Introduction to Data Science Topic-9

- Instructor: Professor Henry Horng-Shing Lu,

  Institute of Statistics, National Yang Ming Chiao Tung University, Taiwan

  Email: henryhslu@nycu.edu.tw
- WWW: http://misg.stat.nctu.edu.tw/hslu/course/DataScience.htm
- Classroom: ED B27 (新竹市大學路1001號工程四館B27教室)
- References:

  M. A. Pathak, Beginning Data Science with R, 2014, Springer-Verlag.

  K.-T. Tsai, Machine Learning for Knowledge Discovery with R: Methodologies for Modeling, Inference, and Prediction, 2021, Chapman and Hall/CRC.
- Evaluation: Homework: 70%, Term Project: 30%
- Office hours: By appointment

Course Outline

10 Topics and 10 Homeworks:

- **Introduction of Data Science**
- **Introduction of R and Python**
- **Cleaning Data into R and Python**
- **Data Visualization**
- **Exploratory Data Analysis**
- **Regression (Supervised Learning)**
- **Classification (Supervised Learning)**
- **Text Mining**
- **Clustering (Unsupervised Learning)**
- **Neural Network and Deep Learning**

Clustering with R

References:
https://www.kaggle.com/code/kashnitsky/topic-7-unsupervised-learning-pca-and-clustering
https://www.kaggle.com/code/dongdongxzoez/r-topic-9/

Clustering Methods

Supervised & Unsupervised Learning

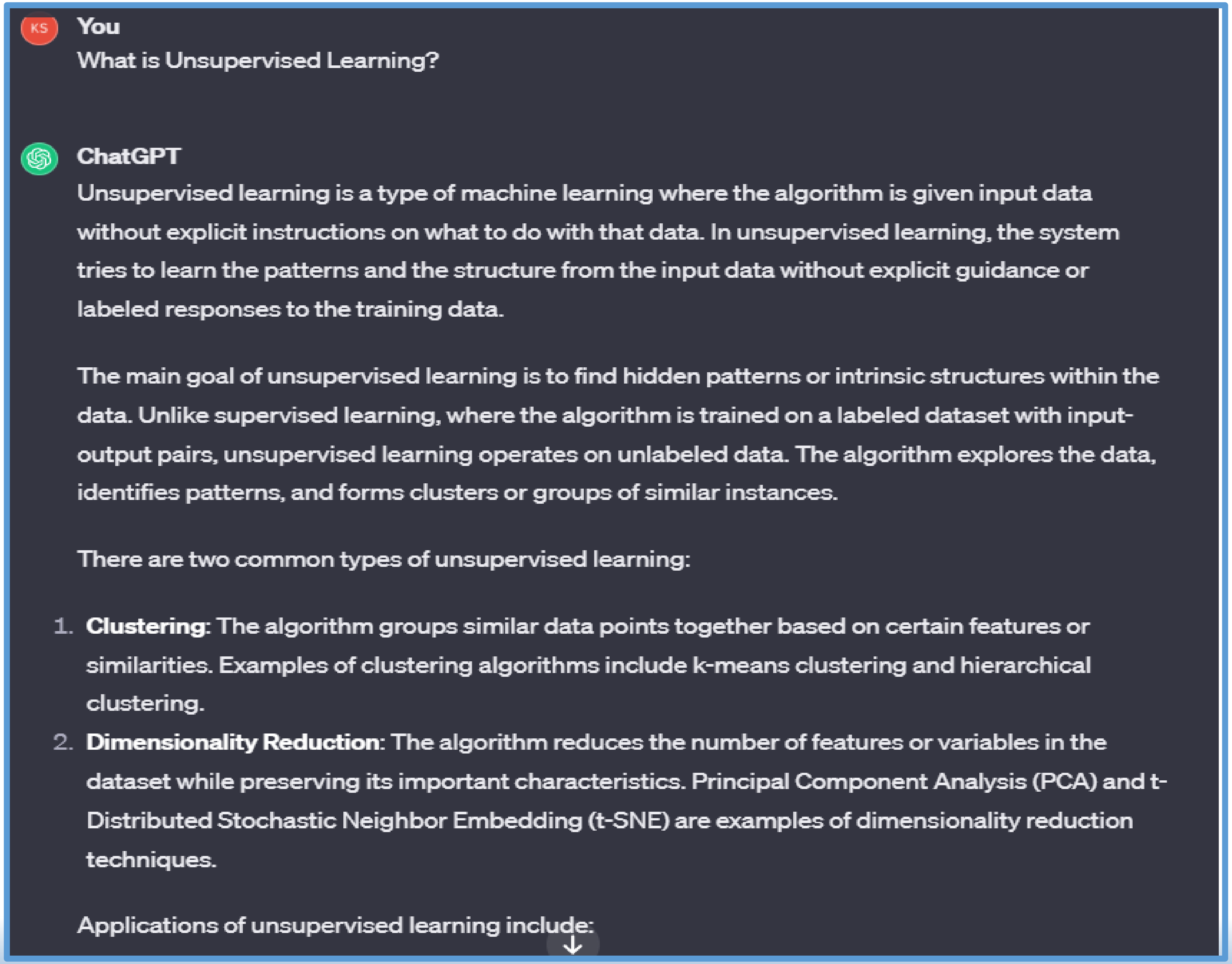
Supervised Learning	**Unsupervised Learning**
Supervised learning is when we teach or train the machine using data that is well labelled.	Unsupervised learning is to group unsorted information according to similarities, patterns, and differences without any prior training of data.
Advantages	**Advantages**
1. Supervised learning allows collecting of data and produces data output from previous experiences. 2. Helps to optimize performance criteria with the help of experience. 3. Supervised machine learning helps to solve various types of real world computation problems.	1. It can see what human minds cannot visualize and has widespread applications in real time. 2. There is lesser complexity compared to the supervised learning task. Here, no one is required to interpret the associated labels and hence it holds lesser complexities. 3. It is reasonably easier to obtain unlabeled data.
Disadvantages	Disadvantages
1. Classifying big data can be challenging. 2. Training for supervised learning needs a lot of computation time. So, it requires a lot of time.	1. Classifying big data can be challenging. 2. Training for supervised learning needs a lot of computation time. So, it requires a lot of time.

Unsupervised Learning

- Unsupervised learning is a type of machine learning that looks for previously undetected patterns in a data set with no pre-existing labels and with a minimum of human supervision.

- In contrast, supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. Each example is a pair consisting of an input object and a output value.

https://en.wikipedia.org/wiki/Unsupervised_learning



You

What is Unsupervised Learning?

ChatGPT

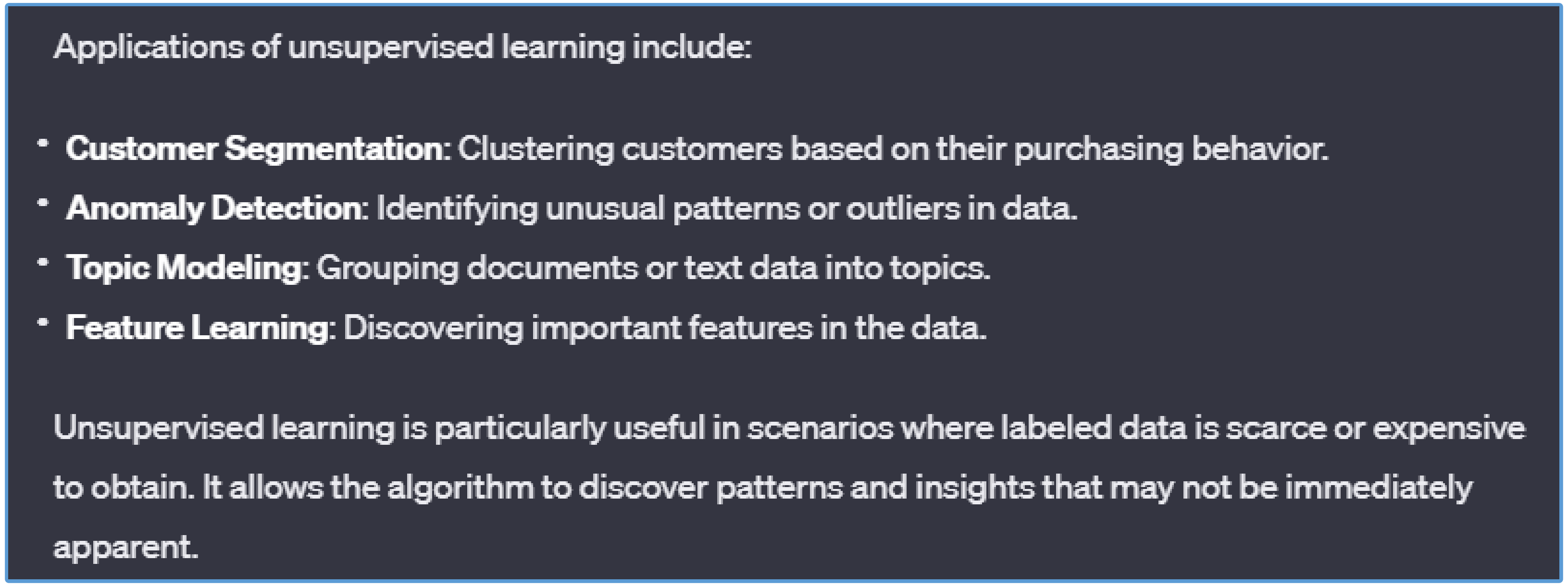
Unsupervised learning is a type of machine learning where the algorithm is given input data without explicit instructions on what to do with that data. In unsupervised learning, the system tries to learn the patterns and the structure from the input data without explicit guidance or labeled responses to the training data.

The main goal of unsupervised learning is to find hidden patterns or intrinsic structures within the data. Unlike supervised learning, where the algorithm is trained on a labeled dataset with input-output pairs, unsupervised learning operates on unlabeled data. The algorithm explores the data, identifies patterns, and forms clusters or groups of similar instances.

There are two common types of unsupervised learning:

1. **Clustering**: The algorithm groups similar data points together based on certain features or similarities. Examples of clustering algorithms include k-means clustering and hierarchical clustering.
2. **Dimensionality Reduction**: The algorithm reduces the number of features or variables in the dataset while preserving its important characteristics. Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) are examples of dimensionality reduction techniques.

Applications of unsupervised learning include:



Applications of unsupervised learning include:

- **Customer Segmentation:** Clustering customers based on their purchasing behavior.
- **Anomaly Detection:** Identifying unusual patterns or outliers in data.
- **Topic Modeling:** Grouping documents or text data into topics.
- **Feature Learning:** Discovering important features in the data.

Unsupervised learning is particularly useful in scenarios where labeled data is scarce or expensive to obtain. It allows the algorithm to discover patterns and insights that may not be immediately apparent.

Unsupervised Learning

Unsupervised learning is classified into two categories of algorithms:

- **Clustering**

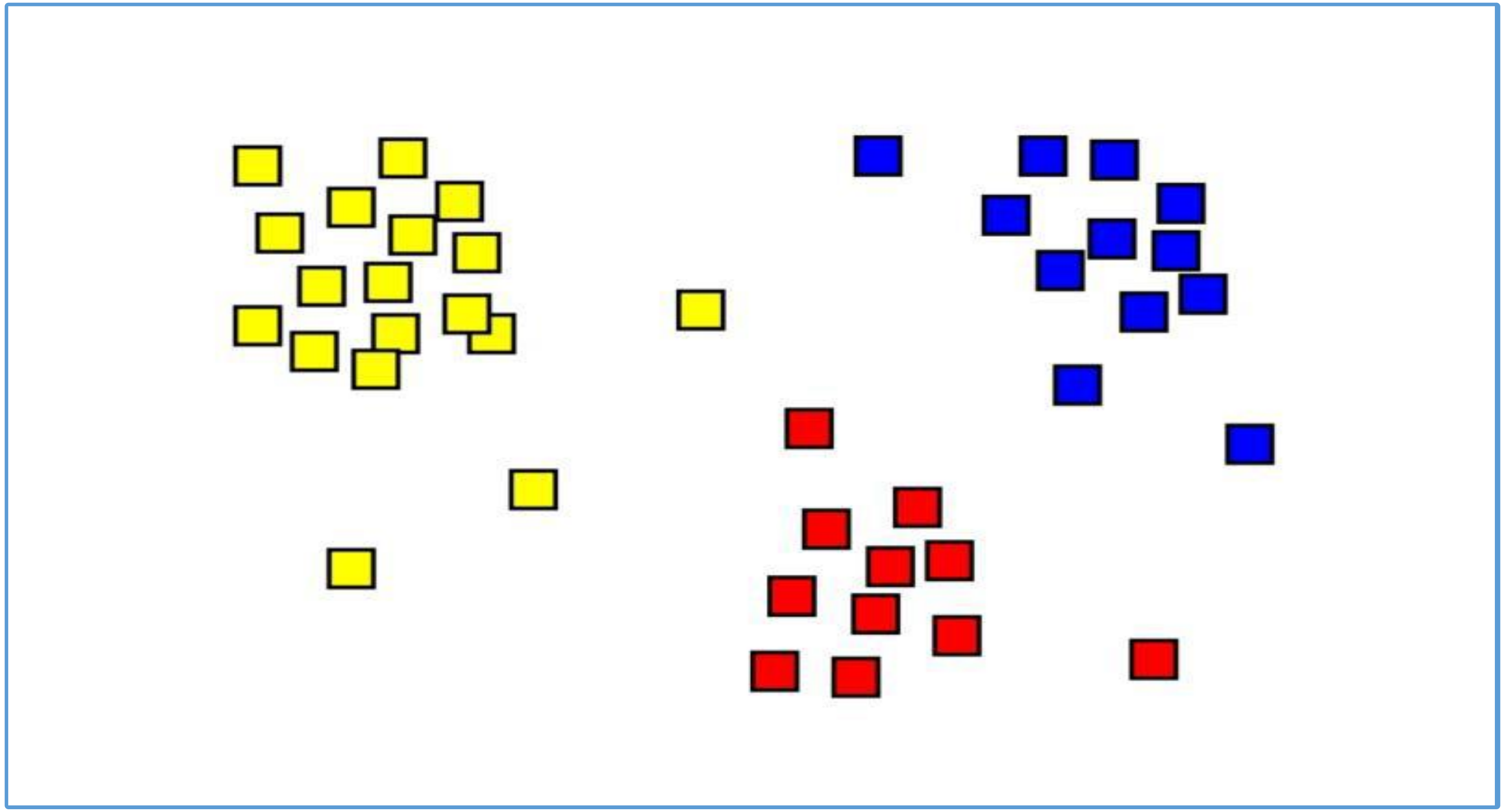
A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

- **Association**

An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy A also tend to buy B.

https://www.geeksforgeeks.org/supervised unsupervised learning/

Introduction Clustering



“Clustering is to make the similar thing together by data feature distance or distribution”

Introduction Clustering

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

Clustering Types

1. Hierarchical Clustering
2. K means Clustering
3. Clustering with Dimension Reduction, such as Principal Component Analysis

Hierarchical clustering

- In order to decide which clusters should be combined, or where a cluster should be split, a measure of dissimilarity between sets of observations is required.

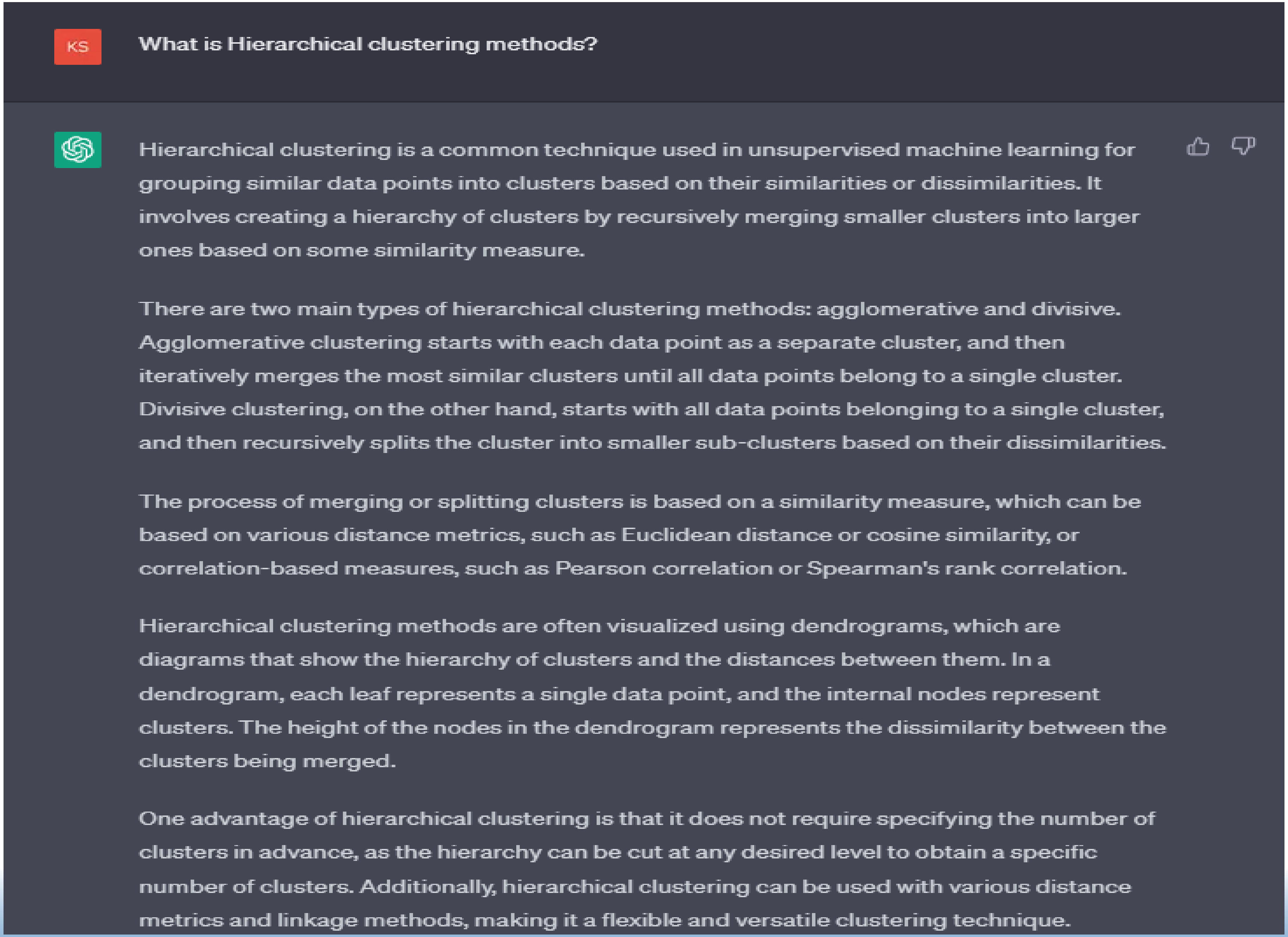
- In most methods of hierarchical clustering, this is achieved by use of an appropriate metric (a measure of distance between pairs of observations), and a linkage criterion which specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets.

https://en.wikipedia.org/wiki/Hierarchical_clustering

Hierarchical Clustering Approaches

Hierarchical clustering (also called hierarchical cluster analysis or HCA) is a method of cluster analysis that seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two categories:

- Agglomerative hierarchical methods: This is a “bottom-up" approach: Each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- Divisive hierarchical methods: This is a “top-down" approach: All observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.
- Both hierarchical methods' results can be displayed in a dendrogram



What is Hierarchical clustering methods?

Hierarchical clustering is a common technique used in unsupervised machine learning for grouping similar data points into clusters based on their similarities or dissimilarities. It involves creating a hierarchy of clusters by recursively merging smaller clusters into larger ones based on some similarity measure.

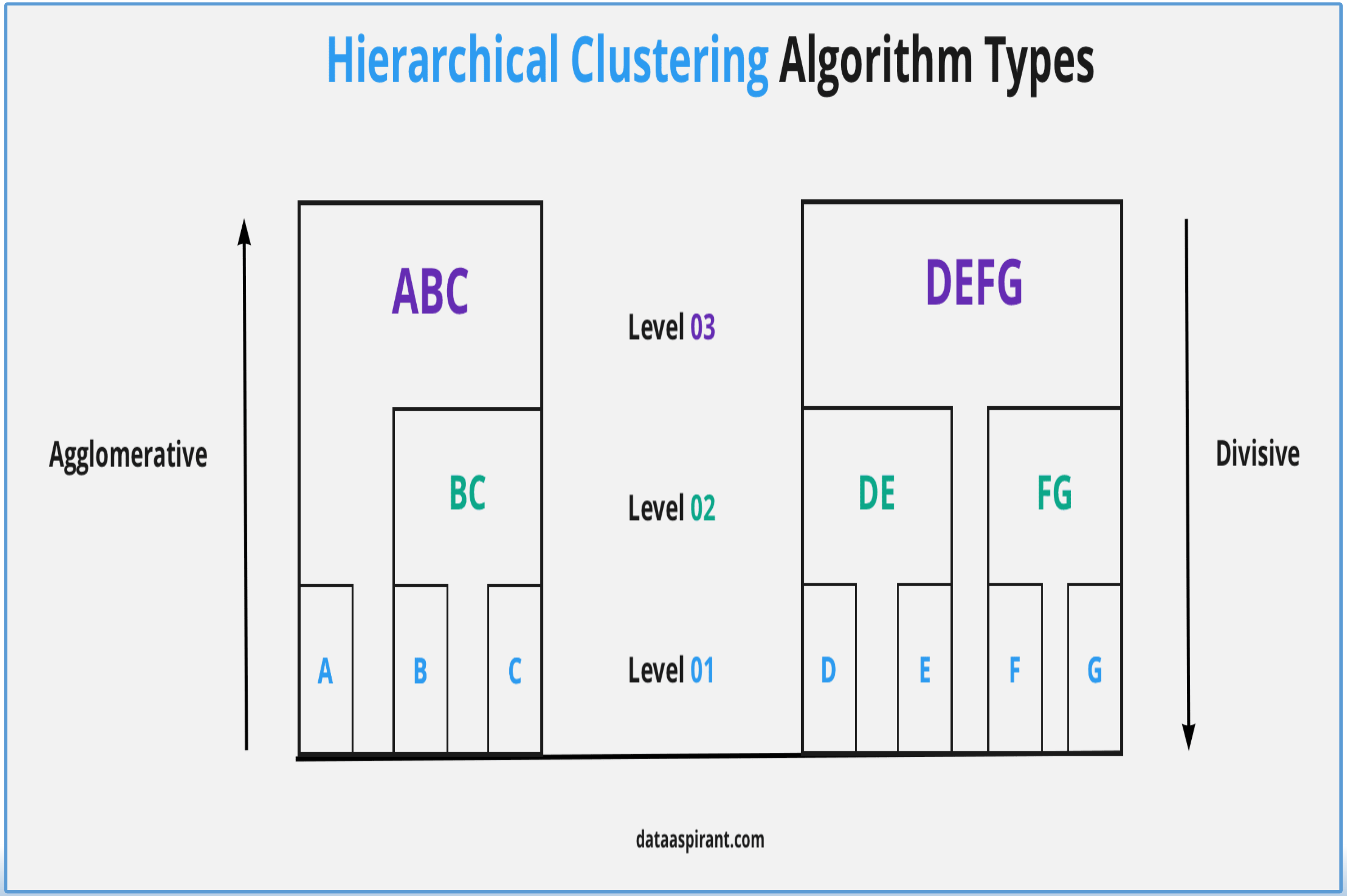
There are two main types of hierarchical clustering methods: agglomerative and divisive. Agglomerative clustering starts with each data point as a separate cluster, and then iteratively merges the most similar clusters until all data points belong to a single cluster. Divisive clustering, on the other hand, starts with all data points belonging to a single cluster, and then recursively splits the cluster into smaller sub-clusters based on their dissimilarities.

The process of merging or splitting clusters is based on a similarity measure, which can be based on various distance metrics, such as Euclidean distance or cosine similarity, or correlation-based measures, such as Pearson correlation or Spearman's rank correlation.

Hierarchical clustering methods are often visualized using dendrograms, which are diagrams that show the hierarchy of clusters and the distances between them. In a dendrogram, each leaf represents a single data point, and the internal nodes represent clusters. The height of the nodes in the dendrogram represents the dissimilarity between the clusters being merged.

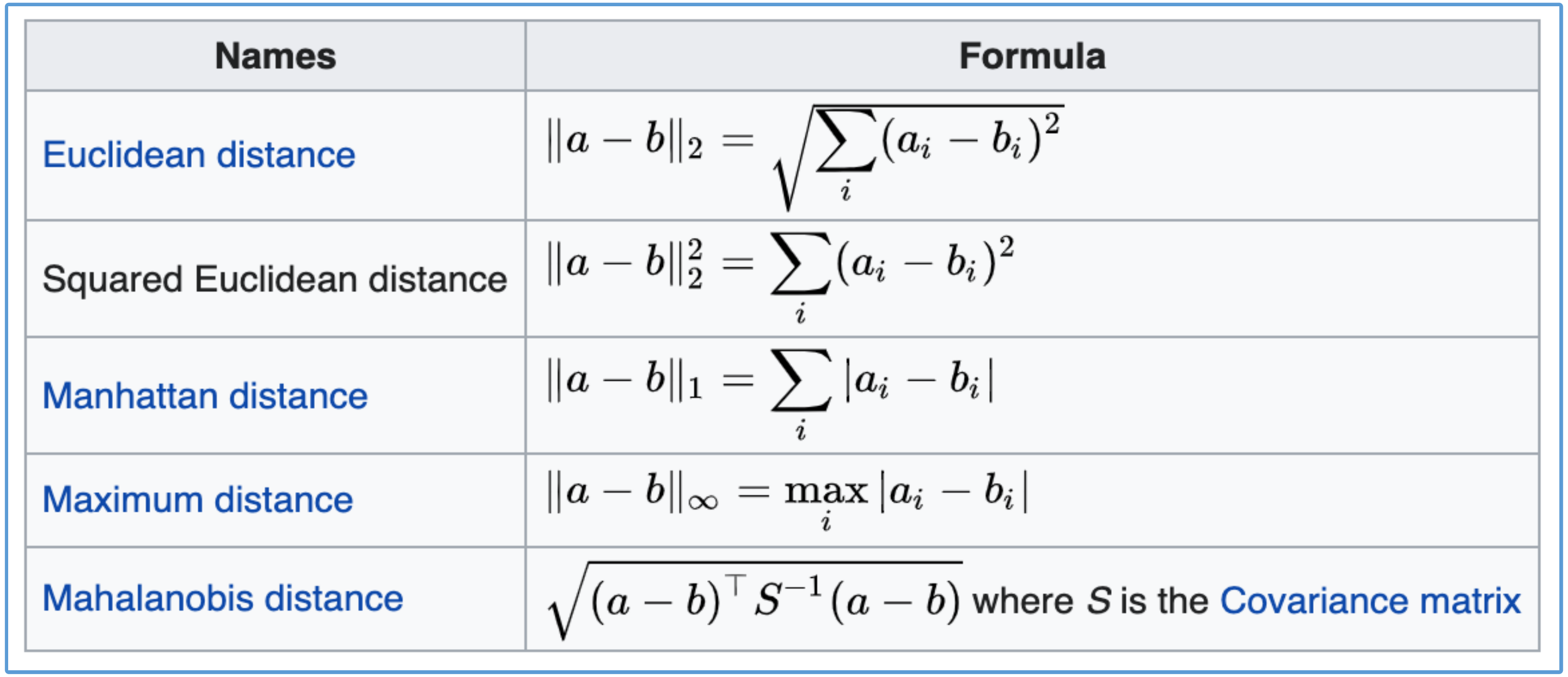
One advantage of hierarchical clustering is that it does not require specifying the number of clusters in advance, as the hierarchy can be cut at any desired level to obtain a specific number of clusters. Additionally, hierarchical clustering can be used with various distance metrics and linkage methods, making it a flexible and versatile clustering technique.

Hierarchical Clustering Types



https://i2.wp.com/dataaspirant.com/wp-content/uploads/2020/12/5-Hierarchical-Clustering-Types-Agglomerative-and-Divisive.png?ssl=1

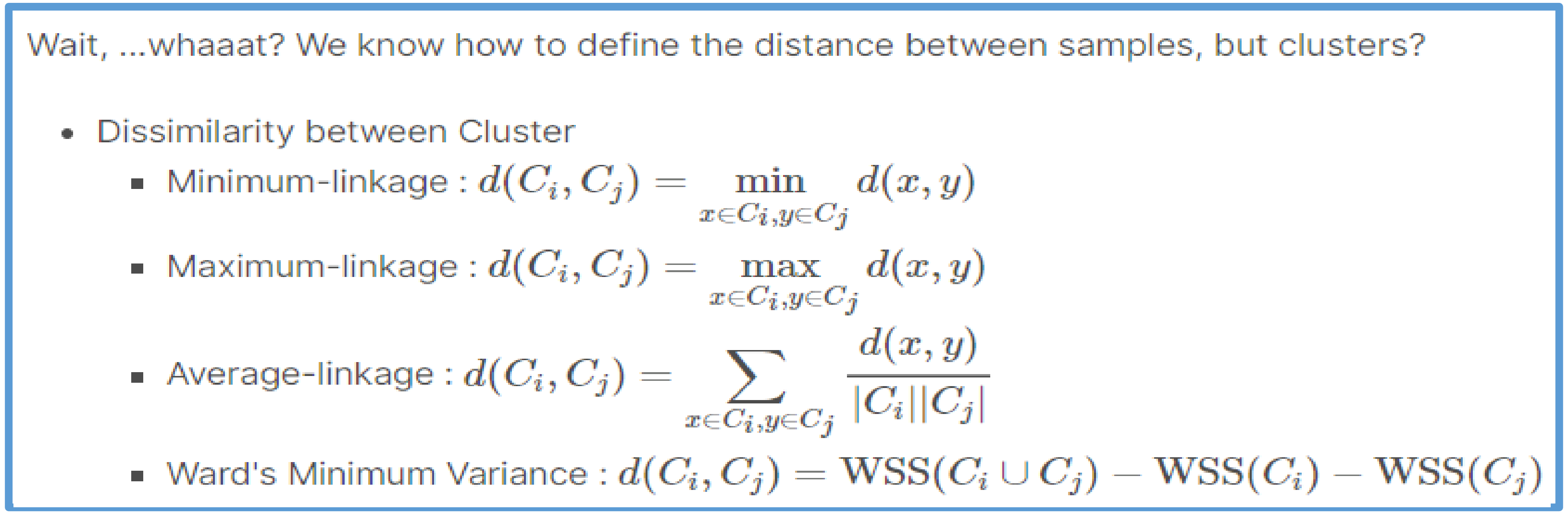
Distance



Names	Formula
Euclidean distance	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidean distance	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan distance	$\|a-b\|_1 = \sum_i \|a_i - b_i\|$
Maximum distance	$\|a-b\|_\infty = \max_i \|a_i - b_i\|$
Mahalanobis distance	$\sqrt{(a-b)^\top S^{-1}(a-b)}$ where S is the Covariance matrix

https://mmuratarat.github.io/2020-05-23/hierarchical_clustering

Distance Between Clusters



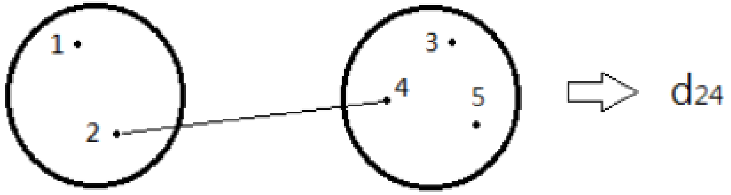
Wait, ...whaaat? We know how to define the distance between samples, but clusters?

- Dissimilarity between Cluster
  - Minimum-linkage : $d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$
  - Maximum-linkage : $d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$
  - Average-linkage : $d(C_i, C_j) = \sum_{x \in C_i, y \in C_j} \frac{d(x, y)}{|C_i||C_j|}$
  - Ward's Minimum Variance : $d(C_i, C_j) = \mathrm{WSS}(C_i \cup C_j) - \mathrm{WSS}(C_i) - \mathrm{WSS}(C_j)$

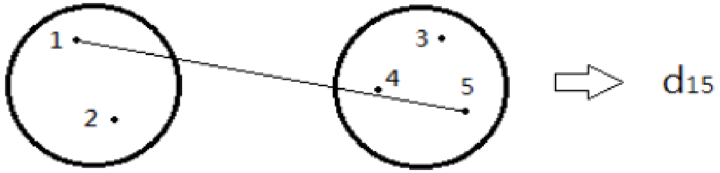
Hierarchical clustering methods: Linkage methods

There are three types of linkage methods for calculating the distance between two groups of objects

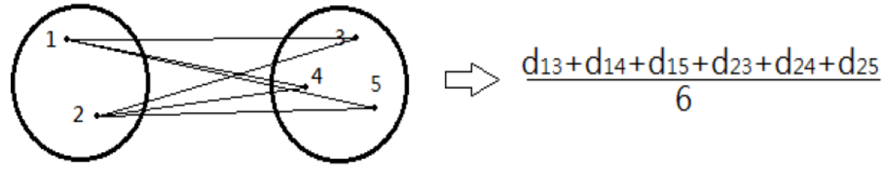
Single Linkage (Minimum distance or Nearest distance)



Complete Linkage (Maximum distance or Farthest neighbor)



Average linkage (average distance) :



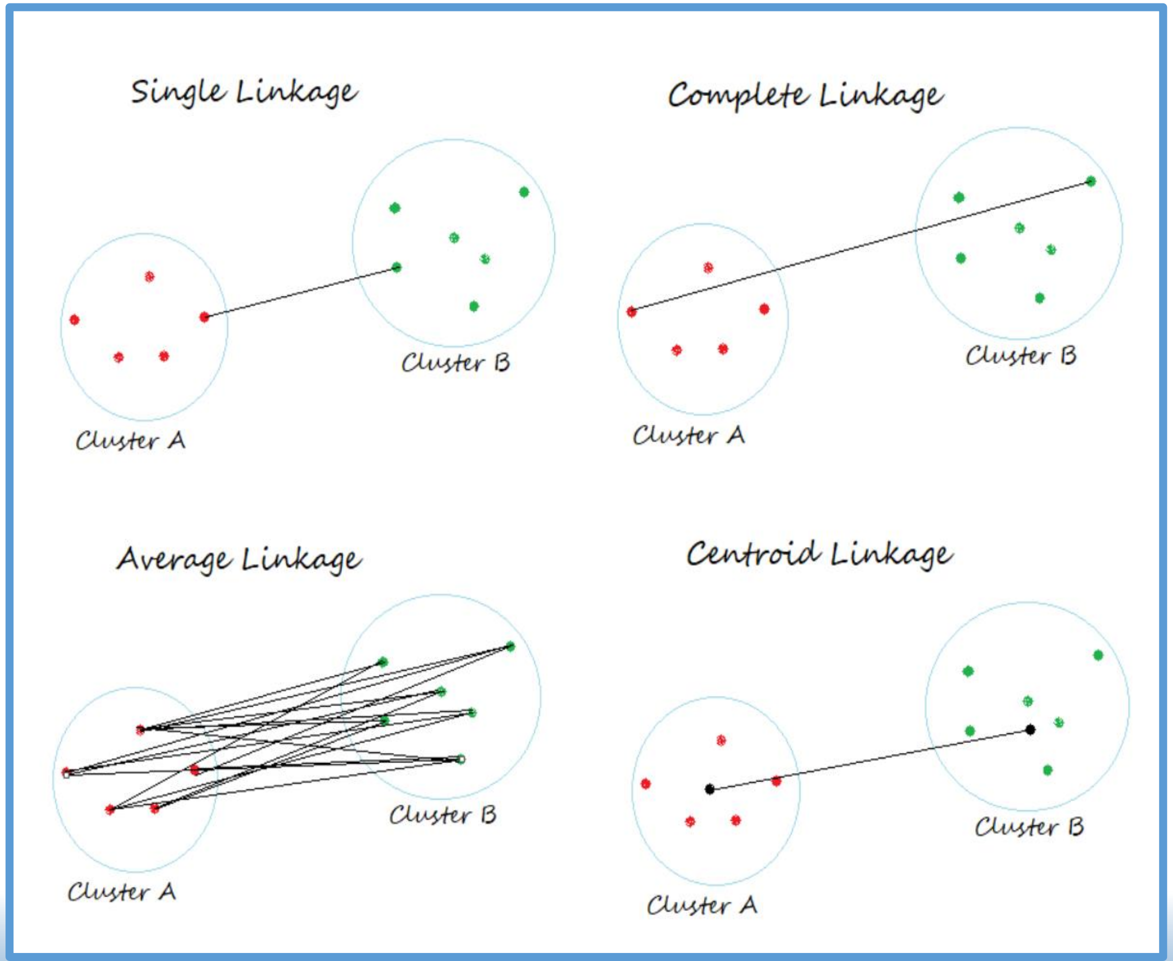
Applied Multivariate Statistical Analysis (6th Edition) by Richard A. Johnson , Dean W.Wichern

Hierarchical clustering methods: Linkage methods

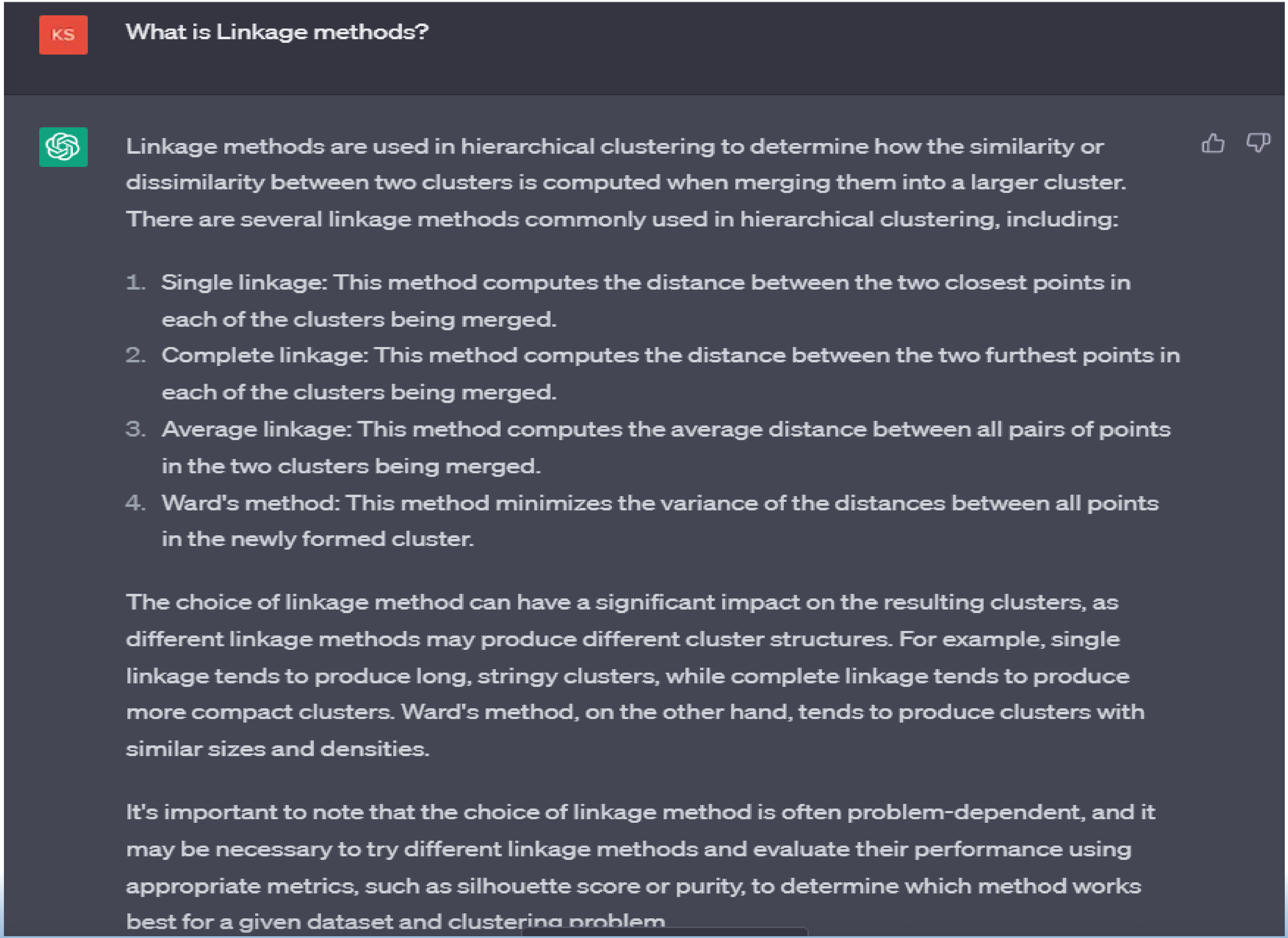
Single Linkage: - The distance between two clusters is the minimum distance between members of the two clusters.

Complete Linkage: - The distance between two clusters is the maximum distance between members of the two clusters

Average Linkage:- The distance between two clusters is the average of all distances between members of the two clusters.



https://www.analyticsvidhya.com/blog/2021/06/single-link-hierarchical-clustering-clearly-explained/



What is Linkage methods?

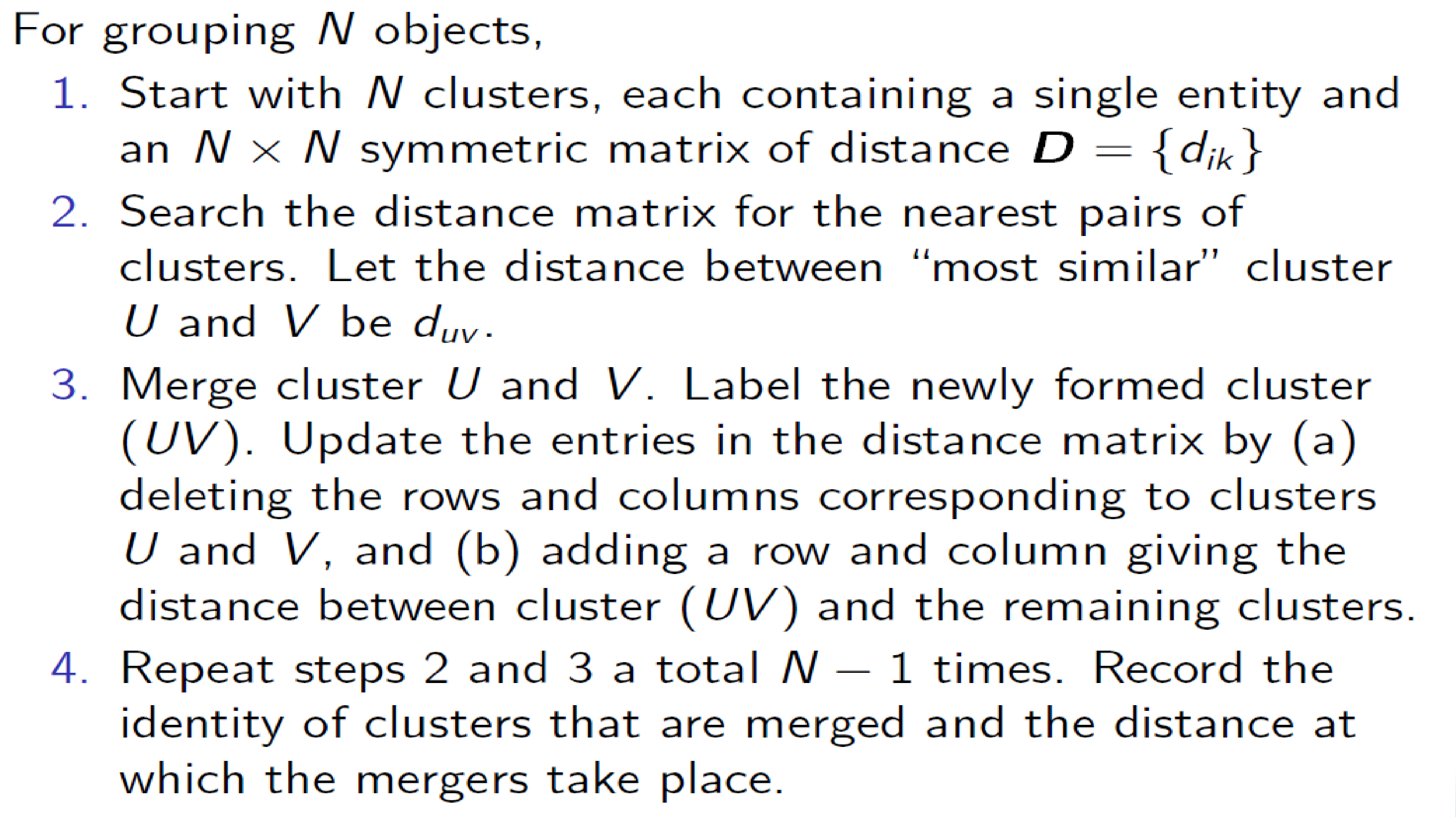
Linkage methods are used in hierarchical clustering to determine how the similarity or dissimilarity between two clusters is computed when merging them into a larger cluster. There are several linkage methods commonly used in hierarchical clustering, including:

1. Single linkage: This method computes the distance between the two closest points in each of the clusters being merged.
2. Complete linkage: This method computes the distance between the two furthest points in each of the clusters being merged.
3. Average linkage: This method computes the average distance between all pairs of points in the two clusters being merged.
4. Ward's method: This method minimizes the variance of the distances between all points in the newly formed cluster.

The choice of linkage method can have a significant impact on the resulting clusters, as different linkage methods may produce different cluster structures. For example, single linkage tends to produce long, stringy clusters, while complete linkage tends to produce more compact clusters. Ward's method, on the other hand, tends to produce clusters with similar sizes and densities.

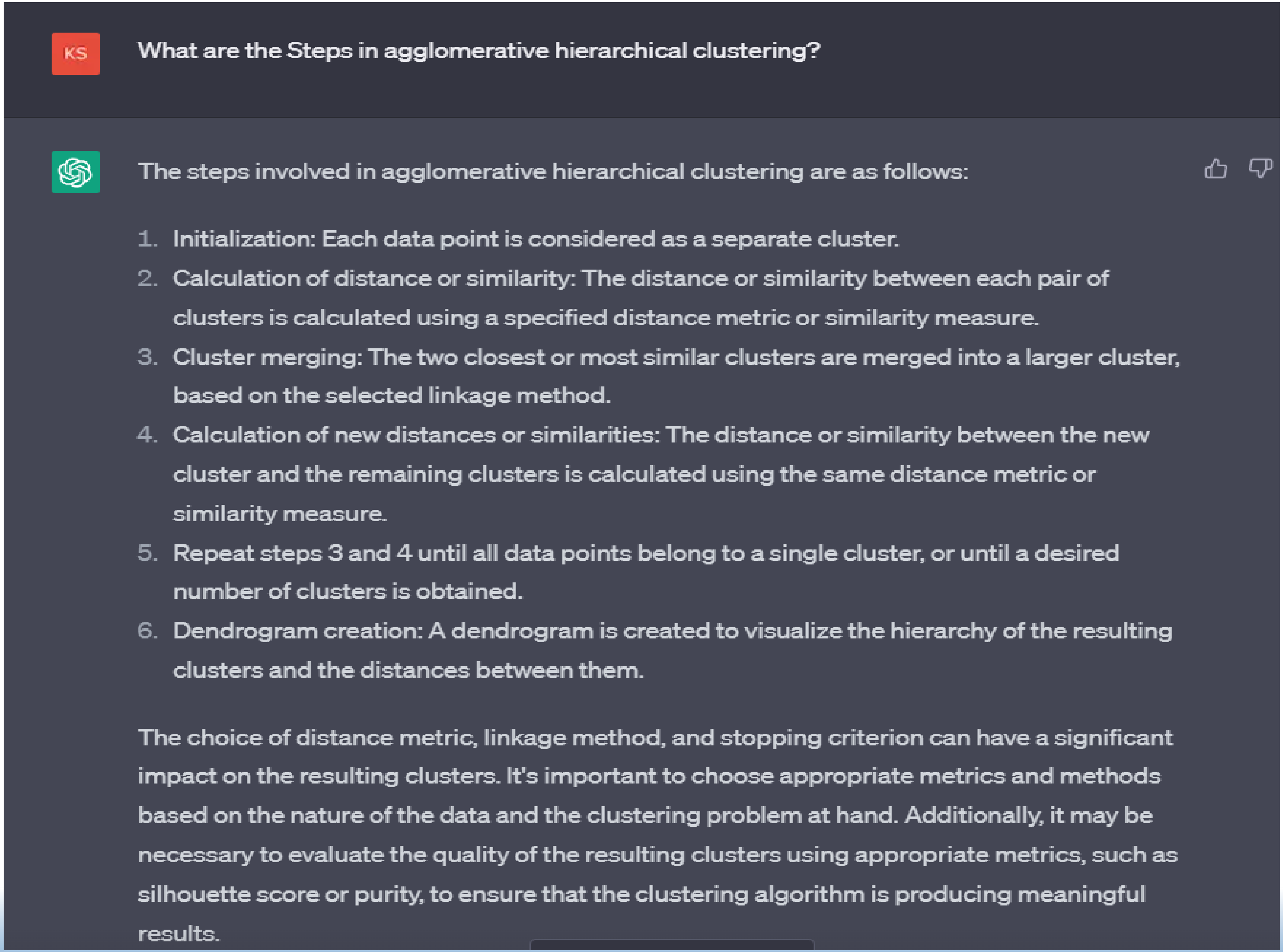
It's important to note that the choice of linkage method is often problem-dependent, and it may be necessary to try different linkage methods and evaluate their performance using appropriate metrics, such as silhouette score or purity, to determine which method works best for a given dataset and clustering problem.

Steps in agglomerative hierarchical clustering



For grouping N objects,

1. Start with N clusters, each containing a single entity and an $N \times N$ symmetric matrix of distance $\mathbf{D} = \{d_{ik}\}$
2. Search the distance matrix for the nearest pairs of clusters. Let the distance between "most similar" cluster U and V be d_{uv}.
3. Merge cluster U and V. Label the newly formed cluster (UV). Update the entries in the distance matrix by (a) deleting the rows and columns corresponding to clusters U and V, and (b) adding a row and column giving the distance between cluster (UV) and the remaining clusters.
4. Repeat steps 2 and 3 a total $N - 1$ times. Record the identity of clusters that are merged and the distance at which the mergers take place.



What are the Steps in agglomerative hierarchical clustering?

The steps involved in agglomerative hierarchical clustering are as follows:

1. Initialization: Each data point is considered as a separate cluster.
2. Calculation of distance or similarity: The distance or similarity between each pair of clusters is calculated using a specified distance metric or similarity measure.
3. Cluster merging: The two closest or most similar clusters are merged into a larger cluster, based on the selected linkage method.
4. Calculation of new distances or similarities: The distance or similarity between the new cluster and the remaining clusters is calculated using the same distance metric or similarity measure.
5. Repeat steps 3 and 4 until all data points belong to a single cluster, or until a desired number of clusters is obtained.
6. Dendrogram creation: A dendrogram is created to visualize the hierarchy of the resulting clusters and the distances between them.

The choice of distance metric, linkage method, and stopping criterion can have a significant impact on the resulting clusters. It's important to choose appropriate metrics and methods based on the nature of the data and the clustering problem at hand. Additionally, it may be necessary to evaluate the quality of the resulting clusters using appropriate metrics, such as silhouette score or purity, to ensure that the clustering algorithm is producing meaningful results.

Example: Single linkage

$$d_{(UV)W} = \min\{d_{UW}, d_{VW}\}$$

$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & \boxed{\mathbf{2}} & 8 & 0 \end{bmatrix} \end{array}$$

merge 3, 5 $\rightarrow$ (3 5)

$$\begin{array}{cc} & \begin{array}{cccc} 1 & 2 & (35) & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ (35) \\ 4 \end{array} & \begin{bmatrix} 0 & & & \\ 9 & 0 & & \\ \boxed{\mathbf{3}} & 7 & 0 & \\ 6 & 5 & 8 & 0 \end{bmatrix} \end{array}$$

merge 1, (35) $\rightarrow$ (135)

$$\begin{array}{cc} & \begin{array}{ccc} (135) & 2 & 4 \end{array} \\ \begin{array}{c} (135) \\ 2 \\ 4 \end{array} & \begin{bmatrix} 0 & & \\ 7 & 0 & \\ 6 & \boxed{\mathbf{5}} & 0 \end{bmatrix} \end{array}$$

merge 2, 4 $\rightarrow$ (24)

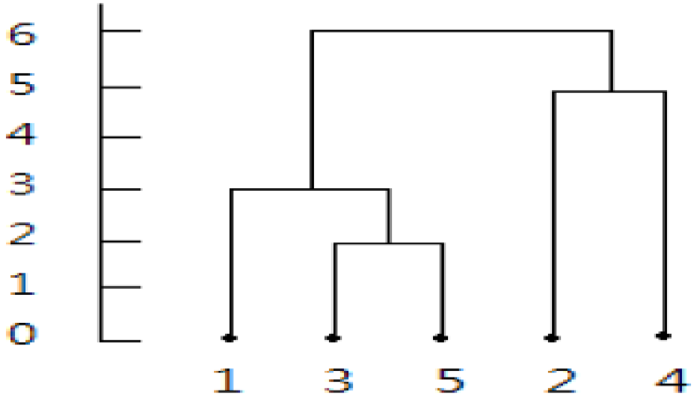
$$\begin{array}{cc} & \begin{array}{cc} (135) & (24) \end{array} \\ \begin{array}{c} (135) \\ (24) \end{array} & \begin{bmatrix} 0 & \\ \boxed{\mathbf{6}} & 0 \end{bmatrix} \end{array}$$

merge (135), (24) $\rightarrow$ (13524)

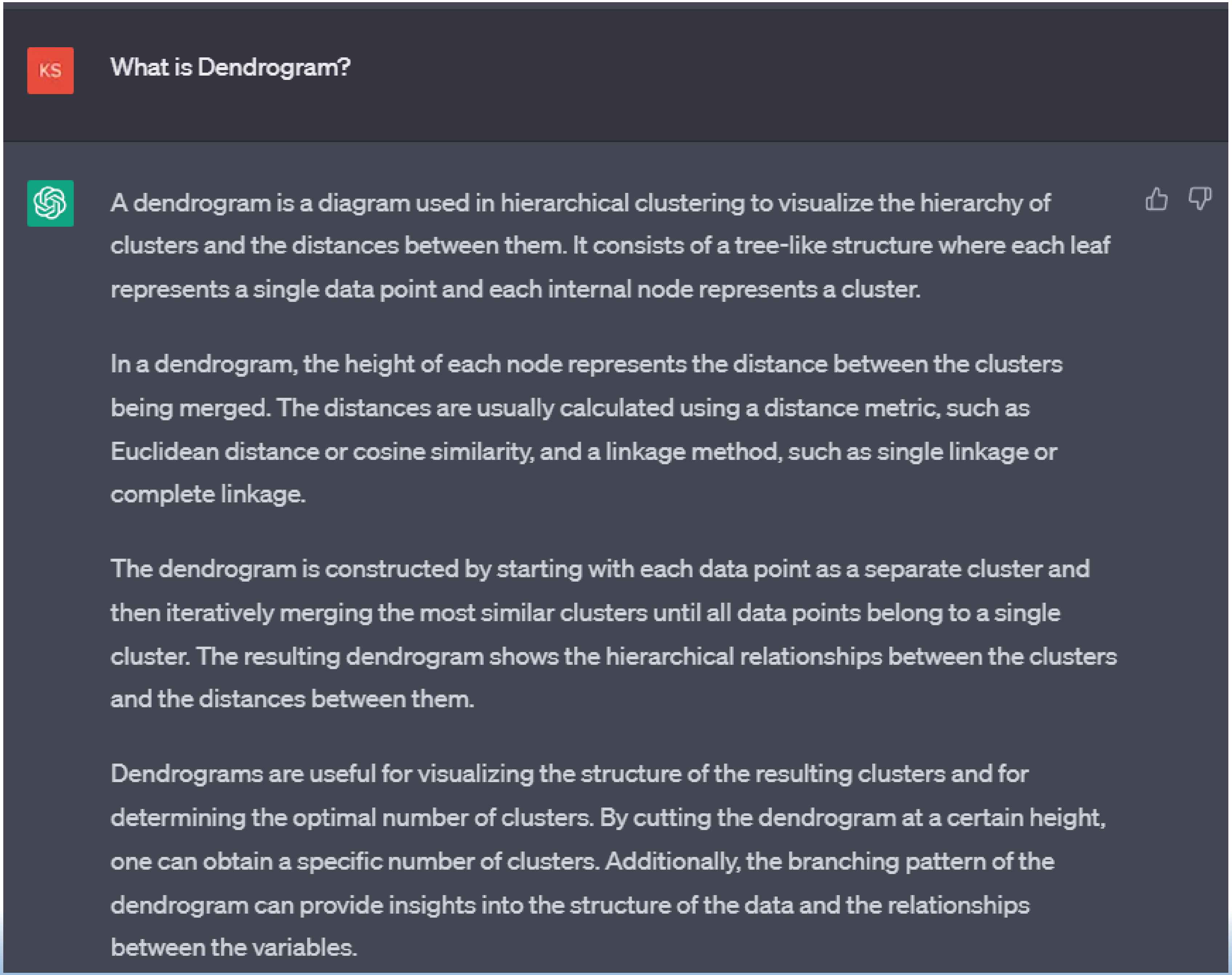
Applied Multivariate Statistical Analysis (6th Edition) by Richard A. Johnson , Dean W.Wichern

Dendrogram

- Hierarchical clustering provides with clusters of every size: where to **cut** the **dendrogram** is user-determined.
- We can then make the dendrogram showing merging.
- The y-axis represents the distance between the groups merged at that point.



Note: The cluster formed by the single linkage will be unchanged by any assignment of distance that gives the same relative orderings as the initial distance.



What is Dendrogram?

A dendrogram is a diagram used in hierarchical clustering to visualize the hierarchy of clusters and the distances between them. It consists of a tree-like structure where each leaf represents a single data point and each internal node represents a cluster.

In a dendrogram, the height of each node represents the distance between the clusters being merged. The distances are usually calculated using a distance metric, such as Euclidean distance or cosine similarity, and a linkage method, such as single linkage or complete linkage.

The dendrogram is constructed by starting with each data point as a separate cluster and then iteratively merging the most similar clusters until all data points belong to a single cluster. The resulting dendrogram shows the hierarchical relationships between the clusters and the distances between them.

Dendrograms are useful for visualizing the structure of the resulting clusters and for determining the optimal number of clusters. By cutting the dendrogram at a certain height, one can obtain a specific number of clusters. Additionally, the branching pattern of the dendrogram can provide insights into the structure of the data and the relationships between the variables.

Example: complete linkage

$$d_{(UV)W} = \max\{d_{UW}, d_{VW}\}$$

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & & & & \\ 2 & 9 & 0 & & & \\ 3 & 3 & 7 & 0 & & \\ 4 & 6 & 5 & 9 & 0 & \\ 5 & 11 & 10 & \boxed{\mathbf{2}} & 8 & 0 \end{array}$$

merge 3, 5 → (35)

$$\begin{array}{c|cccc} & 1 & 2 & (35) & 4 \\ \hline 1 & 0 & & & \\ 2 & 9 & 0 & & \\ (35) & 11 & 10 & 0 & \\ 4 & 6 & \boxed{\mathbf{5}} & 8 & 0 \end{array}$$

merge 2, 4 → (24)

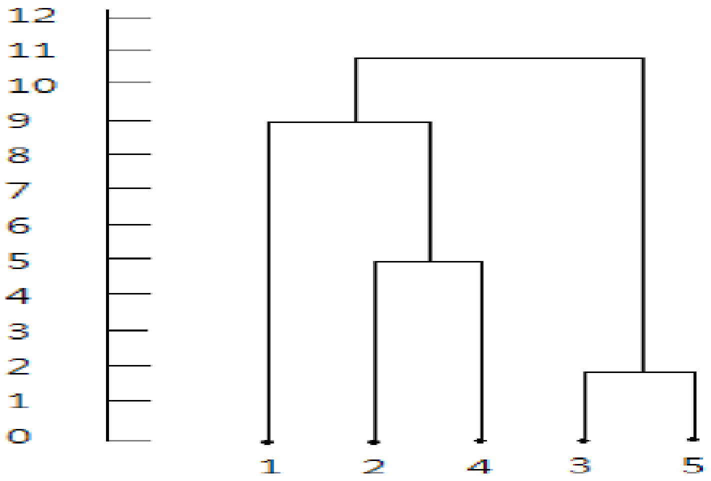
$$\begin{array}{c|ccc} & 1 & (24) & (35) \\ \hline 1 & 0 & & \\ (24) & \boxed{\mathbf{9}} & 0 & \\ (35) & 11 & 10 & 0 \end{array}$$

merge 1, (24) → (124)

$$\begin{array}{c|cc} & (124) & (35) \\ \hline (124) & 0 & \\ (35) & \boxed{\mathbf{11}} & 0 \end{array}$$

merge (124), (35) → (12435)

Dendrogram



1. A “new” assignment of distances that have the same relative orderings as the initial distances will not change the configuration of the complete linkage clusters.
2. When grouping variables and the sample correlation is used as similarity measures, the complete linkage takes the smallest similarity between members of corresponding clusters.

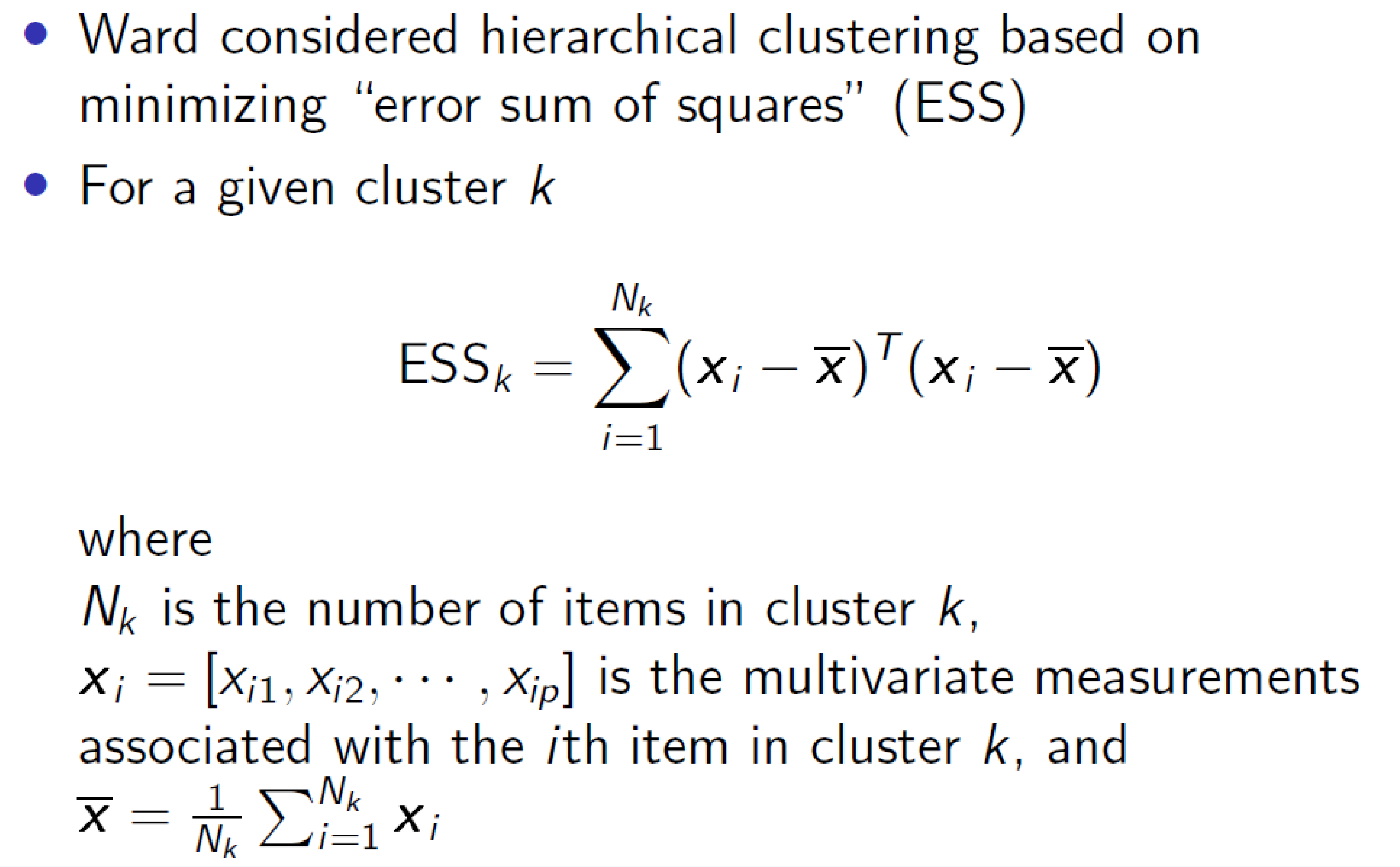
Average linkage

$$d_{(UV)W} = \frac{\sum_{i \in (UV)} \sum_{k \in W} d_{ik}}{N_{(UV)} N_W}$$

where $\boldsymbol{D} = \{d_{ik}\}$ = the distance of matrix
$N_{(UV)}$ = the number of items in cluster (UV)
N_W = the number of items in cluster W

Note: For average linkage clustering, changes in the assignment of distances can affect the arrangement of the final configuration of clusters, even through the changes preserve relative orderings.

Ward's hierarchical clustering method



- Ward considered hierarchical clustering based on minimizing "error sum of squares" (ESS)
- For a given cluster k

$$\mathrm{ESS}_k = \sum_{i=1}^{N_k} (\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T (\boldsymbol{x}_i - \overline{\boldsymbol{x}})$$

where
N_k is the number of items in cluster k,
$\boldsymbol{x}_i = [x_{i1}, x_{i2}, \cdots, x_{ip}]$ is the multivariate measurements associated with the ith item in cluster k, and
$\overline{\boldsymbol{x}} = \frac{1}{N_k} \sum_{i=1}^{N_k} \boldsymbol{x}_i$

Steps of Ward's method:

1. Initially, each cluster consists of a single item. Then

   $$\text{ESS}_k = 0, k = 1, \cdots, N \text{ (total number of items)}$$

   So the sum of all ESS_k's

   $$\text{ESS} = \text{ESS}_1 + \text{ESS}_2 + \cdots + \text{ESS}_N = 0$$

2. The union of every possible pair of clusters is considered, and the two clusters whose combination results in the smallest increase in ESS are jointed.
3. Repeat step 2.
4. At the extreme, all the clusters are combined in a single cluster of N items, the value of

   $$\text{ESS} = \sum_{i=1}^{N} (\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T (\boldsymbol{x}_i - \overline{\boldsymbol{x}})$$

- The results of Ward's method can be displayed as a dendrogram. The vertical axis gives the values of ESS at which the mergers occur.

Example

item	X_1	X_2
1	5	3
2	-1	1
3	1	-2
4	-3	-2

Step 1: $$\underset{(=0)}{\overset{1}{\mathrm{ESS}_1}} + \underset{(=0)}{\overset{2}{\mathrm{ESS}_2}} + \underset{(=0)}{\overset{3}{\mathrm{ESS}_3}} + \underset{(=0)}{\overset{4}{\mathrm{ESS}_4}} = 0$$

Step2: Start with 1,2,3,4

	$\overline{X}_1$	$\overline{X}_2$
(12)	(5-1)/2=2	(3+1)/2=2
3	1	-2
4	-3	-2

$$\Rightarrow \begin{array}{ccccccc} (12) & & 3 & & 4 & \\ \mathrm{ESS}_{(12)} & + & \mathrm{ESS}_3 & + & \mathrm{ESS}_4 & = 20 \\ \begin{array}{c}(5-2)^2+(3-2)^2+ \\ (-1-2)^2+(1-2)^2=20\end{array} & & (=0) & & (=0) & \end{array}$$

	$\overline{X}_1$	$\overline{X}_2$
(13)	3	0.5
2	-1	1
4	-3	-2

$$\Rightarrow \begin{array}{ccccccc} (13) & & 2 & & 4 & \\ \mathrm{ESS}_{(13)} & + & \mathrm{ESS}_2 & + & \mathrm{ESS}_4 & = 20.5 \\ 20.5 & & (=0) & & (=0) & \end{array}$$

$$\overset{(14)}{\mathrm{ESS}_{(14)}} + \overset{2}{\mathrm{ESS}_2} + \overset{3}{\mathrm{ESS}_3} = 44.5$$

$$\overset{1}{\mathrm{ESS}_1} + \overset{(23)}{\mathrm{ESS}_{(23)}} + \overset{4}{\mathrm{ESS}_4} = 6.5 \text{ (smallest, randomly selected)}$$

$$\overset{1}{\mathrm{ESS}_1} + \overset{(24)}{\mathrm{ESS}_{(24)}} + \overset{3}{\mathrm{ESS}_3} = 6.5 \text{ (smallest)}$$

$$\overset{1}{\mathrm{ESS}_1} + \overset{2}{\mathrm{ESS}_2} + \overset{(34)}{\mathrm{ESS}_{(34)}} = 8$$

Step 3: start with 1, (23), 4

	$\bar{X}_1$	$\bar{X}_2$
(123)	1.67	0.67
4	-3	-2

$$\Rightarrow \overset{(123)}{ESS_{(123)}} + \overset{4}{ESS_4} = 31.82$$

	$\bar{X}_1$	$\bar{X}_2$
(14)	1	0.5
(23)	0	-0.5

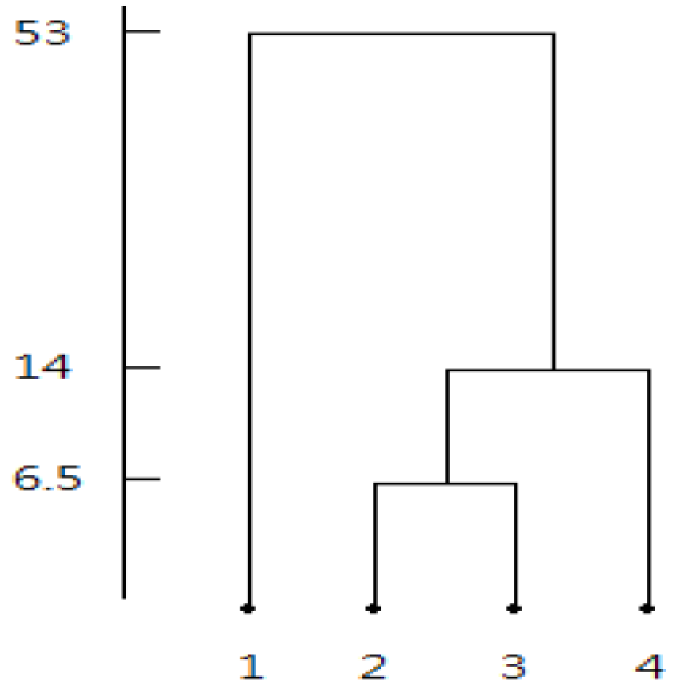
$$\Rightarrow \overset{(14)}{ESS_{(14)}} + \overset{(23)}{ESS_{(23)}} = 51$$

$$\Rightarrow \overset{1}{ESS_{(1)}} + \overset{(234)}{ESS_{(234)}} = 14 \text{ (smallest)}$$

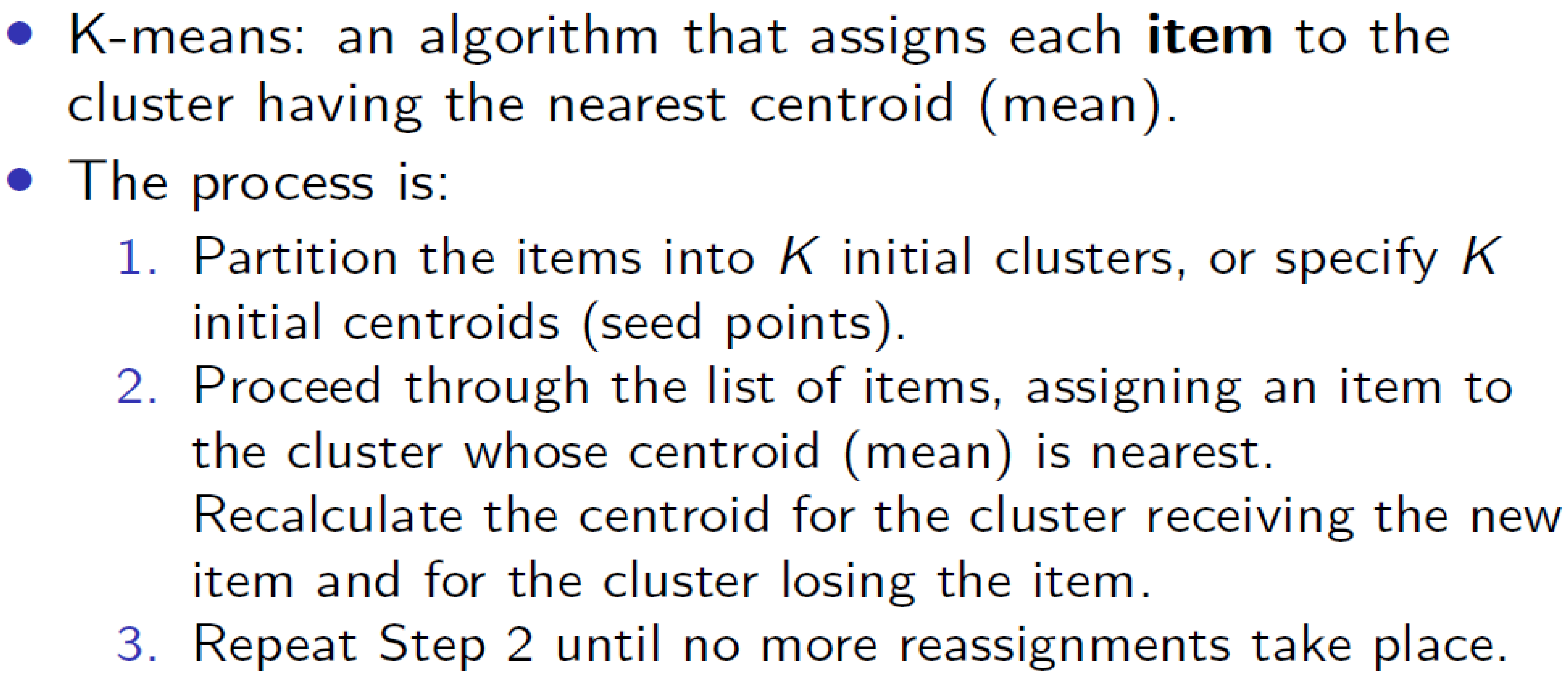
Step 4: start with 1, (234)

$$\overset{(1234)}{ESS_{(1234)}} = 53$$

Dendrogram

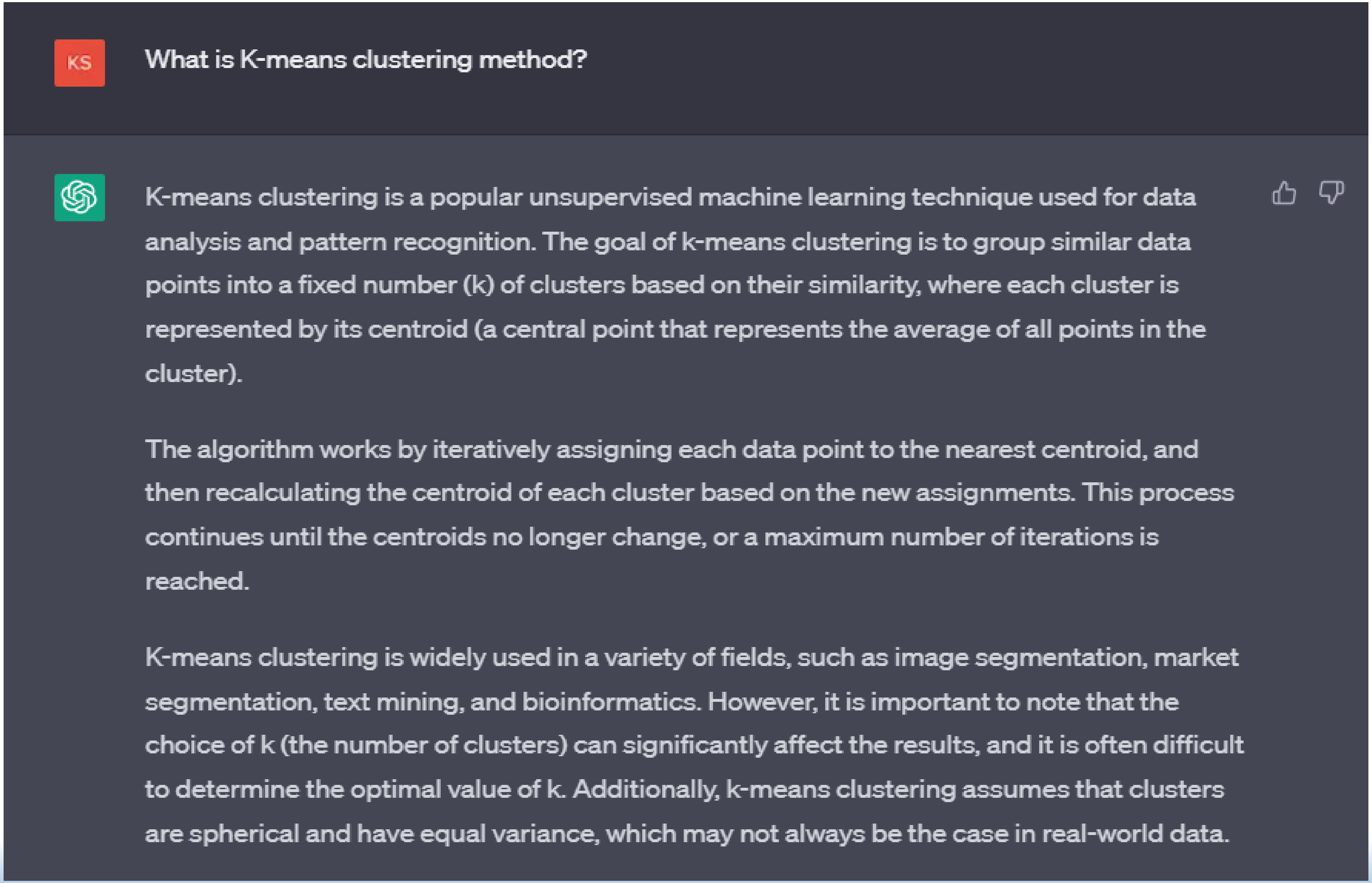


K-means clustering method



- K-means: an algorithm that assigns each **item** to the cluster having the nearest centroid (mean).
- The process is:
  1. Partition the items into K initial clusters, or specify K initial centroids (seed points).
  2. Proceed through the list of items, assigning an item to the cluster whose centroid (mean) is nearest. Recalculate the centroid for the cluster receiving the new item and for the cluster losing the item.
  3. Repeat Step 2 until no more reassignments take place.

https://en.wikipedia.org/wiki/Cluster_analysis



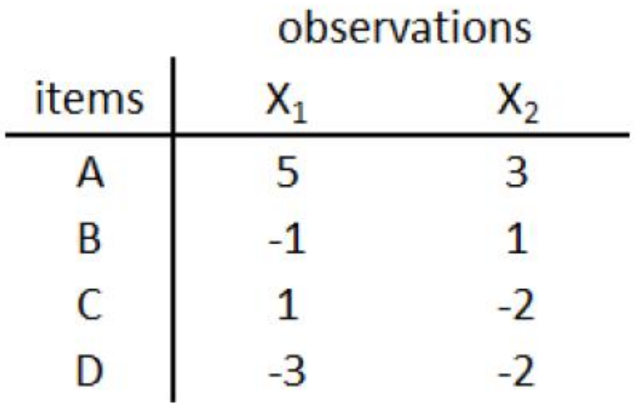
What is K-means clustering method?

K-means clustering is a popular unsupervised machine learning technique used for data analysis and pattern recognition. The goal of k-means clustering is to group similar data points into a fixed number (k) of clusters based on their similarity, where each cluster is represented by its centroid (a central point that represents the average of all points in the cluster).

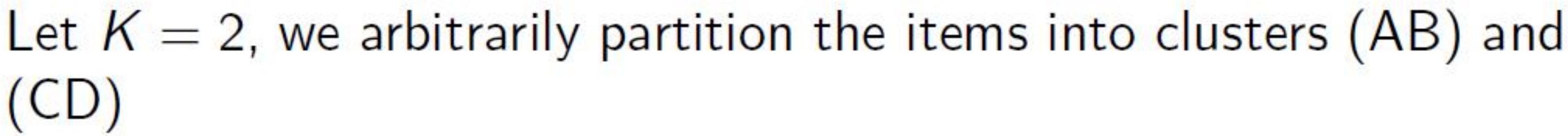
The algorithm works by iteratively assigning each data point to the nearest centroid, and then recalculating the centroid of each cluster based on the new assignments. This process continues until the centroids no longer change, or a maximum number of iterations is reached.

K-means clustering is widely used in a variety of fields, such as image segmentation, market segmentation, text mining, and bioinformatics. However, it is important to note that the choice of k (the number of clusters) can significantly affect the results, and it is often difficult to determine the optimal value of k. Additionally, k-means clustering assumes that clusters are spherical and have equal variance, which may not always be the case in real-world data.

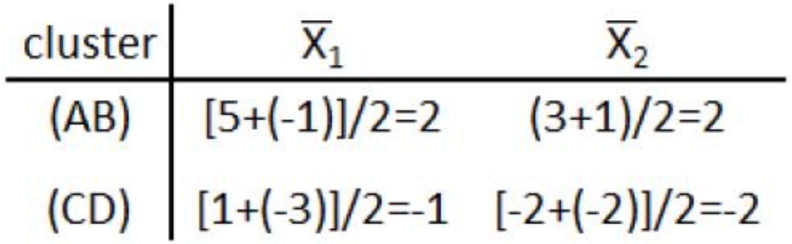
Example



	observations	
items	X_1	X_2
A	5	3
B	-1	1
C	1	-2
D	-3	-2



Let $K = 2$, we arbitrarily partition the items into clusters (AB) and (CD)



cluster	$\overline{X}_1$	$\overline{X}_2$
(AB)	[5+(-1)]/2=2	(3+1)/2=2
(CD)	[1+(-3)]/2=-1	[-2+(-2)]/2=-2

Using the Euclidean distance

$$d^2(A,(AB)) = (5-2)^2 + (3-2)^2 = 10$$
$$d^2(A,(CD)) = (5+1)^2 + (3+2)^2 = 61$$

(A is not reassigned)

$$d^2(B,(AB)) = 10$$
$$d^2(B,(CD)) = 9$$

(B is reassigned)

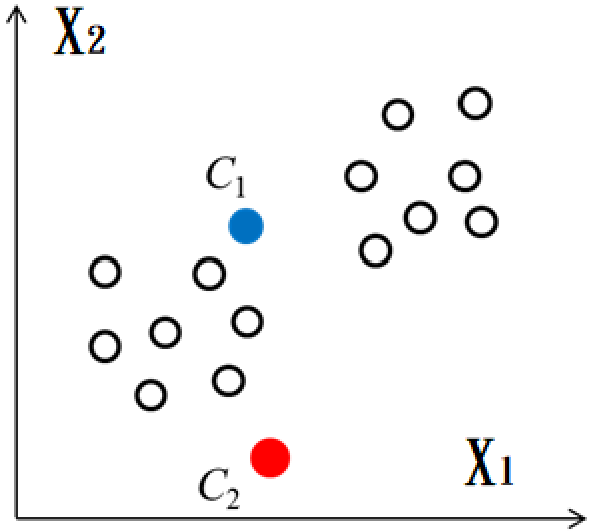
cluster	$\bar{x}_1$	$\bar{x}_2$
A	5	3
(BCD)	-1	-1

squared distances to cluster centroids

	item			
cluster	A	B	C	D
A	0	40	41	89
(BCD)	52	4	5	5

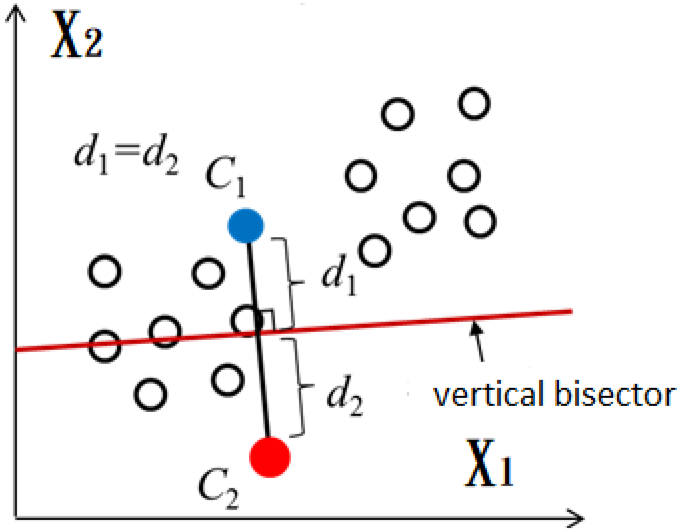
$\Rightarrow$ no reassignment is needed

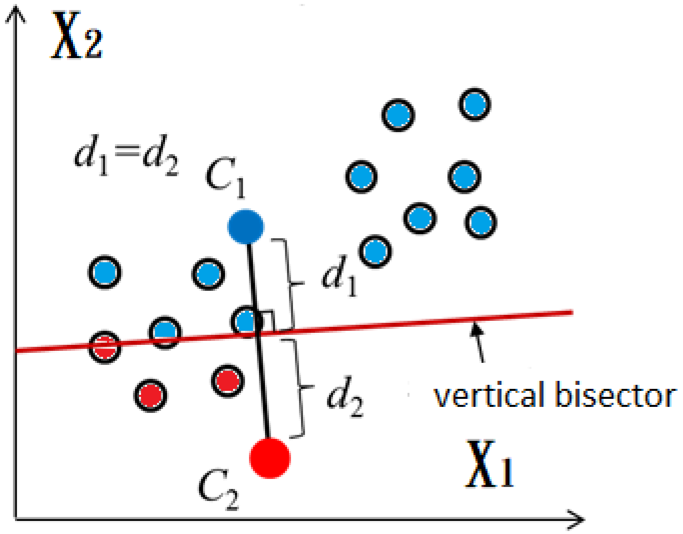
Step 1: Randomly select c1 and c2



$$\mu_c^{(0)} \in R^d, c = 1,2,\ldots,K$$

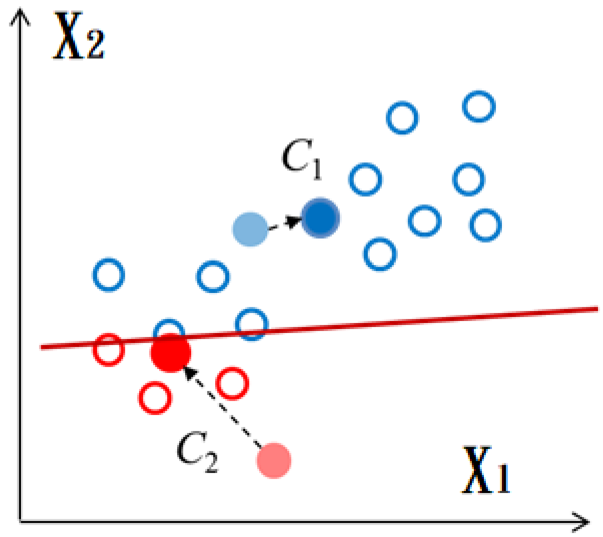
Step 2: Draw the vertical bisector of the center of the group, classify each sample





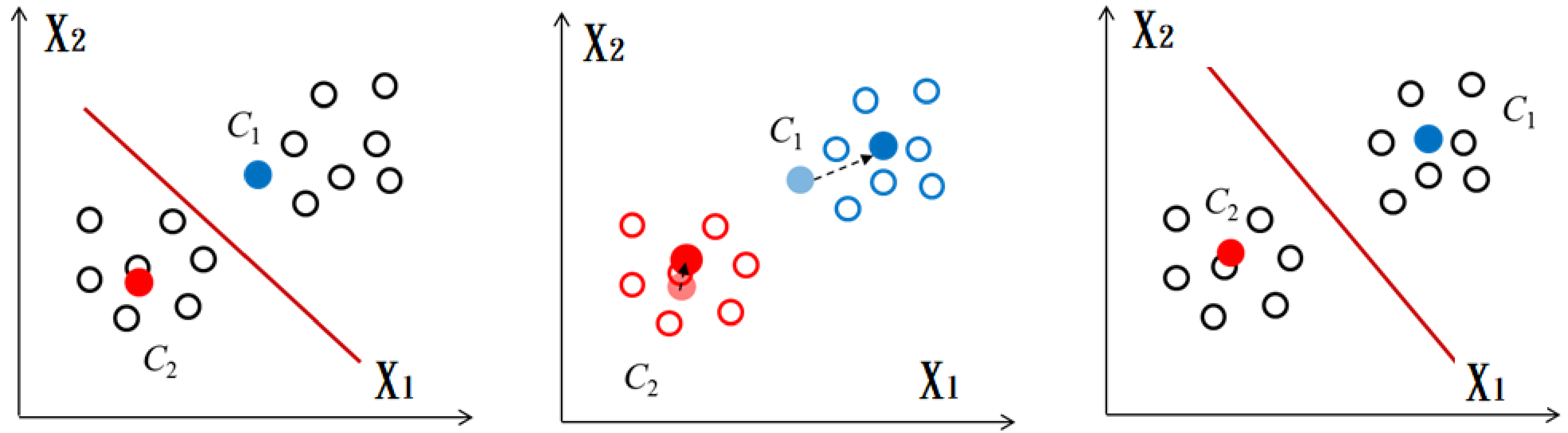
$$S_c^{(t)} = \left\{ x_i : \left\| x_i - \mu_c^{(t)} \right\| \le \left\| x_i - \mu_{c^*}^{(t)} \right\|, \forall i = 1, \dots, n \right\}$$

Step 3: Update cluster center



$$\mu_c^{(t+1)} = \frac{sum(S_c^{(t)})}{n_c} = \sum_{i=1}^{n_c} x_i \Big|_{x_i \in S_c^{(t)}}$$

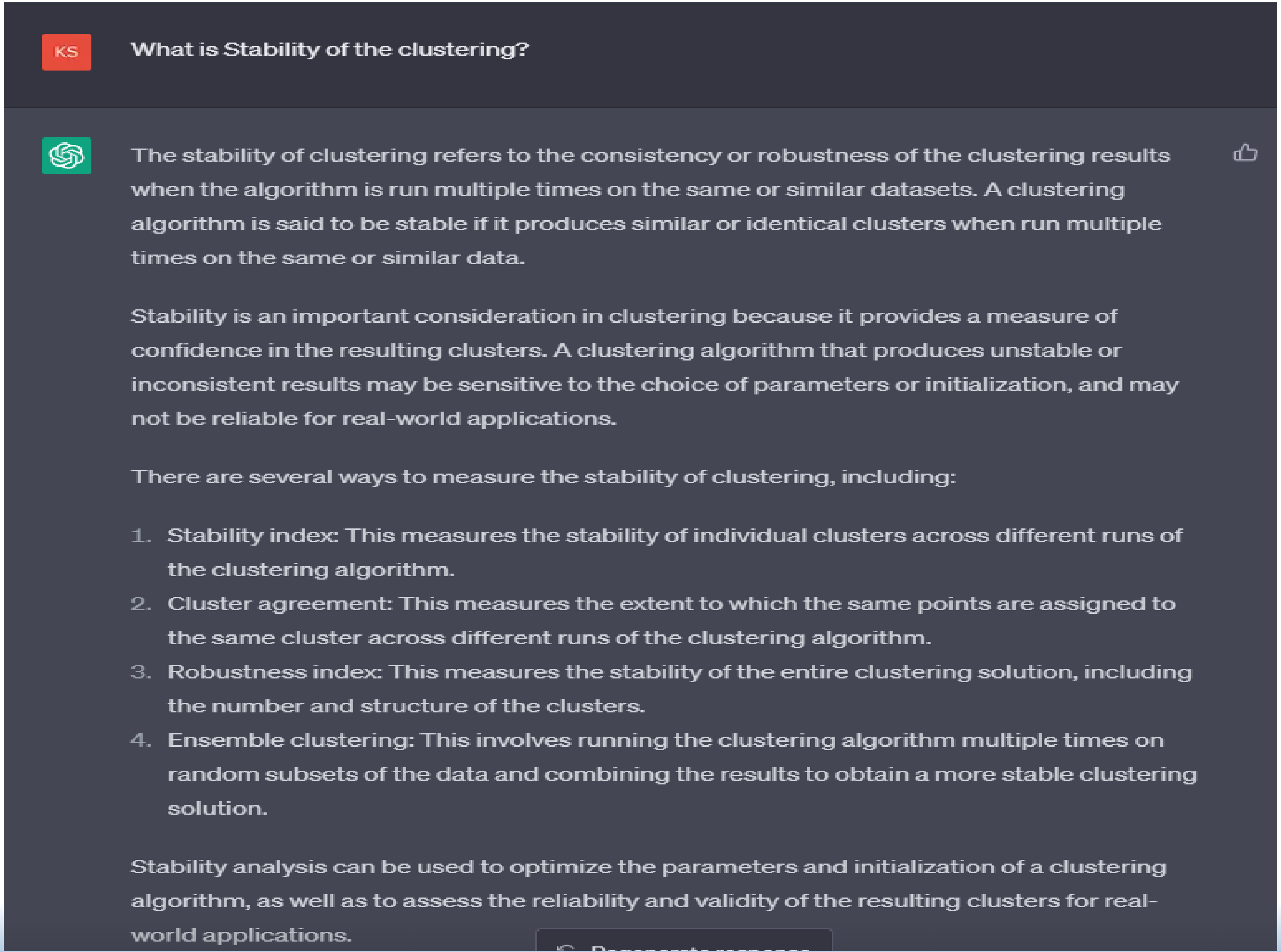
Repeat steps 2–3 until the center position does not change



$$S_c^{(t+1)} = S_c^{(t)}, \forall c = 1, \ldots, K$$

Stability of the clustering

- Good choice of the initial partition or the initial selection of seed points should be free of biases. One way to start is to randomly select seed points among the items or to randomly partition the items into initial groups.
- To check the stability of the clustering, it is desirable to rerun the algorithm with a new initial partition or new initial seed points.



What is Stability of the clustering?

The stability of clustering refers to the consistency or robustness of the clustering results when the algorithm is run multiple times on the same or similar datasets. A clustering algorithm is said to be stable if it produces similar or identical clusters when run multiple times on the same or similar data.

Stability is an important consideration in clustering because it provides a measure of confidence in the resulting clusters. A clustering algorithm that produces unstable or inconsistent results may be sensitive to the choice of parameters or initialization, and may not be reliable for real-world applications.

There are several ways to measure the stability of clustering, including:

1. Stability index: This measures the stability of individual clusters across different runs of the clustering algorithm.
2. Cluster agreement: This measures the extent to which the same points are assigned to the same cluster across different runs of the clustering algorithm.
3. Robustness index: This measures the stability of the entire clustering solution, including the number and structure of the clusters.
4. Ensemble clustering: This involves running the clustering algorithm multiple times on random subsets of the data and combining the results to obtain a more stable clustering solution.

Stability analysis can be used to optimize the parameters and initialization of a clustering algorithm, as well as to assess the reliability and validity of the resulting clusters for real-world applications.

Determine K for K-Means Clustering

Elbow method:

Silhouette method:

Cross-validation:

The gap statistics:

Others:

https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set
https://www.analyticsvidhya.com/blog/2021/05/k mean getting the optimal number of clusters
https://www.datanovia.com/en/lessons/determining the optimal number of clusters 3 must know methods

Choosing K for K-means by Elbow Method

In contrast to the supervised learning tasks such as classification and regression, clustering requires more effort to choose the optimization criterion. Usually, when working with k-means, we optimize the sum of squared distances between the observations and their centroids.

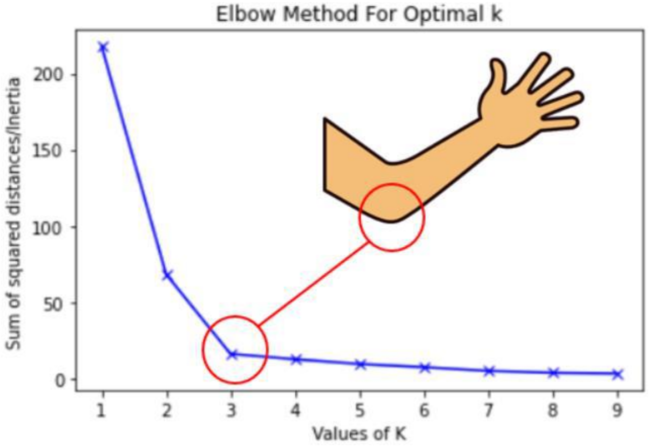
$$J(C) = \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \mu_k|| \to \min_{C},$$

where C – is a set of clusters with power K, μ_k is a centroid of a cluster C_k.

This definition seems reasonable -- we want our observations to be as close to their centroids as possible. But, there is a problem -- the optimum is reached when the number of centroids is equal to the number of observations, so you would end with every single observation as its own separate cluster.

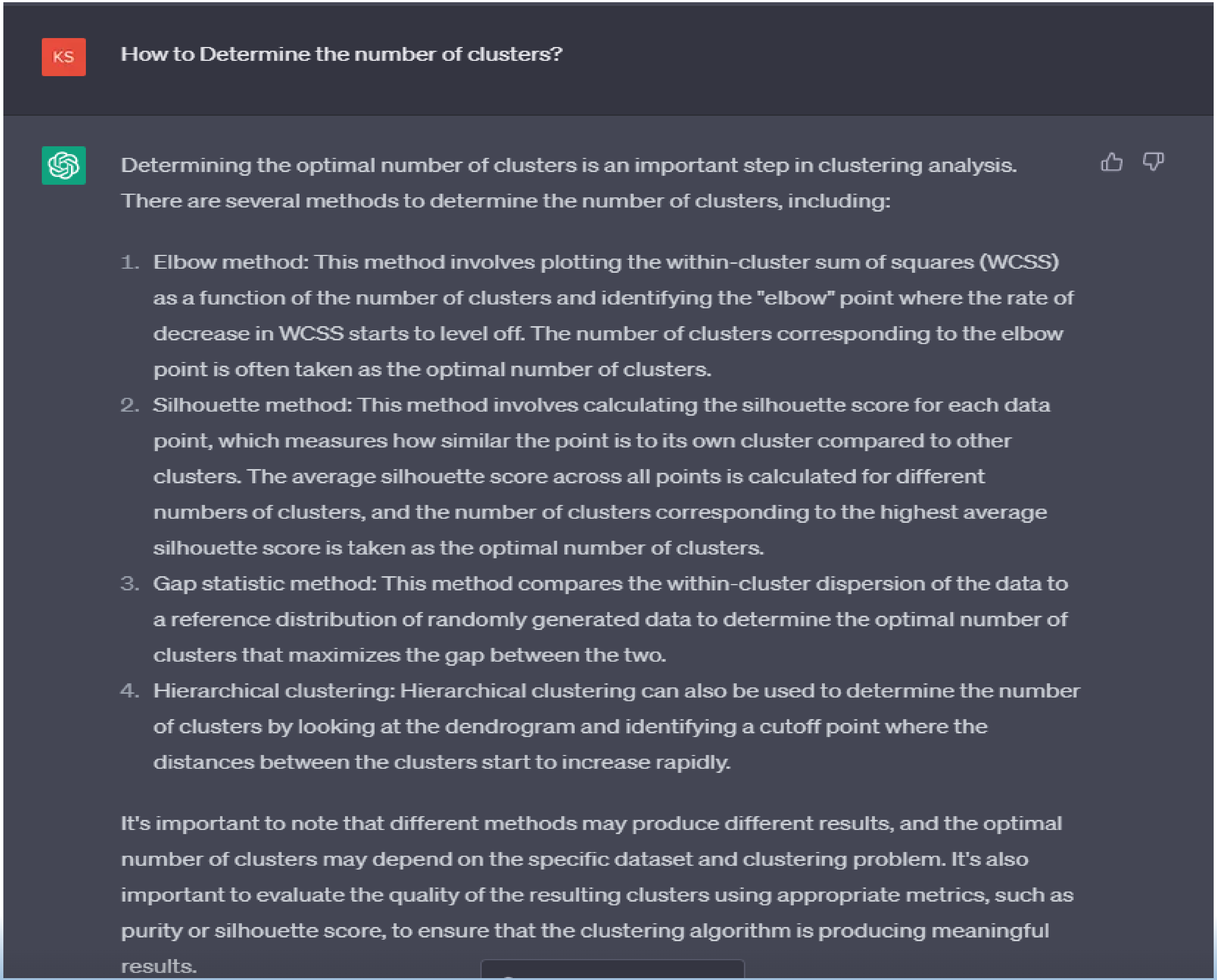
In order to avoid that case, we should choose a number of clusters after which a function $J(C_k)$ is decreasing less rapidly. More formally,

$$D(k) = \frac{|J(C_k) - J(C_{k+1})|}{|J(C_{k-1}) - J(C_k)|} \to \min_{k}$$



Line plot between K and inertia

https://learn.saylor.org/mod/book/tool/print/index.php?id=55625&chapterid=41475
https://www.analyticsvidhya.com/blog/2021/05/k mean getting the optimal number of clusters



How to Determine the number of clusters?

Determining the optimal number of clusters is an important step in clustering analysis. There are several methods to determine the number of clusters, including:

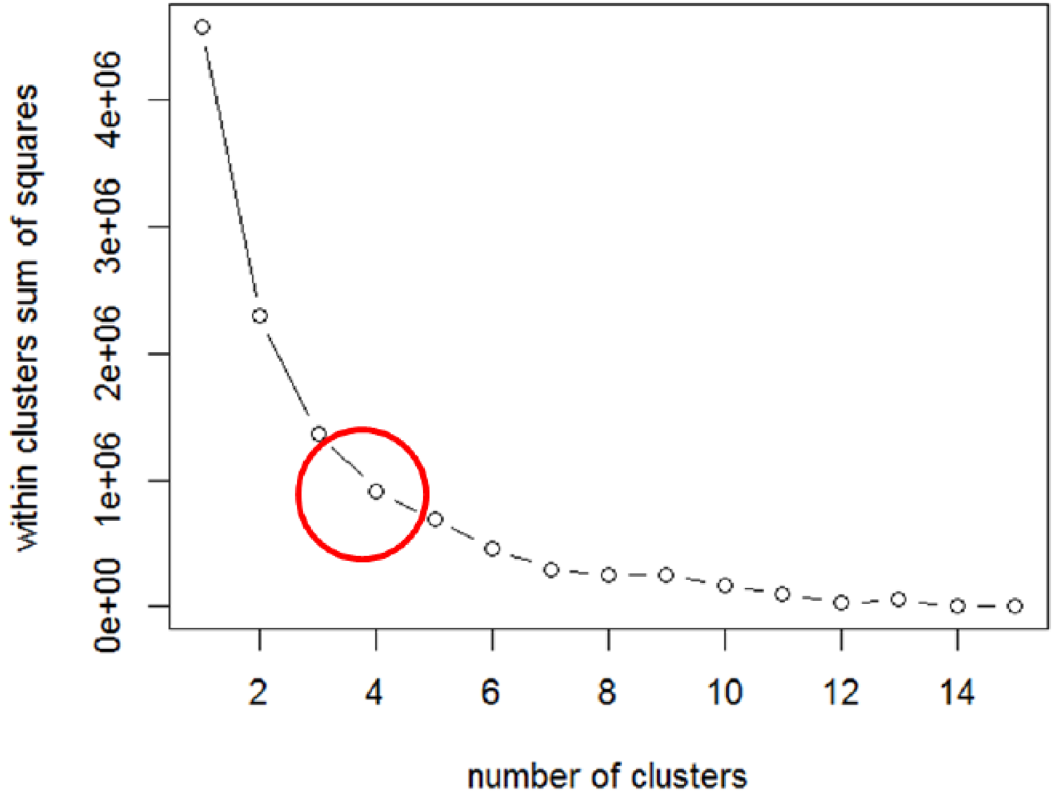
1. Elbow method: This method involves plotting the within-cluster sum of squares (WCSS) as a function of the number of clusters and identifying the "elbow" point where the rate of decrease in WCSS starts to level off. The number of clusters corresponding to the elbow point is often taken as the optimal number of clusters.
2. Silhouette method: This method involves calculating the silhouette score for each data point, which measures how similar the point is to its own cluster compared to other clusters. The average silhouette score across all points is calculated for different numbers of clusters, and the number of clusters corresponding to the highest average silhouette score is taken as the optimal number of clusters.
3. Gap statistic method: This method compares the within-cluster dispersion of the data to a reference distribution of randomly generated data to determine the optimal number of clusters that maximizes the gap between the two.
4. Hierarchical clustering: Hierarchical clustering can also be used to determine the number of clusters by looking at the dendrogram and identifying a cutoff point where the distances between the clusters start to increase rapidly.

It's important to note that different methods may produce different results, and the optimal number of clusters may depend on the specific dataset and clustering problem. It's also important to evaluate the quality of the resulting clusters using appropriate metrics, such as purity or silhouette score, to ensure that the clustering algorithm is producing meaningful results.

Plot the value of the clustering measure against the number of clusters

- —Large changes of levels in the plot are taken as suggestive of a particular number of clusters

Clustering criterion: within-cluster sum of squares



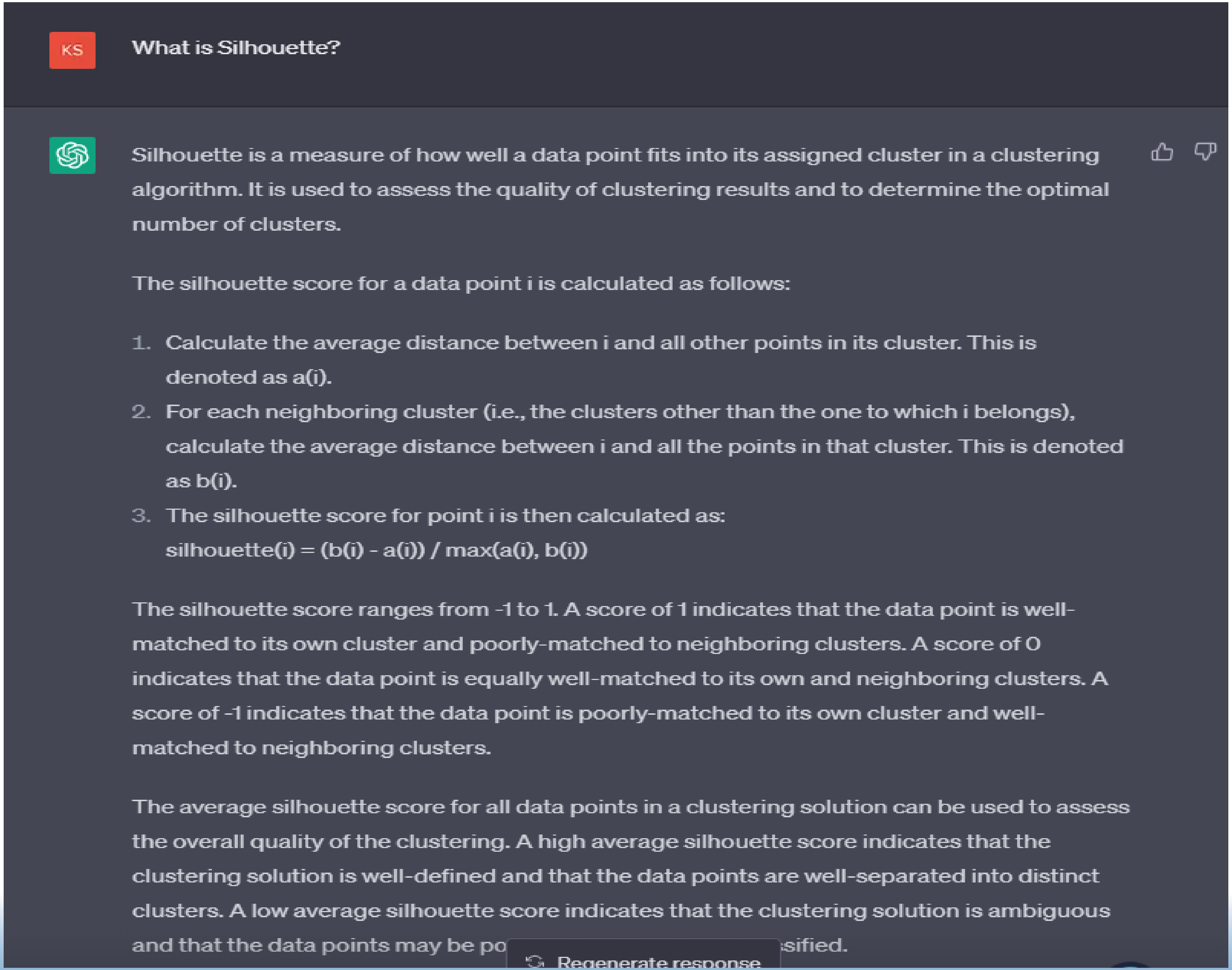
Silhouette

Silhouette refers to a method of interpretation and validation of consistency within clusters of data. The technique provides a succinct graphical representation of how well each object has been classified.[1] It was proposed by Belgian statistician Peter Rousseeuw in 1987.

The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well-matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

The silhouette can be calculated with any distance metric, such as the Euclidean distance or the Manhattan distance.

https://en.wikipedia.org/wiki/Silhouette_(clustering)#:~:text=The%20silhouette%20value%20is%20a,poorly%20matched%20to%20neighboring%20clusters.



What is Silhouette?

Silhouette is a measure of how well a data point fits into its assigned cluster in a clustering algorithm. It is used to assess the quality of clustering results and to determine the optimal number of clusters.

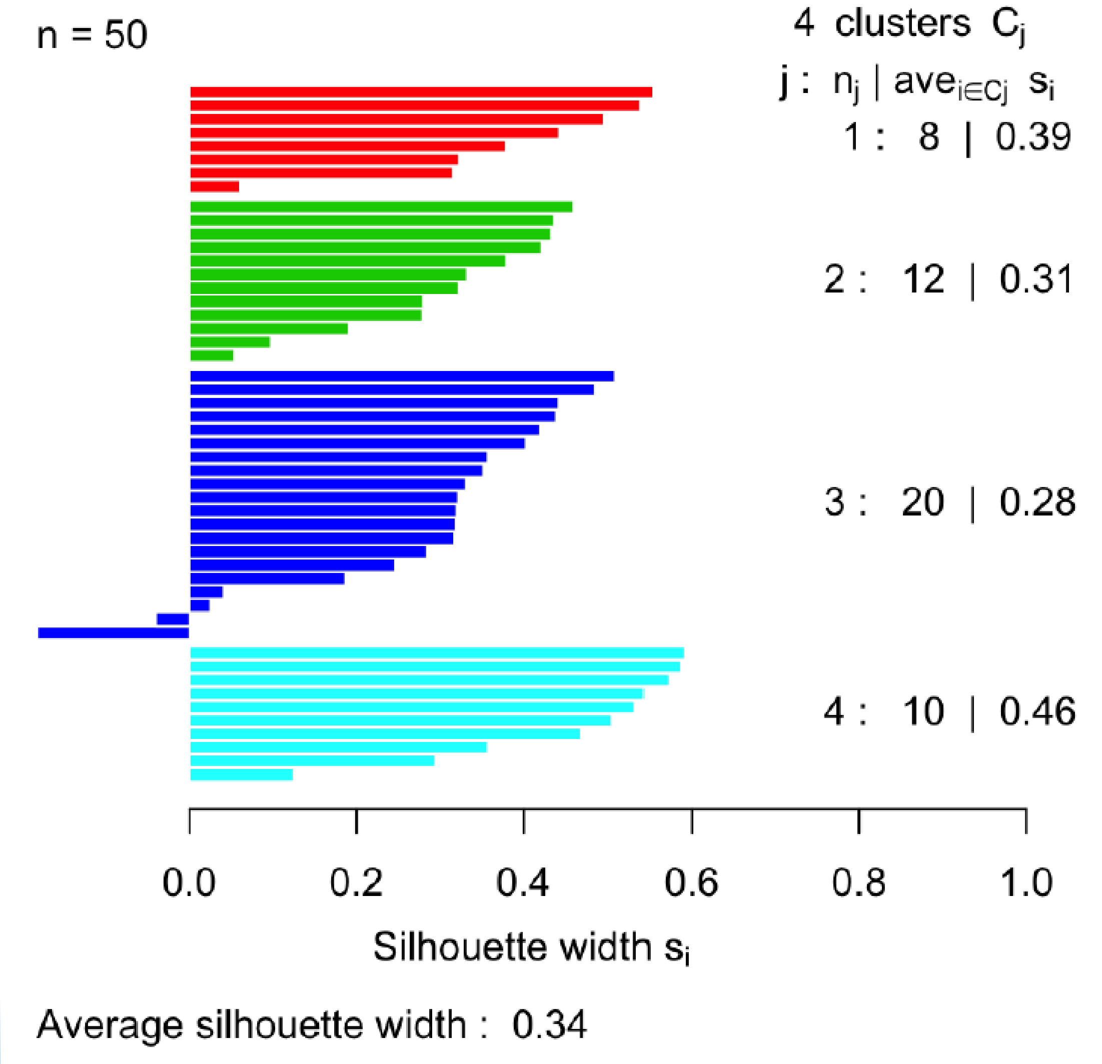
The silhouette score for a data point i is calculated as follows:

1. Calculate the average distance between i and all other points in its cluster. This is denoted as a(i).
2. For each neighboring cluster (i.e., the clusters other than the one to which i belongs), calculate the average distance between i and all the points in that cluster. This is denoted as b(i).
3. The silhouette score for point i is then calculated as:
   silhouette(i) = (b(i) - a(i)) / max(a(i), b(i))

The silhouette score ranges from -1 to 1. A score of 1 indicates that the data point is well-matched to its own cluster and poorly-matched to neighboring clusters. A score of 0 indicates that the data point is equally well-matched to its own and neighboring clusters. A score of -1 indicates that the data point is poorly-matched to its own cluster and well-matched to neighboring clusters.

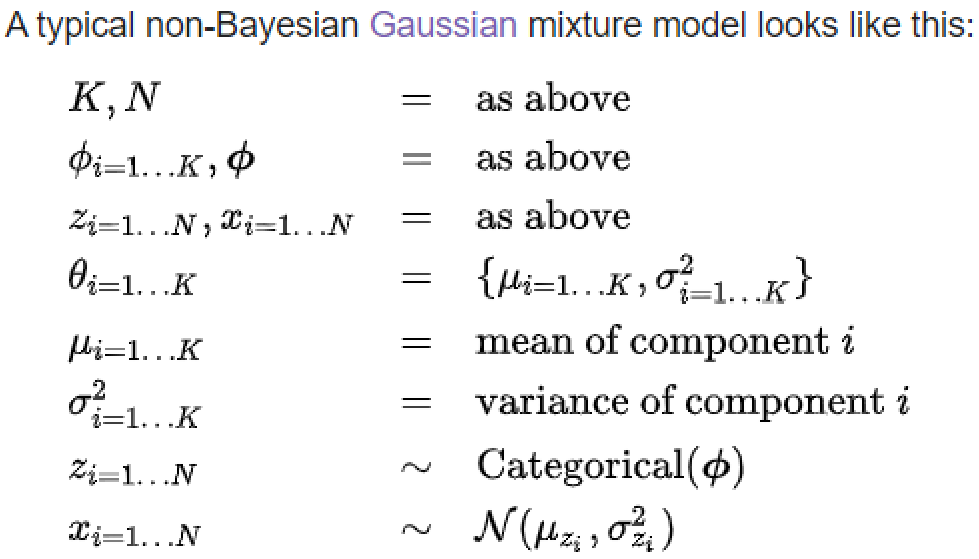
The average silhouette score for all data points in a clustering solution can be used to assess the overall quality of the clustering. A high average silhouette score indicates that the clustering solution is well-defined and that the data points are well-separated into distinct clusters. A low average silhouette score indicates that the clustering solution is ambiguous and that the data points may be po sified.

Silhouette plot



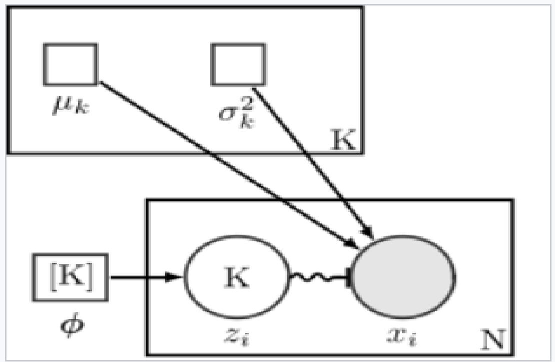
Gaussian Mixture Models (GMMs)

- Gaussian Mixture Models (GMMs) are a probabilistic model used for clustering and density estimation. In a GMM, data is assumed to come from a mixture of Gaussian distributions, each with its own mean and covariance matrix. The goal is to estimate the parameters of these Gaussians, as well as the mixing coefficients that describe the relative contributions of each Gaussian to the overall mixture.



A typical non-Bayesian Gaussian mixture model looks like this:

$$\begin{array}{lcl}
K, N & = & \text{as above} \\
\phi_{i=1\dots K}, \boldsymbol{\phi} & = & \text{as above} \\
z_{i=1\dots N}, x_{i=1\dots N} & = & \text{as above} \\
\theta_{i=1\dots K} & = & \{\mu_{i=1\dots K}, \sigma^2_{i=1\dots K}\} \\
\mu_{i=1\dots K} & = & \text{mean of component } i \\
\sigma^2_{i=1\dots K} & = & \text{variance of component } i \\
z_{i=1\dots N} & \sim & \operatorname{Categorical}(\boldsymbol{\phi}) \\
x_{i=1\dots N} & \sim & \mathcal{N}(\mu_{z_i}, \sigma^2_{z_i})
\end{array}$$

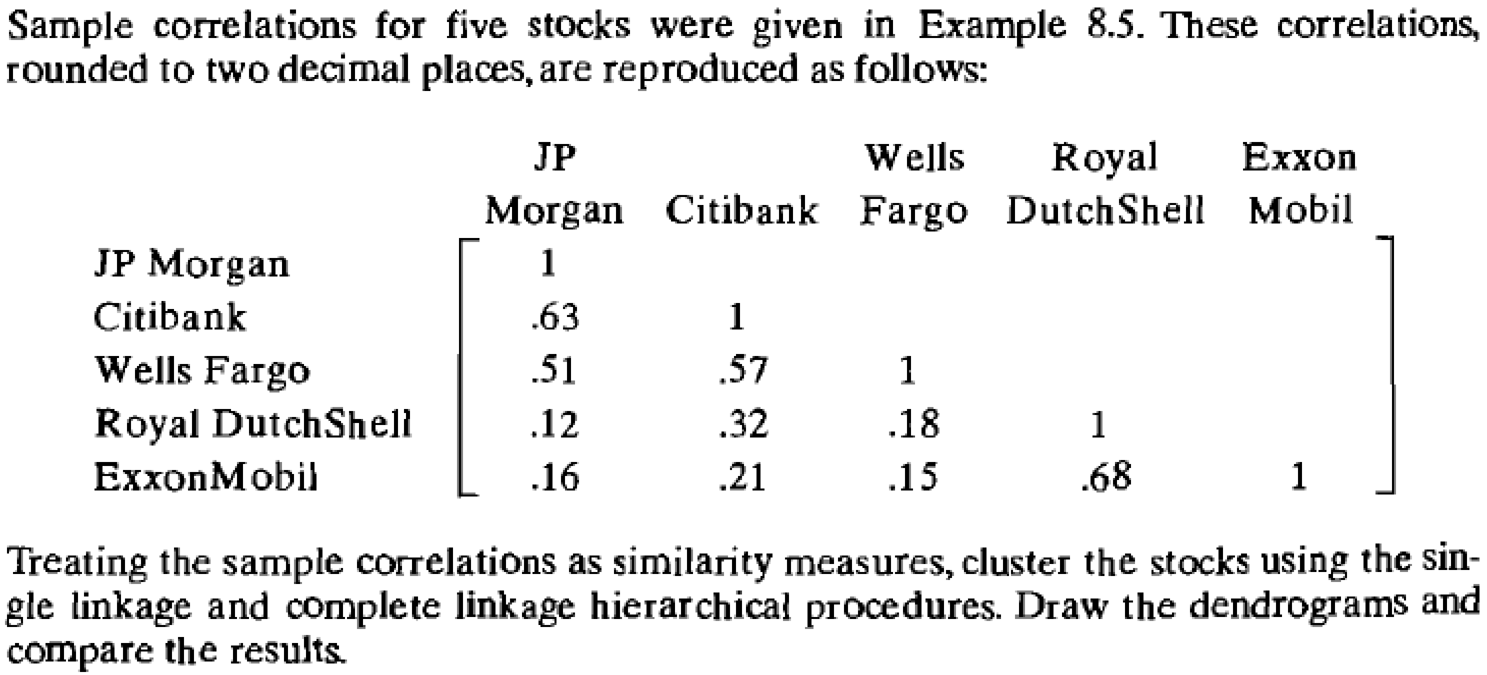


Non-Bayesian Gaussian mixture model using plate notation. Smaller squares indicate fixed parameters; larger circles indicate random variables. Filled-in shapes indicate known values. The indication [K] means a vector of size *K*.

https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

https://en.wikipedia.org/wiki/Mixture_model#Further_reading

Clustering demo:

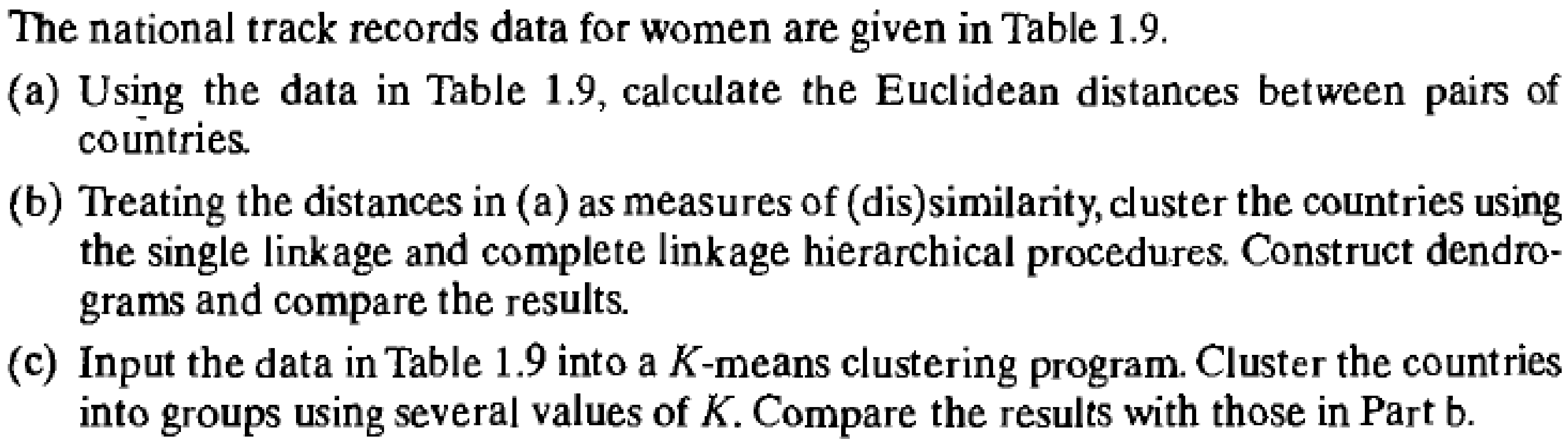


Sample correlations for five stocks were given in Example 8.5. These correlations, rounded to two decimal places, are reproduced as follows:

$$\begin{array}{l} \\ \\ \text{JP Morgan} \\ \text{Citibank} \\ \text{Wells Fargo} \\ \text{Royal DutchShell} \\ \text{ExxonMobil} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{JP} & & \text{Wells} & \text{Royal} & \text{Exxon} \\ \text{Morgan} & \text{Citibank} & \text{Fargo} & \text{DutchShell} & \text{Mobil} \end{array} \\ \left[\begin{array}{ccccc} 1 & & & & \\ .63 & 1 & & & \\ .51 & .57 & 1 & & \\ .12 & .32 & .18 & 1 & \\ .16 & .21 & .15 & .68 & 1 \end{array}\right] \end{array}$$

Treating the sample correlations as similarity measures, cluster the stocks using the single linkage and complete linkage hierarchical procedures. Draw the dendrograms and compare the results.

Applied Multivariate Statistical Analysis (6th Edition) by Richard A. Johnson , Dean W.Wichern



The national track records data for women are given in Table 1.9.

(a) Using the data in Table 1.9, calculate the Euclidean distances between pairs of countries.

(b) Treating the distances in (a) as measures of (dis)similarity, cluster the countries using the single linkage and complete linkage hierarchical procedures. Construct dendrograms and compare the results.

(c) Input the data in Table 1.9 into a *K*-means clustering program. Cluster the countries into groups using several values of *K*. Compare the results with those in Part b.

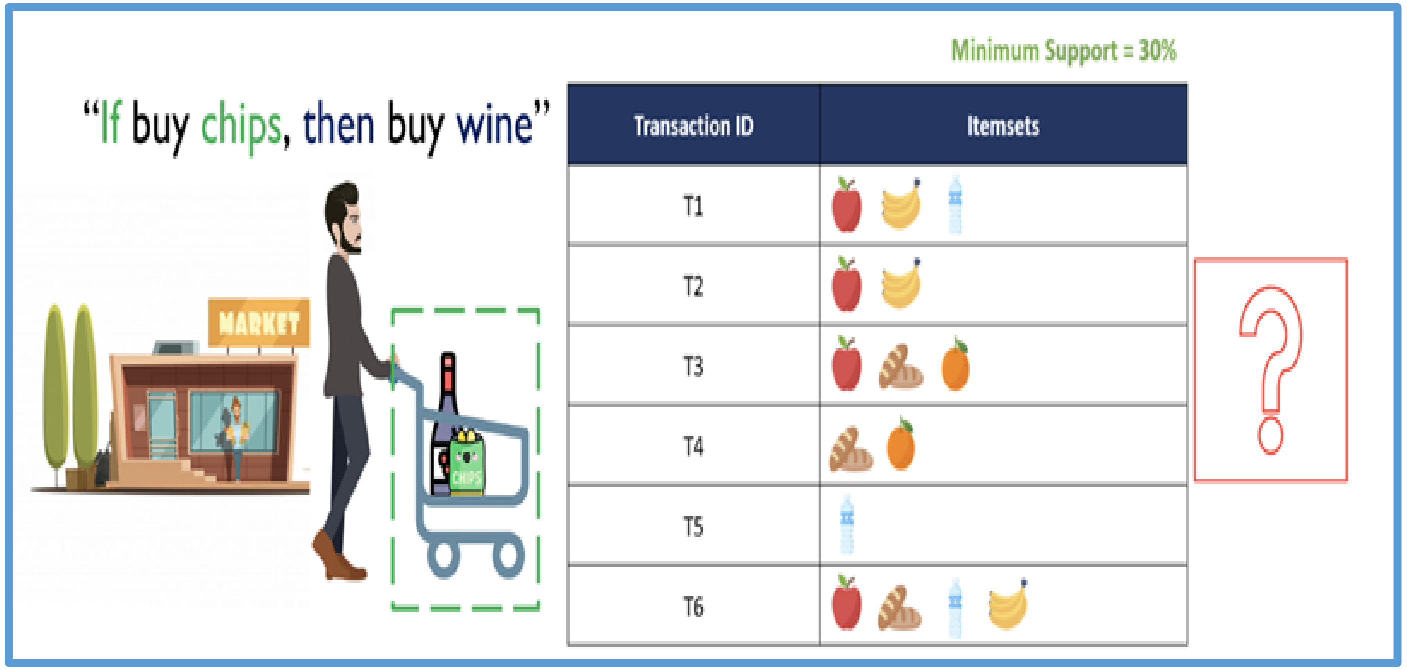
Association Rules

Data Mining: Concepts and Techniques, 3rd Edition
by Jiawei Han, Micheline Kamber, Jian Pei, 2011

https://https://en.wikipedia.org/wiki/Association_rule_learning
https://hanj.cs.illinois.edu/bk3/

Introduction to Association

Association analysis is the task of finding interesting relationships in large datasets. These interesting relationships can take two forms: frequent item sets or association rules. Frequent item sets are a collection of items that frequently occur together. Association rules suggest that a strong relationship exists between the two items.

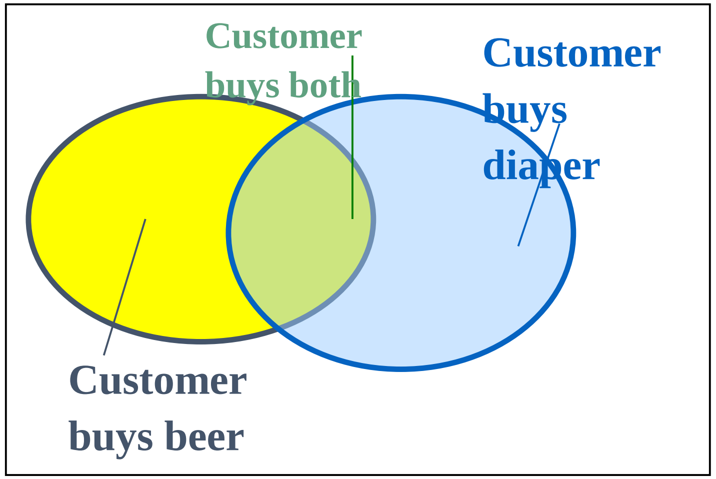


Transaction ID	Itemsets
T1	🍎 🍌 🧴
T2	🍎 🍌
T3	🍎 🥖 🍊
T4	🥖 🍊
T5	🧴
T6	🍎 🥖 🧴 🍌

https://rpubs.com/anikbibib/Arules
https://livebook.manning.com/book/machine-learning-in-action/chapter-11/17

Basic Concepts: Association Rules

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, *s*, probability that a transaction contains $X \cup Y$
  - confidence, *c*, conditional probability that a transaction having X also contains *Y*

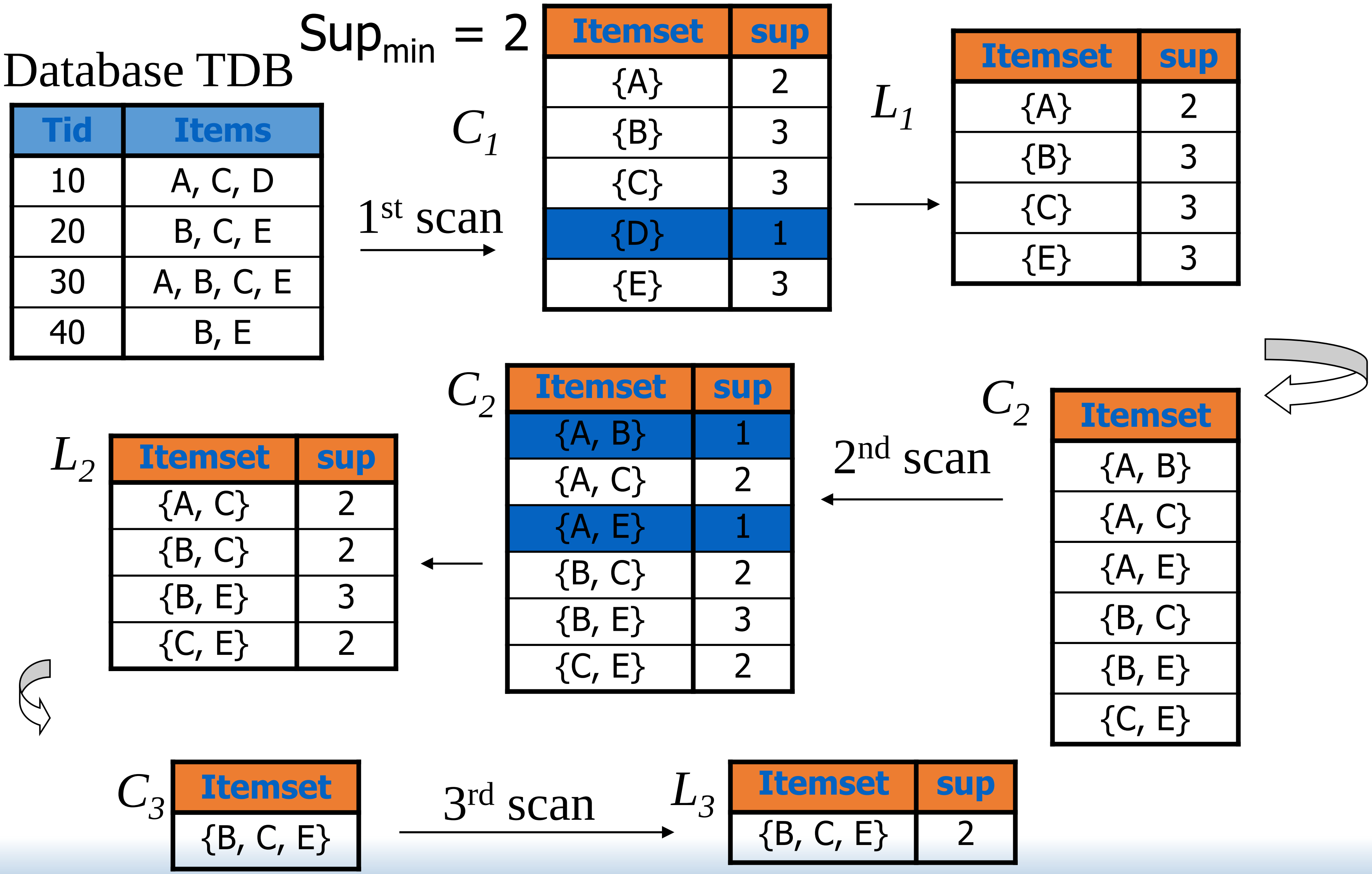
Let minsup = 50%, minconf = 50%

Freq. Pat.: Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

- **Association rules: (many more!)**
  - ***Beer → Diaper*** **(60%, 100%)**
  - ***Diaper → Beer*** **(60%, 75%)**

Source: https://hanj.cs.illinois.edu/bk3/bk3_slides/06FPBasic.ppt

The Apriori Algorithm—An Example



$Sup_{min} = 2$

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

Source: https://hanj.cs.illinois.edu/bk3/bk3_slides/06FPBasic.ppt

The Apriori Algorithm (Pseudo-Code)

C_k: Candidate itemset of size k

L_k : frequent itemset of size k

```
$L_1$ = {frequent items};
for ($k$ = 1; $L_k$ !=$\emptyset$; $k$++) do begin
    $C_{k+1}$ = candidates generated from $L_k$;
    for each transaction $t$ in database do
        increment the count of all candidates in $C_{k+1}$ that are contained in $t$
    $L_{k+1}$ = candidates in $C_{k+1}$ with min_support
    end
return $\cup_k L_k$;
```

Source: https://hanj.cs.illinois.edu/bk3/bk3_slides/06FPBasic.ppt

Other Methods for Graph Models

- Graph Models: https://en.wikipedia.org/wiki/Graphical_model

- Boolean Networks: https://en.wikipedia.org/wiki/Boolean_network

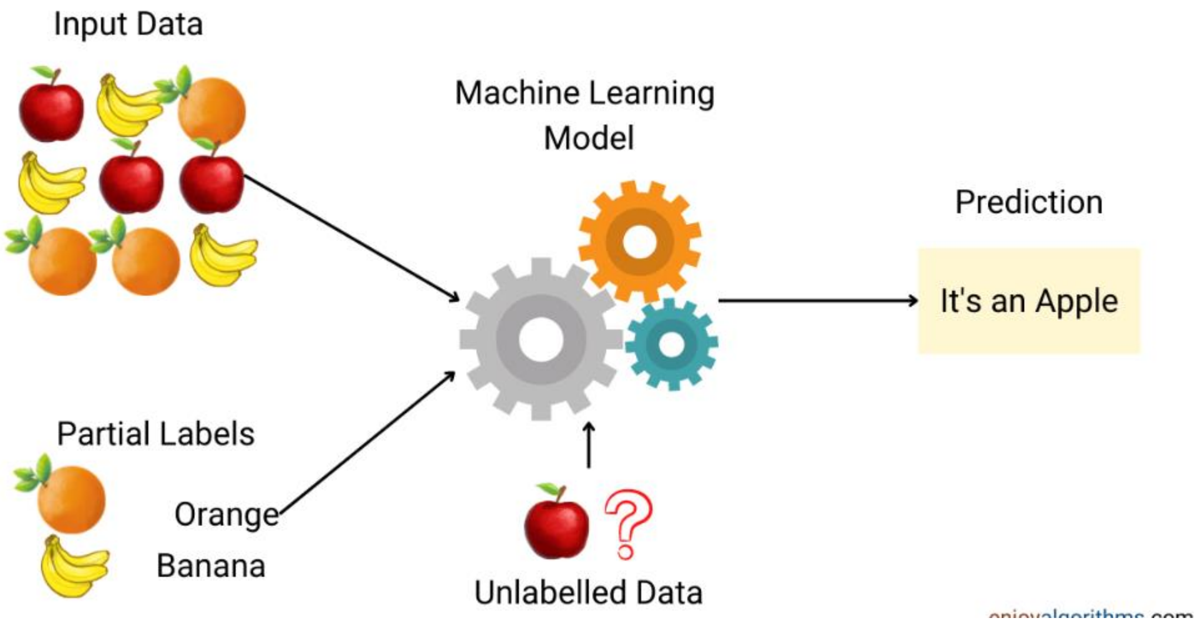
- Time Delay Boolean Networks: Chueh T-H, Lu HH-S (2012) Inference of Biological Pathway from Gene Expression Profiles by Time Delay Boolean Networks. PLoS ONE 7(8): e42095. https://doi.org/10.1371/journal.pone.0042095

- Bayesian Networks: https://en.wikipedia.org/wiki/Bayesian_network

- Boolean-Bayesian Networks: Tsai, MY., Lu, H.HS. (2022). Integration of Boolean and Bayesian Networks. In: Lu, H.HS., Schölkopf, B., Wells, M.T., Zhao, H. (eds) Handbook of Statistical Bioinformatics. Springer Handbooks of Computational Statistics. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-65902-1_9

semi-supervised learning

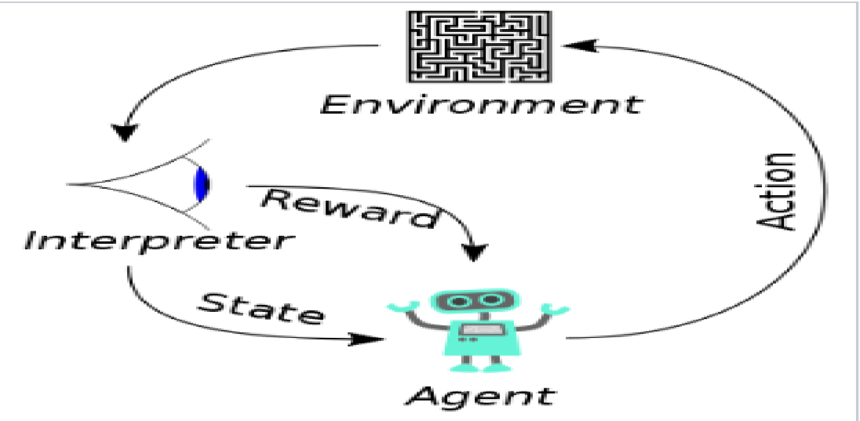
- **semi-supervised learning**:- is a branch of machine learning that combines a small amount of labeled data with a large amount of unlabeled data during training. Semi-supervised learning falls between unsupervised learning(with no labeled training data) and supervised learning (with only labeled training data). Semi-supervised learning aims to alleviate the issue of having limited amounts of labeled data available for training.



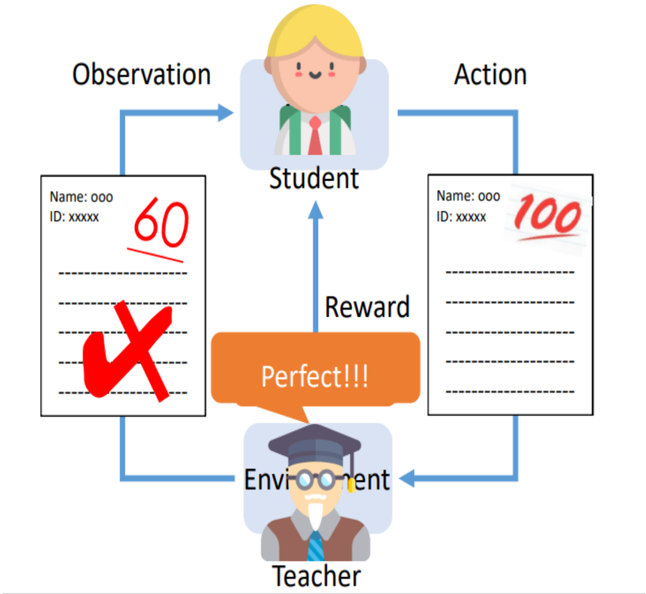
https://www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning

Reinforcement learning

- **Reinforcement learning (RL)** is an area of machine learning concerned with how intelligent agents ought to take action in an environment in order to maximize the notion of cumulative reward. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning.



The typical framing of a Reinforcement Learning (RL) scenario: an agent takes actions in an environment, which is interpreted into a reward and a representation of the state, which are fed back into the agent.



https://en.wikipedia.org/wiki/Reinforcement_learning

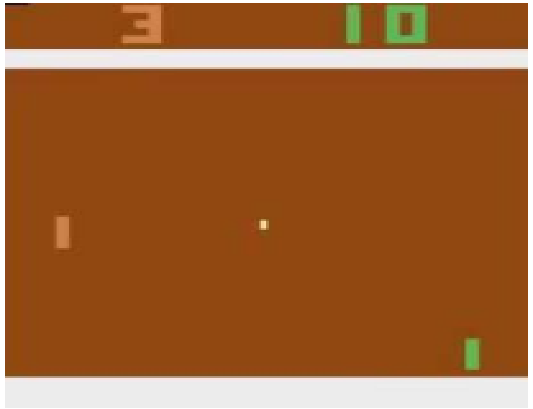
Example 1: Reinforcement Learning

AlphaGo

https://www.youtube.com/watch?v=MgowR4pq3e8

Example 2: Reinforcement Learning

Pong from Pixels



Pong AI with Policy Gradients

[Pong AI with Policy Gradients - YouTube]

[An introduction to Reinforcement Learning - YouTube]

Homework 9 (submitted to e3.nycu.edu.tw before Dec 13, 2023)

Use R, Python, and suitable computer packages to perform different types of Clustering Models.

1. Conduct the Hierarchical clustering and discuss results.
2. Conduct the K-means and discuss results.
3. Conduct the Silhouette plot and choose K in K-means.

Possible sources of open datasets:

- UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets.php)
- Kaggle Datasets (https://www.kaggle.com/datasets)
- World Health Organization Datasets (https://www.who.int/)