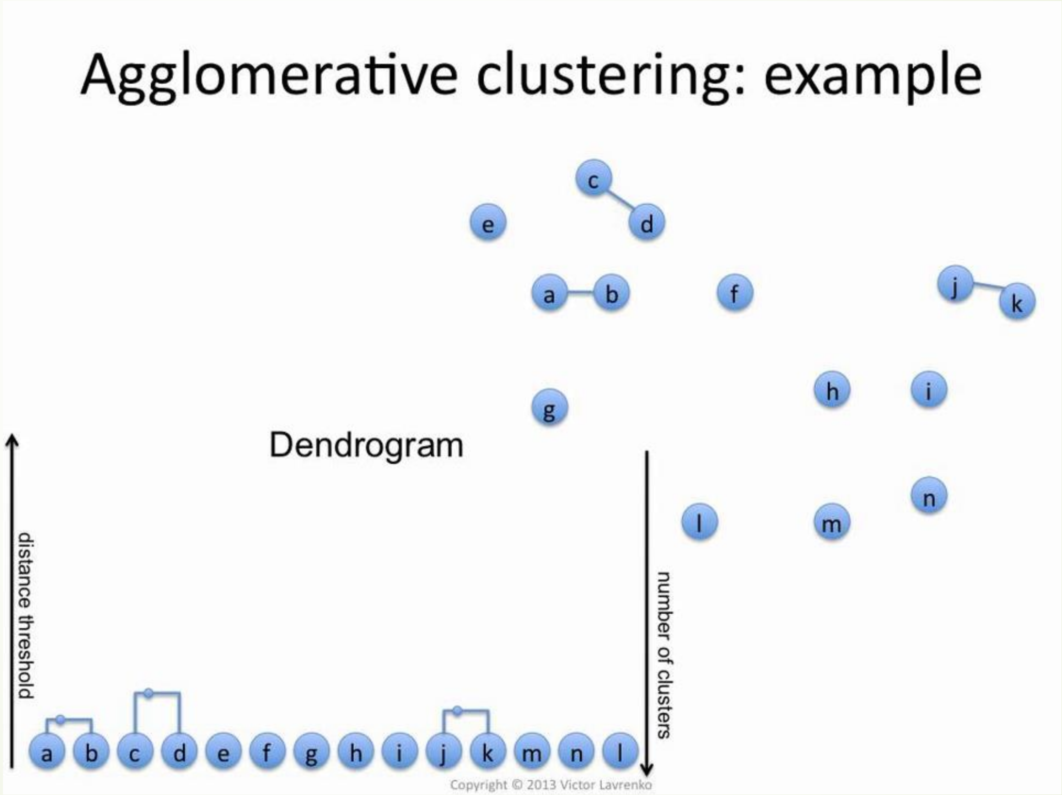
Ref: https://www.youtube.com/watch?v=XJ3194AmH40



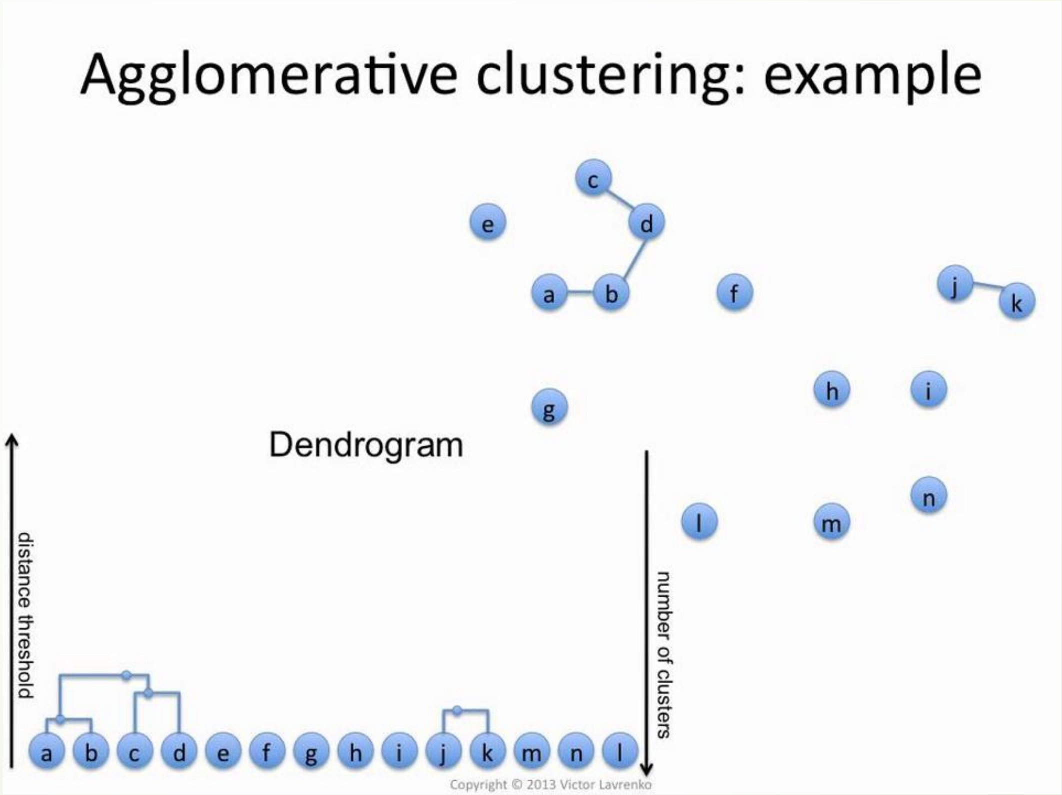
Ref: https://www.youtube.com/watch?v=XJ3194AmH40



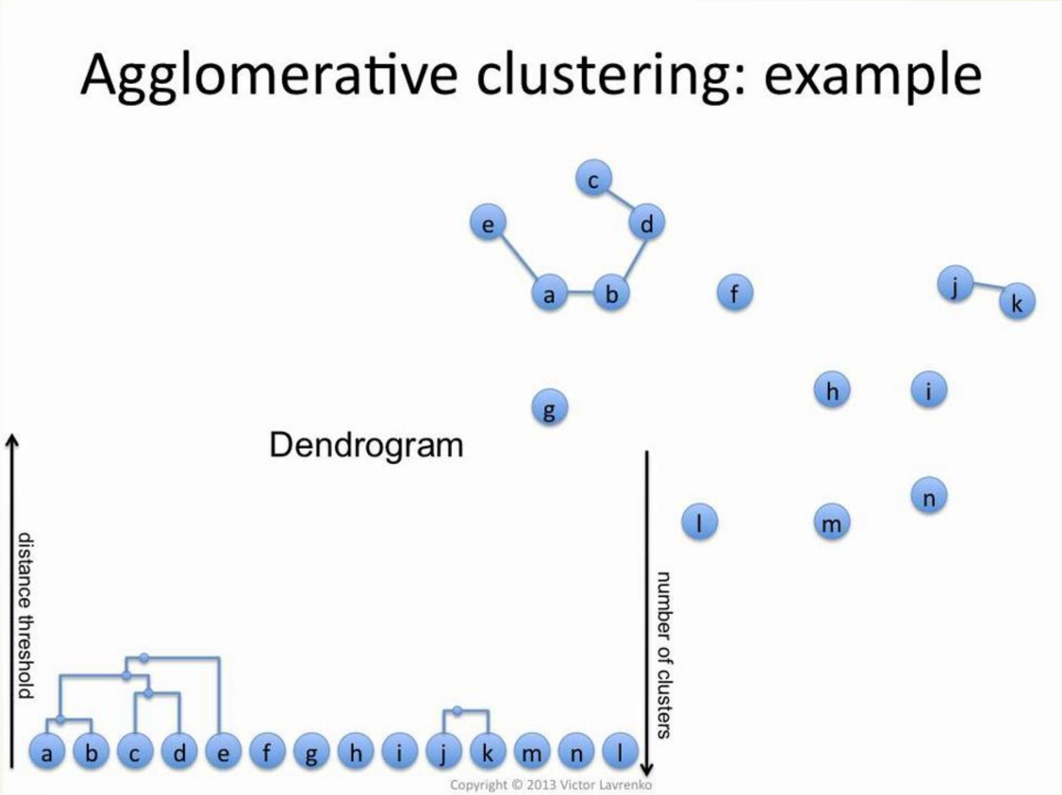
Ref: https://www.youtube.com/watch?v=XJ3194AmH40



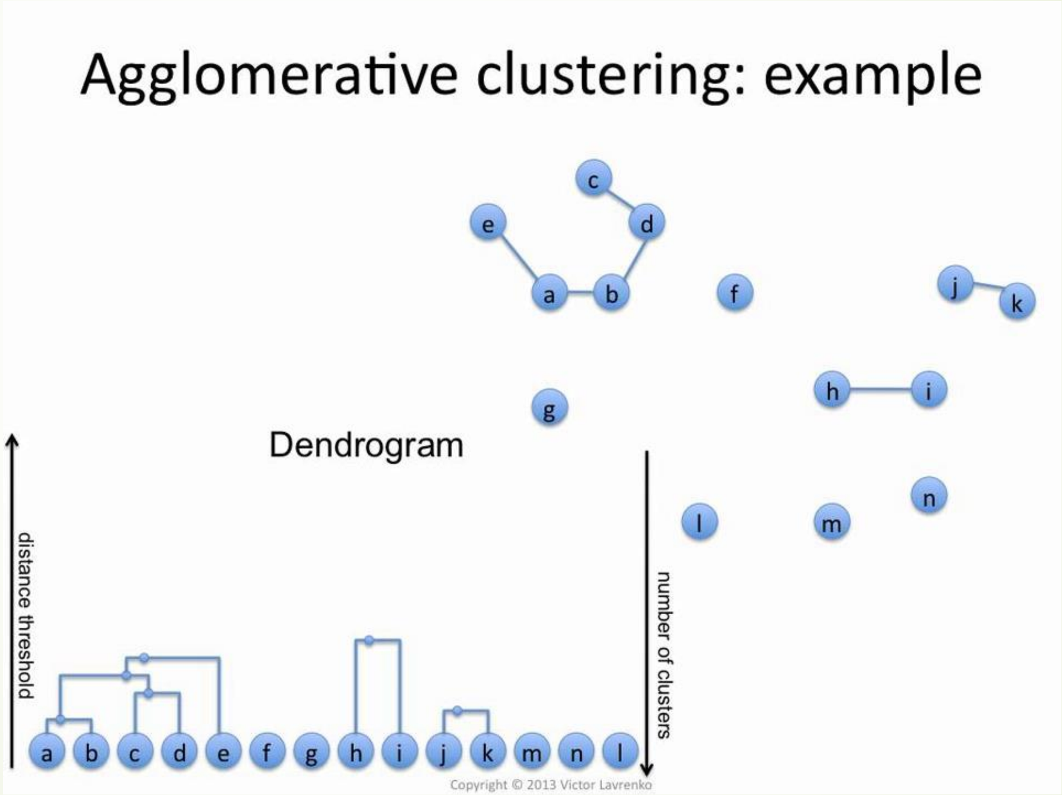
Ref: https://www.youtube.com/watch?v=XJ3194AmH40



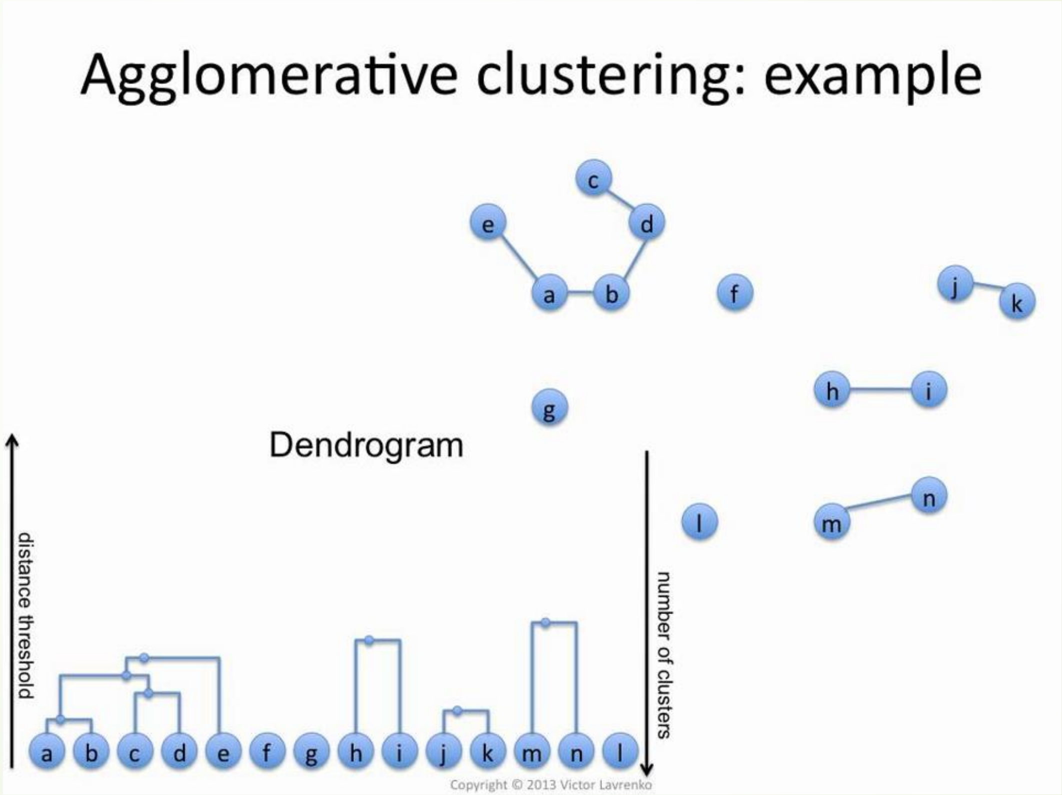
Ref: https://www.youtube.com/watch?v=XJ3194AmH40



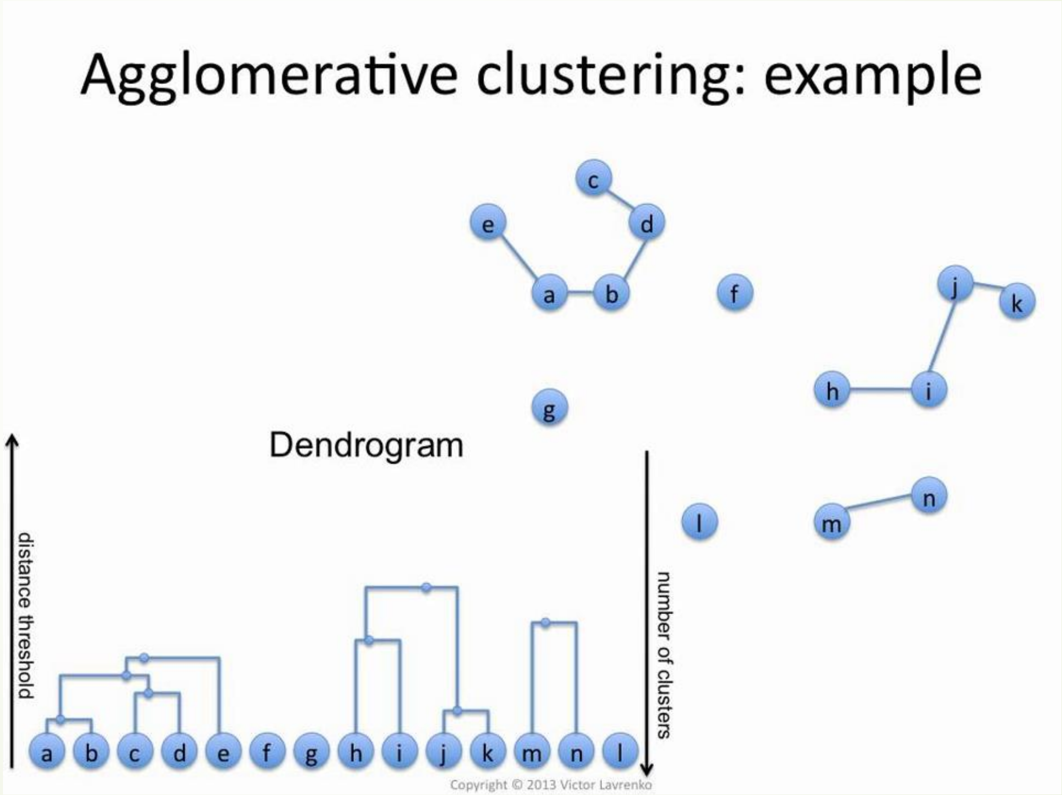
Ref: https://www.youtube.com/watch?v=XJ3194AmH40



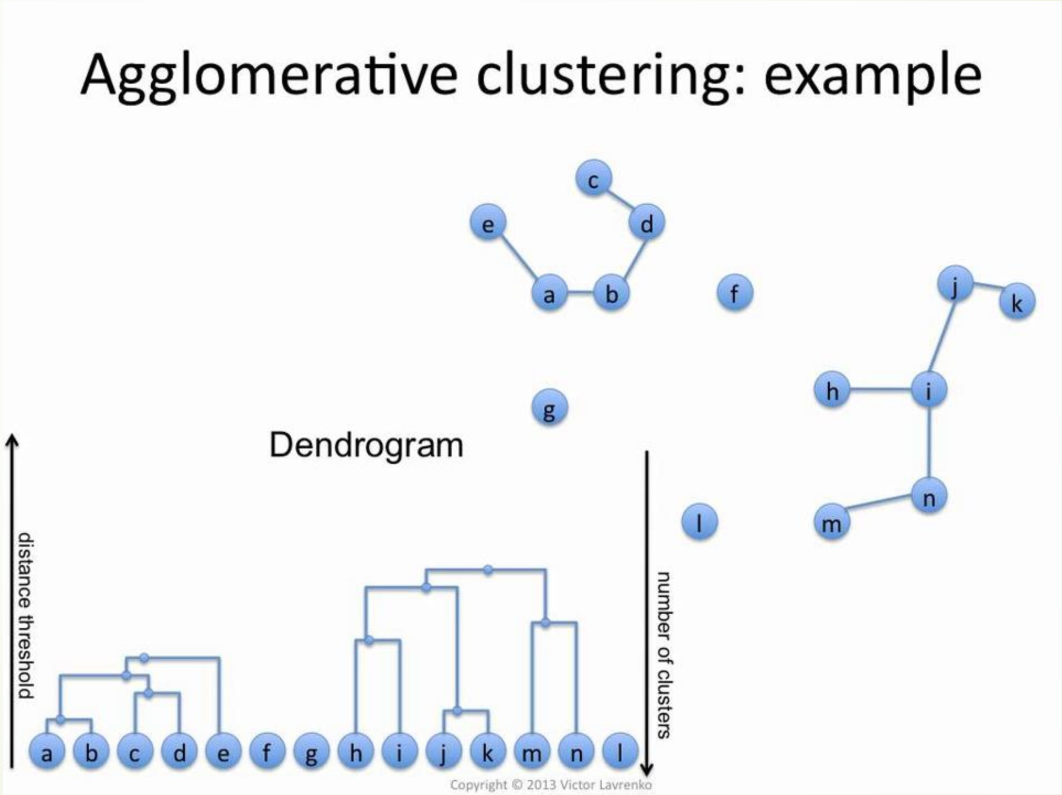
Ref: https://www.youtube.com/watch?v=XJ3194AmH40



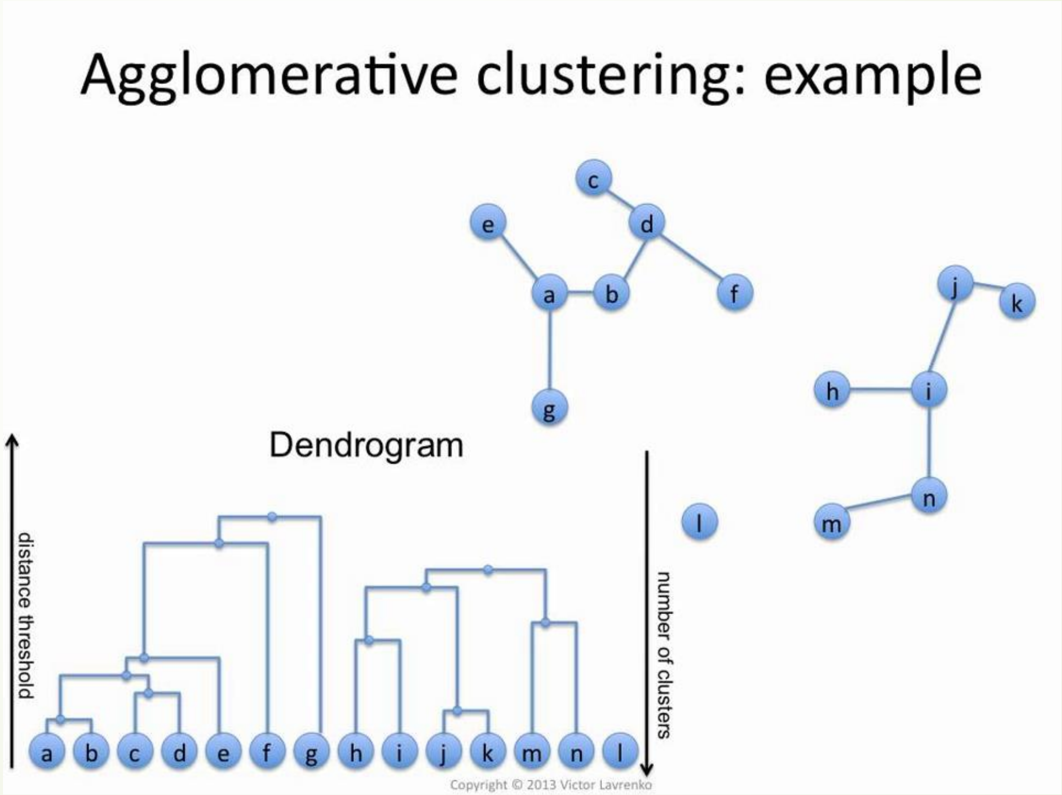
Ref: https://www.youtube.com/watch?v=XJ3194AmH40



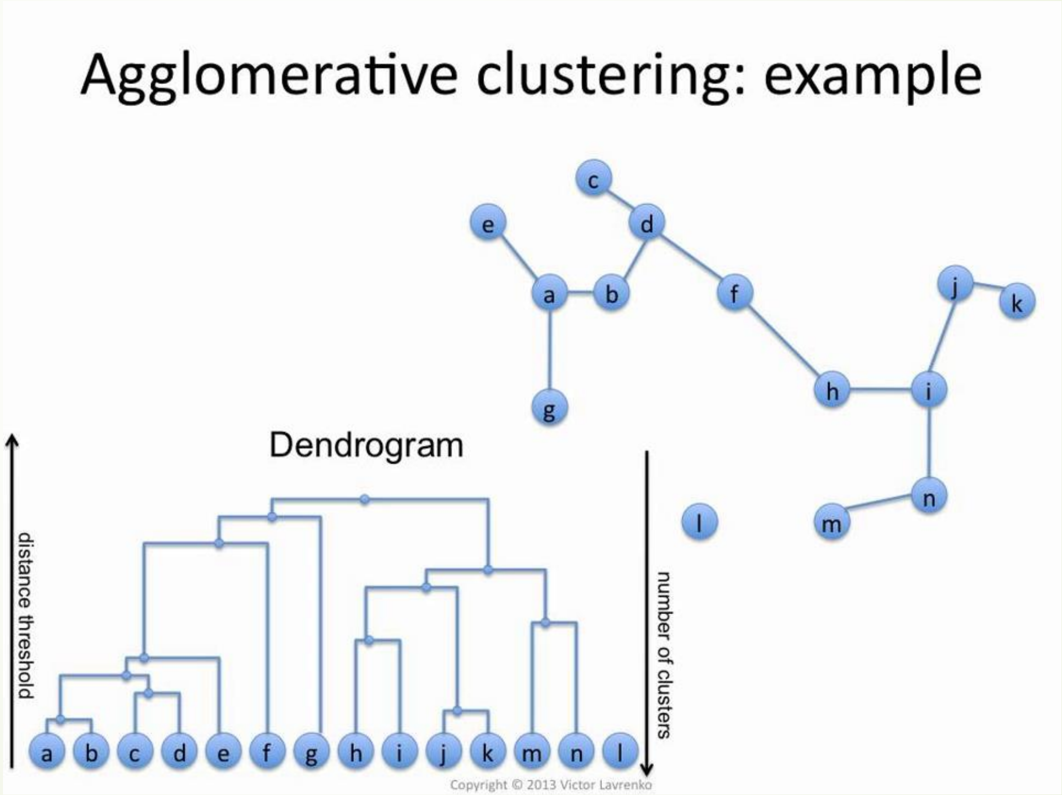
Ref: https://www.youtube.com/watch?v=XJ3194AmH40



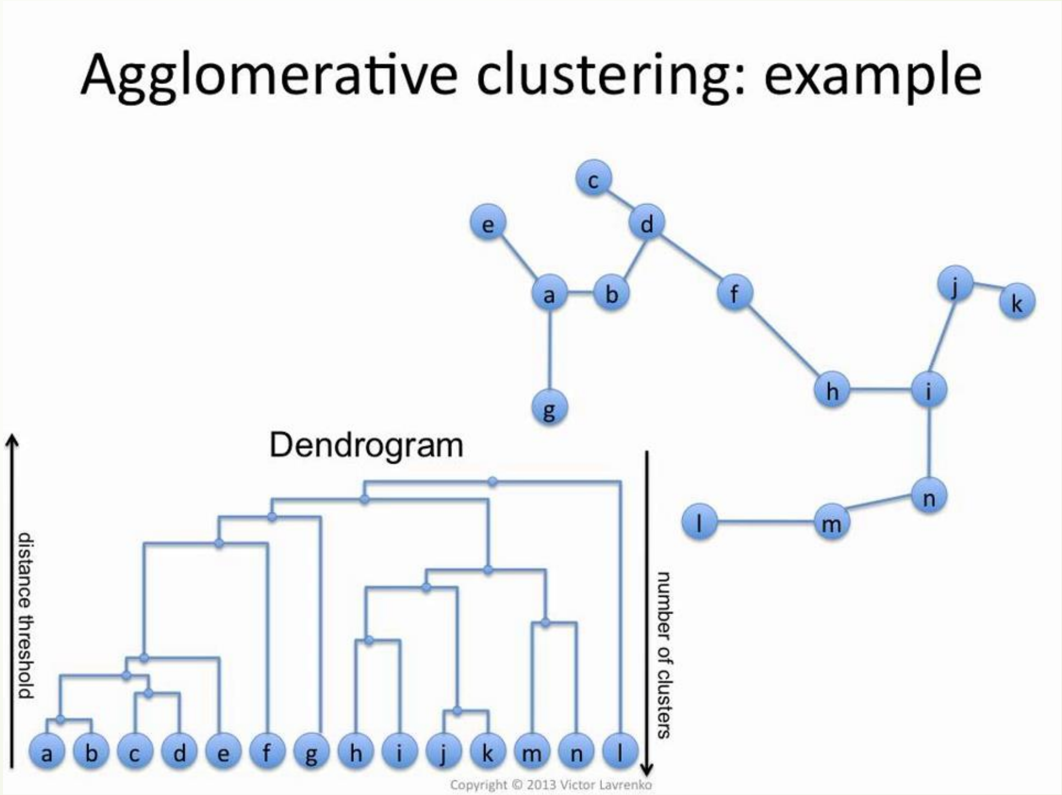
Ref: https://www.youtube.com/watch?v=XJ3194AmH40



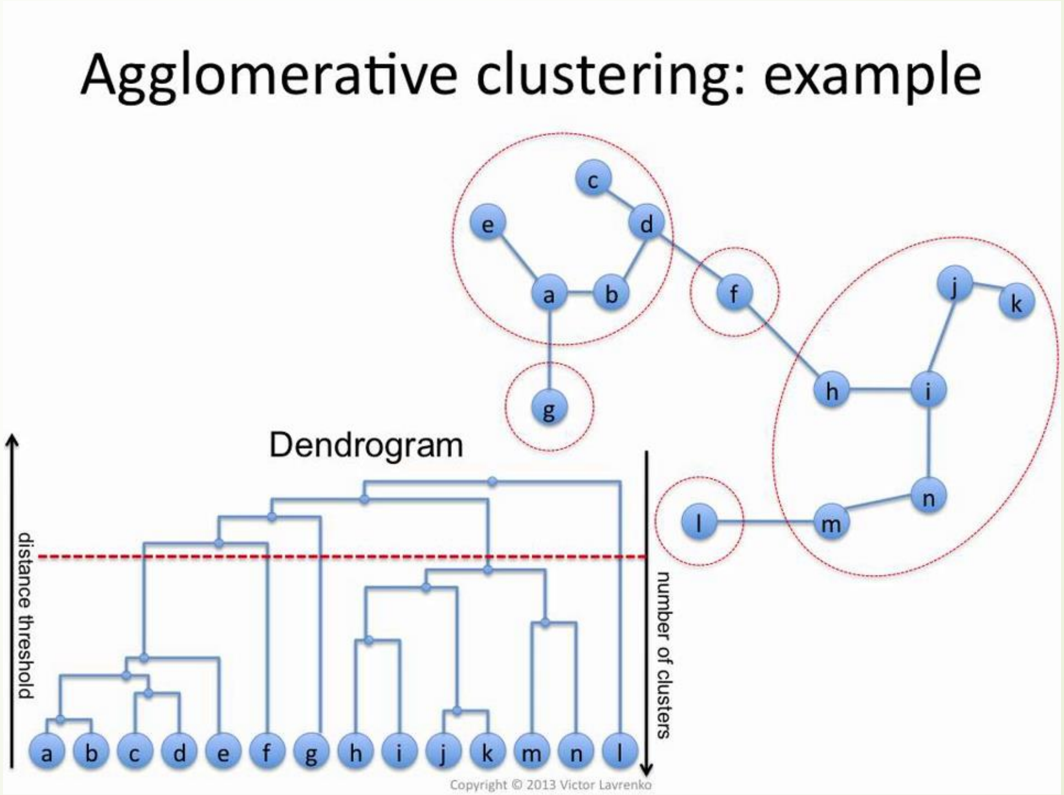
Ref: https://www.youtube.com/watch?v=XJ3194AmH40



Ref: https://www.youtube.com/watch?v=XJ3194AmH40



Ref: https://www.youtube.com/watch?v=XJ3194AmH40



Ref: https://www.youtube.com/watch?v=XJ3194AmH40

Hierarchical Clustering: Top-Down (Divisive)

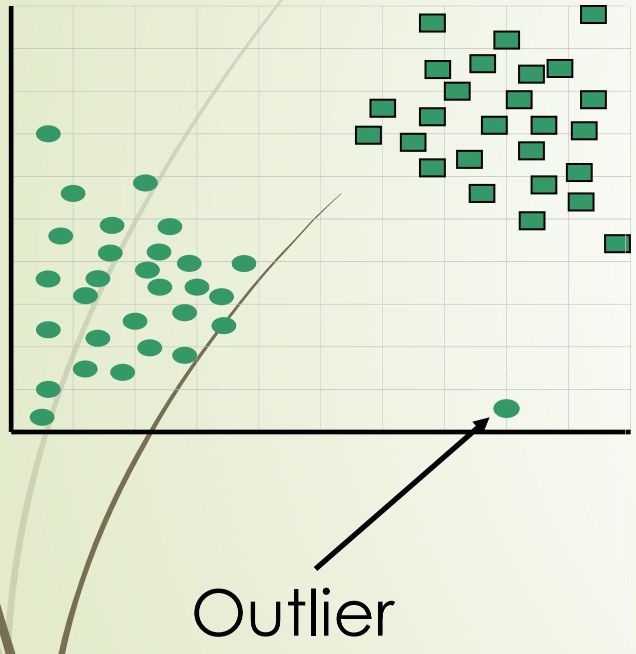
- In Top-Down clustering, the entire dataset starts as a single cluster. This cluster is then recursively split into smaller clusters.
- One common method used for divisive clustering is Recursive K-Means, where K-Means is applied iteratively to split clusters.
- The process continues until either:
  - Each data point is in its own cluster, or
  - A predefined minimum number of data points per cluster is reached.
- At each step, carefully decide how many clusters you want to form in the next iteration based on the structure of the data.

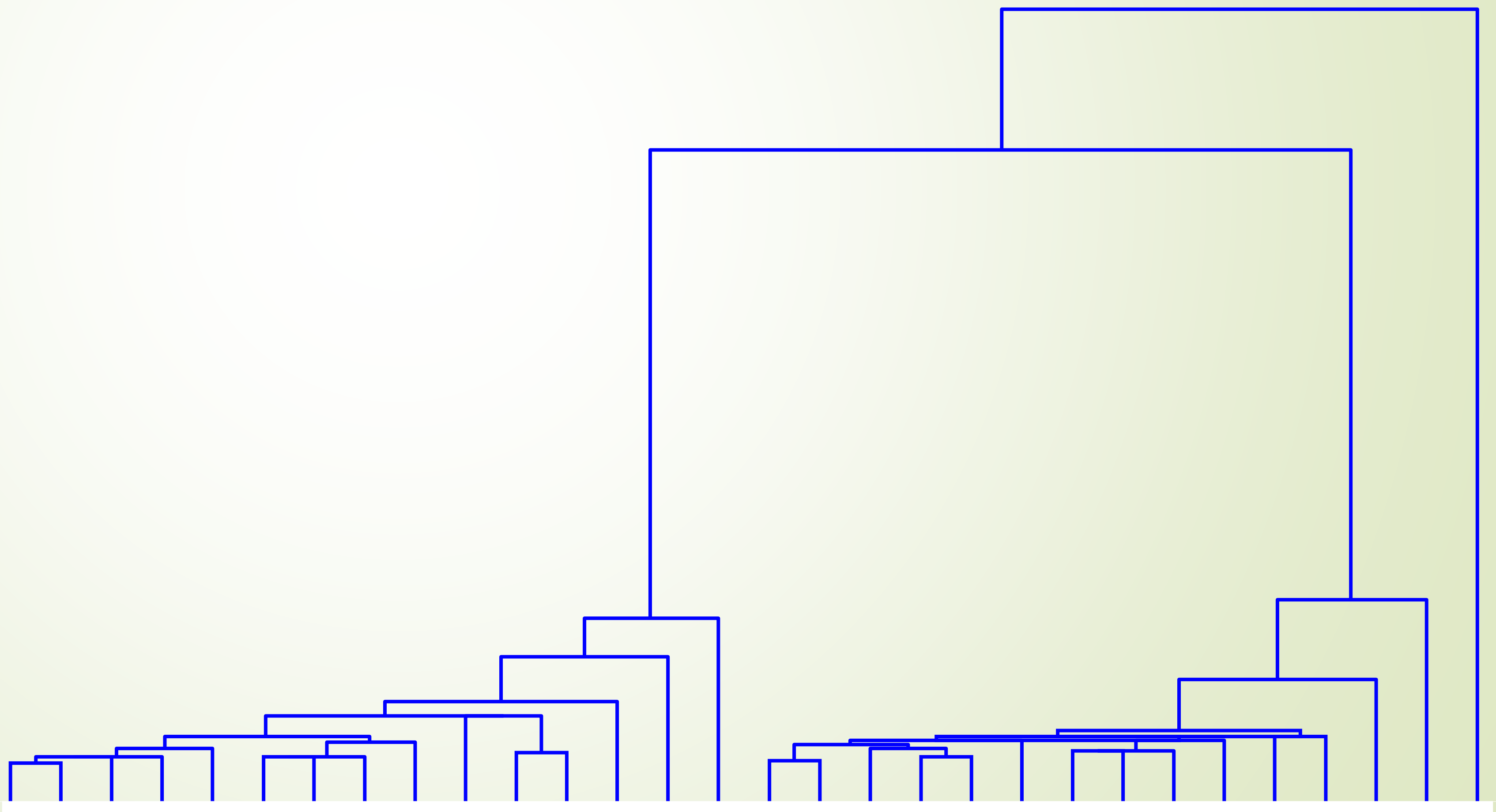
Calculating Distance in Hierarchical Clustering

- Single Linkage (Nearest Neighbor):
  - Measures the distance between the two closest points in different clusters (nearest neighbors).
  - This method tends to form long, chain-like clusters.
- Complete Linkage (Furthest Neighbor):
  - Measures the distance between the two farthest points in different clusters (furthest neighbors).
  - This method prefers compact, spherical clusters.
- Group Average Linkage:
  - Calculates the distance between clusters by averaging the distances between all pairs of points across the two clusters.
  - Balances between single and complete linkage methods.

Detecting Outliers

The single isolated branch is suggestive of a data point that is very different to all others





Hierarchical Clustering Summary

- No need to predefine the number of clusters: The number of clusters is determined after the dendrogram is constructed.
- Intuitive structure: The hierarchical approach aligns well with human intuition for grouping, making it useful in certain domains.
- Scalability issues: Hierarchical clustering can be computationally expensive.
- Prone to local optima: Like many heuristic algorithms, hierarchical clustering may get stuck in local optima, leading to suboptimal clustering results.
- Subjective interpretation: Results can be highly subjective, and different analysts may interpret the same dendrogram in different ways.
- Robust to Noise/Outliers (Handles noise and irrelevant data).

Partitional Clustering

K-Means Clustering

- K-Means clustering minimizes the Euclidean distance between points and their respective cluster centroids.
- The cluster quality is measured using an intra-cluster measure, which sums the distances from each point to the cluster centroid.

Steps of the K-Means Algorithm

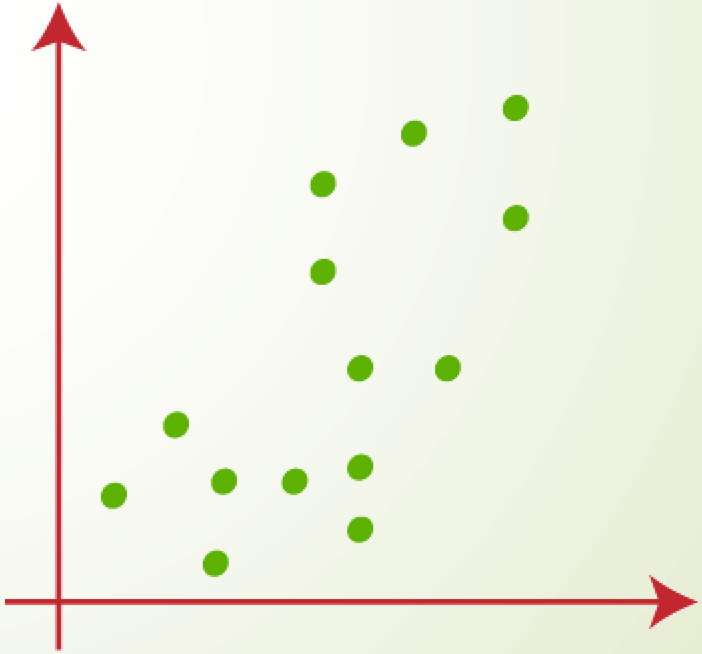
1. Choose the number of clusters k.
2. Initialize k cluster centroids (randomly or using K-Means++ initialization).
3. Assign each point to the nearest centroid.
4. Update the centroids by averaging the points in each cluster.
5. Repeat steps 3 and 4 until the centroids do not change or a predefined stopping criterion is met.

Stopping Criteria for K-Means

- No (or minimal) reassignments of data points to different clusters.
- No (or minimal) change in centroids.
- Minimal decrease in the sum of squared error (SSE) between iterations.

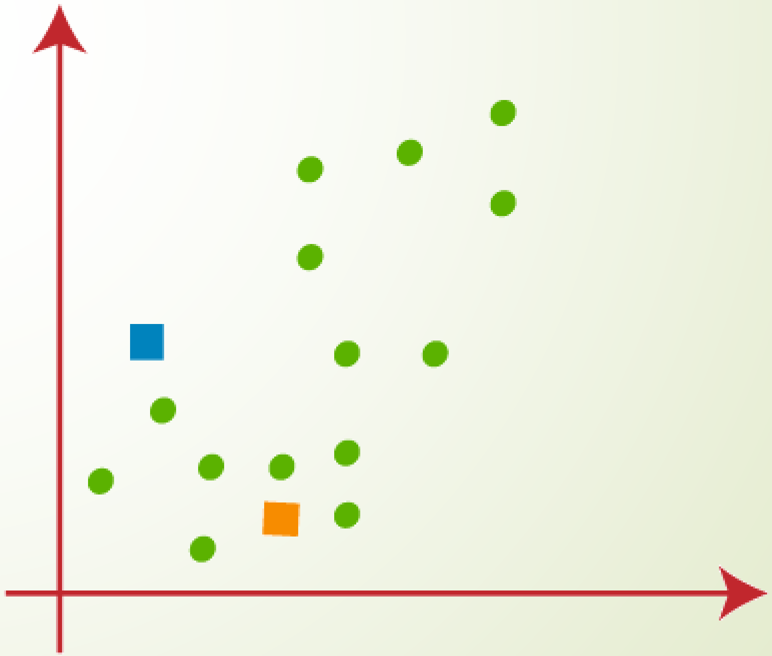
Example Scatter Plot for Clustering

- Suppose we have two variables, M1 and M2.
- The scatter plot shows a distribution of data points that we aim to cluster into groups.



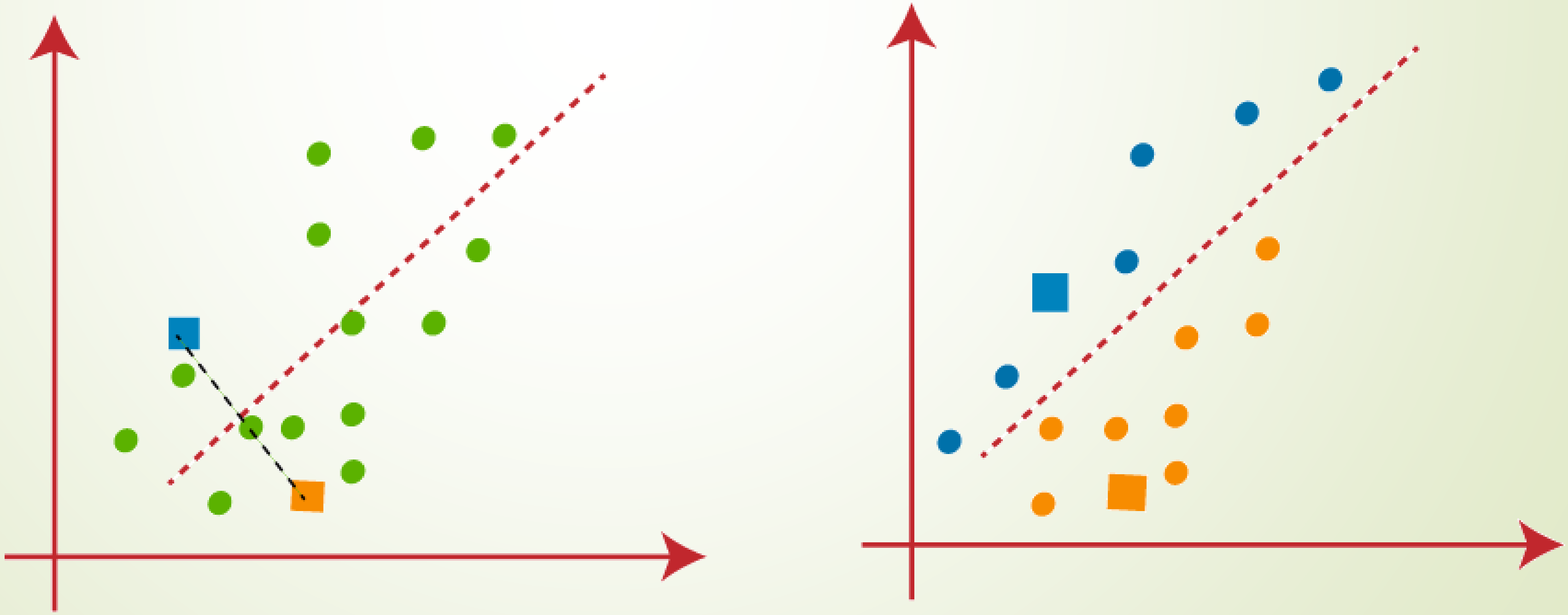
Initializing Random Centroids

- Step 1: We select two random points, not necessarily part of the dataset, as initial cluster centroids.
- These points will act as the reference for the clusters.



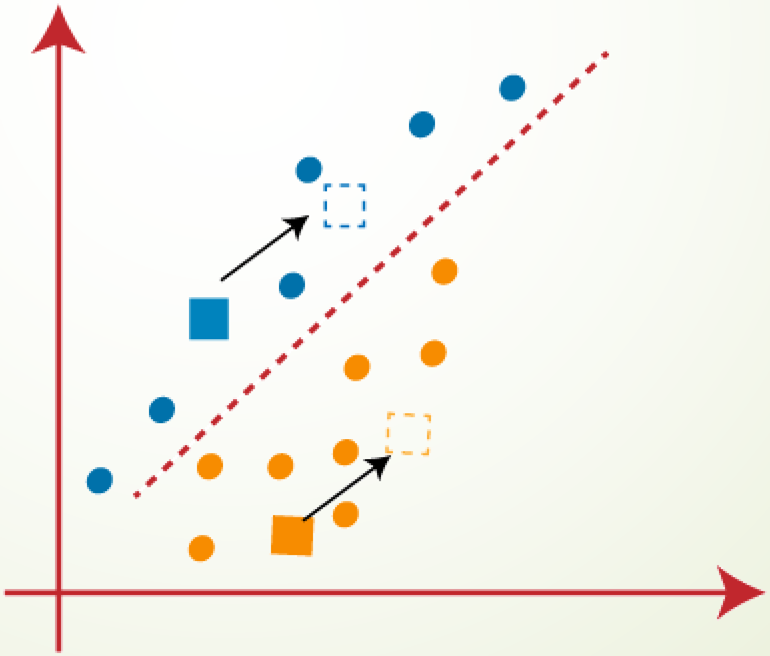
First Assignment of Points

- Now, assign each data point to its nearest centroid using Euclidean distance.



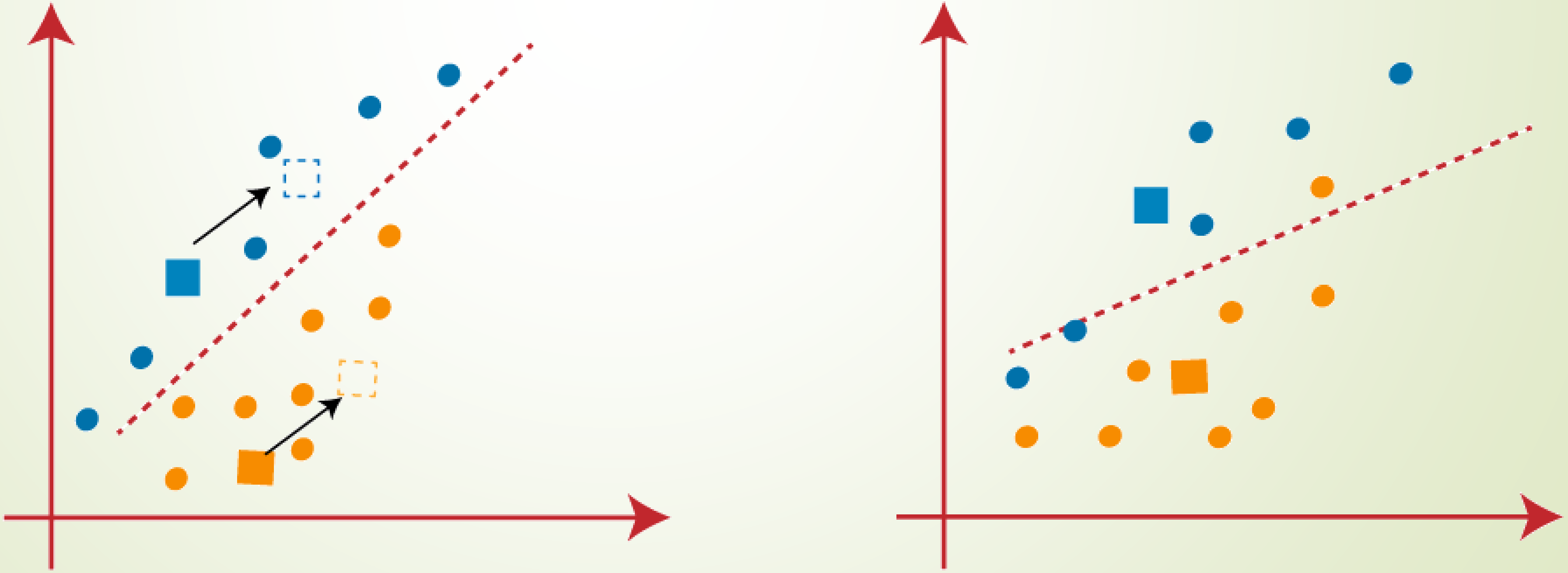
Recomputing Centroids

- After the initial assignment, we recompute the centroids.
- The new centroids will be the mean of the points assigned to each cluster.



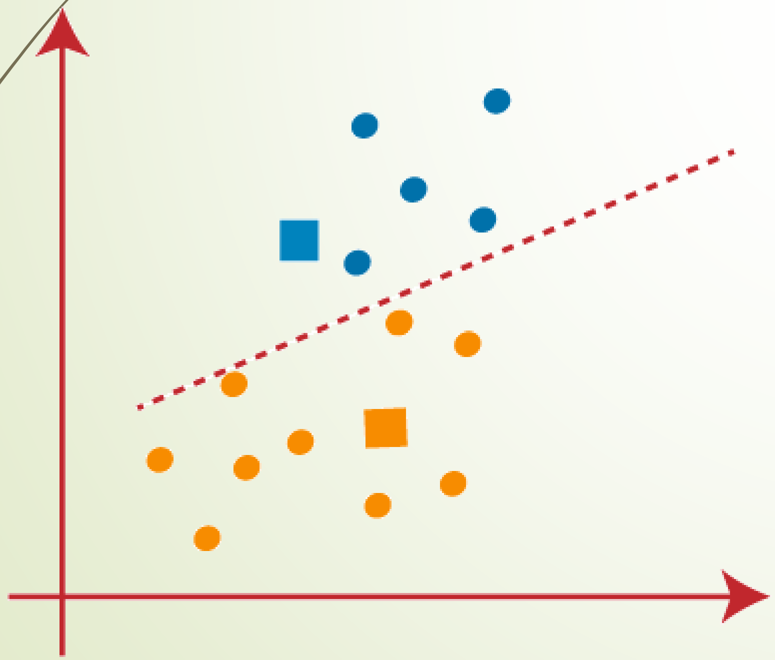
Iteration and Final Clusters

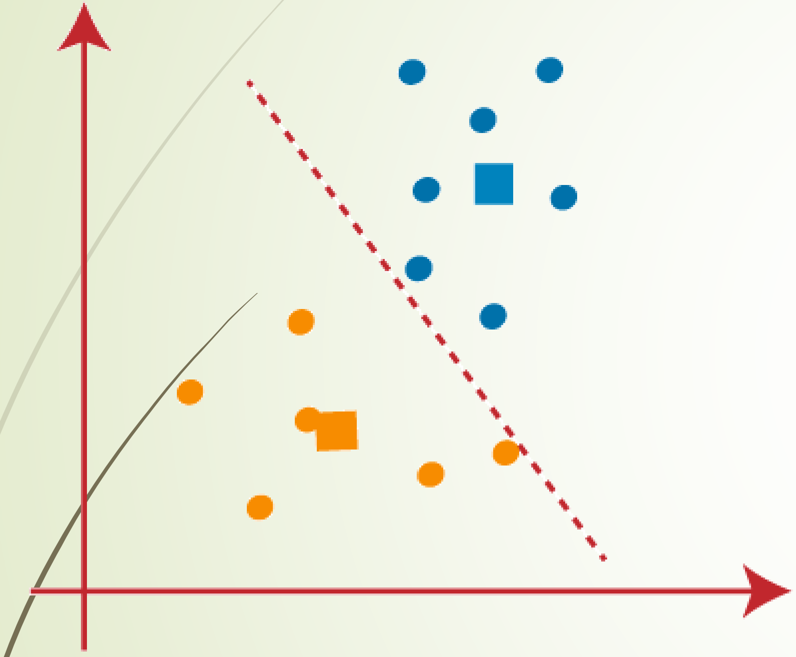
- We repeat the process of assigning data points and updating centroids until the assignments do not change.
- This indicates that the algorithm has converged, and the final clusters are stable.

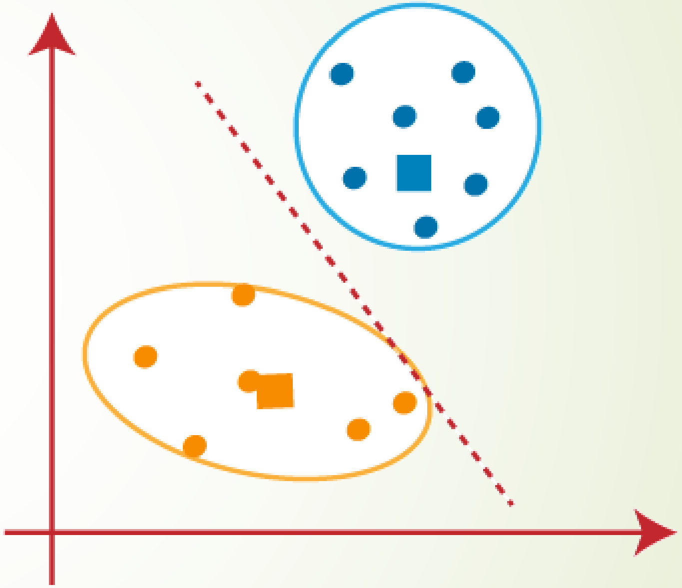


Visualizing the Final Result

- n the final step, visualize the clusters. Each cluster should now contain points that are closer to its centroid than to any other.
- This concludes the K-Means clustering process.







K-Means Convergence

Objective

$$\min_{\mu}\min_{C}\sum_{i=1}^{k}\sum_{x\in C_i}|x-\mu_i|^2$$

1. Fix μ, optimize *C:*

Step 1 of kmeans

$$\min_{C}\sum_{i=1}^{k}\sum_{x\in C_i}|x-\mu_i|^2=\min_{c}\sum_{i}^{n}\left|x_i-\mu_{x_i}\right|^2$$

2. Fix *C,* optimize μ:

$$\min_{\mu}\sum_{i=1}^{k}\sum_{x\in C_i}|x-\mu_i|^2$$

– Take partial derivative of μ_i and set to zero, we have

$$\mu_i=\frac{1}{|C_i|}\sum_{x\in C_i}x$$

Step 2 of kmeans

Kmeans takes an alternating optimization approach, each step is guaranteed to decrease the objective – thus guaranteed to converge

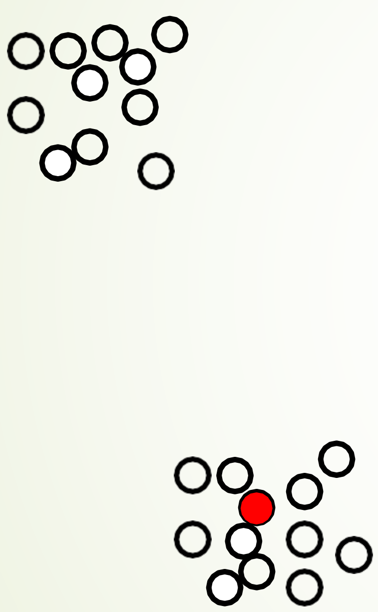
Strengths

- K-Means has a time complexity of O(tkn), where:
  - n is the number of data points,
  - k is the number of clusters,
  - t is the number of iterations.
  - In practice, both k and t are much smaller than n, making K-Means relatively fast and scalable.
- Converges quickly:
  - The algorithm generally converges within a small number of iterations, often reaching a local optimum.

Weaknesses

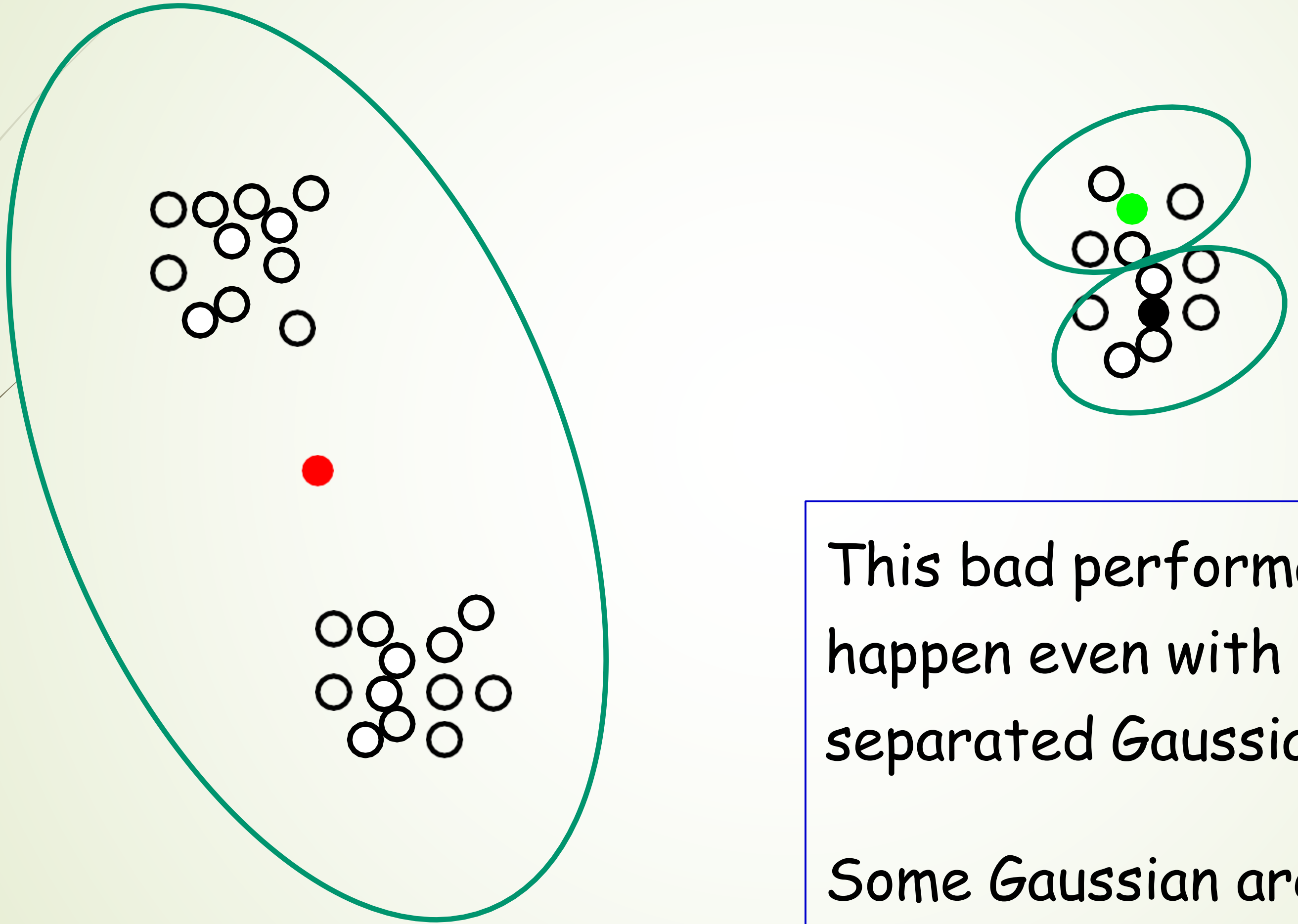
- Depends on mean calculation:
  - K-Means is only applicable when the mean of the data points is defined.
- Requires pre-determined k:
  - The number of clusters (k) must be specified beforehand, which may not always be intuitive or easy to determine.
- Sensitive to noise and outliers:
  - K-Means struggles with noisy data and outliers, as they can significantly affect the cluster centroids.
- Not suitable for non-convex clusters:
  - The algorithm assumes clusters are spherical and cannot easily identify clusters with non-convex shapes.

Lloyd's method: Performance



This bad performance, can happen even with well separated Gaussian clusters.

Lloyd's method: Performance



This bad performance, can happen even with well separated Gaussian clusters.

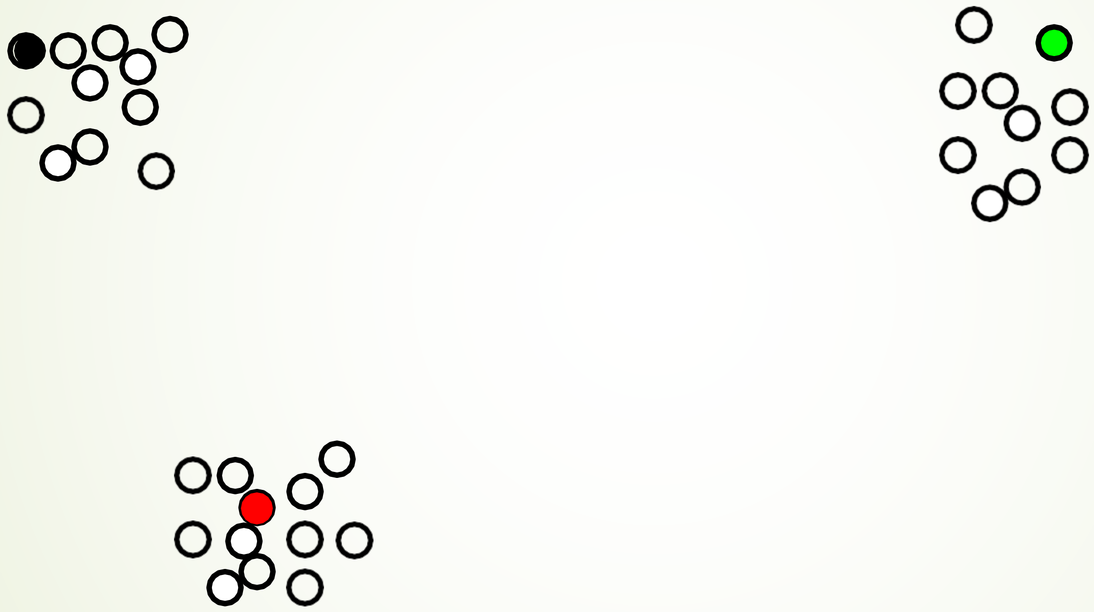
Some Gaussian are combined.....

Another Initialization Idea: Furthest Point Heuristic

Choose $\mathbf{c_1}$ arbitrarily (or at random).

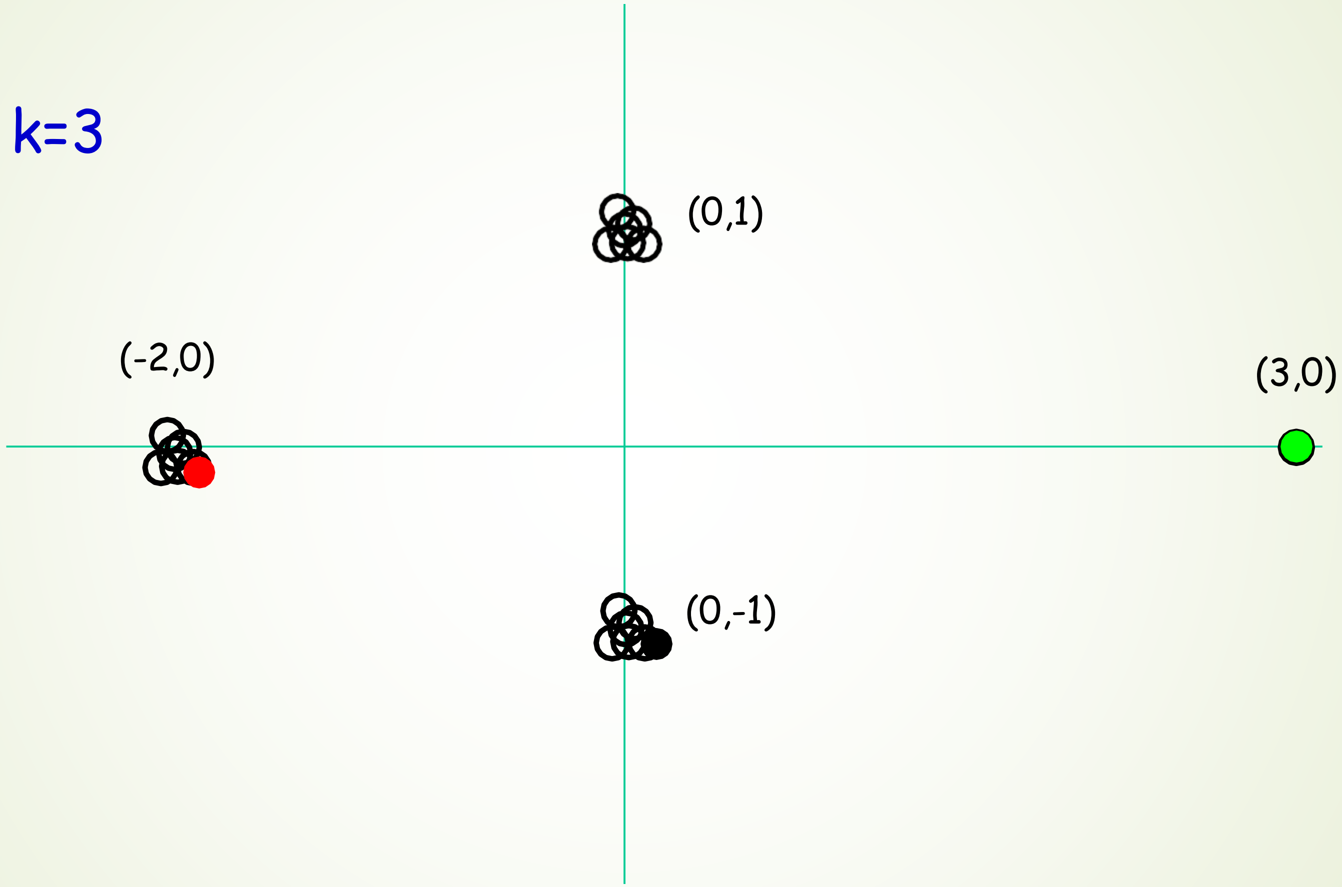
- For $j = 2, \ldots, k$
  - Pick $\mathbf{c_j}$ among datapoints $\mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^n}$ that is farthest from previously chosen $\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c}_{j-1}$

Furthest point heuristic does well on previous example



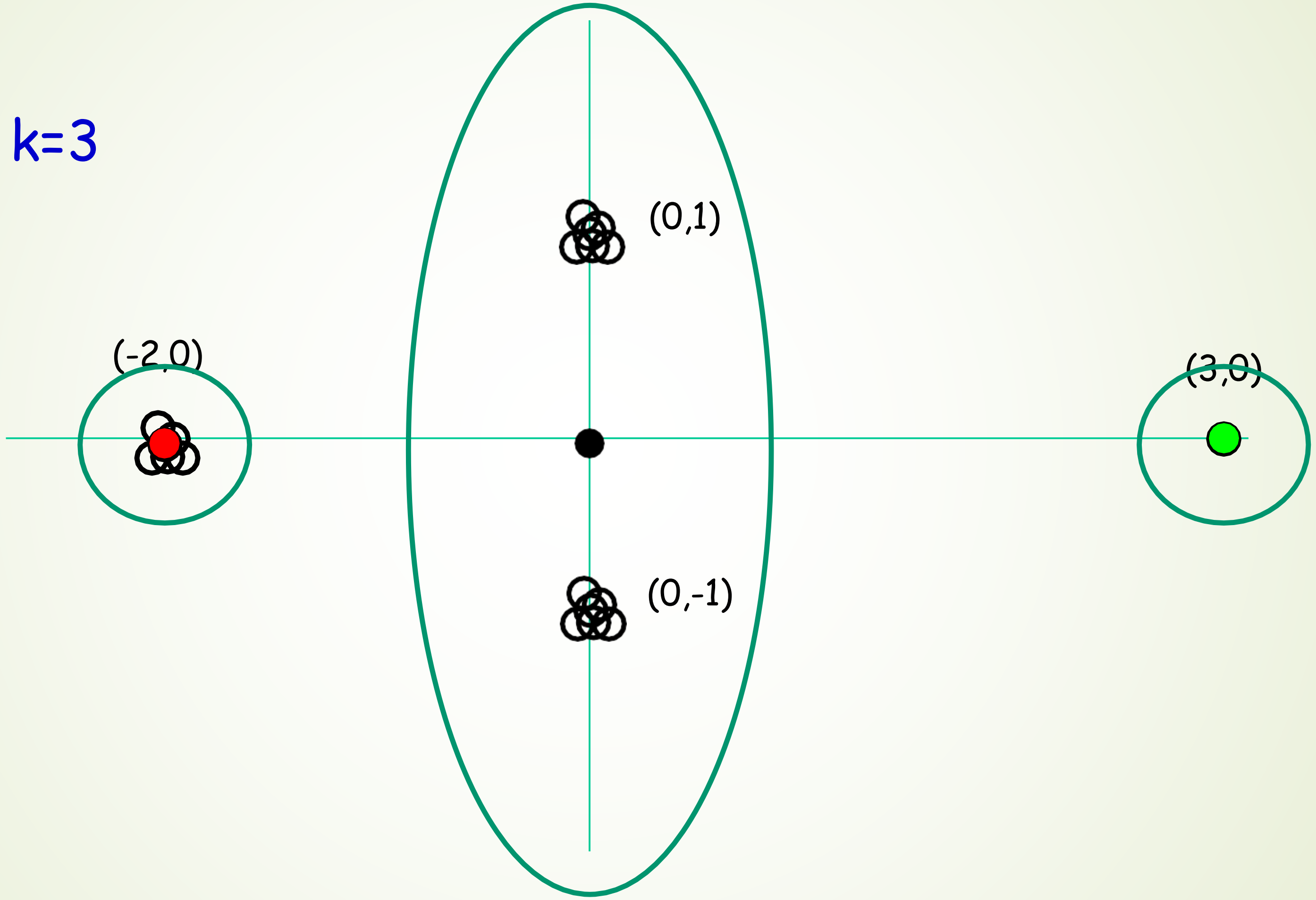
Furthest point initialization heuristic sensitive to outliers

Assume k=3



Furthest point initialization heuristic sensitive to outliers

Assume k=3



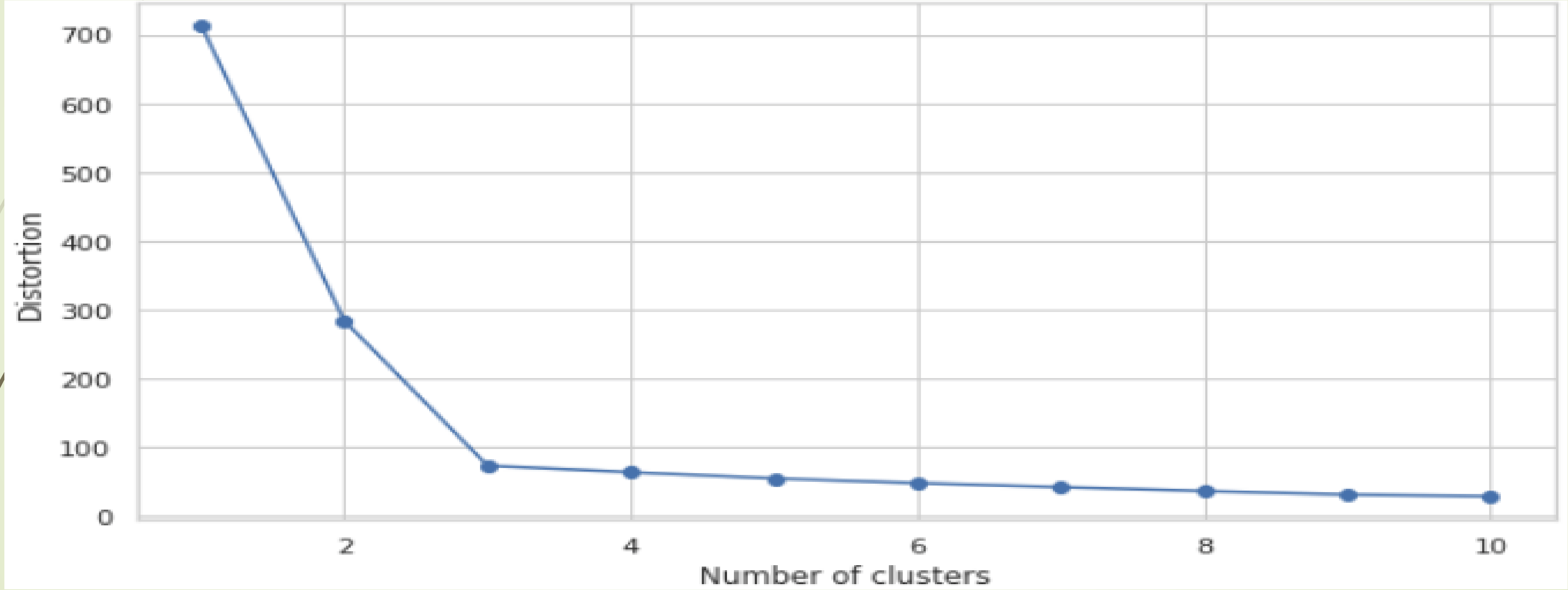
K-means ++ Fix

Elbow Method for Optimal Clustering

- What is the Elbow Method?
  - A graphical tool used to estimate the optimal number of clusters (k) in K-Means clustering.
  - Helps balance between a low number of clusters and minimizing within-cluster variance.
- Key Idea:
  - As the number of clusters increases, the within-cluster SSE (Sum of Squared Errors), also known as distortion, decreases.
  - Beyond a certain point, increasing the number of clusters provides diminishing returns in reducing distortion.

How the Elbow Method Works

- A higher k reduces distortion because points are closer to their centroids.
- However, after a certain number of clusters, adding more clusters does not significantly improve the model.
- The Elbow Point:
  - The point where the rate of reduction in distortion sharply decreases, forming an "elbow" in the plot.
  - This indicates the optimal number of clusters where the trade-off between model simplicity and clustering performance is balanced.



700
600
500
400
300
200
100
0
Distortion
2
4
6
8
10
Number of clusters

References

- https://utsavdesai26.medium.com/the-beginners-guide-to-clustering-in-machine-learning-331987a7ceaf
- https://www.researchgate.net/figure/Dendrogram-showing-the-hierarchical-clustering-result-from-average-linkage-clustering_fig15_262025025
- https://www.slideshare.net/slideshow/agglomarative/254548084
- https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8
- https://www.youtube.com/watch?v=XJ3194AmH40