Introduction to Unsupervised Learning

Ana Maria Simion & Irina Mocanu

Part of the slides were adapted from content by Alexandru Sorici and lectures at Stanford (CS231n, 2022) and University of Michigan (EECS 598, 2022)

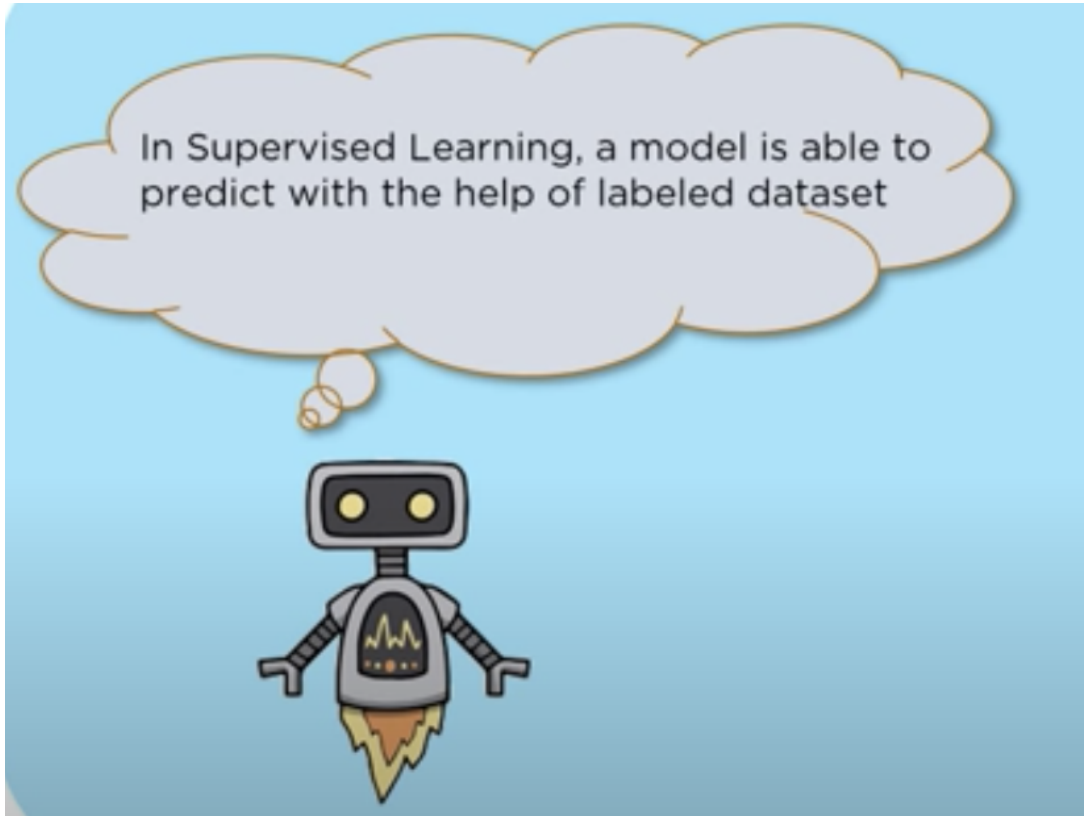
Machine Learning

Supervised Learning

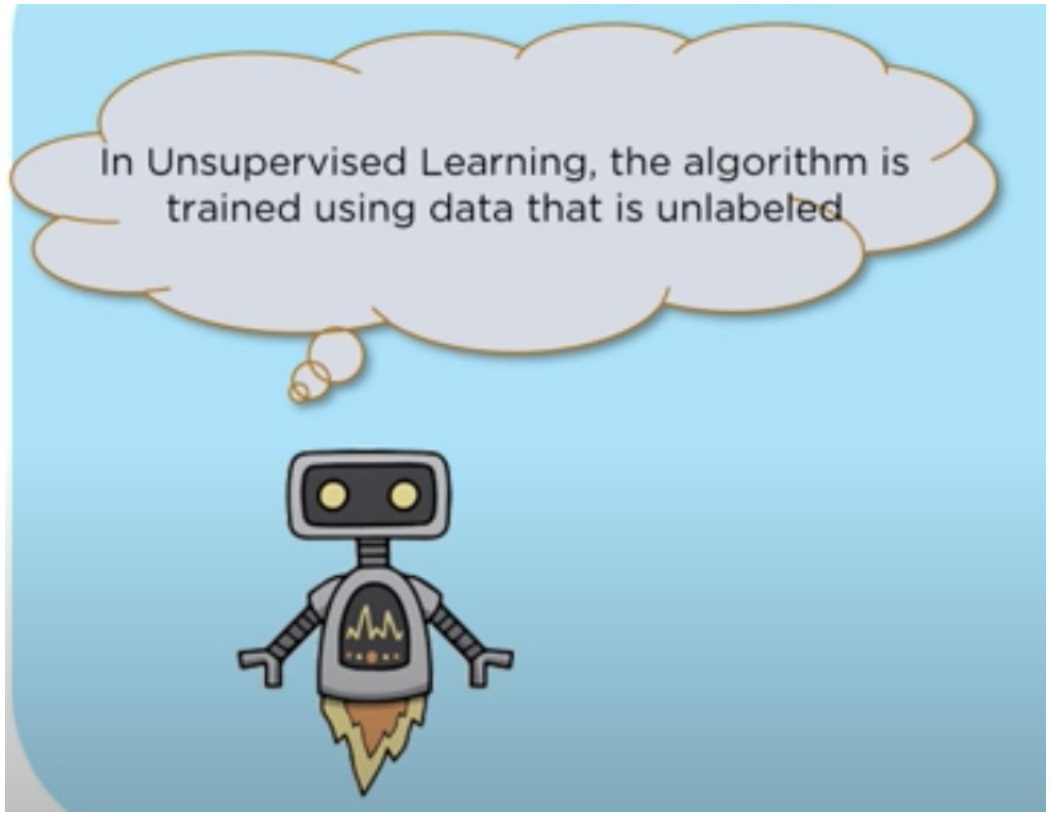
Unsupervised Learning

Machine Learning

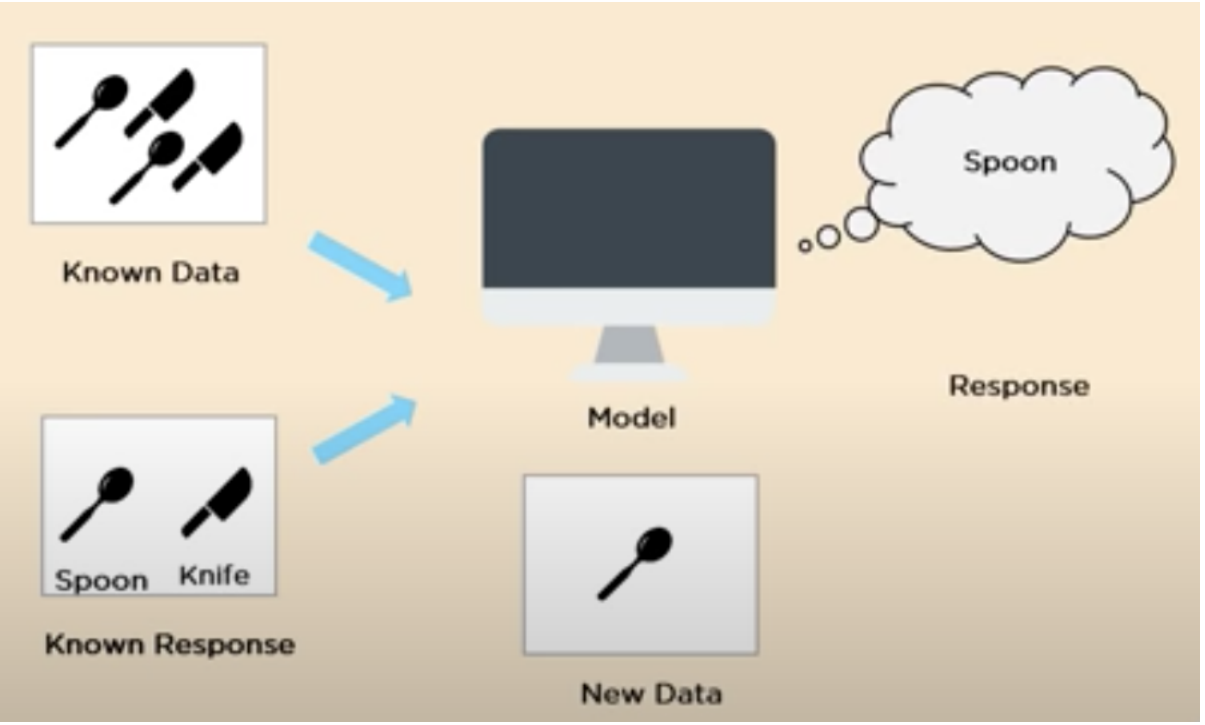
Supervised Learning



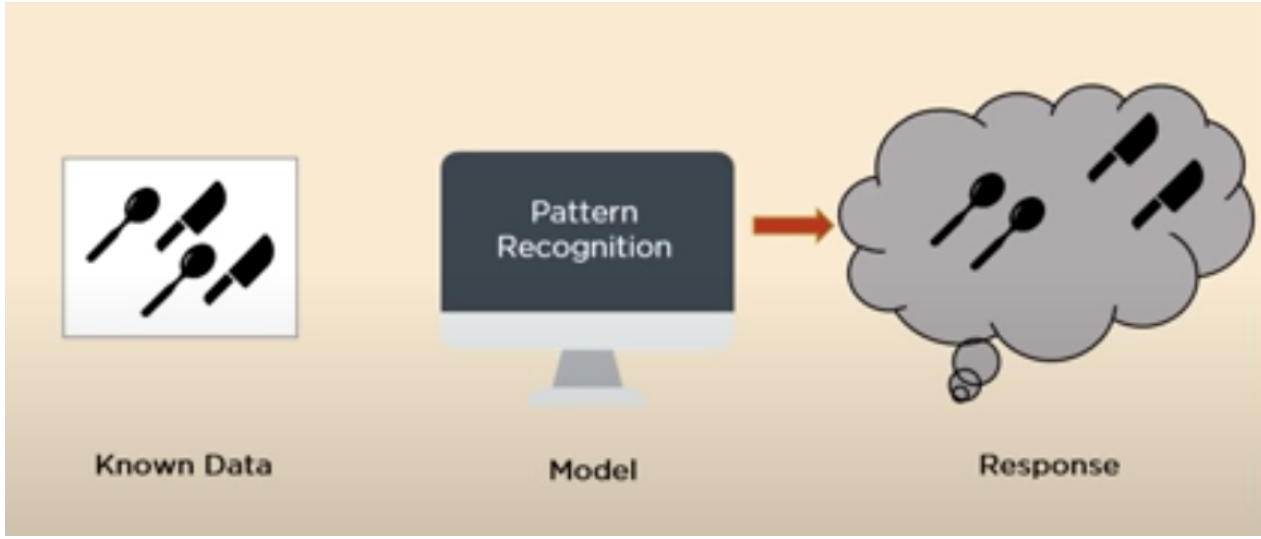
Unsupervised Learning



Supervised Learning



Unsupervised Learning



Supervised vs. Unsupervised Learning

	Supervised Learning	Unsupervised Learning
Objective	To **approximate a function** that maps inputs to outputs based on **example input-output pairs**.	To **build a concise representation of the data** and generate imaginative content from it.
Accuracy	Highly accurate and reliable.	Less accurate and reliable.
Complexity	Simpler method.	**Computationally complex.**
Classes	Number of classes is **known**.	Number of classes is **unknown**.
Output	A **desired output** value (also called the **supervisory** signal).	No corresponding output values.

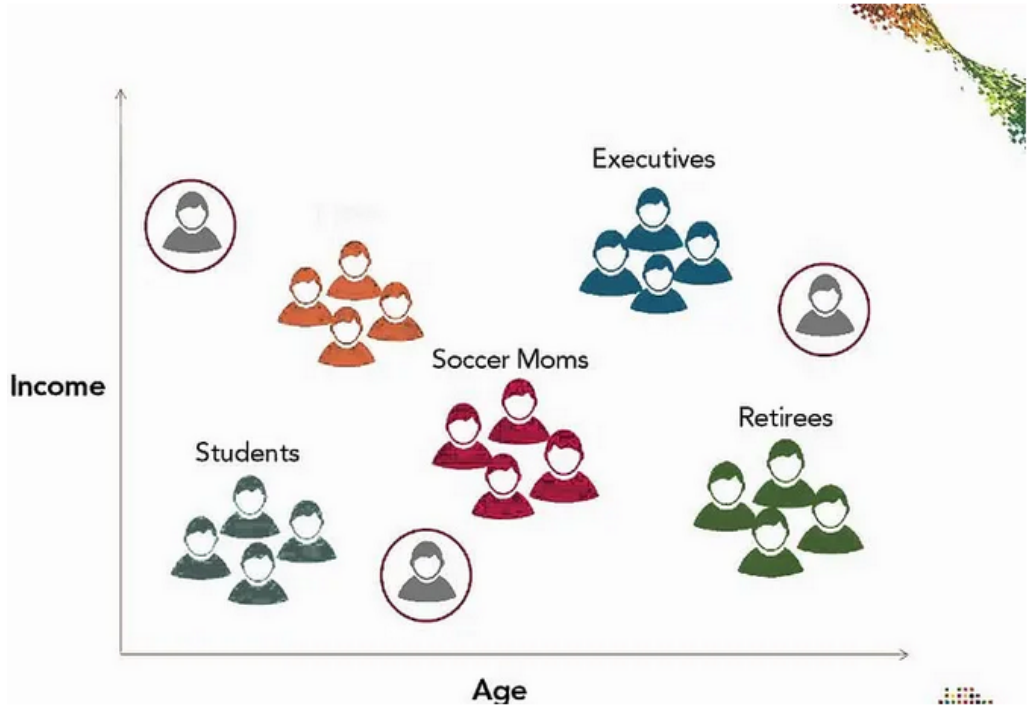
Types of Unsupervised Learning

- **Clustering**
  - Grouping similar data points together based on their features
  - **K-Means** – partitions data into *k* clusters
  - **Hierarchical Clustering** – builds a tree of clusters
- **Examples**
  - Customer segmentation in marketing
  - Image segmentation
  - Grouping similar documents

Types of Unsupervised Learning

- **Clustering**
  - Marget Segmentation: businesses
  - group *customers* into *types*

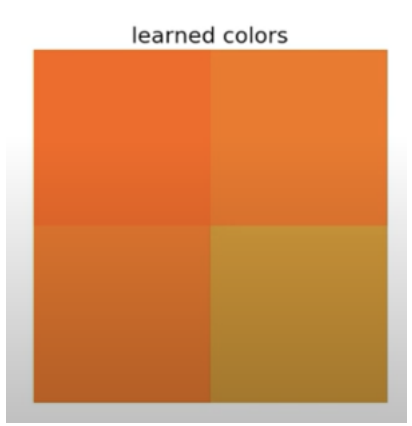
Market segmentation types			
Demographic	**Behavioral**	**Psychographic**	**Geographic**
• Age • Gender • Family size • Income • Occupation • Religion/Race • Nationality	• Purchases • Usage (heavy, moderate, lesser) • Special events • Benefits (customer's needs)	• social class • Personality: reluctant, receptive, impulsive,.. • Lifestyle • Activities • Interests • Opinions & posts	• Location (nations, states, regions, countries, cities) • Population density • Weather • language



Images source: https://medium.com/@ashim.maity8/customer-segmentation-using-kmeans-clustering-cd8edc0a8e87

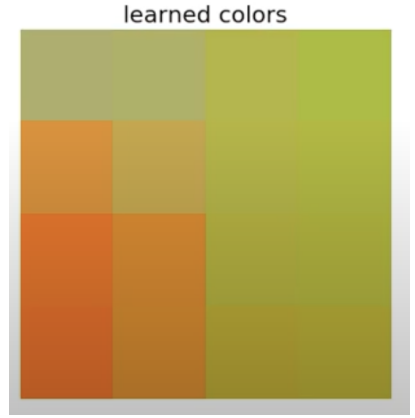
Types of Unsupervised Learning

- **Clustering**
  - Image segmentation: Self-organizing Maps (Kohonen networks)

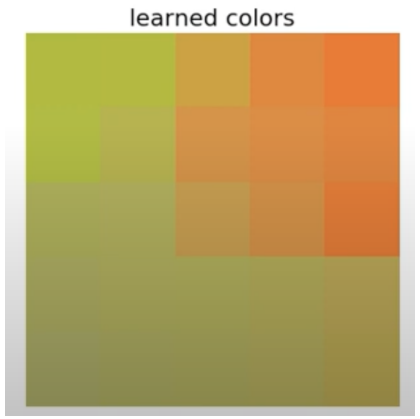


2x2

3x3



4x4



5x5

6x6

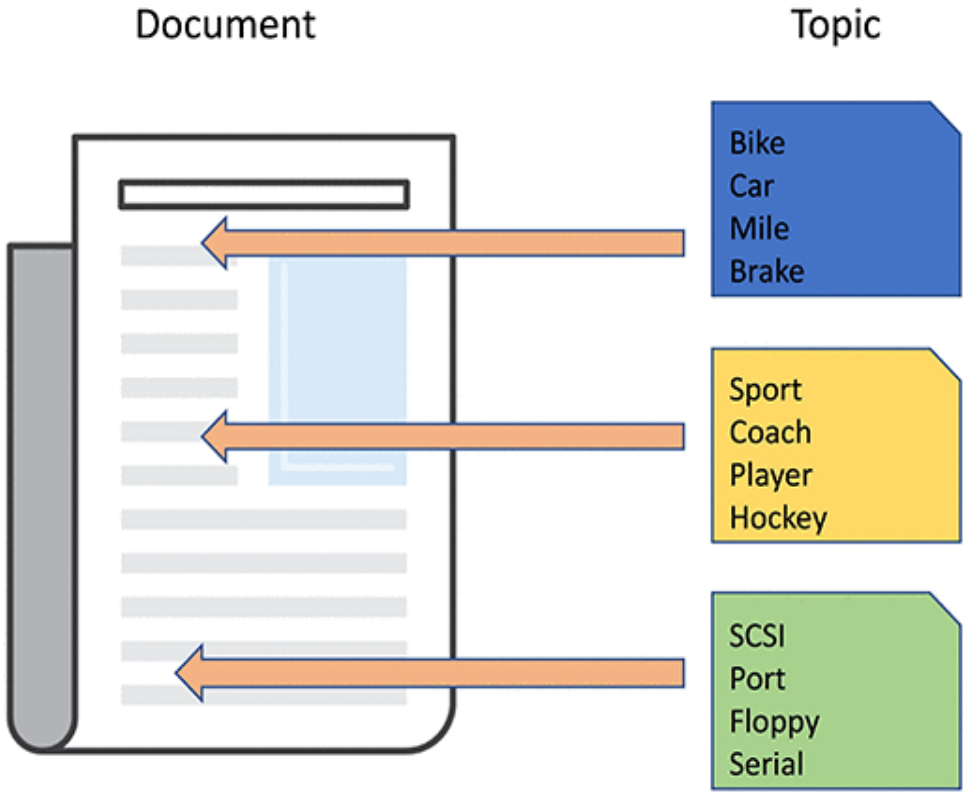
7x7

8x8

https://www.youtube.com/watch?v=-_d0h3VhpAs

Types of Unsupervised Learning

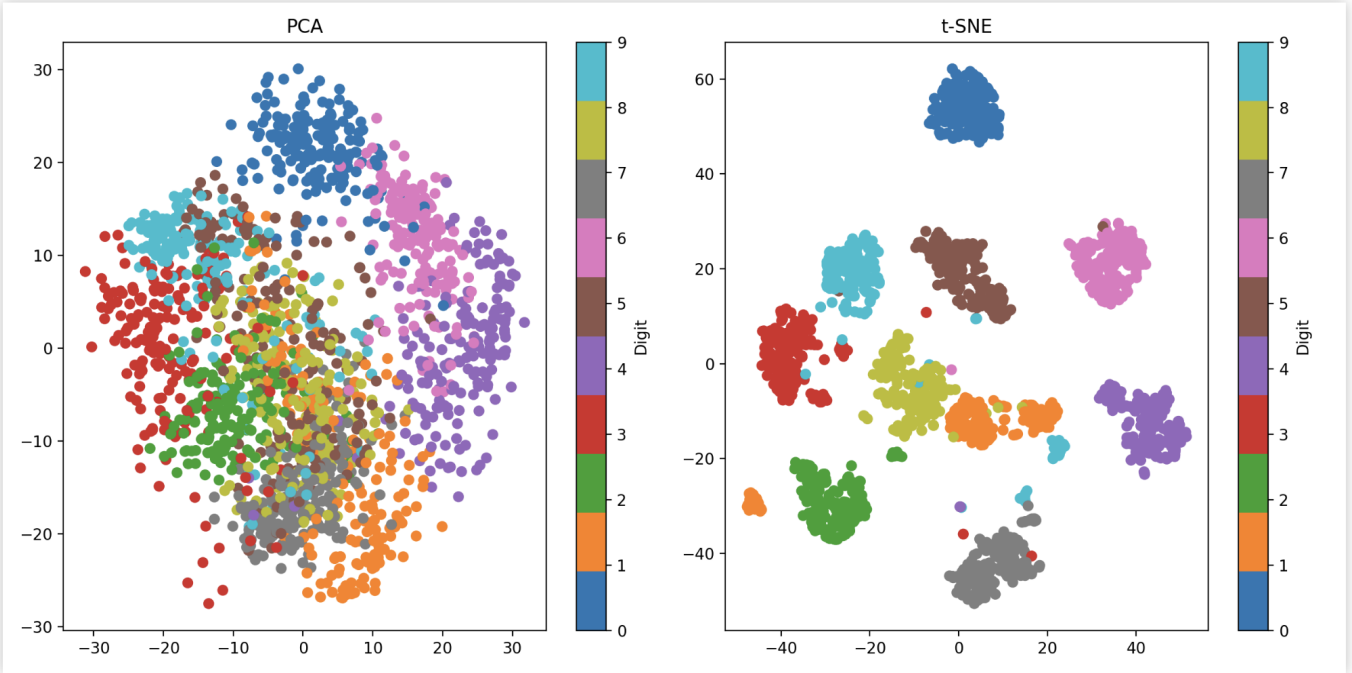
- **Clustering**
  - Grouping similar documents
    - Topic Modelling – cluster *documents* into *topics of discussion*
    - Linguistics – identify different uses of ambiguous words (e.g. “set”, “run”, “track”)



https://aws.amazon.com/blogs/machine-learning/introduction-to-the-amazon-sagemaker-neural-topic-model/

Types of Unsupervised Learning

- **Dimensionality Reduction**
  - **PCA (Principal Component Analysis)** – projects data to a lower-dimensional space.
  - **t-SNE** – good for 2D/3D visualization of clusters.

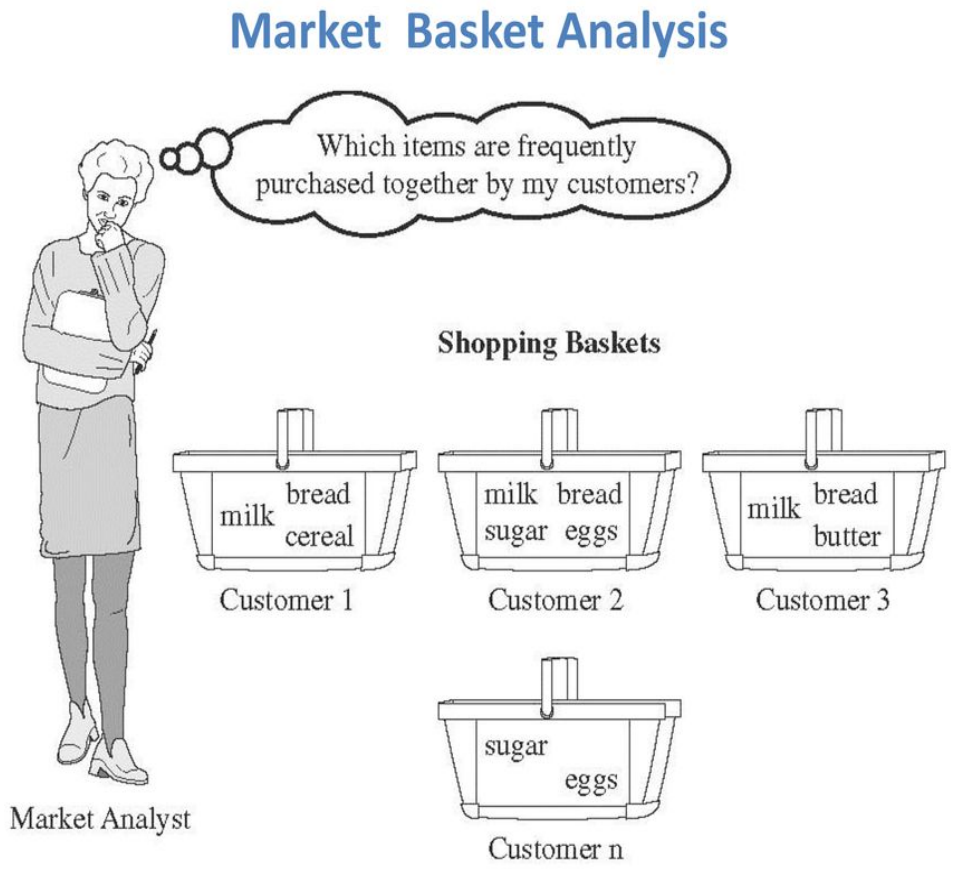


PCA (left) and t-SNE (right) applied to the handwritten digits dataset.

https://www.fabriziomusacchio.com/blog/2023-06-12-tsne_vs_pca/

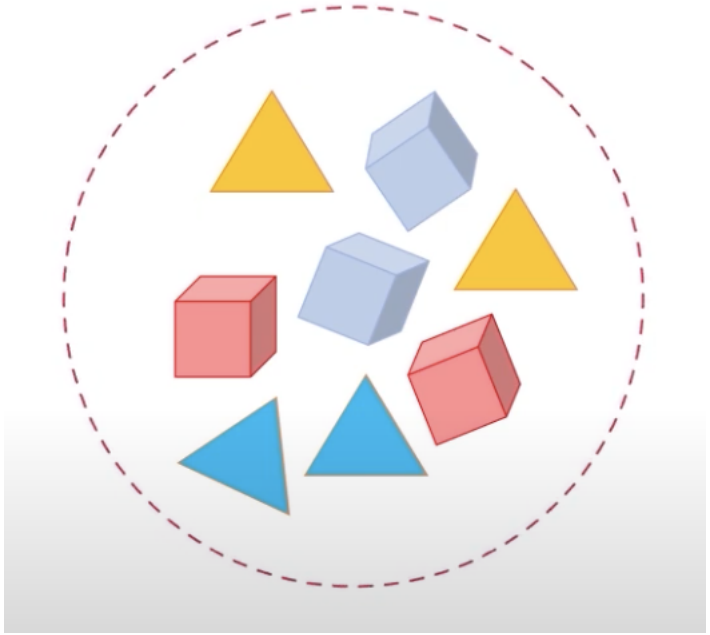
Types of Unsupervised Learning

- **Association Rule Mining**
  - Finding interesting relationships (rules) between variables in large datasets
  - Apriori algorithm – finds frequent itemsets in transactions
  - Eclat algorithm – a faster approach to finding associations
- **Examples**
  - Market basket analysis (e.g., "Customers who buy bread also buy butter").
  - Recommendation systems.

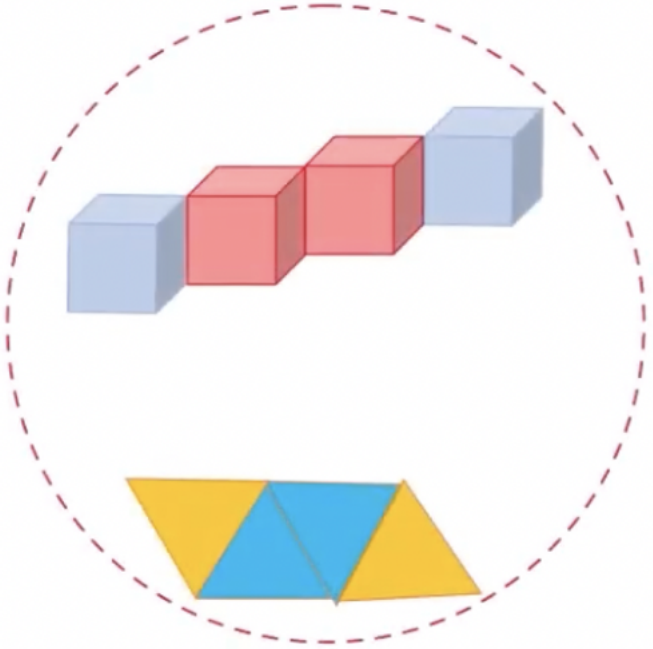


Source: Han & Kamber (2006)

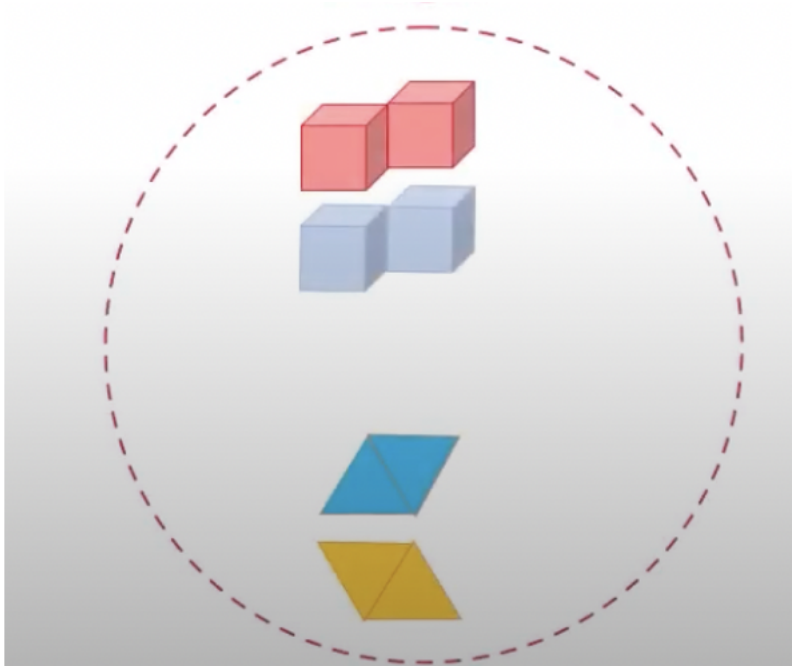
Clustering



similarity



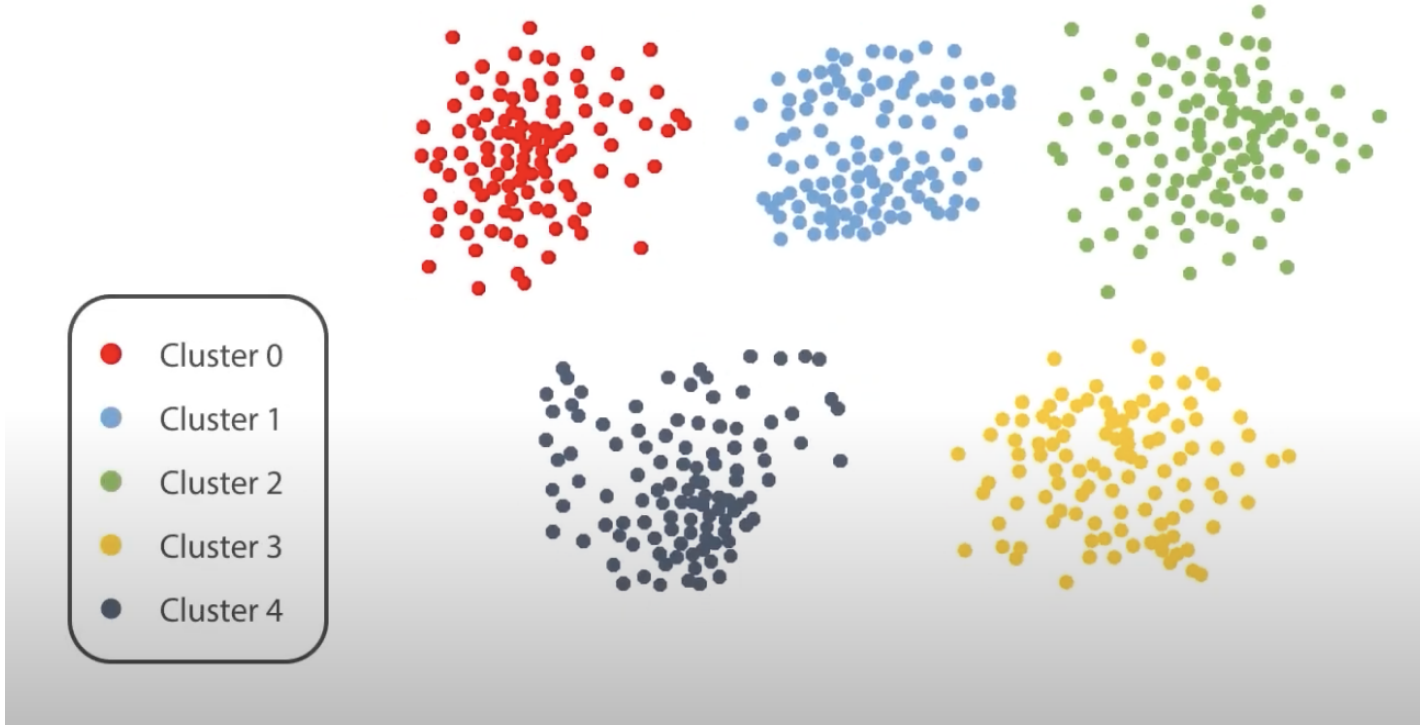
shape

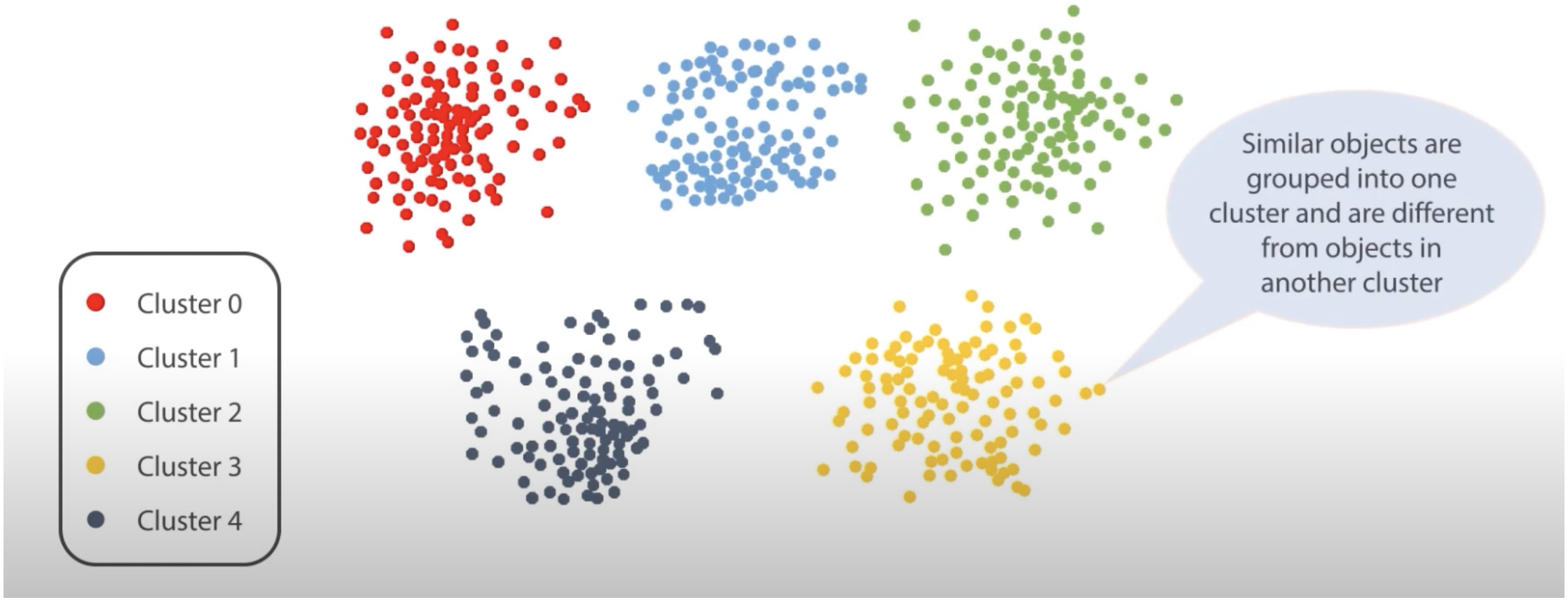


color

Clustering

- Splitting data in similar groups: clusters
- Cluster = collection of similar data (that are different from data existing in other clusters).




Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Similar objects are grouped into one cluster and are different from objects in another cluster

Clustering

- Partinional clustering
  - K-means

A division of the set of data objects into non-overlapping sets or clusters such that every data object is in just one subset.

- Hierarchical clustering
  - Agglomerative
  - Divisive

A tree structure that has a set of nested clusters.

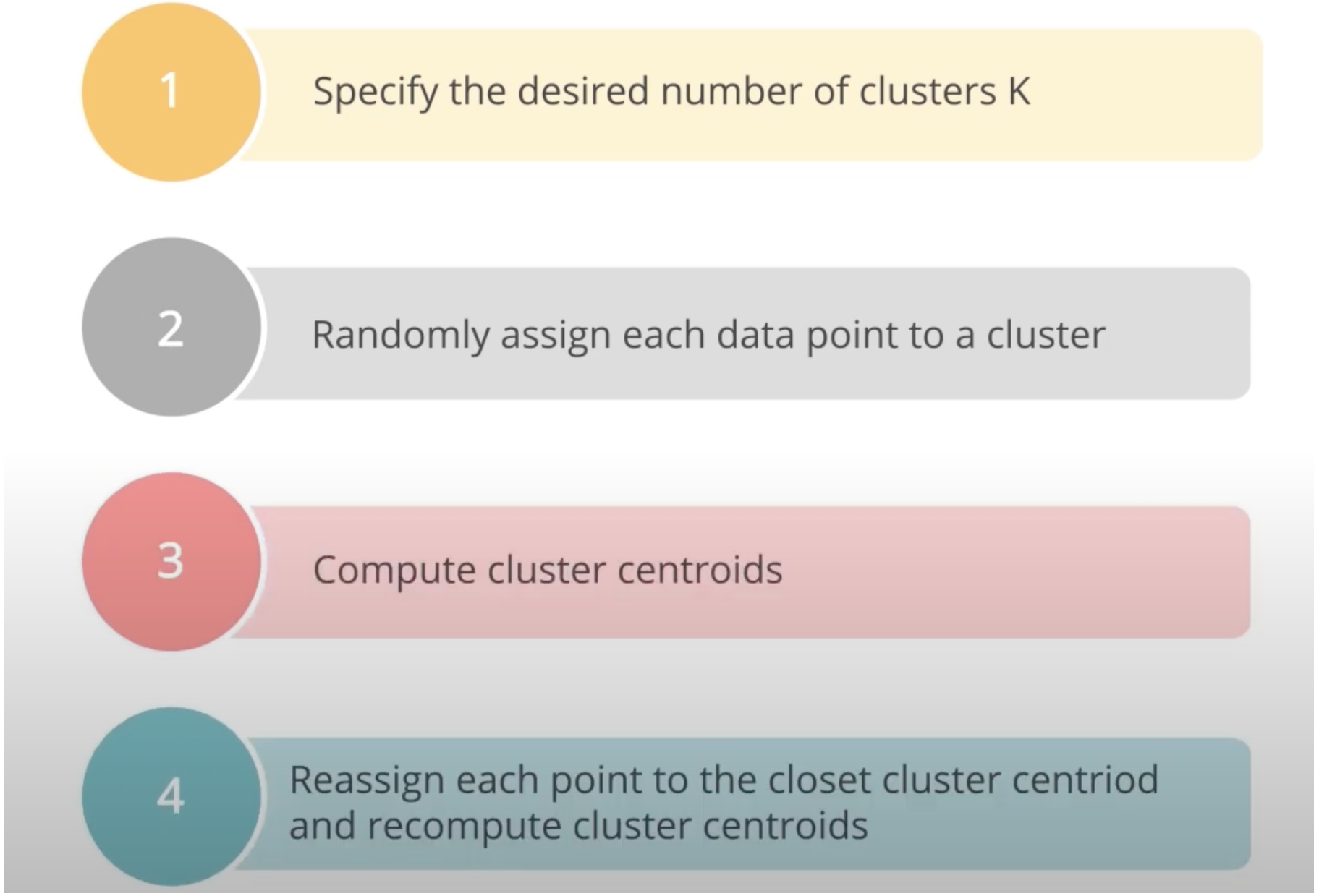
K-Means

- K-means (MacQueen, 1967) – partitioning algorithm using fixed k number of clusters
- Dataset D: $\{x_1 , x_2 , \ldots, x_n \}$,

  $x_i = (x_{i1} , x_{i2} , \ldots, x_{ir})$ array with r elements

- K-means algorithm makes k clusters:
  - Each cluster is characterised by its centroid
  - k is fixed from the beginning

K-means


1
Specify the desired number of clusters K
2
Randomly assign each data point to a cluster
3
Compute cluster centroids
4
Reassign each point to the closet cluster centriod and recompute cluster centroids

K-Means

Divide the dataset into k clusters (k fixed):

1. Randomly initialize k centroids (from the datasets)
2. Associate each element (from the dataset) to a cluster
3. Update centroids (using the current values of the centroids)
4. Repeat from step 2

Convergence criteria

Divide the dataset into k clusters (fixed k):

- Centroids are fixed (minimal updates)

or

- Elements are fixed to clusters (minimal distribution)

or

- Minimum decrease of the error (Sum of Squared Error):

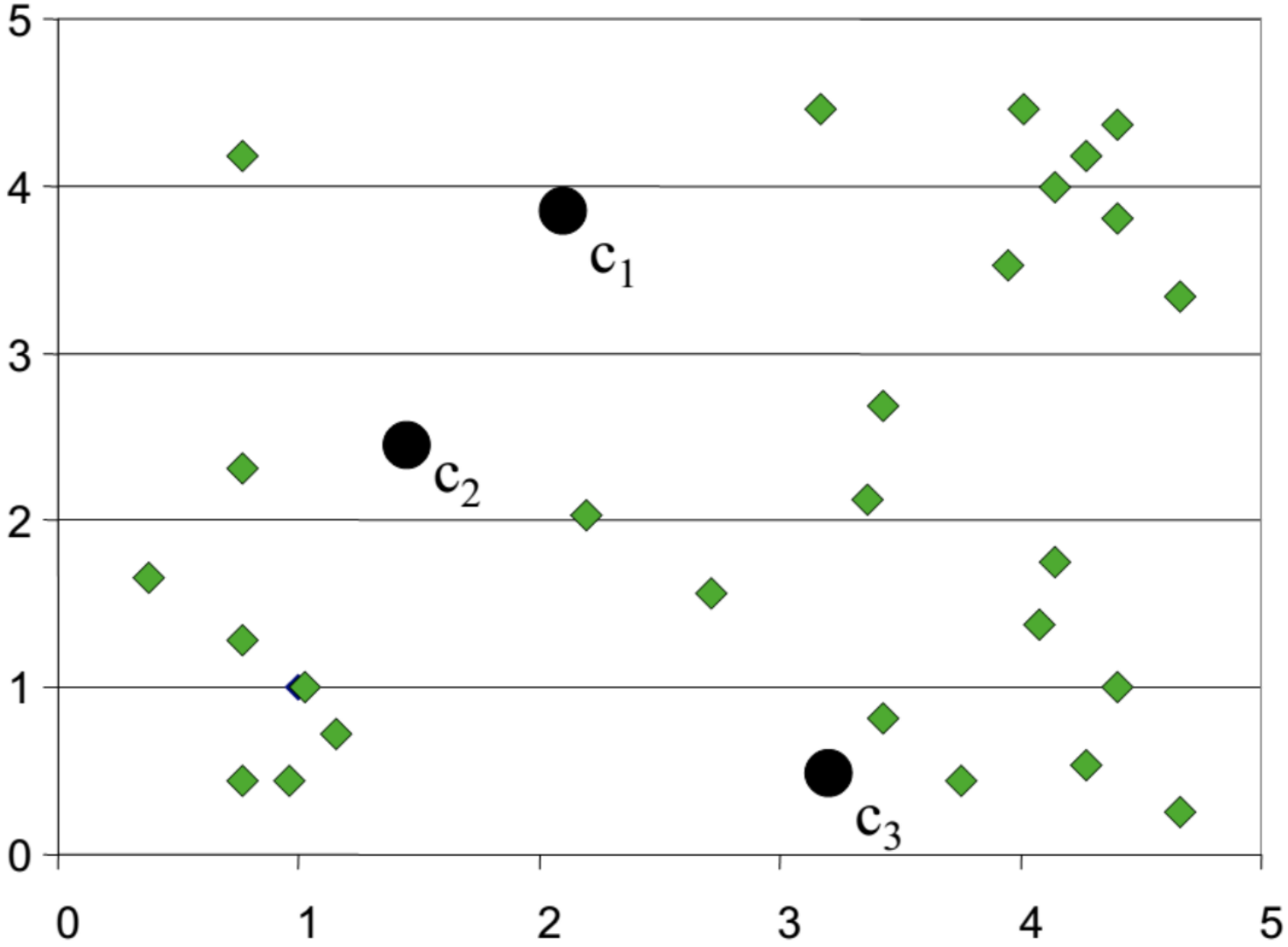
$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} d(\mathbf{x}, \mathbf{m}_j)^2$$

C_j – cluster j

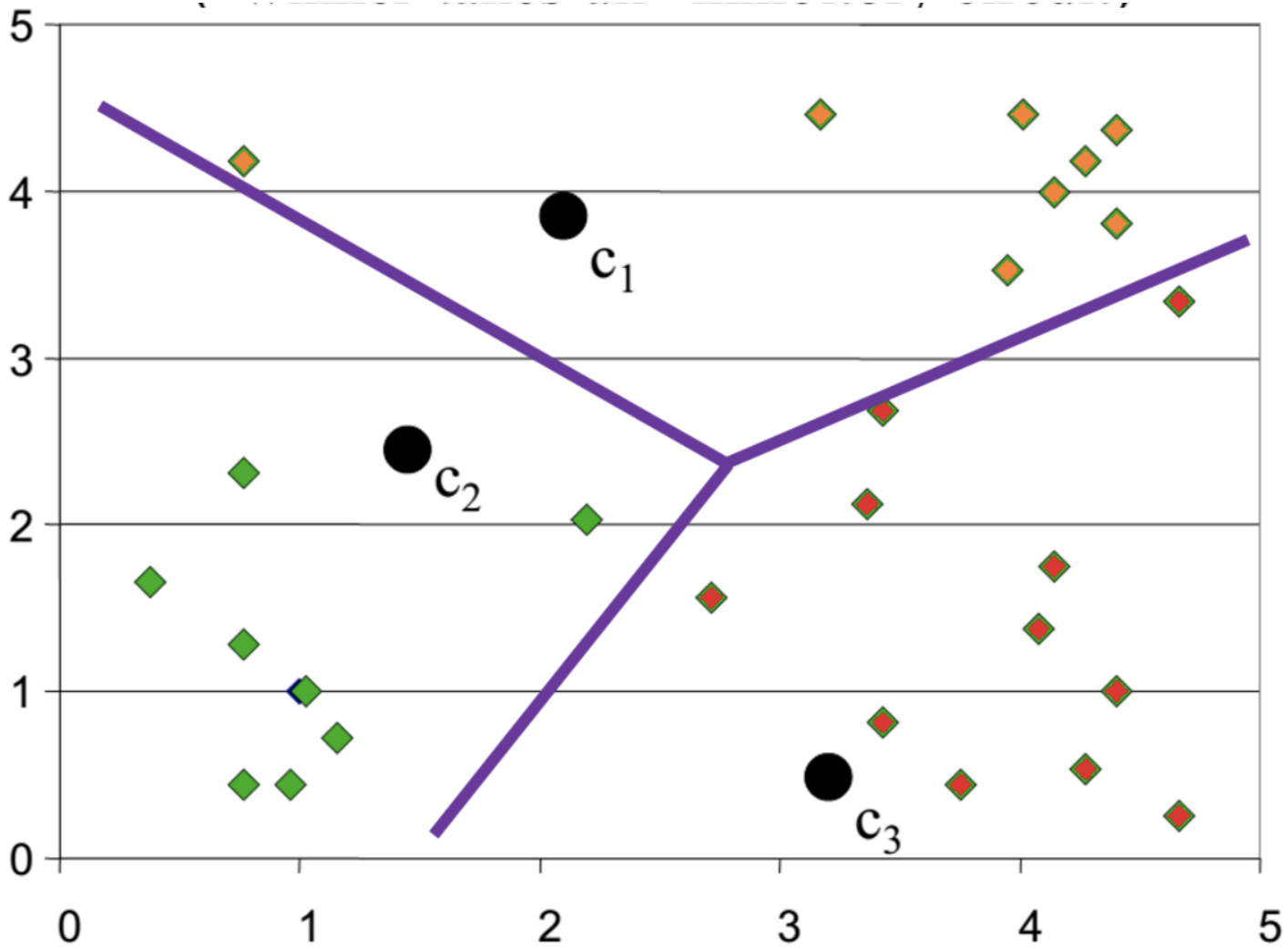
m_j – centroid for cluster j

$d(x,m_j)$ – distance between element x and centroid m_j

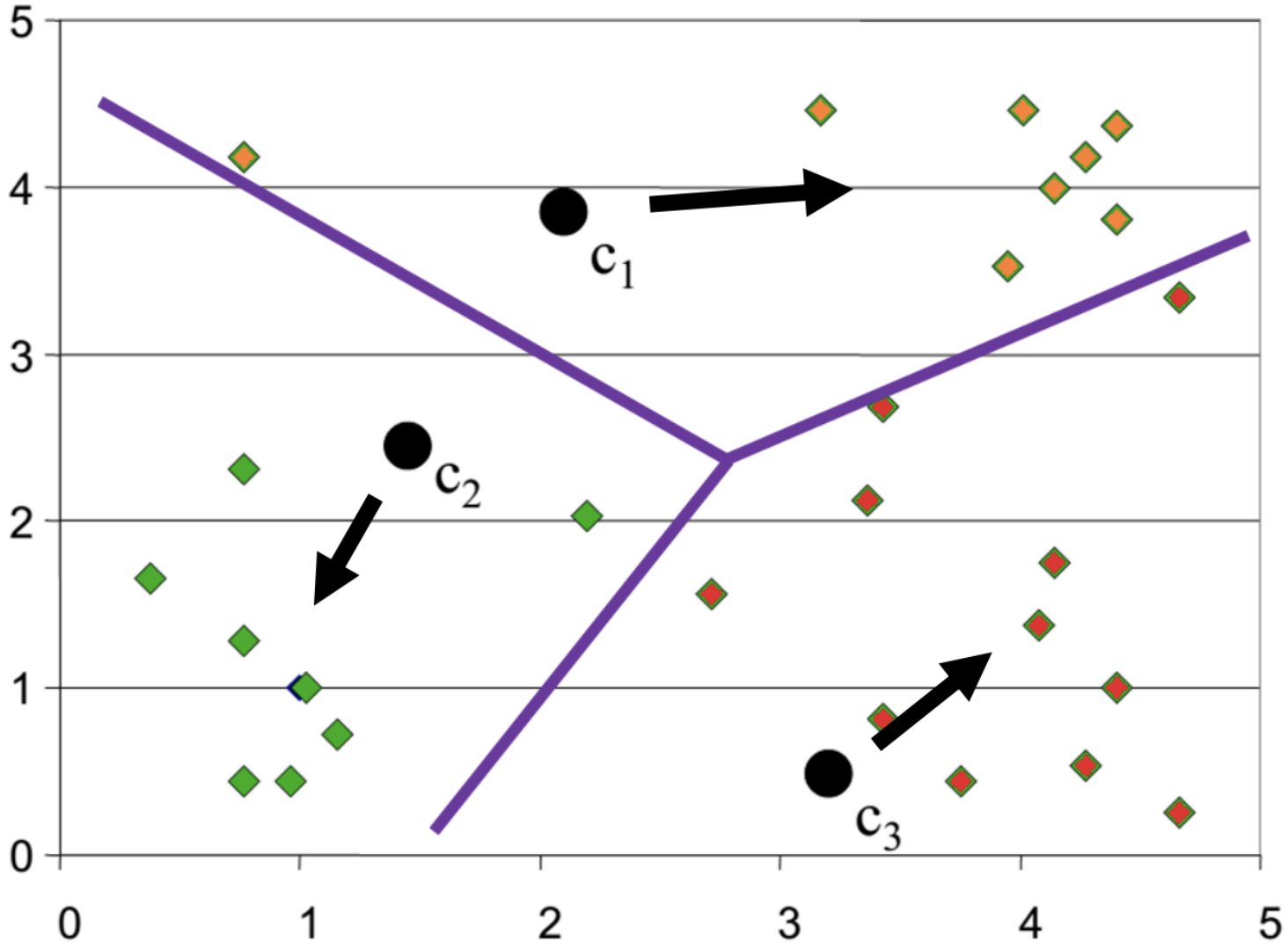
K-Means



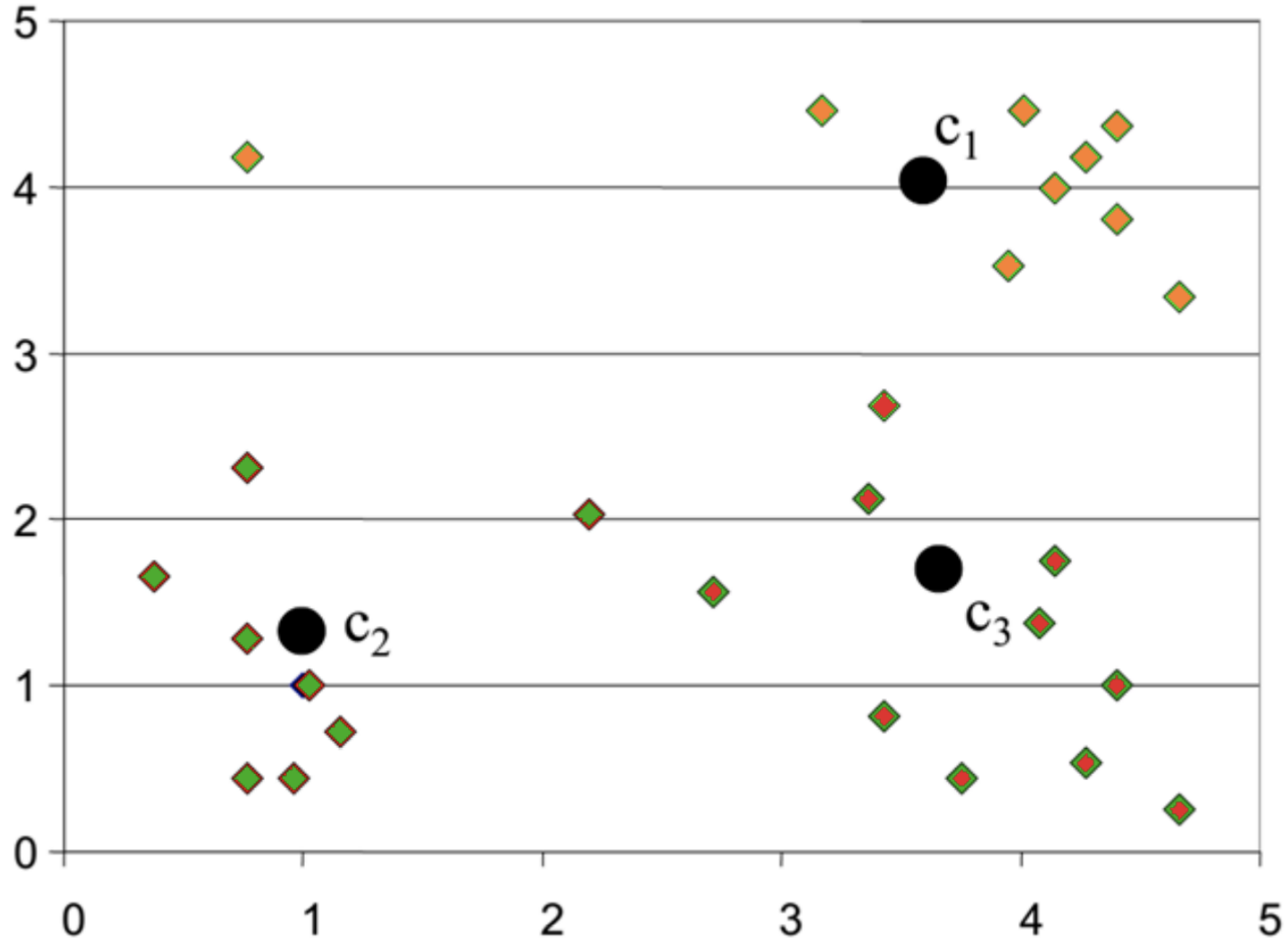
K-Means



K-Means



K-Means example



K-Means

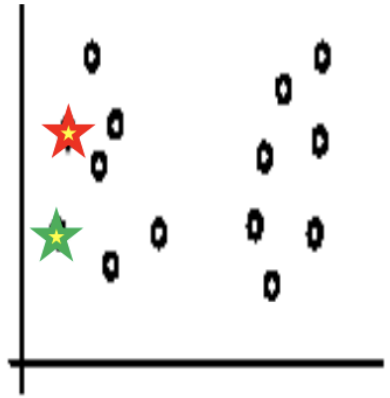
- Advantages:
  - Easy of use and implementation
  - Efficient:
    - Complexity: O(tkn)
      - n: number of elements from the datasets
      - k: number of clusters
      - t: number of iterations.

- k and t have low values ? k-means is a linear algorithm

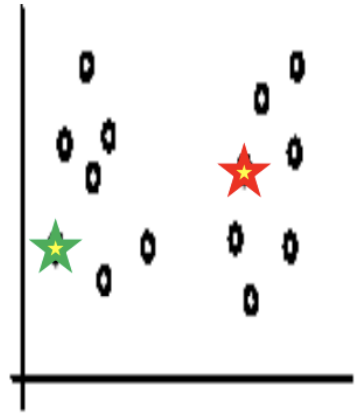
K-Means

- Disadvantages:
  - K specified from the beginning
  - Sensitive for outliers
    - Elements that are far away from other elements
    - Elements with very different values
    - Sensitive to initial centroids

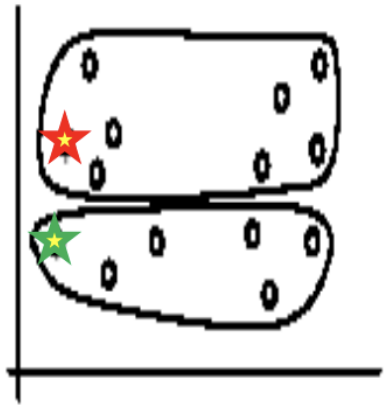
Sensitive at initial centroids



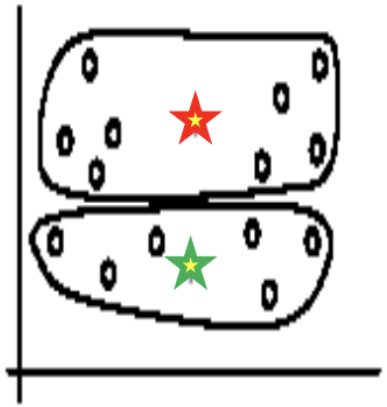
Chosing randomly centroids



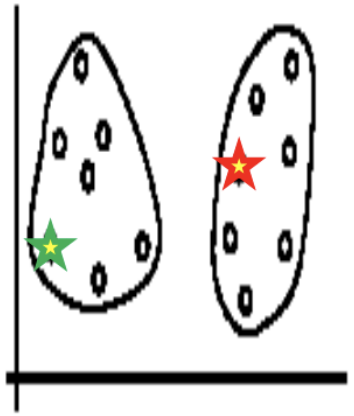
Chosing randomly centroids



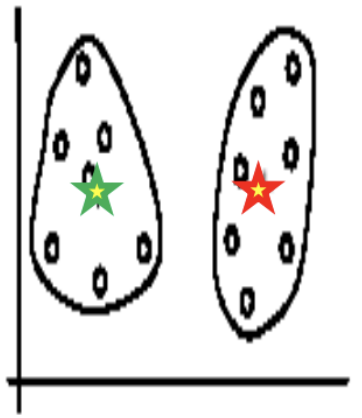
Iteration 1



Iteration 2



Iteration 1



Iteration 2

K-Means++

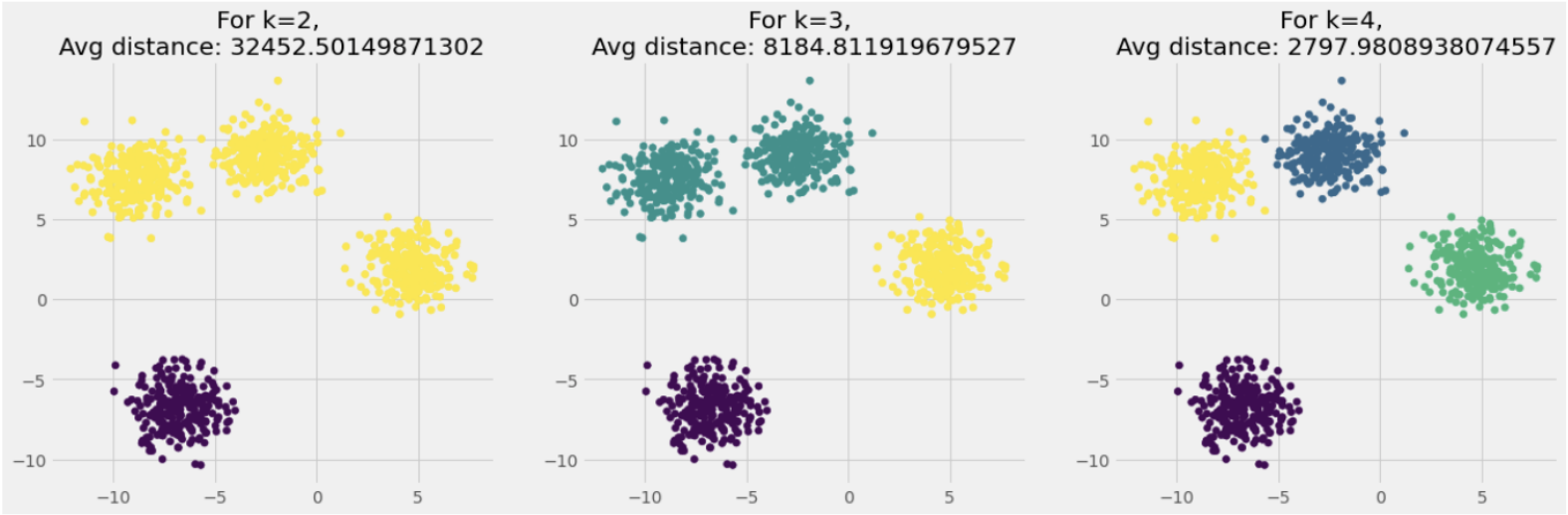
- **Initialise centroids**
- Select the first centroid
- Repeat for the rest of k-1 centroids
  - Compute the minimum distance for each element and each centroids (that are already selected)
  - Select the next centroid from the data points such that the probability of choosing a point as centroid is directly proportional to its distance from the nearest, previously chosen centroid. (i.e. the point having maximum distance from the nearest centroid is most likely to be selected next as a centroid)
- Repeat until k centroids have beer

$$p_i = \frac{D(x_i)^2}{\sum_{x \in X_S} D(x)^2}$$

Evaluate K-means

- **Elbow method**

- *K = ? using SSE (*sum of squared distance between elements and their centroids).
- select *k* where SSE become constant.



Elbow Plot

Evaluate K-Means

- **Silhouette method**

- **Silhouette Method** to find the best k
- **Better alternative to Elbow Method**

For each cluster:

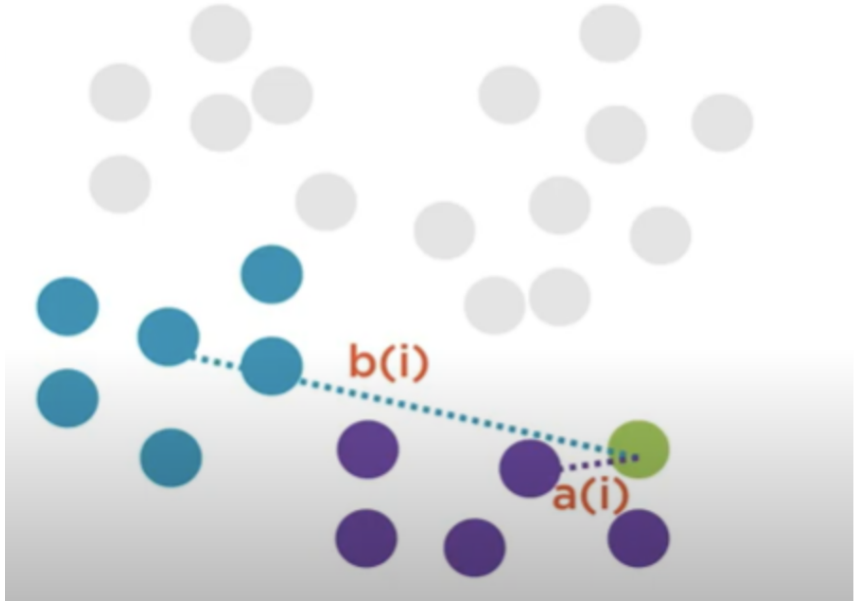
- Compute the mean distance between the elements of each cluster (a^i).
- Compute the mean distance between the elements from similar clusters (closest cluster) (b^i).

- After computing the silhouette coefficient for each element, average it out to get the silhouette score

Evaluate K-Means

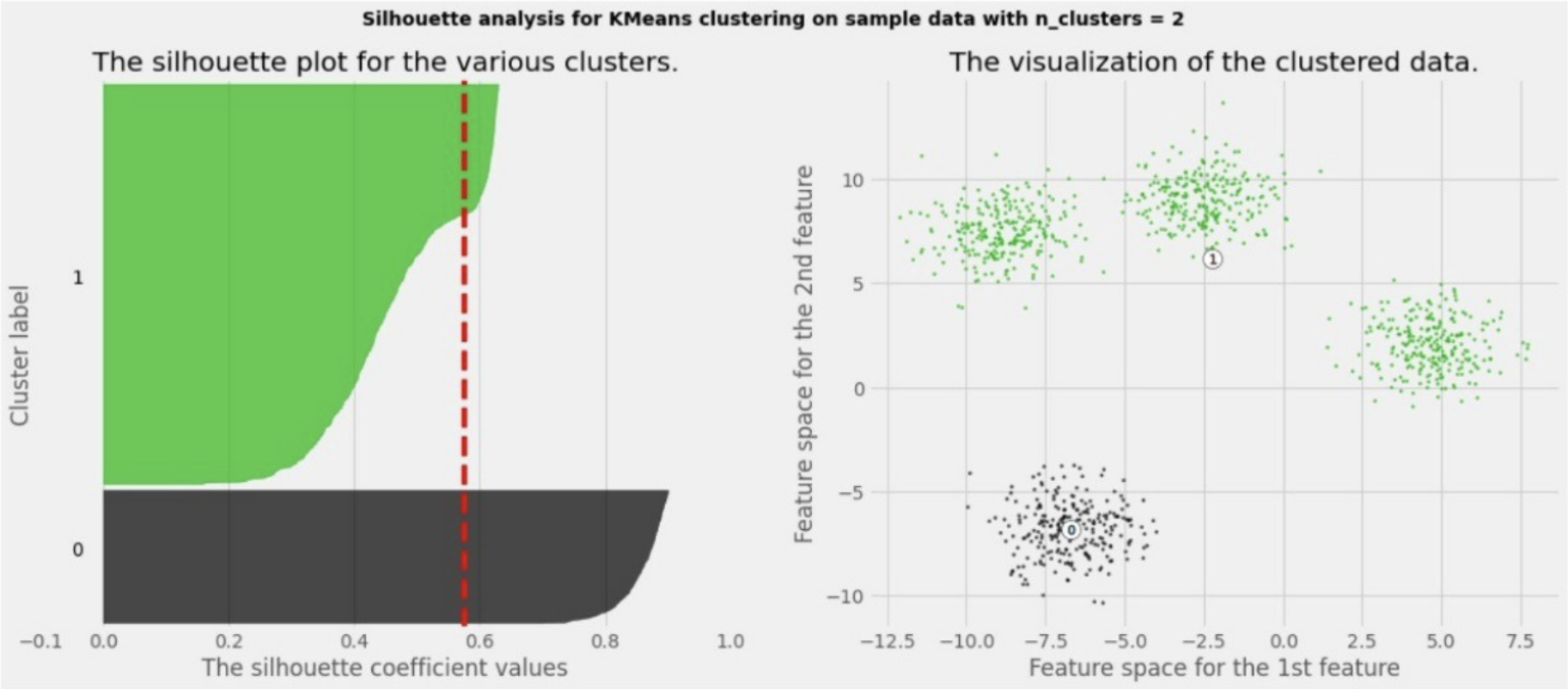
- Coefficient:

$$\frac{b^i - a^i}{max(a^i, b^i)}$$



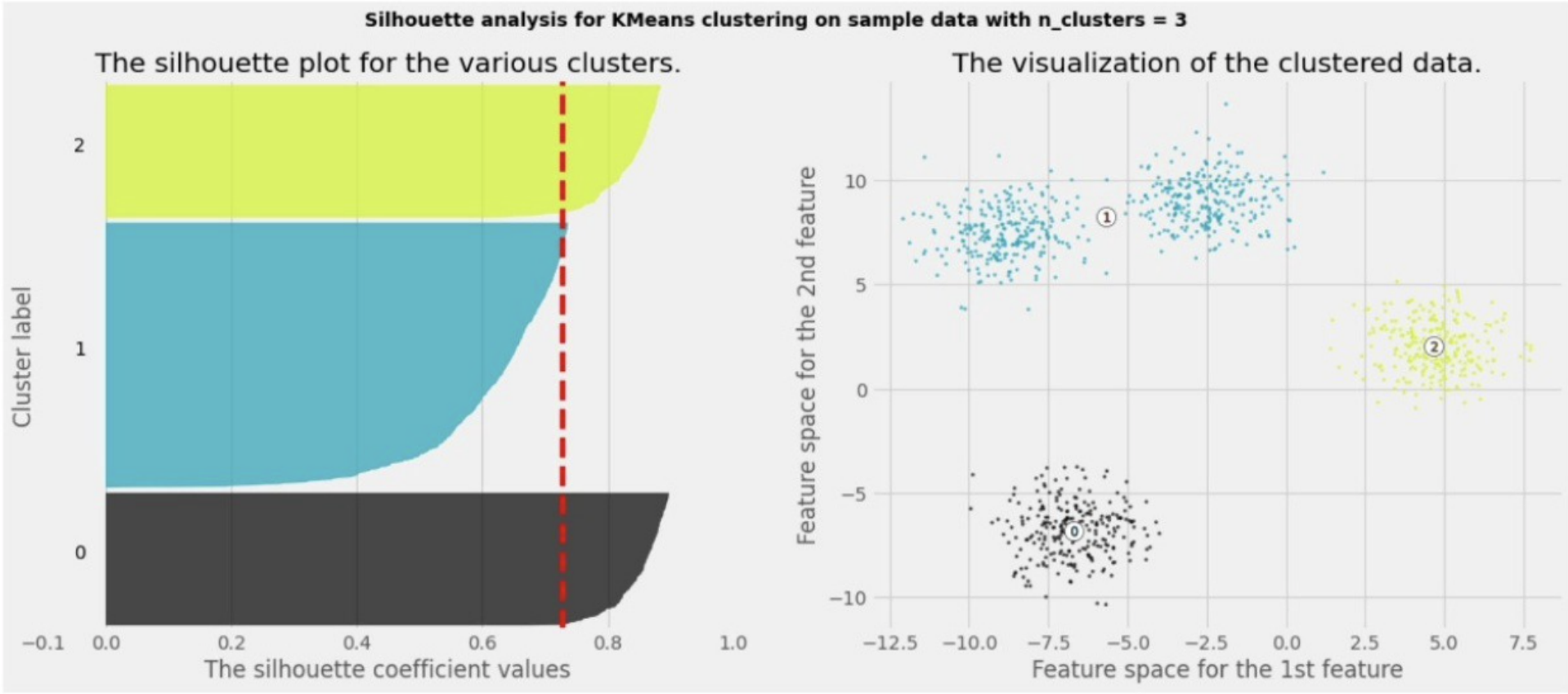
- Coefficient: [-1, 1].
  - almost 1: good clustering
  - <0: bad clustering (an example is associated to a wrong cluster – there is a similar cluster)
  - 0: overlap clusters

Evaluate K-Means



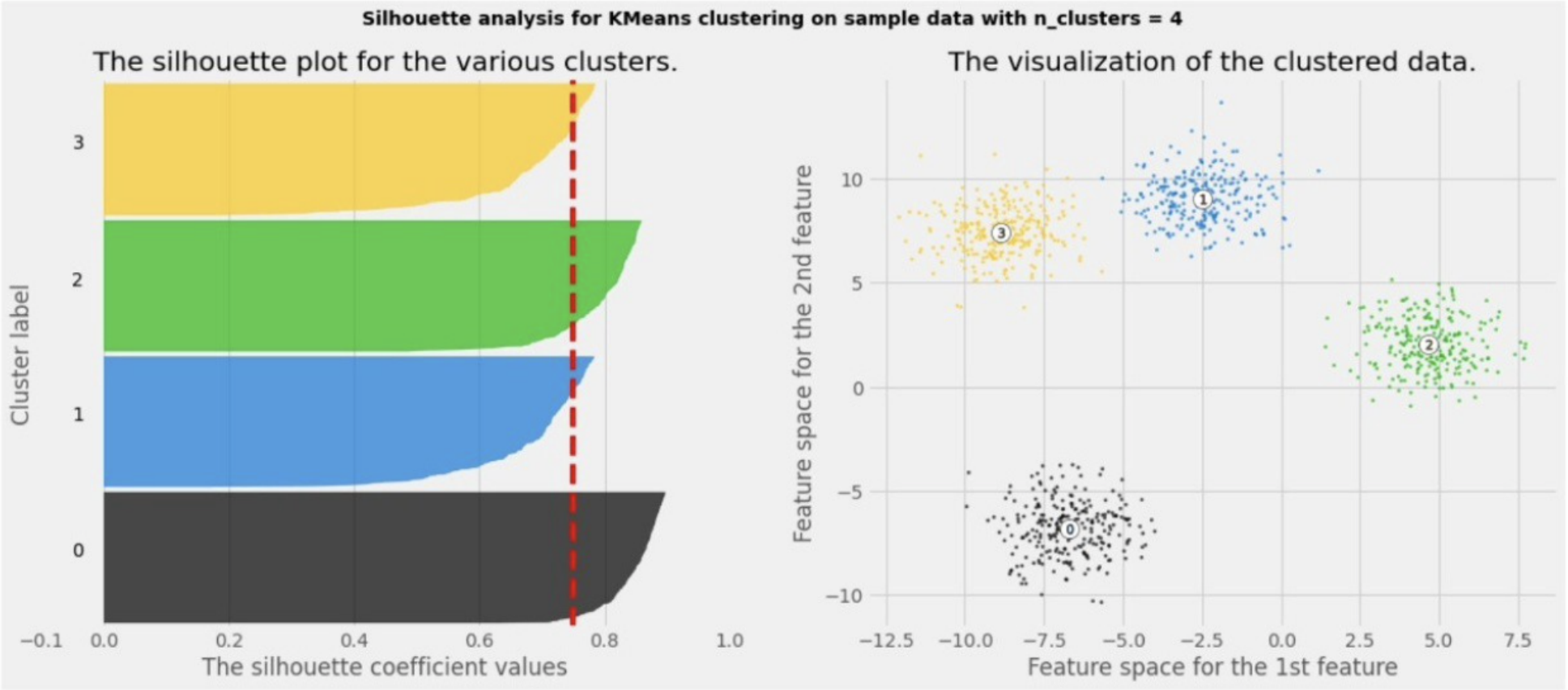
Average silhouette score

Evaluate K-Means

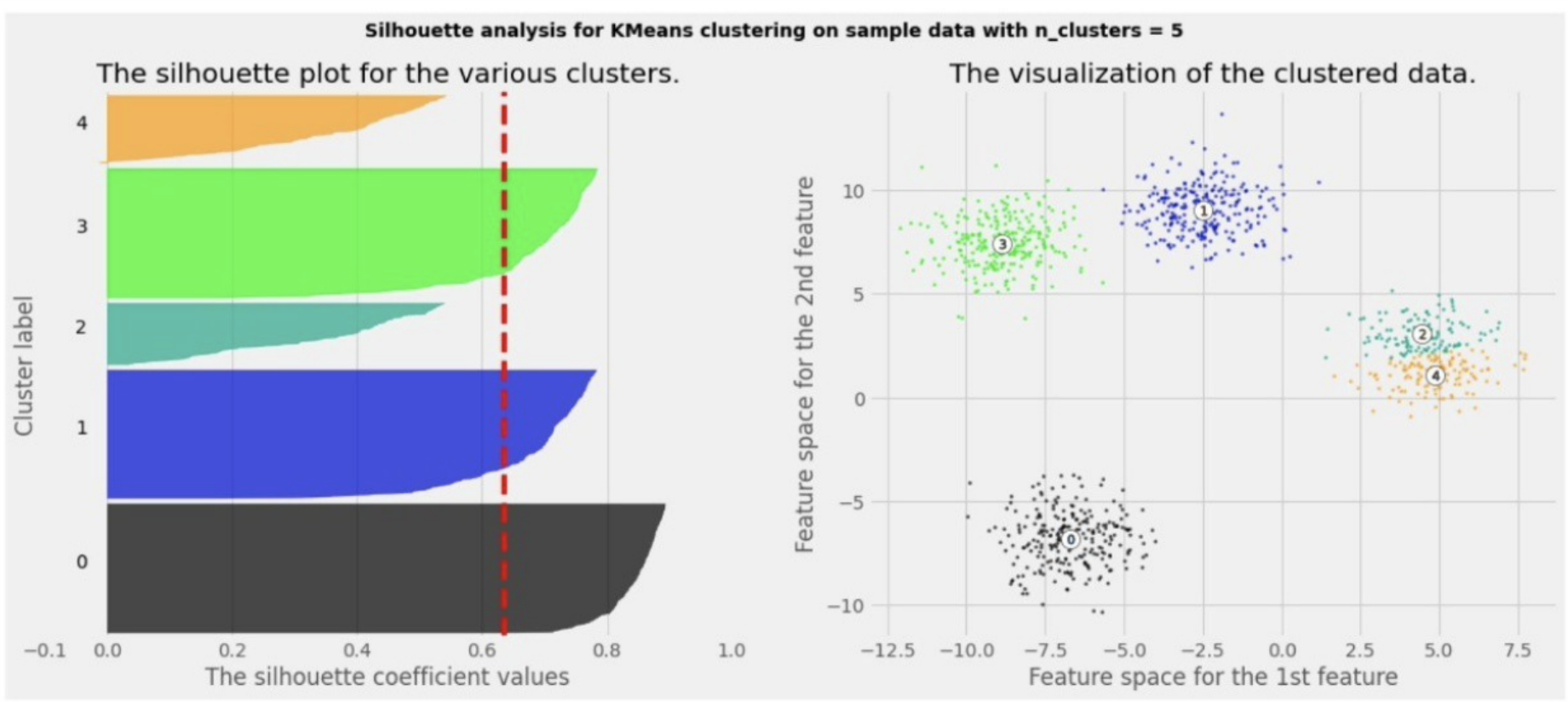


all the points in the cluster with cluster label=1 are below-average silhouette scores

Evaluate K-Means

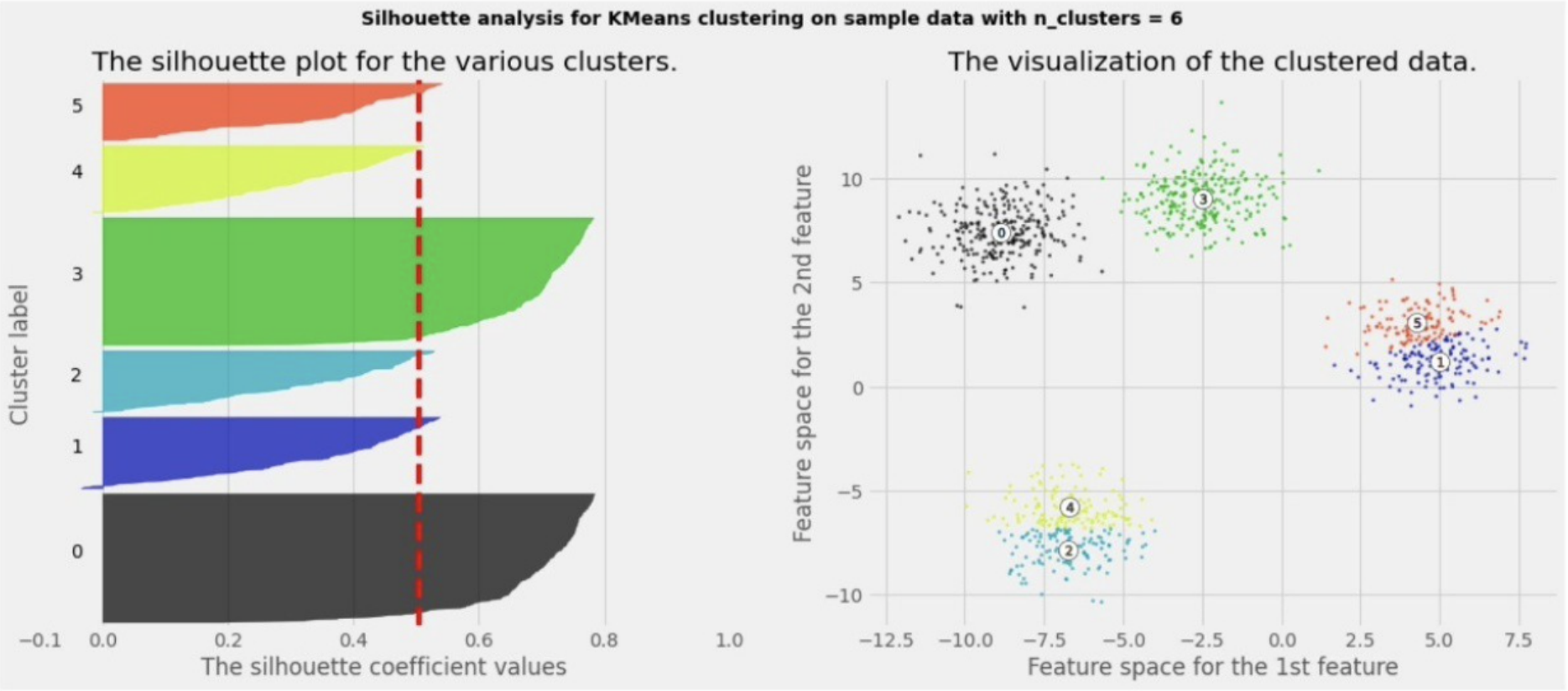


Evaluate K-Means



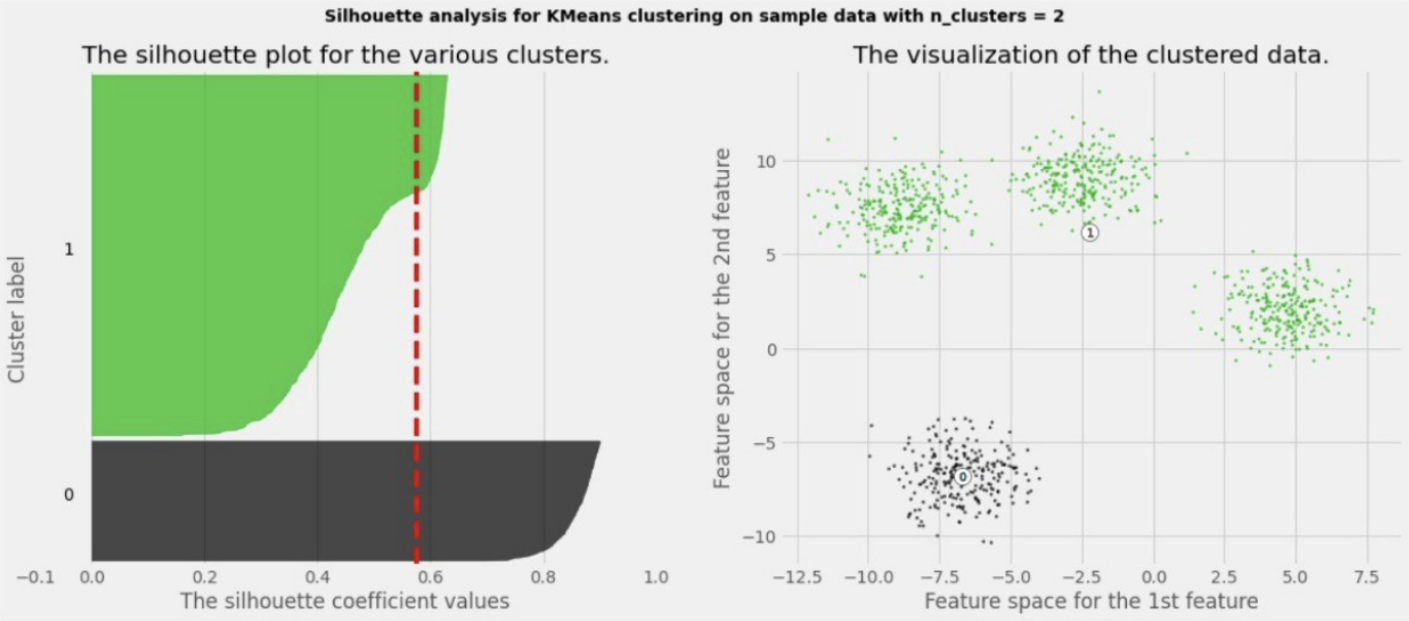
all the points in the cluster with cluster label=2 and 4 are below-average silhouette scores

Evaluate K-Means

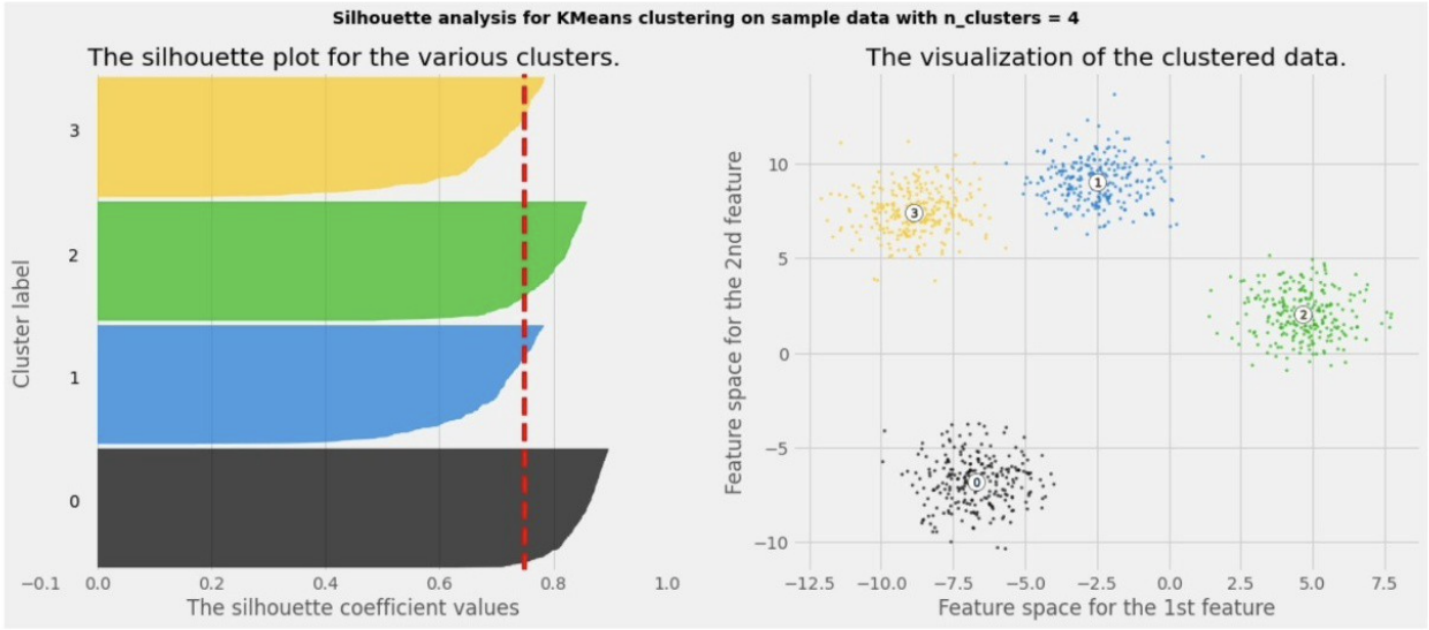


all the points in the cluster with cluster label=1,2,4 and 5 are below-average silhouette scores

Evaluate K-Means

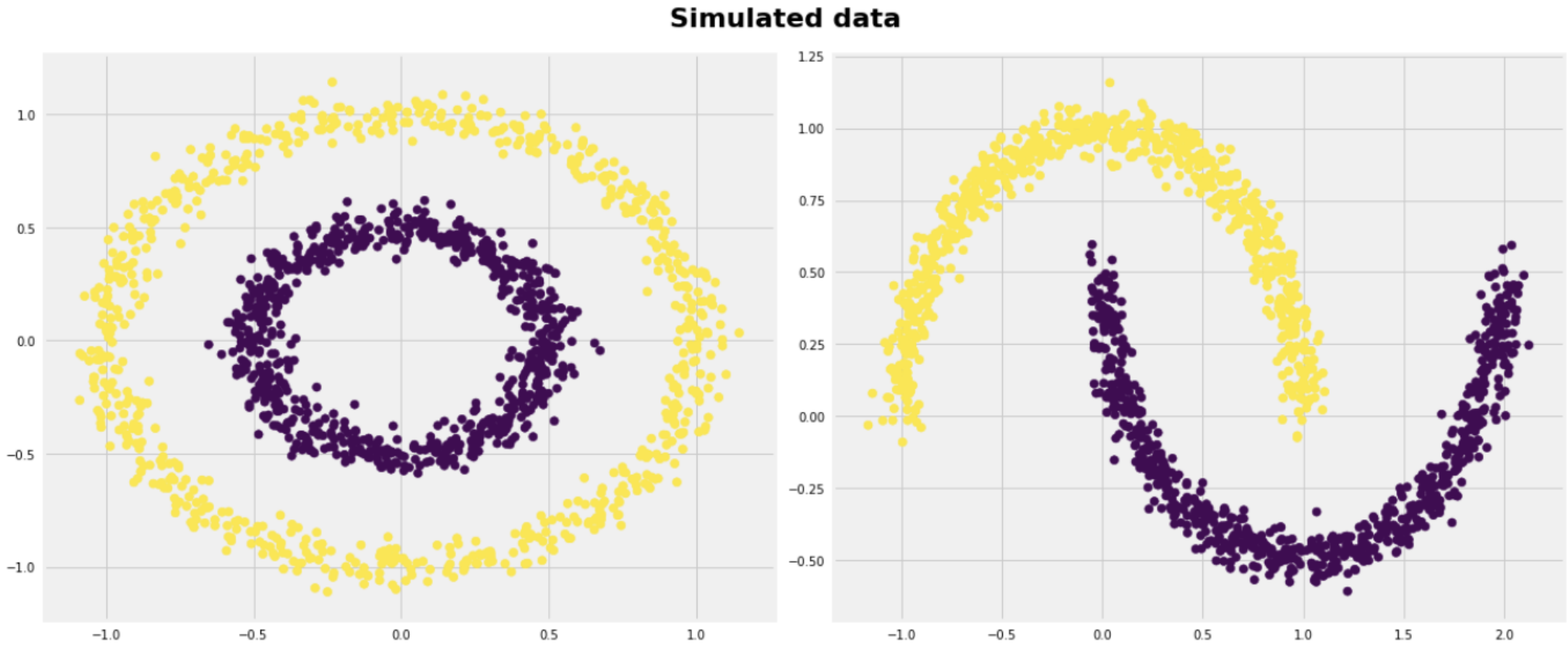


cluster label=1 is bigger in size owing to the grouping of the 3 sub-clusters into one big cluster

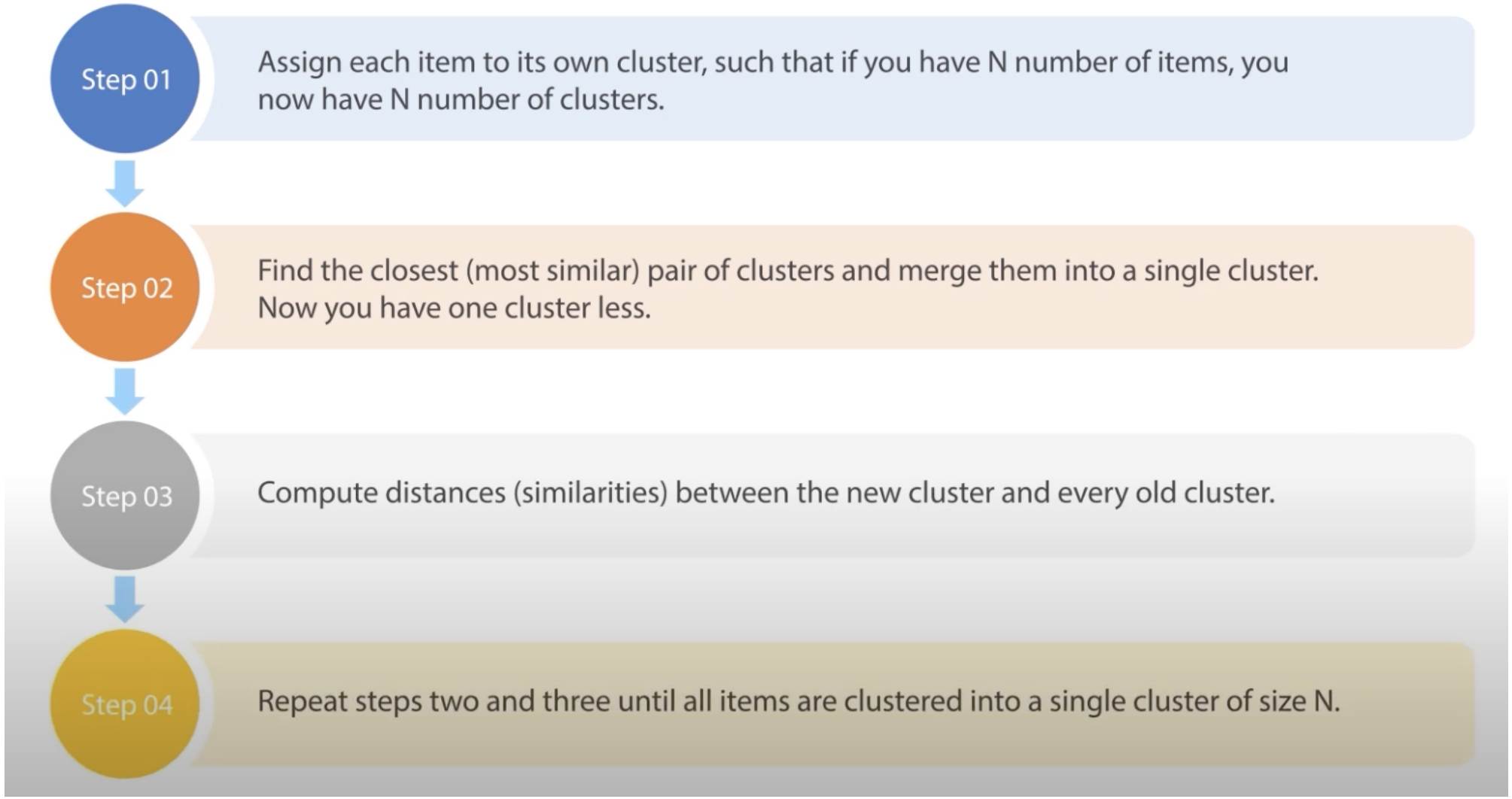


all the plots are more or less of similar thickness and hence are of similar sizes, as can be considered as **best 'k'**

Bad clustering

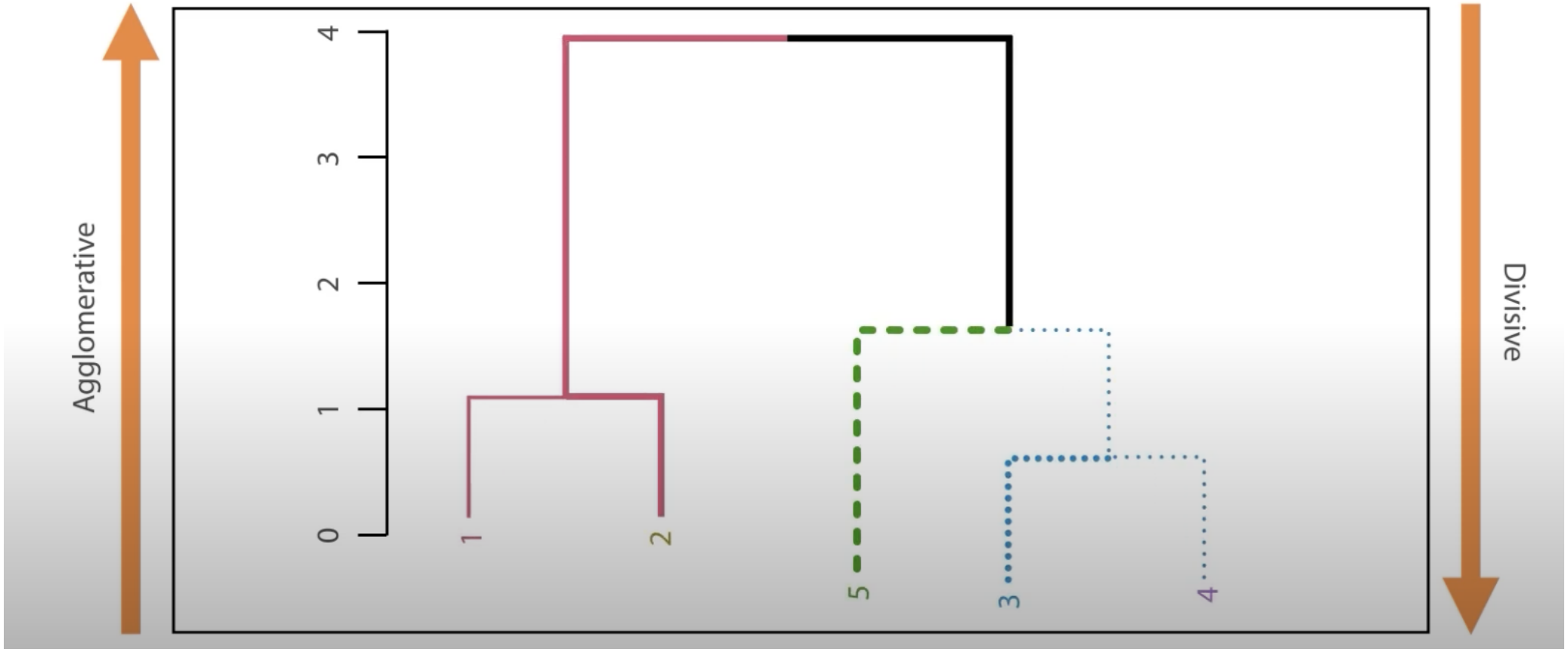


Hierarchical Clustering

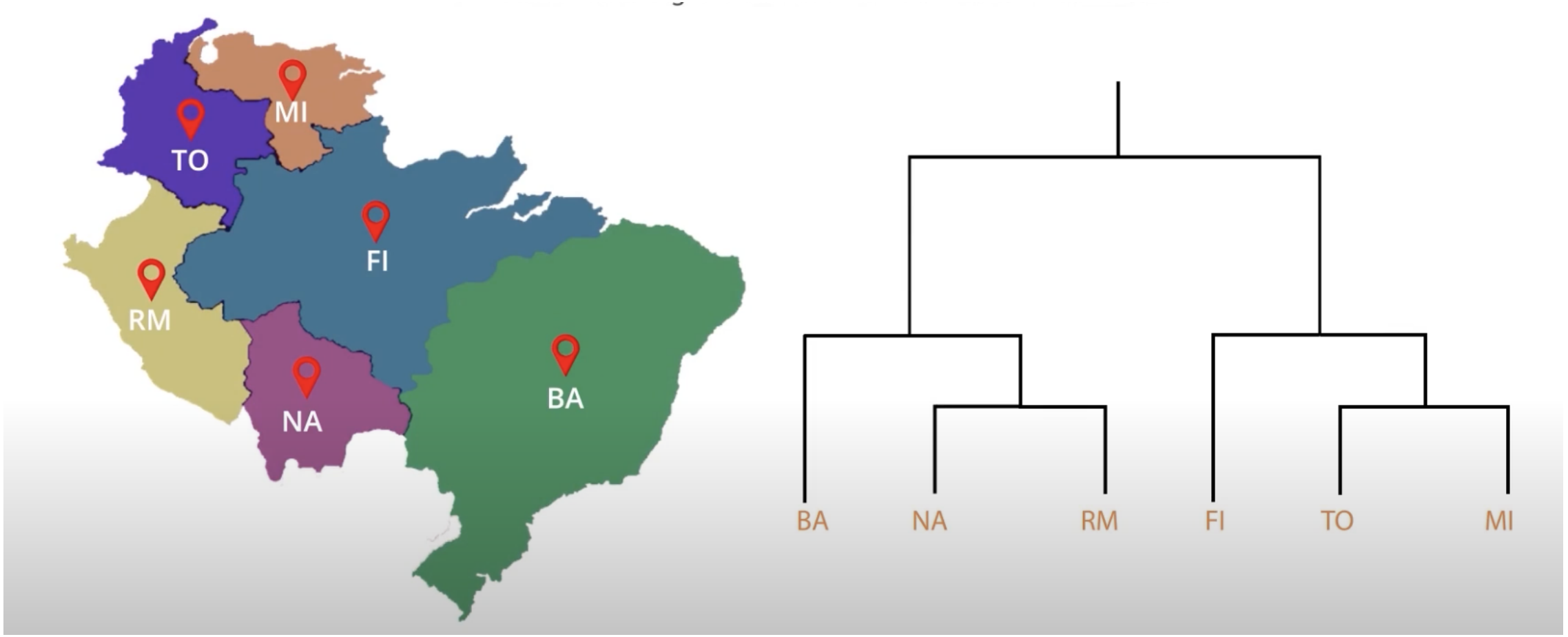


Dendrogram

A tree diagram used to illustrate the clusters produced by hierarchical clustering



A hierarchical clustering of disntances between cities in km



MI column has the lower values than TO column

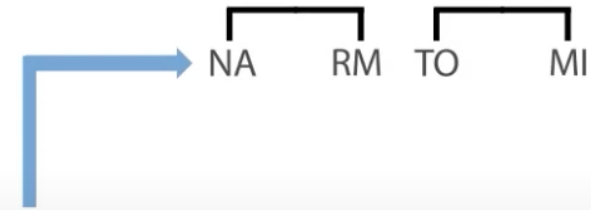
	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

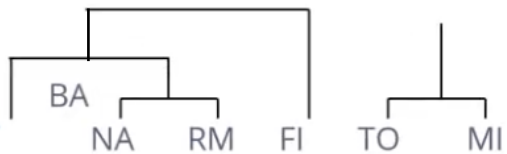


	BA	FI	MI/TO	NA/RM
BA	0	662	877	255
FI	662	0	295	268
MI/TO	877	295	0	564
NA/RM	255	268	564	0

	BA	FI	MI/TO	NA/RM
BA	0	662	877	255
FI	662	0	295	268
MI/TO	877	295	0	564
NA/RM	255	268	564	0

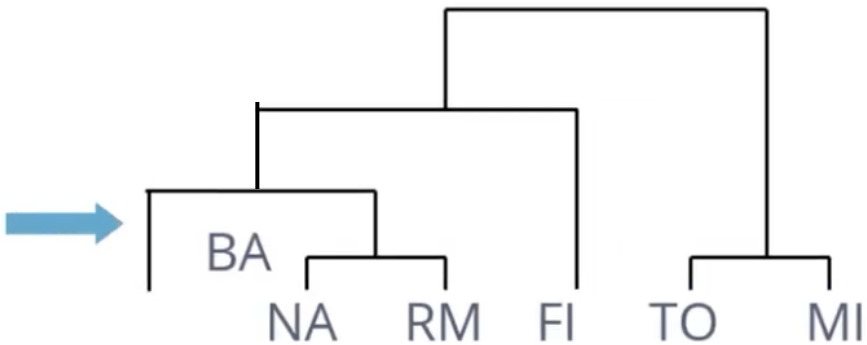
	BA/(NA/RM)	FI	MI/TO
BA/(NA/RM)	0	268	564
FI	268	0	295
MI/TO	564	295	0

	BA/(NA/RM)	FI	MI/TO
BA/(NA/RM)	0	268	564
FI	268	0	295
MI/TO	564	295	0

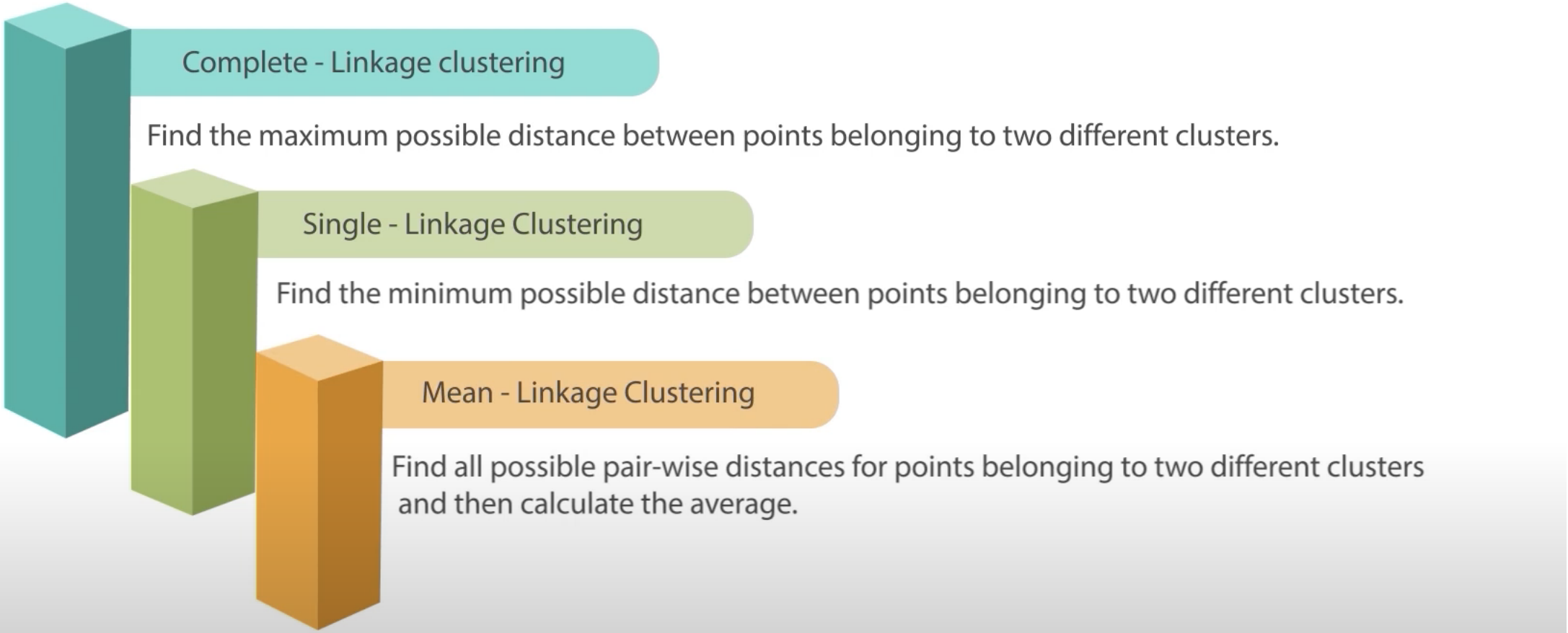


	BA/(NA/RM)/FI	(MI/TO)
BA/(NA/RM)/FI	0	295
(MI/TO)	295	0

	BA/(NA/RM)/FI	(MI/TO)
BA/(NA/RM)/FI	0	295
(MI/TO)	295	0

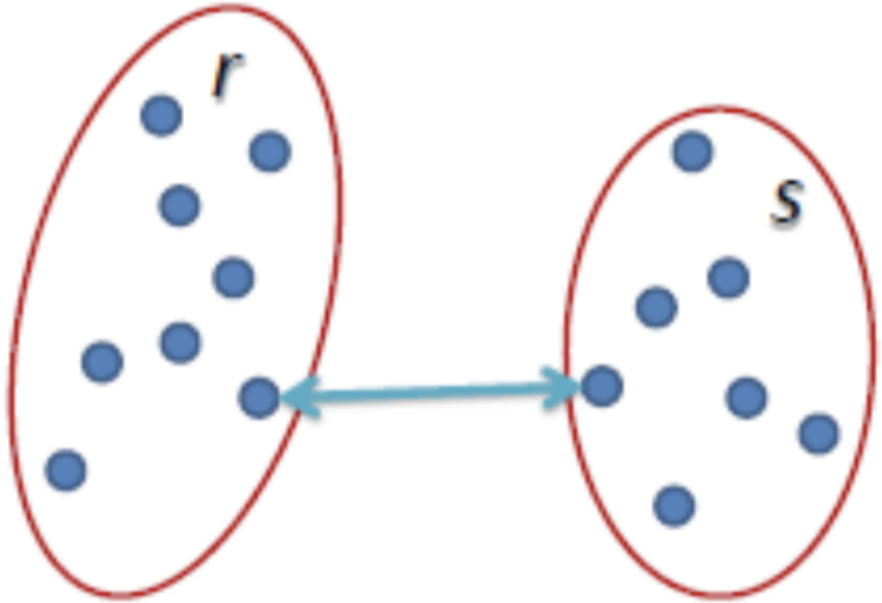


Distance Measures



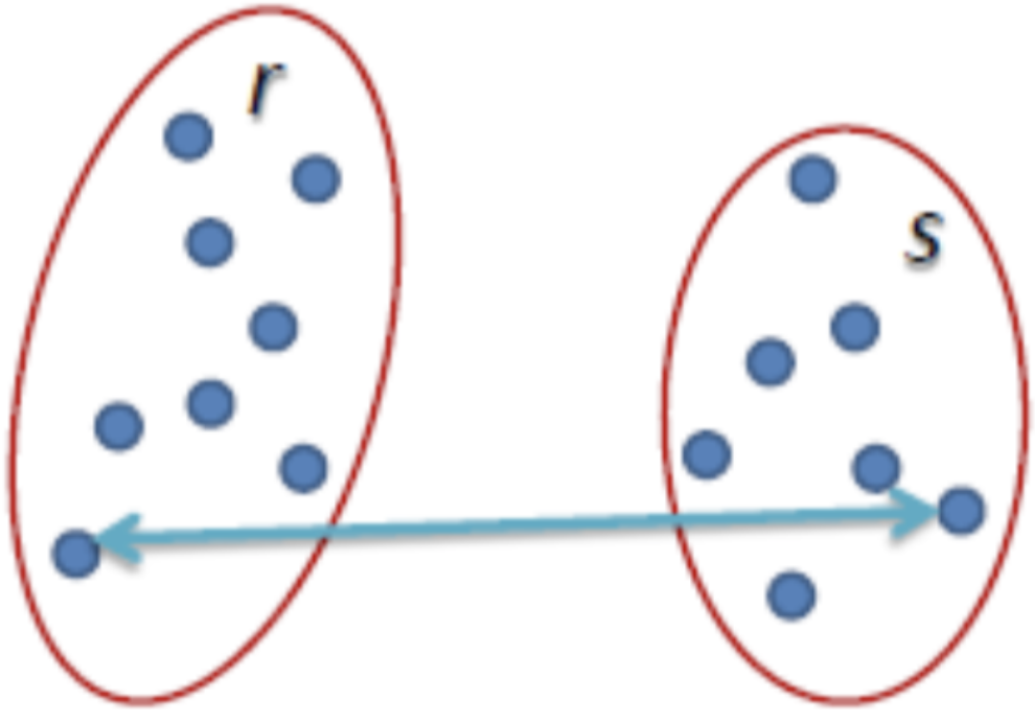
Distance between clusters

- **Single linkage**: The distance between two clusters is the shortest distance between two points in each cluster



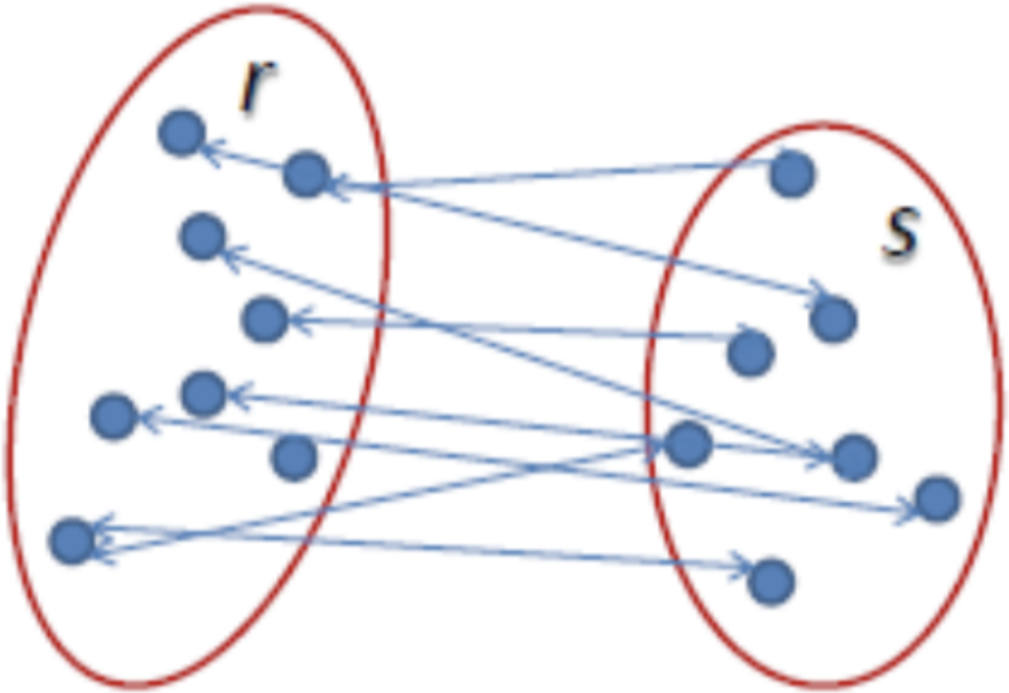
Distance between clusters

- **Complete Linkage**: The distance between two clusters is the longest distance between two points in each cluster

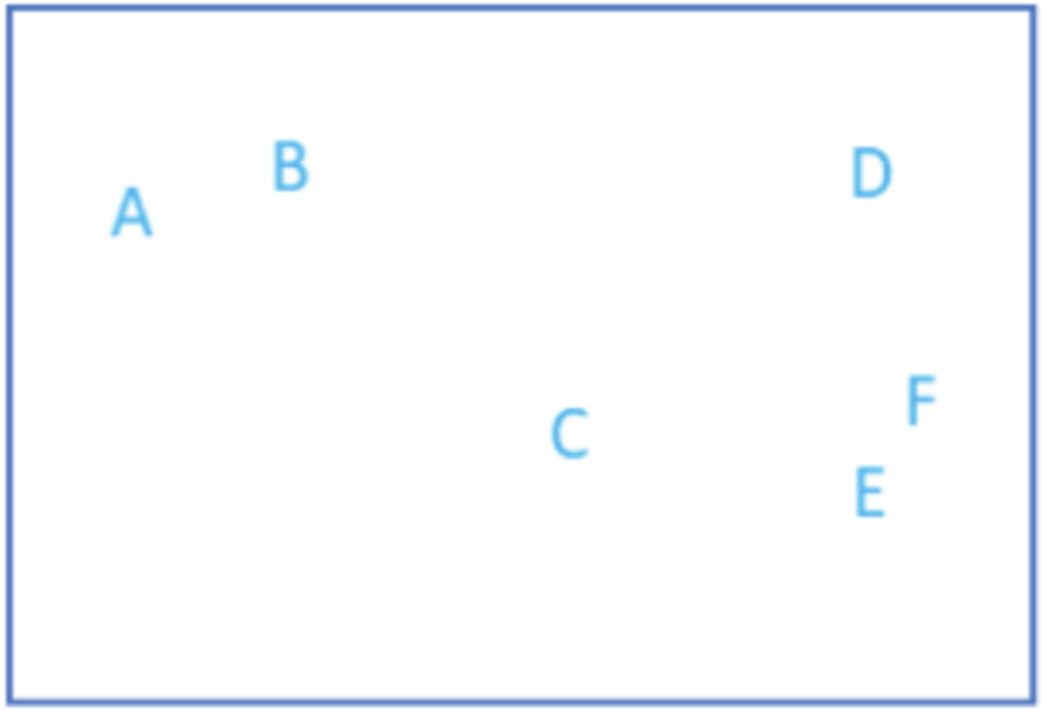


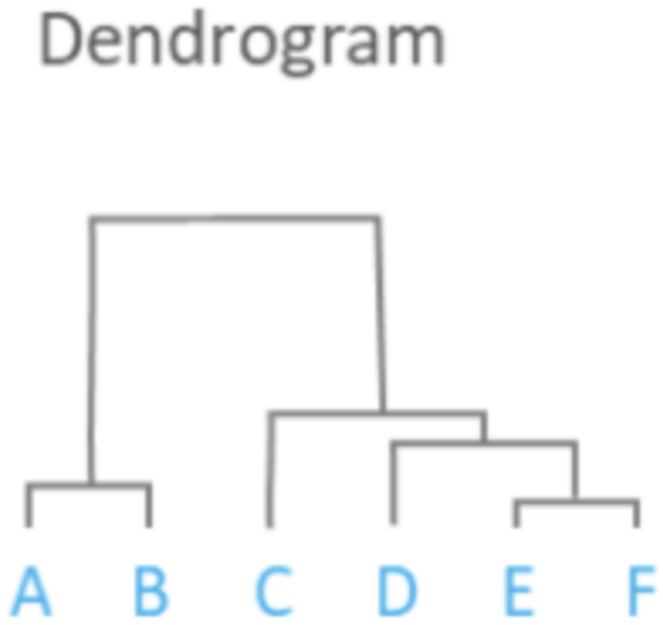
Distance between clusters

- **Average Linkage**: The distance between clusters is the average distance between each point in one cluster to every point in other cluster

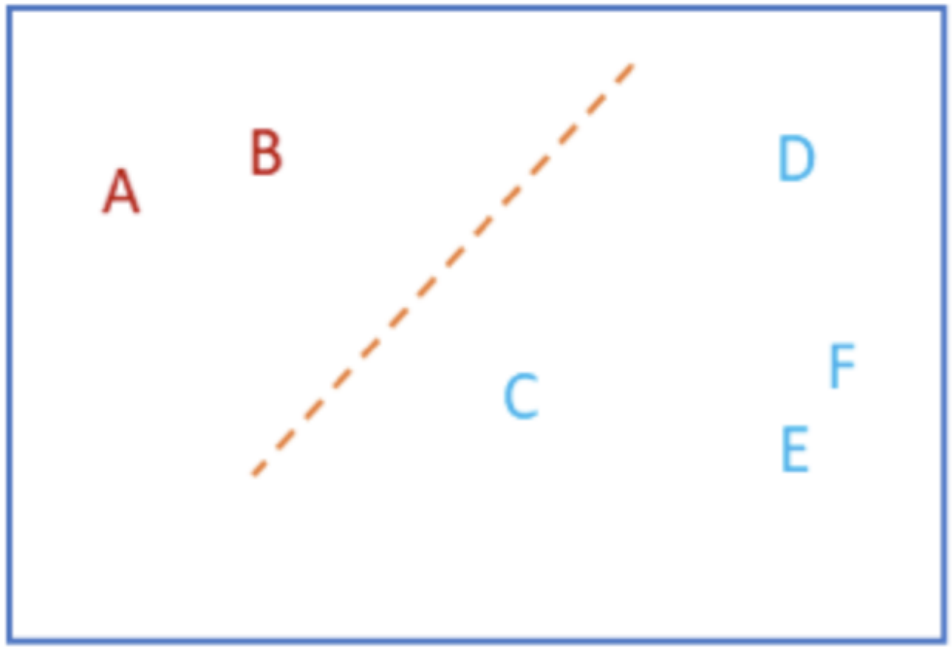


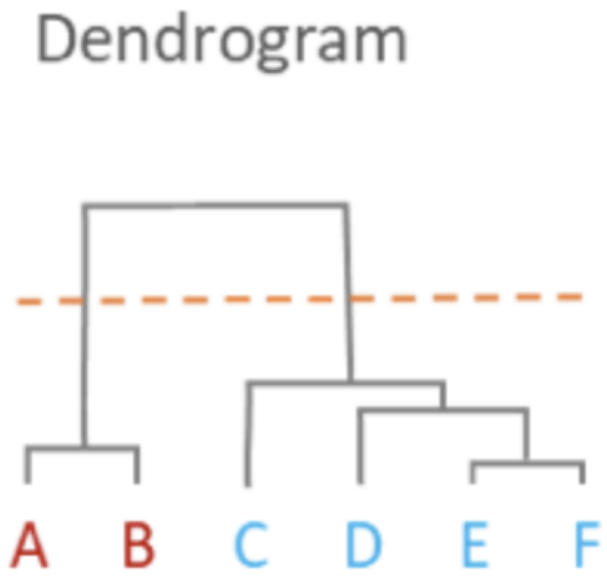
Dendrogram Analysis





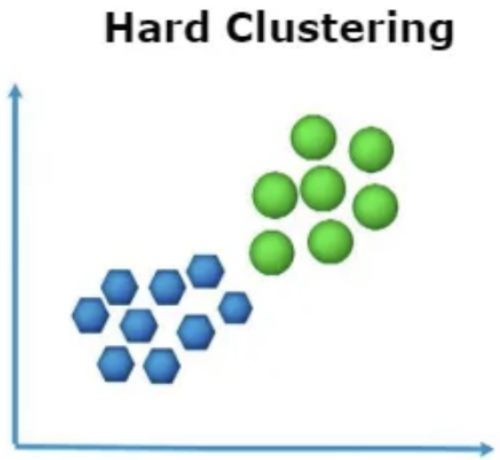
Dendrogram Analysis

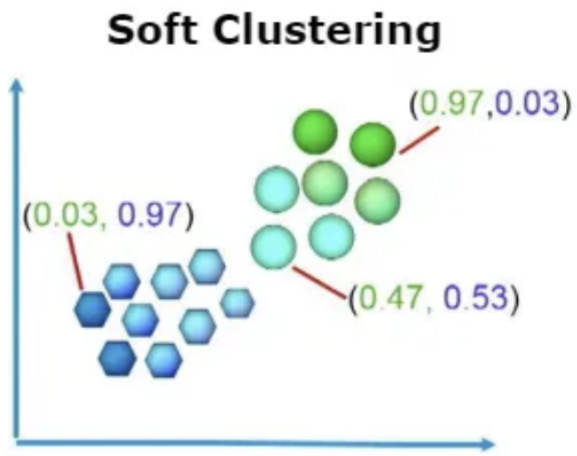




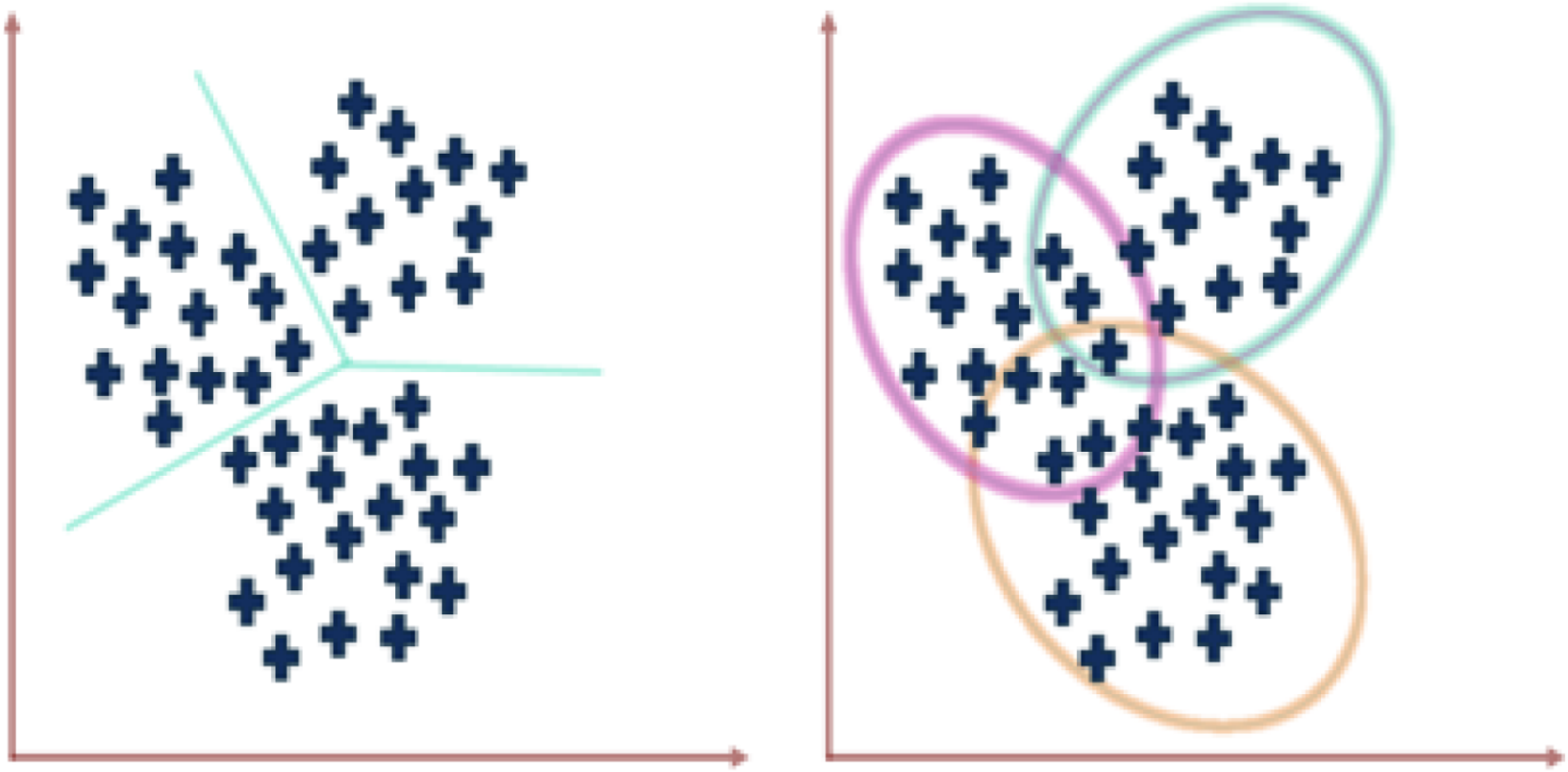
Fuzzy k-means

- Hard Clustering: K-Means
- Soft Clustering: Fuzzy K-Means (Fuzzy C-Means: FCM):
  - Useful when data boundaries are ambiguous





Fuzzy k-means



Fuzzy k-means

1. Set k and m > 1 (fuzziness)
2. Init cluster centroids (eg. random)
3. Compute membership values:

$$\mu_{ij} = \left(\sum_{k=1}^{c} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}} \right)^{-1} \qquad \mu_{ij} \in [0,1], \sum_{i=1}^{c} \mu_{ij} = 1$$

- μ_{ij} membership value for example i to cluster j
- x_i example i
- v_j centroid of cluster j
- c number of clusters
- m fuzziness (m>1)

Fuzzy k-means

4. Update centroids

$$v_j = \frac{\sum_{i=1}^{n} (\mu_{ij})^m \cdot x_i}{\sum_{i=1}^{n} (\mu_{ij})^m}$$

4. Repeat for step 3 until convergence (clusters centroids and membership values does not change significantly)

Fuzzy k-means

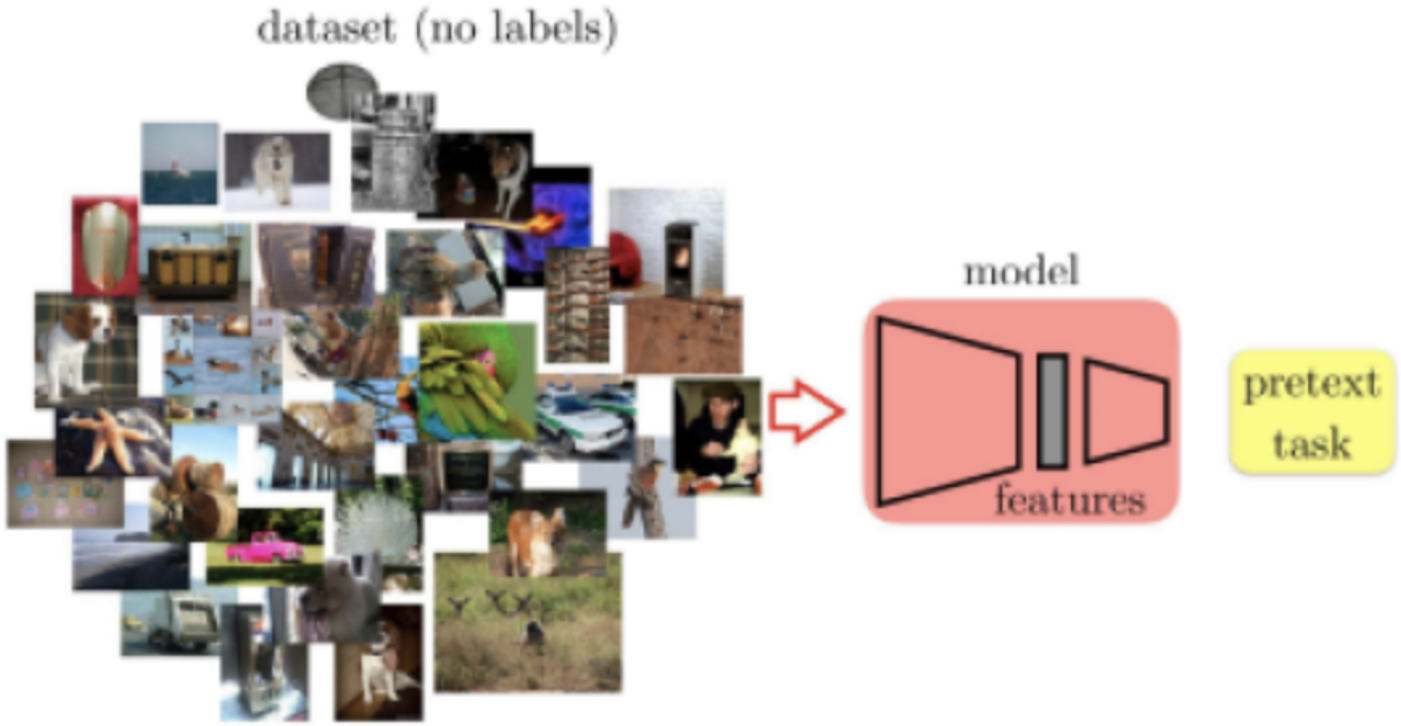
- **m low**: stronger membership to one cluster / fewer clusters per point
- **m high**: softer membership across multiple clusters

- Advantages
  - Handles overlapping data naturally.
  - More flexible than K-Means.
  - Better for datasets where boundaries between clusters are not well defined, eg. medical domain – segmentation of regions with ambiguous boundaries between areas."
- Disadvantages:
  - Computationally more expensive than K-Means.
  - Sensitive to initial conditions and parameter m.

Self-supervised learning

"Learn by comparing.": Instead of asking "What is this?" like in classification, contrastive learning asks: "Is this like that?"

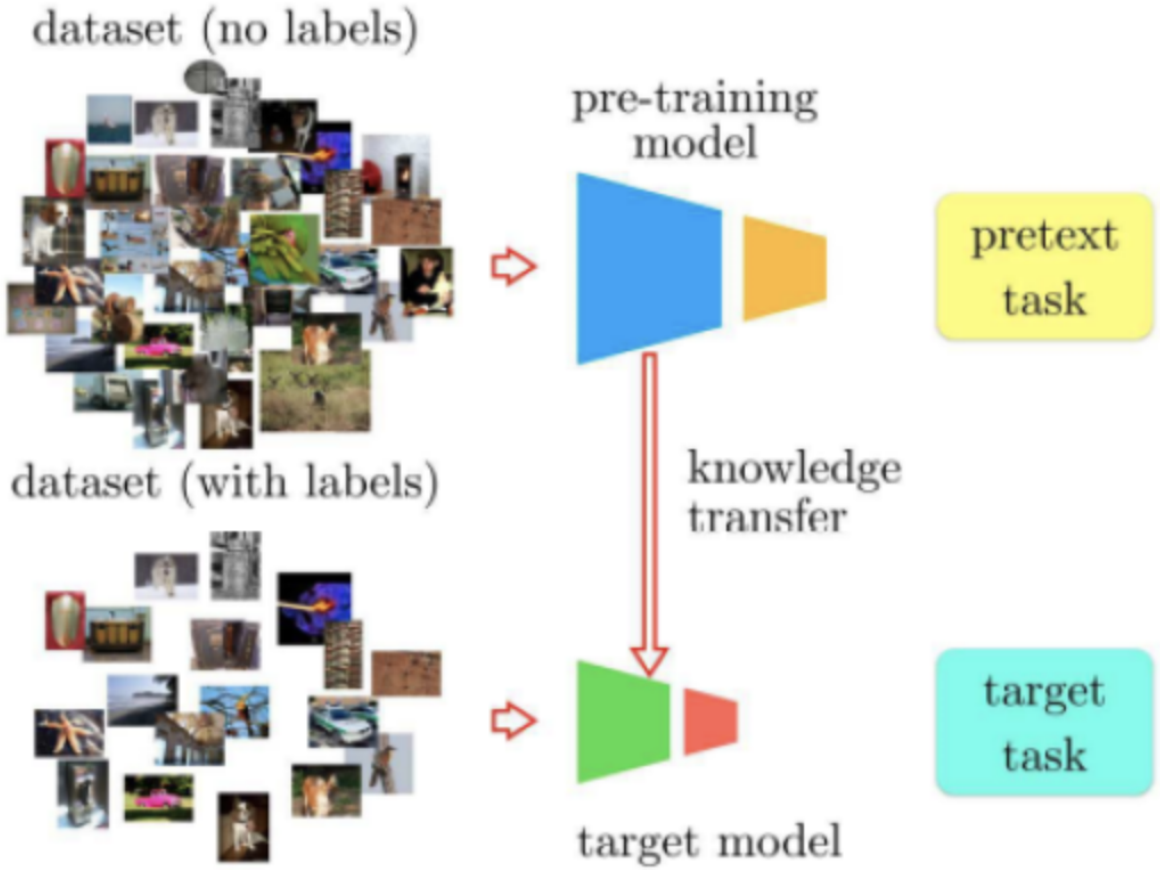
Pretext task: guide the model to learn intermediate representations of data.



https://neptune.ai/blog/self-supervised-learning

Self-supervised learning

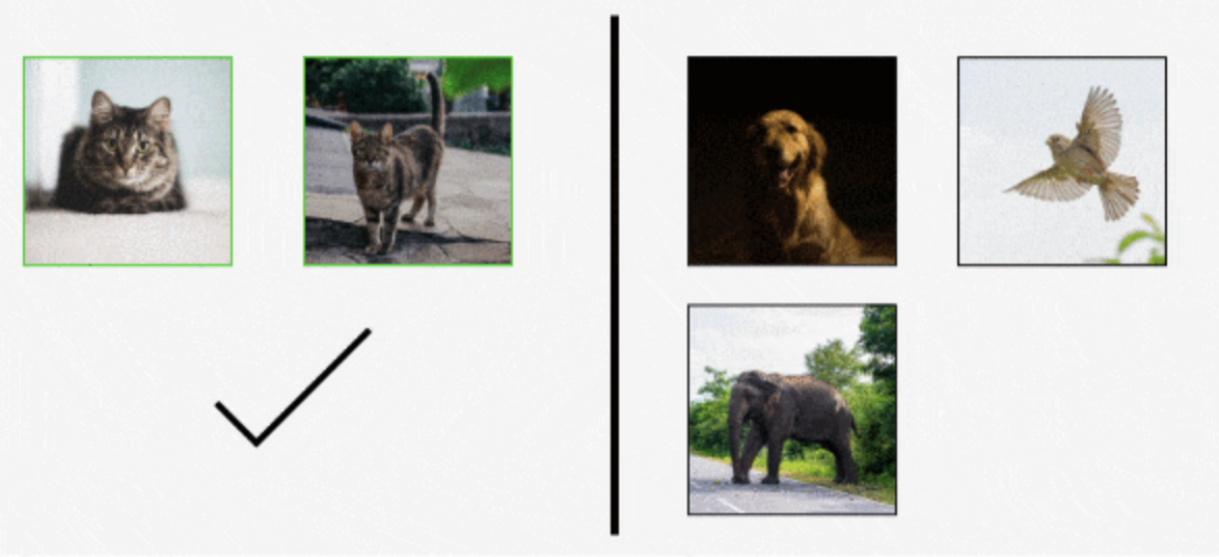
Downstream tasks (target tasks) - object recognition, object classification, object reidentification, etc., finetuned on the pretext model.



https://neptune.ai/blog/self-supervised-learning

Contrastive Learning

- An instance of **self-supervised learning**
- Does not require **labels** of the data, but a means to know whether **pairs of instances from a dataset** are **similar** or **dissimilar**



https://arxiv.org/pdf/2002.05709

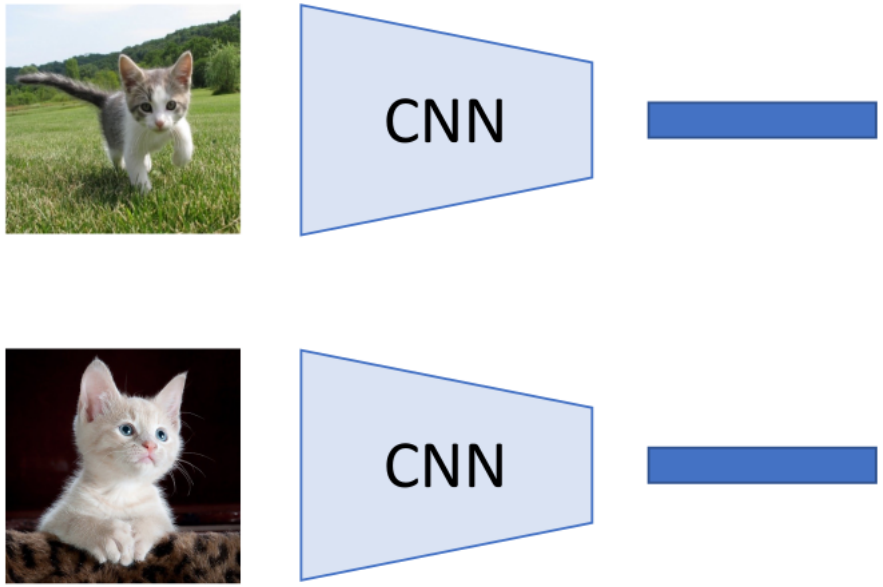
Contrastive Learning

- Learning similar / different images

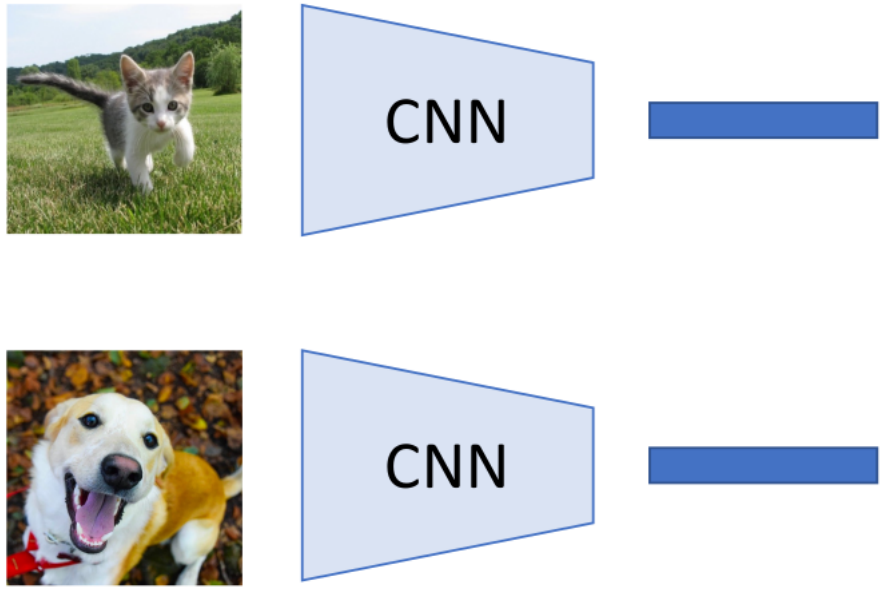
Contrastive Learning

- An instance of **self-supervised learning**
- Does not require **labels** of the data, but a means to know whether **pairs of instances from a dataset** are **similar** or **dissimilar**

Similar images should have similar features



Dissimilar images should have dissimilar features



Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006

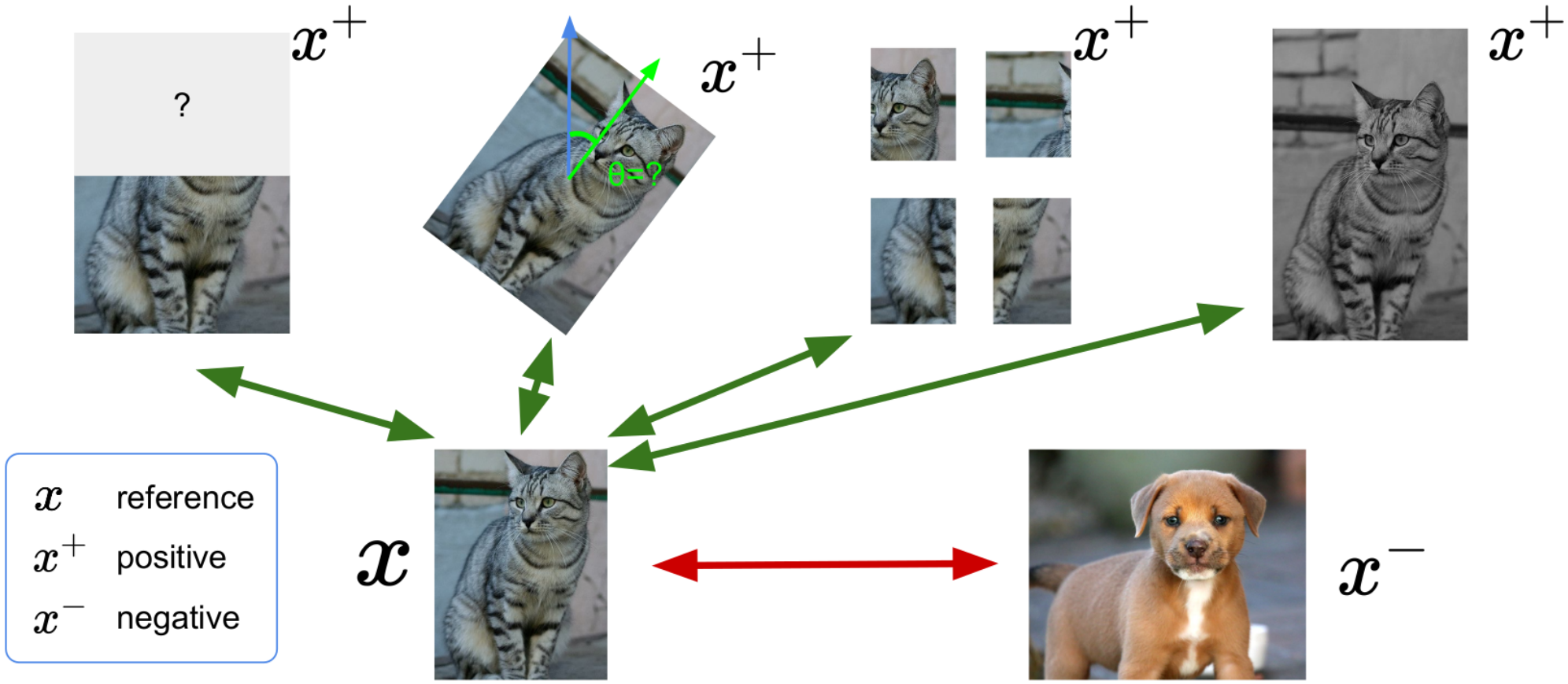
White kitten image is free for commercial use under the Pixabay license

Contrastive Learning with Data Augmentation

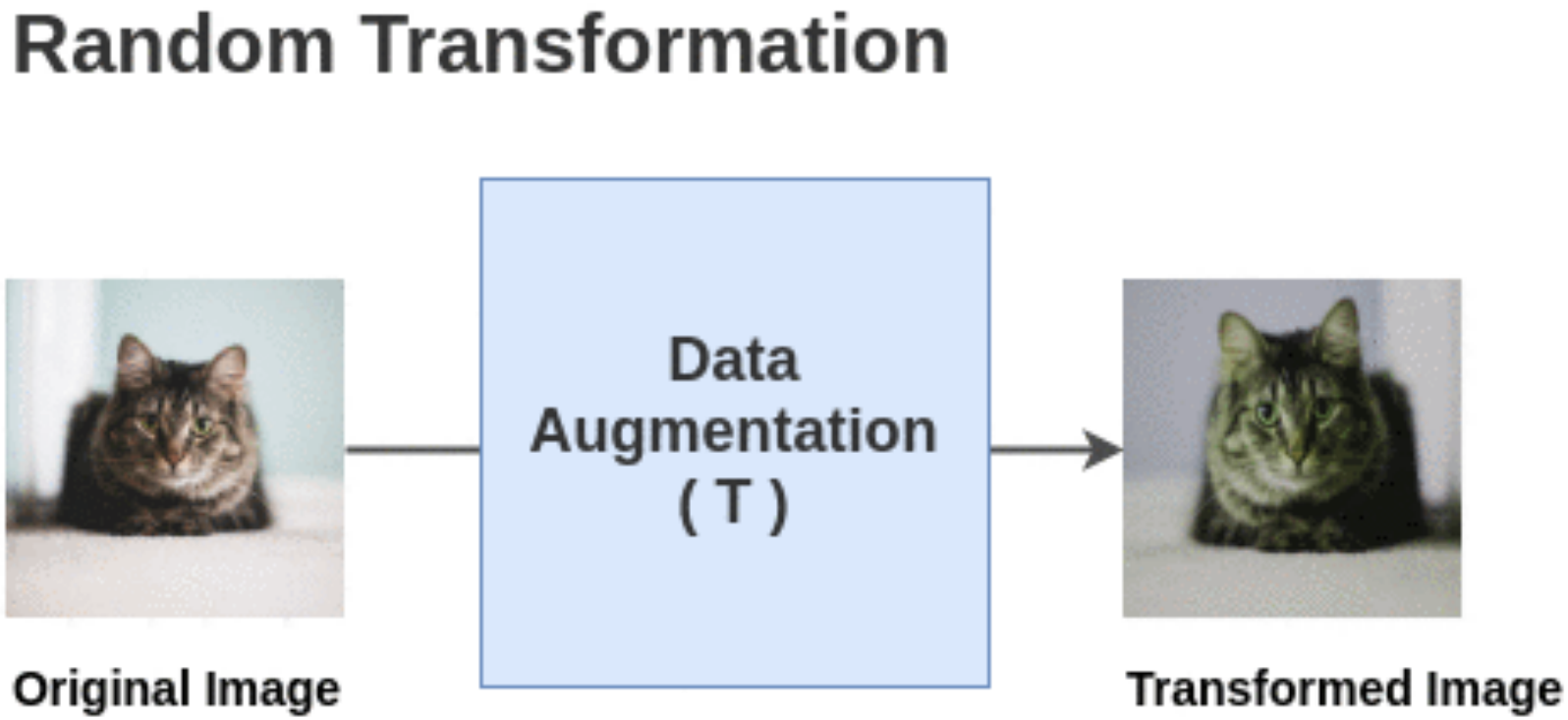
- An instance of **self-supervised learning**
- Does not require **labels** of the data, but a means to know whether **pairs of instances from a dataset** are **similar** or **dissimilar**
- **Problem**: how do we get pairs of **similar** and dissimilar images?
- **Answer:** generate **augmentations** of **data samples** that and bring their representations close together in a **latent representation space,** while distancing them from **many other** (negative) **samples**
  - **Remember!** Augmentations will be **application** and **data type dependent!**
  - **Which augmentations to choose for images?**

Contrastive Learning with Data Augmentation

- **Which augmentations to choose for images?**

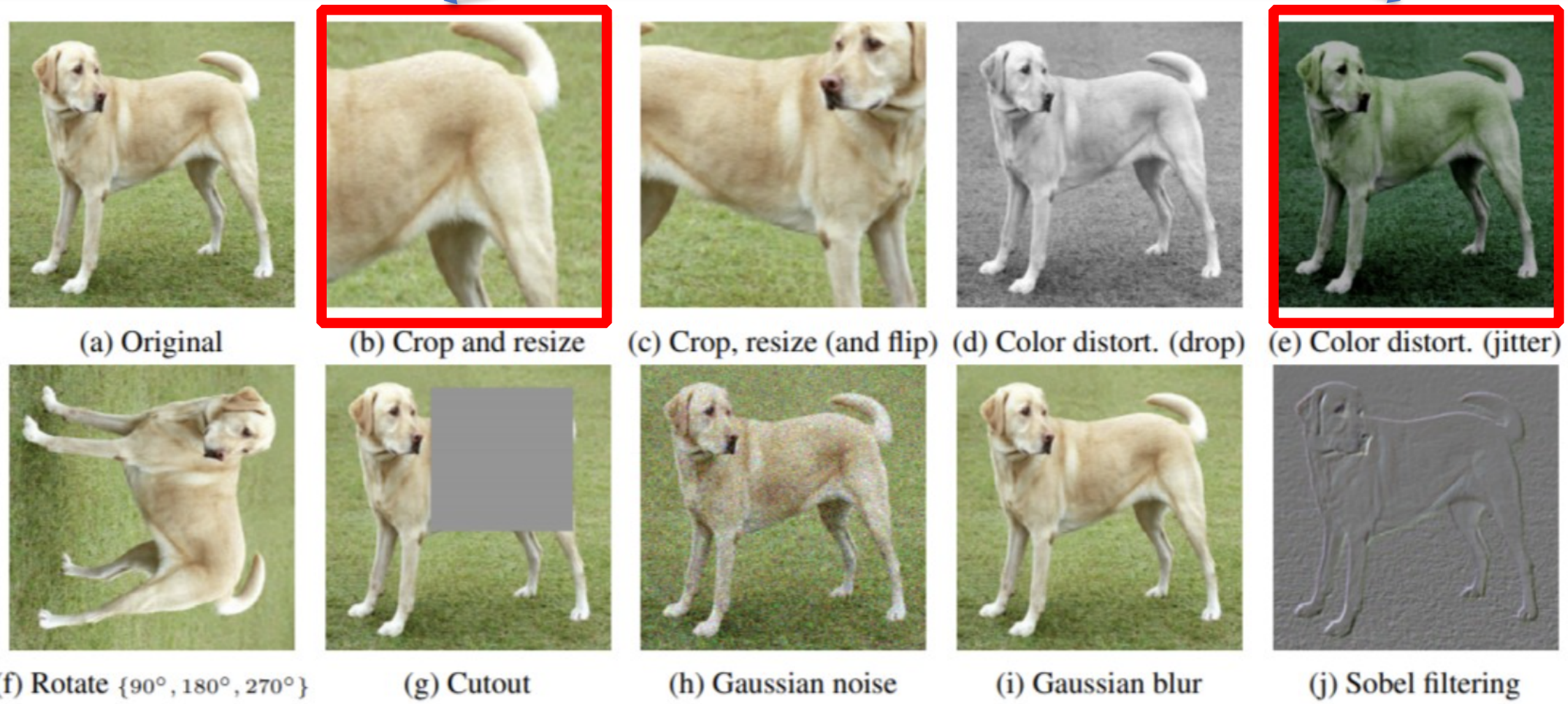


Data augmentation



Data Augmentation

used

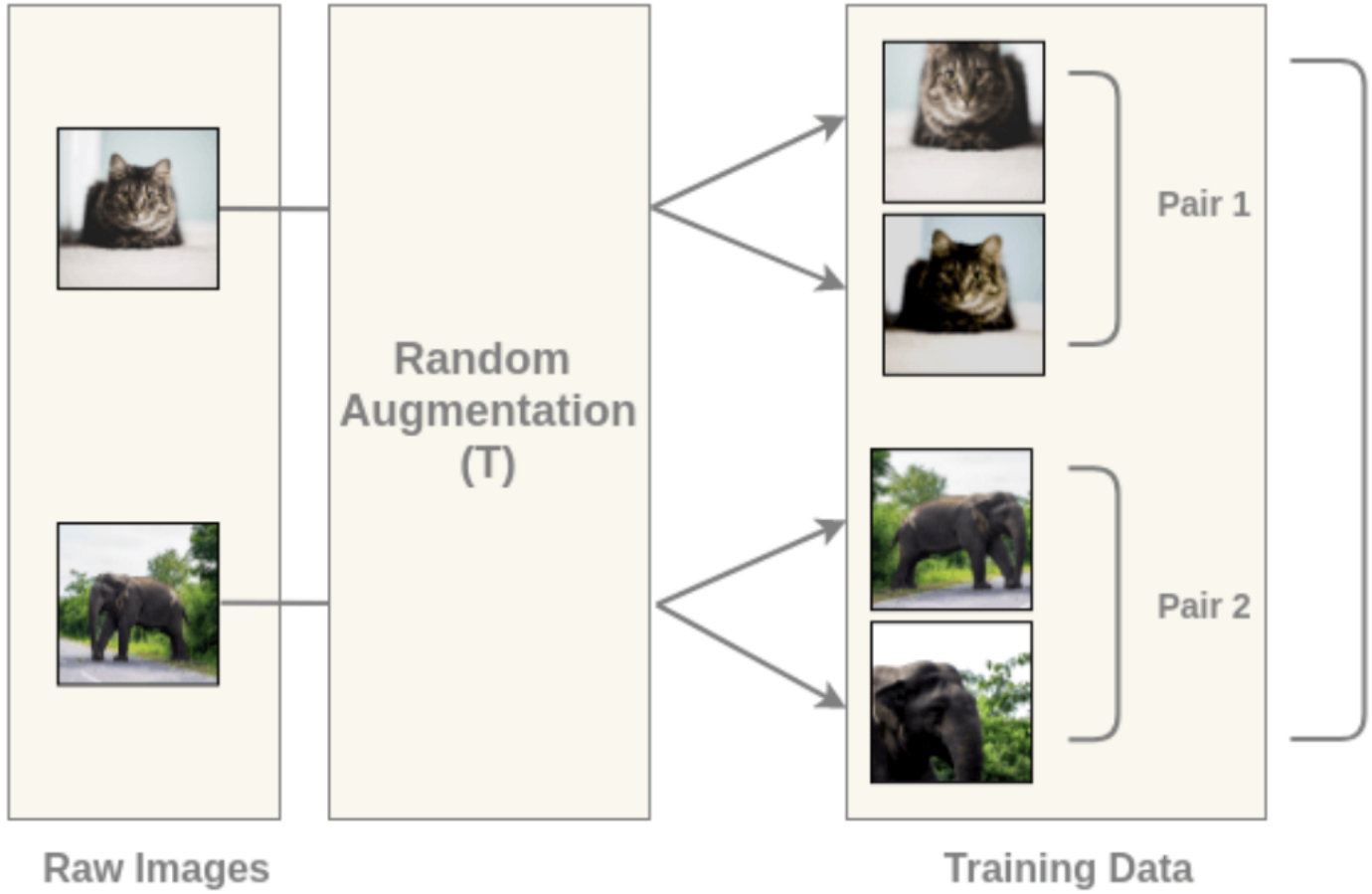


(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Data Augmentation

N=2



2*N images