ITSAR ISTITUTO TECNICO SUPERIORE
ANGELO RIZZOLI

Algoritmi per il Machine Learning

Ing Andrea Colleoni

Clustering

In machine learning too, we often group examples as a first step to understand a subject (data set) in a machine learning system. Grouping unlabeled examples is called clustering.

Clustering has a myriad of uses in a variety of industries. Some common applications for clustering include the following:

- market segmentation
- social network analysis
- search result grouping
- medical imaging
- image segmentation
- anomaly detection

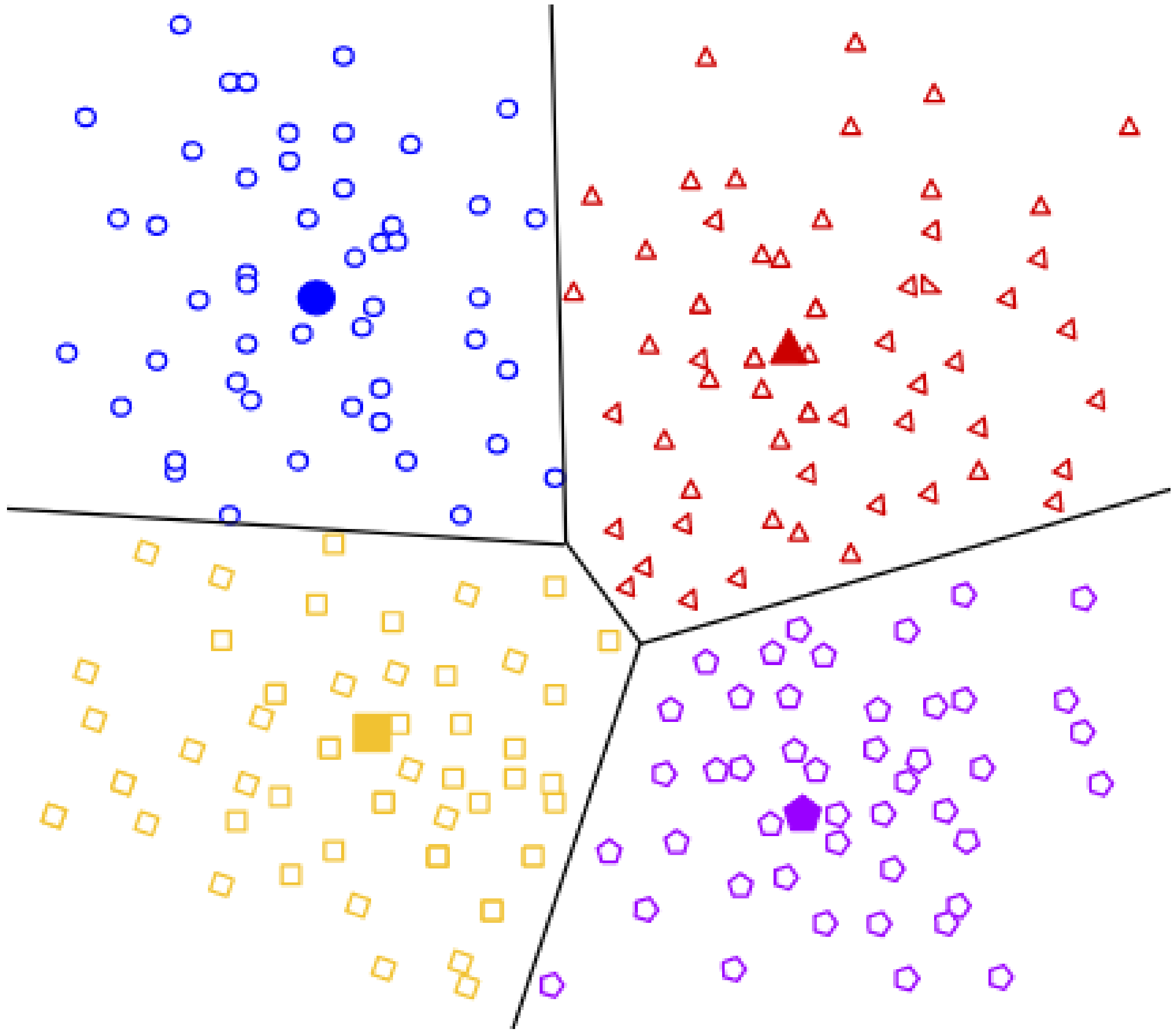
Utilizzo in Google

- Generalization
  When some examples in a cluster have missing feature data, you can infer the missing data from other examples in the cluster.
- Data Compression
  Feature data for all examples in a cluster can be replaced by the relevant cluster ID. This replacement simplifies the feature data and saves storage.
- Privacy Preservation
  You can preserve privacy by clustering users, and associating user data with cluster IDs instead of specific users.

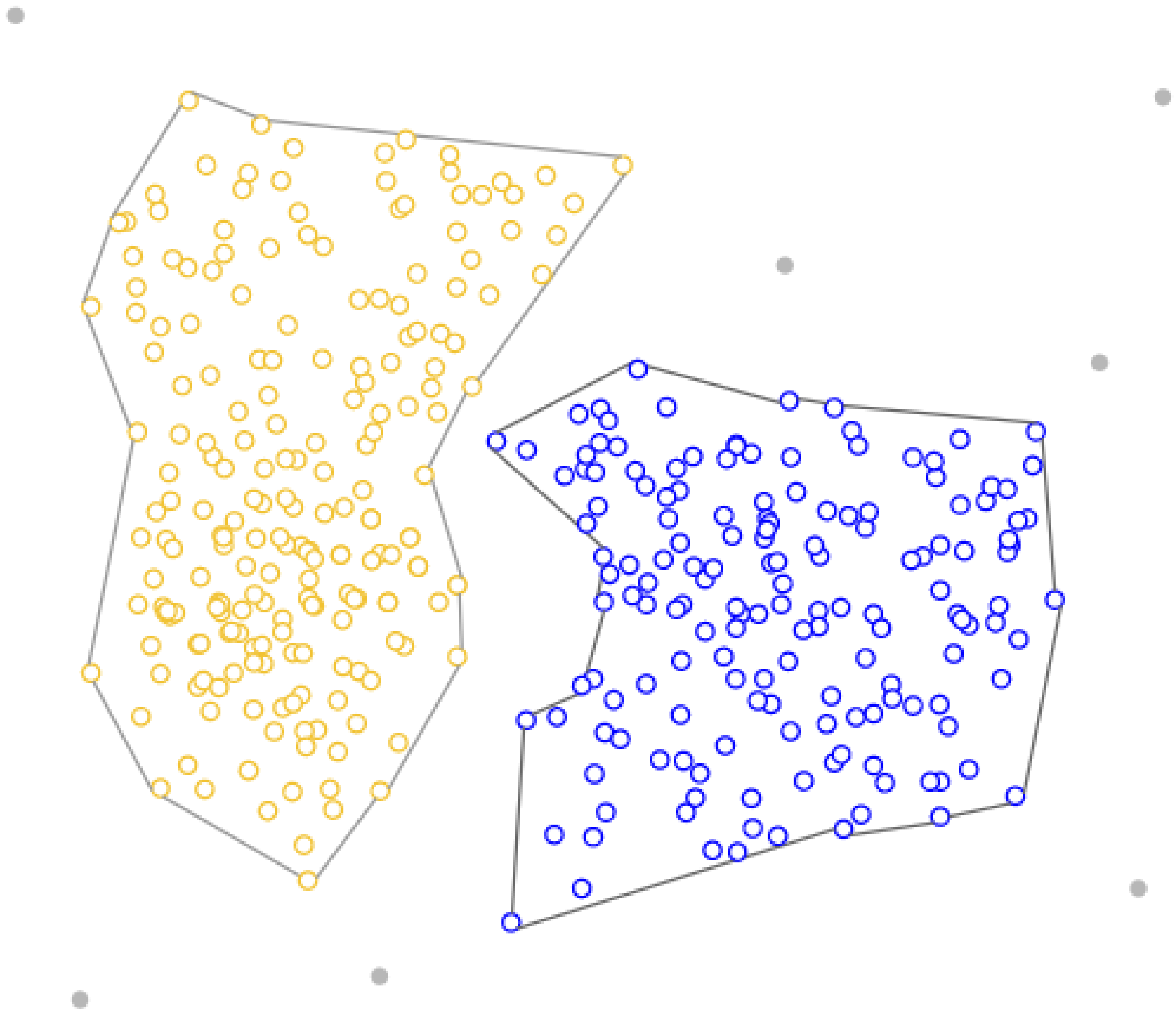
Tipi di clustering

- Centroid-based clustering
  - k-means is the most widely-used centroid-based clustering algorithm.
  - Centroid-based algorithms are efficient but sensitive to initial conditions and outliers.
- Density-based Clustering
  - connects areas of high example density into clusters; This allows for arbitrary-shaped distributions as long as dense areas can be connected
  - These algorithms have difficulty with data of varying densities and high dimensions. Further, by design, these algorithms do not assign outliers to clusters
- Distribution-based Clustering
  - This clustering approach assumes data is composed of distributions, suc h as Gaussian distributions
- Hierarchical Clustering
  - Hierarchical clustering creates a tree of clusters.
  - Hierarchical clustering, not surprisingly, is well suited to hierarchical dat a, such as taxonomies.

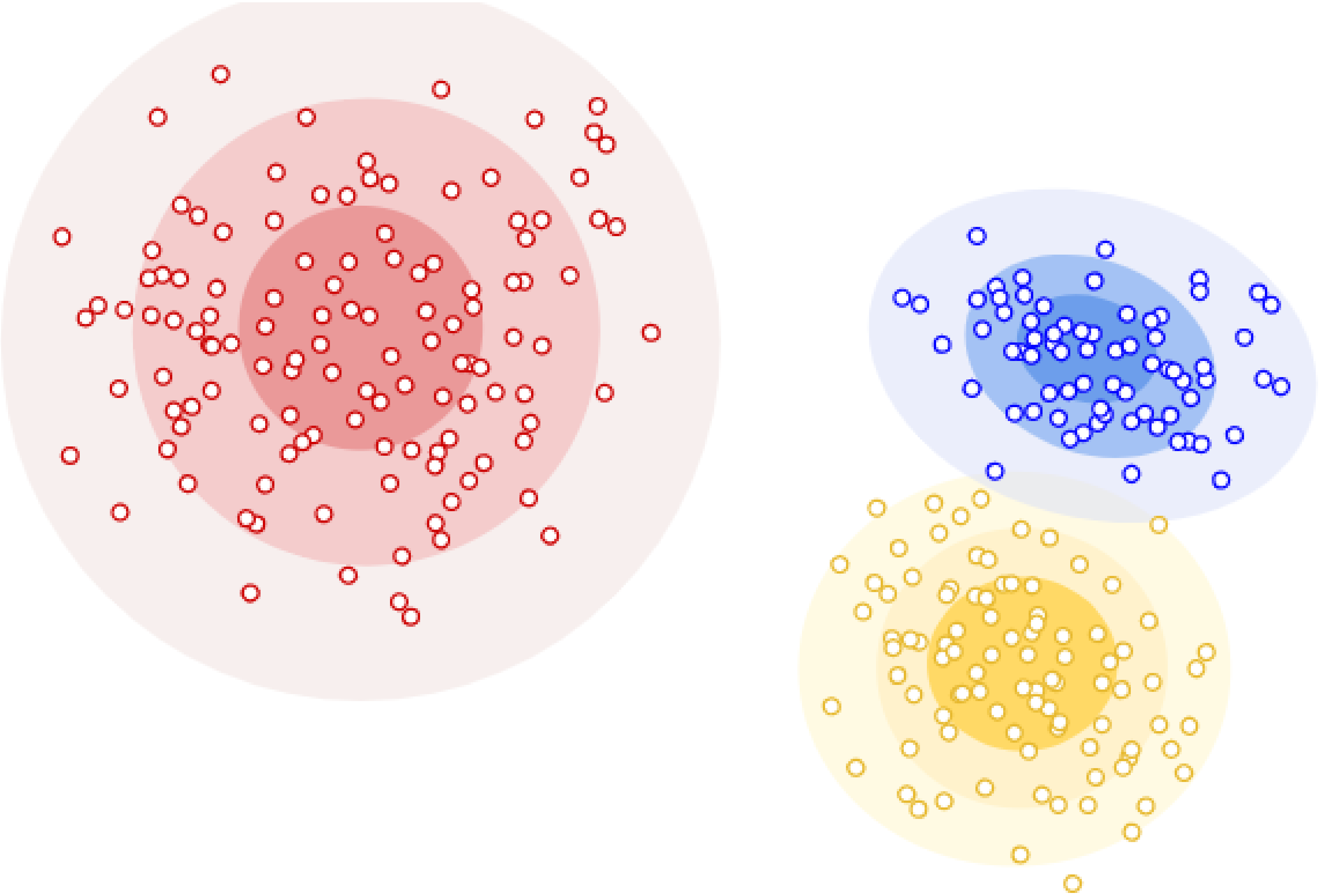
Centroid-based clustering



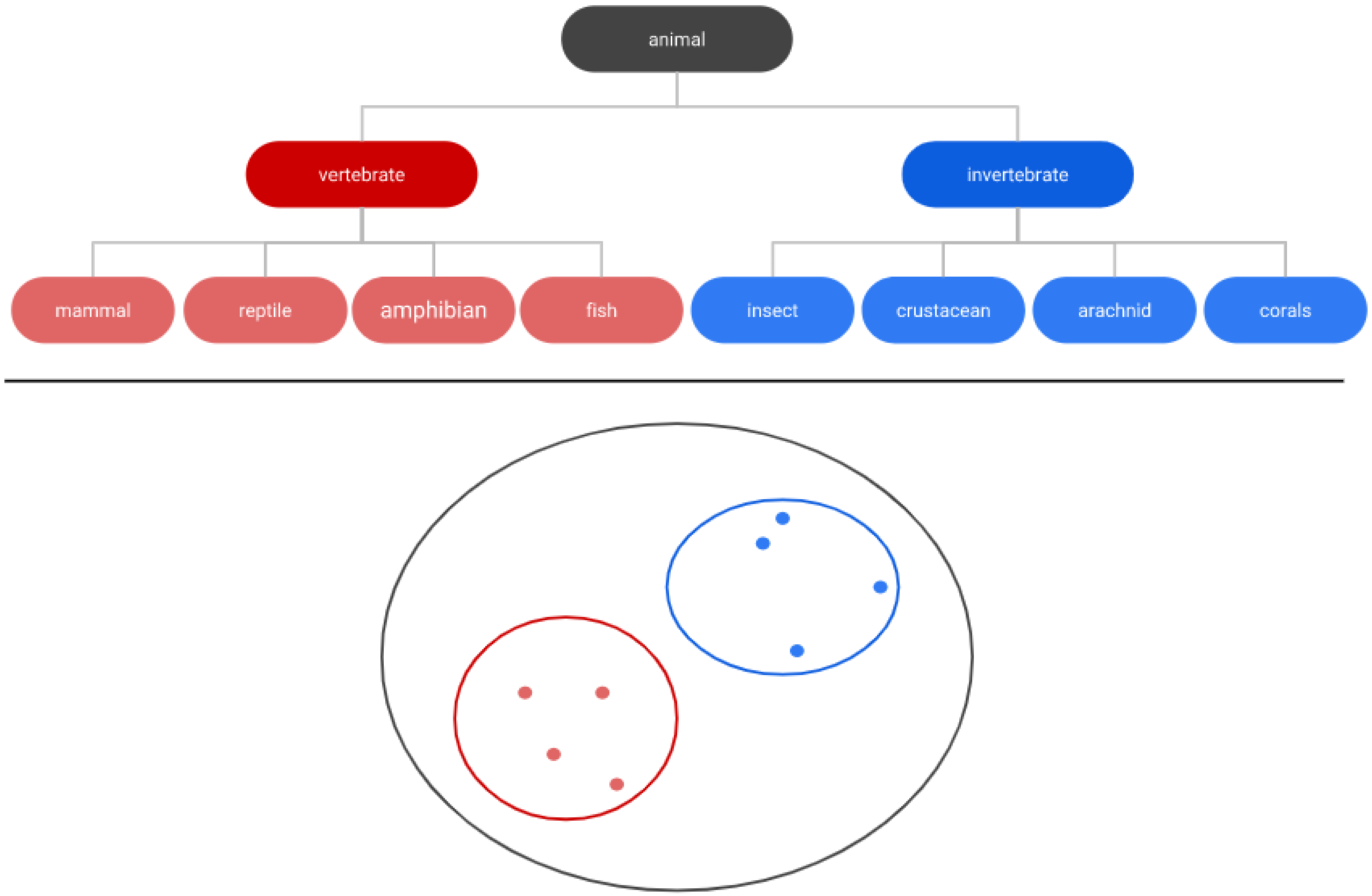
- # Density-based Clustering



Distribution-based clustering

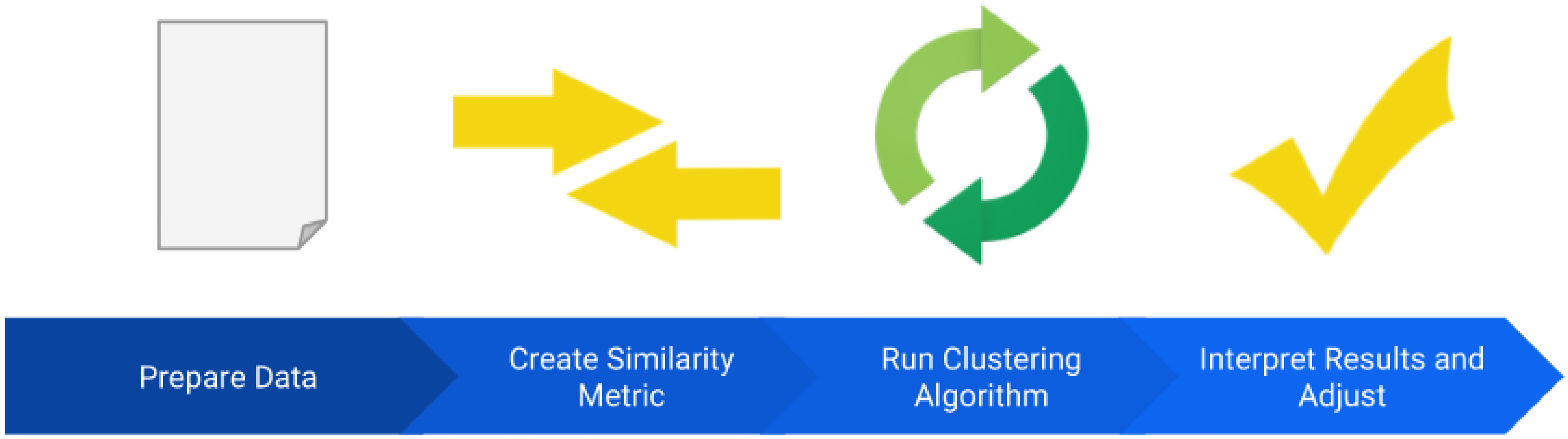


Hyerarchical-clustering



Organizzazione di un algoritmo di clustering

1. Preparazione dei dati
2. Creazione di metriche di similitudine
3. Esecuzione dell'algoritmo
4. Interpretazione dei risultati e adattamento dell'algoritmo

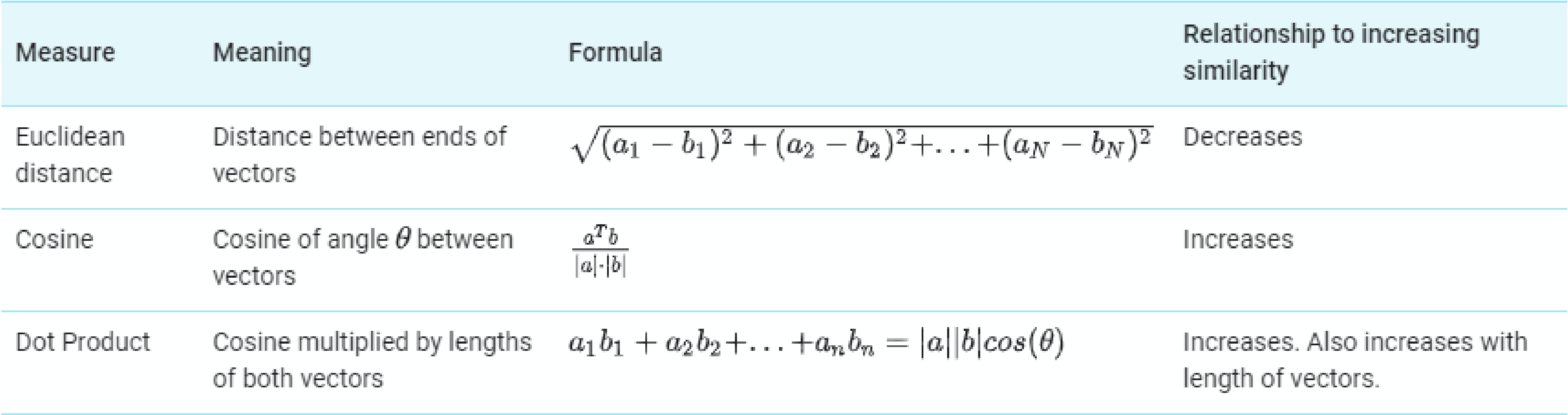


Preparazione dei dati

- Normalizzazione, Utilizzo dei quantili, Sistemazione dei dati mancanti, ecc.
- In generale le tecniche di preparazione dei dati viste lo scorso anno

Creazione di metriche di similitudine

- Combine all the feature data for those two examples into a single numeric value
- Prepare numerical data



Measure	Meaning	Formula	Relationship to increasing similarity
Euclidean distance	Distance between ends of vectors	$\sqrt{(a_1-b_1)^2+(a_2-b_2)^2+\ldots+(a_N-b_N)^2}$	Decreases
Cosine	Cosine of angle θ between vectors	$\frac{a^Tb}{\lvert a\rvert\cdot\lvert b\rvert}$	Increases
Dot Product	Cosine multiplied by lengths of both vectors	$a_1b_1+a_2b_2+\ldots+a_nb_n=\lvert a\rvert\lvert b\rvert cos(\theta)$	Increases. Also increases with length of vectors.

Esecuzione dell’algoritmo

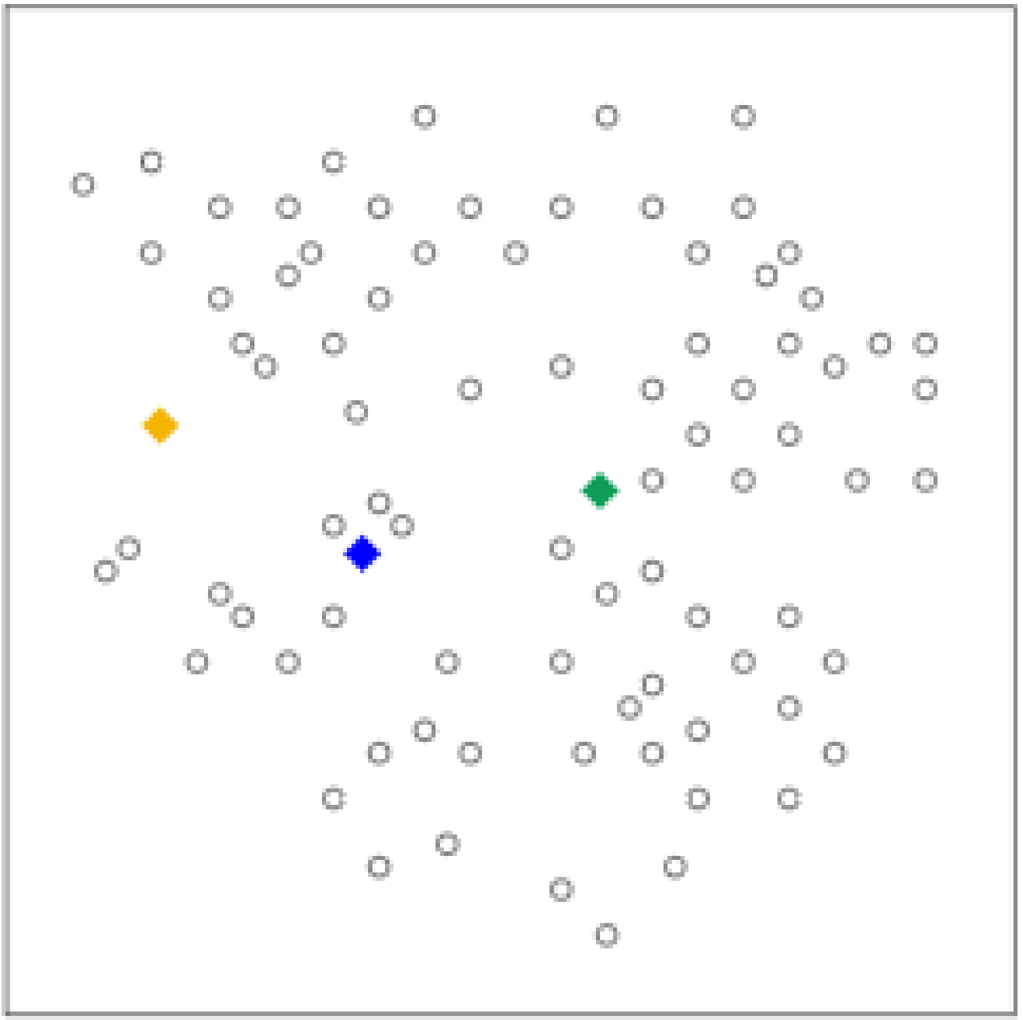
- In machine learning, you sometimes encounter datasets that can have millions of examples.
- ML algorithms must scale efficiently to these large datasets.
- However, many clustering algorithms do not scale because they need to compute the similarity between all pairs of points.
- This means their runtimes increase as the square of the number of points, denoted as $O(n^2)$.

K-means

- k-means scales as O(nk), where k is the number of clusters. k-means groups points into k clusters by minimizing the distances between points and their cluster’s centroid.
  The **centroid** of a cluster is the mean of all the points in the cluster.
- Before running k-means, you must choose the number of clusters, k. Initially, start with a guess for k.

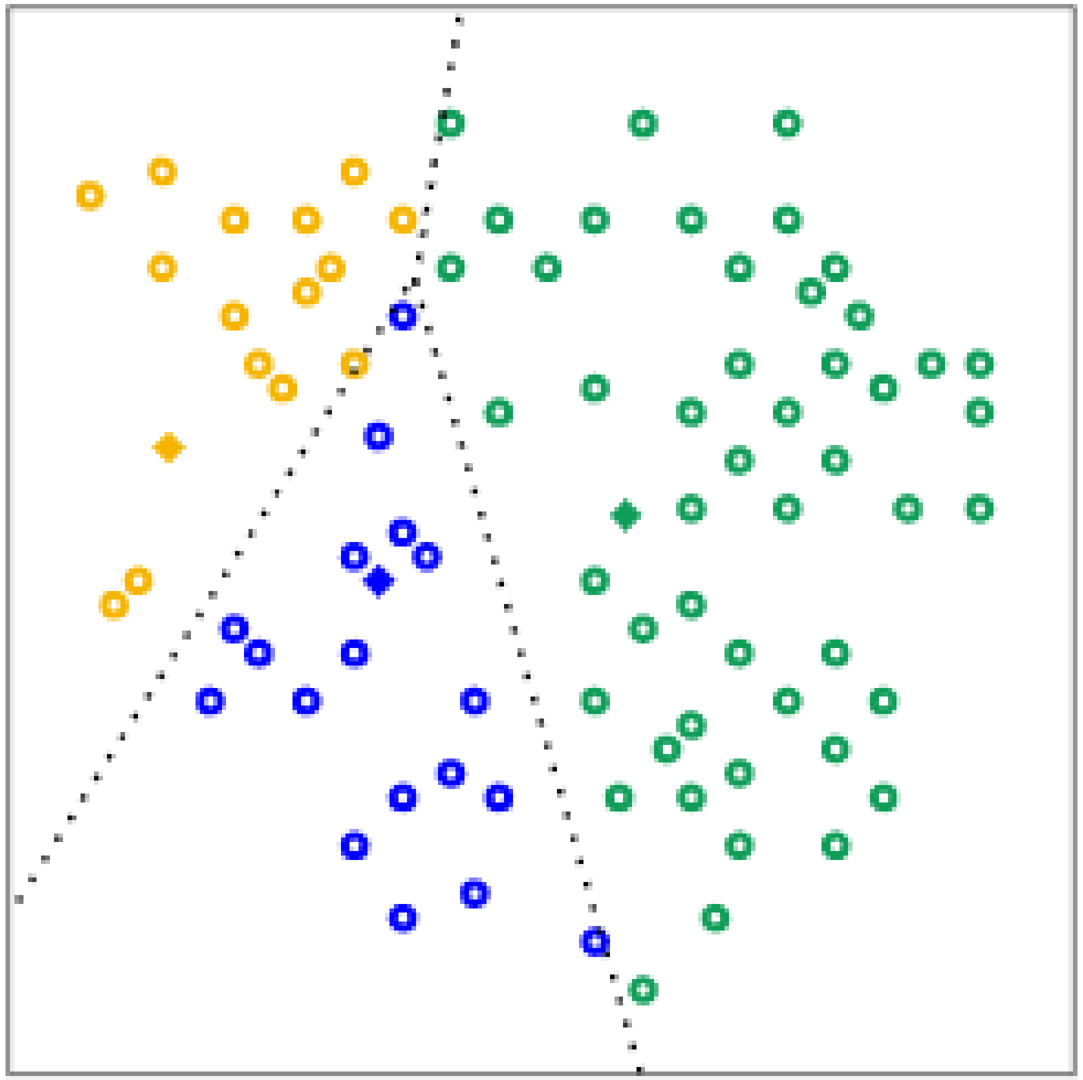
Step 1

The algorithm randomly chooses a centroid for each cluster. In our example, we choose a k of 3, and therefore the algorithm randomly picks 3 centroids.



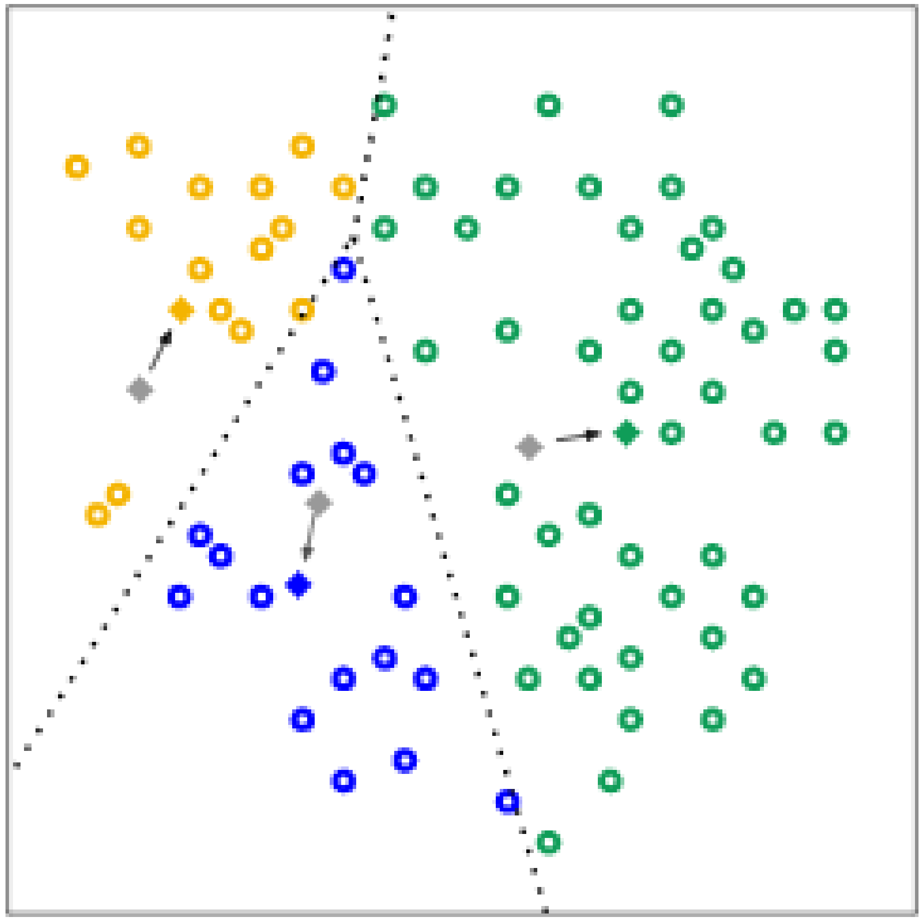
Step 2

Using the chosen similarity measure, the algorithm assigns each point to the closest centroid to get k initial clusters.



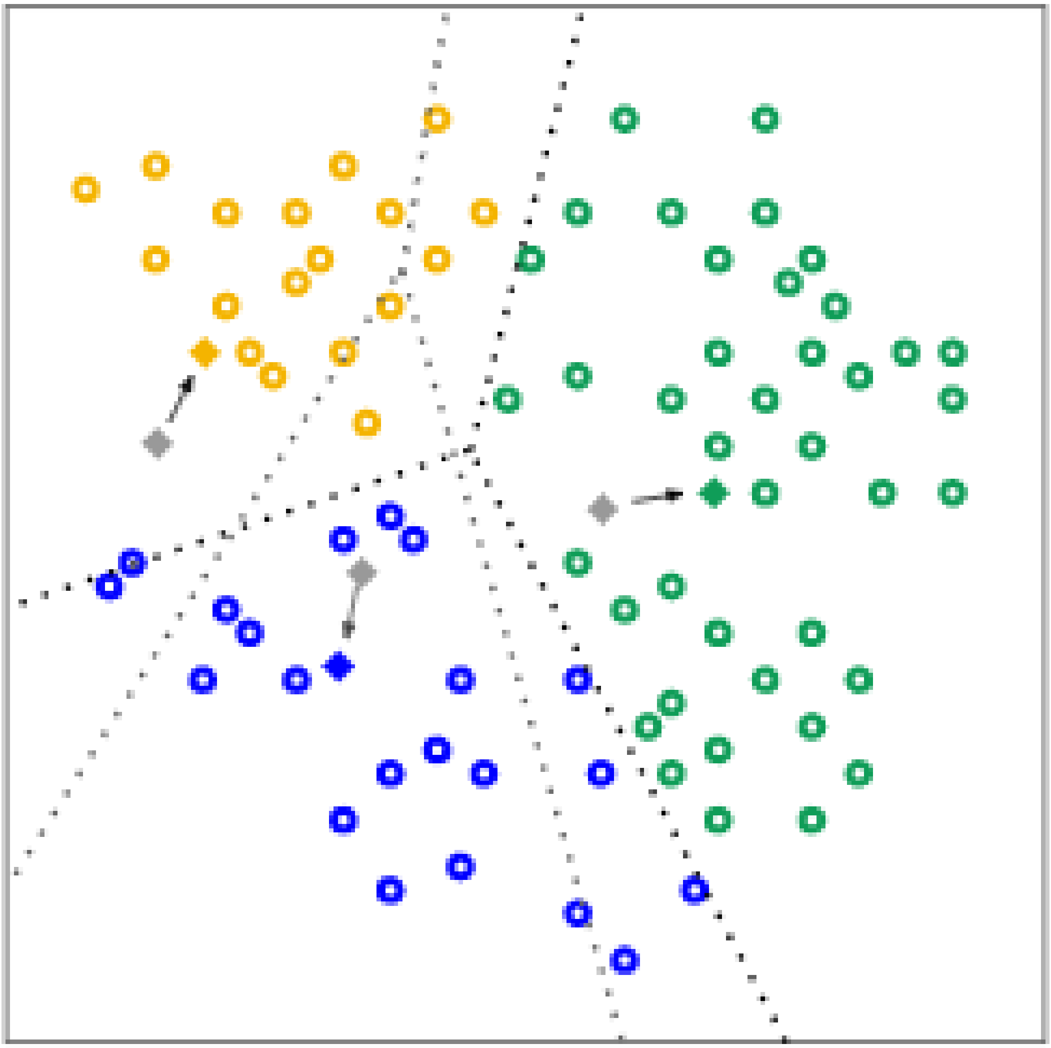
Step 3

For every cluster, the algorithm recomputes the centroid by taking the average of all points in the cluster. The changes in centroids are shown in Figure 3 by arrows. Since the centroids change, the algorithm then re-assigns the points to the closest centroid.

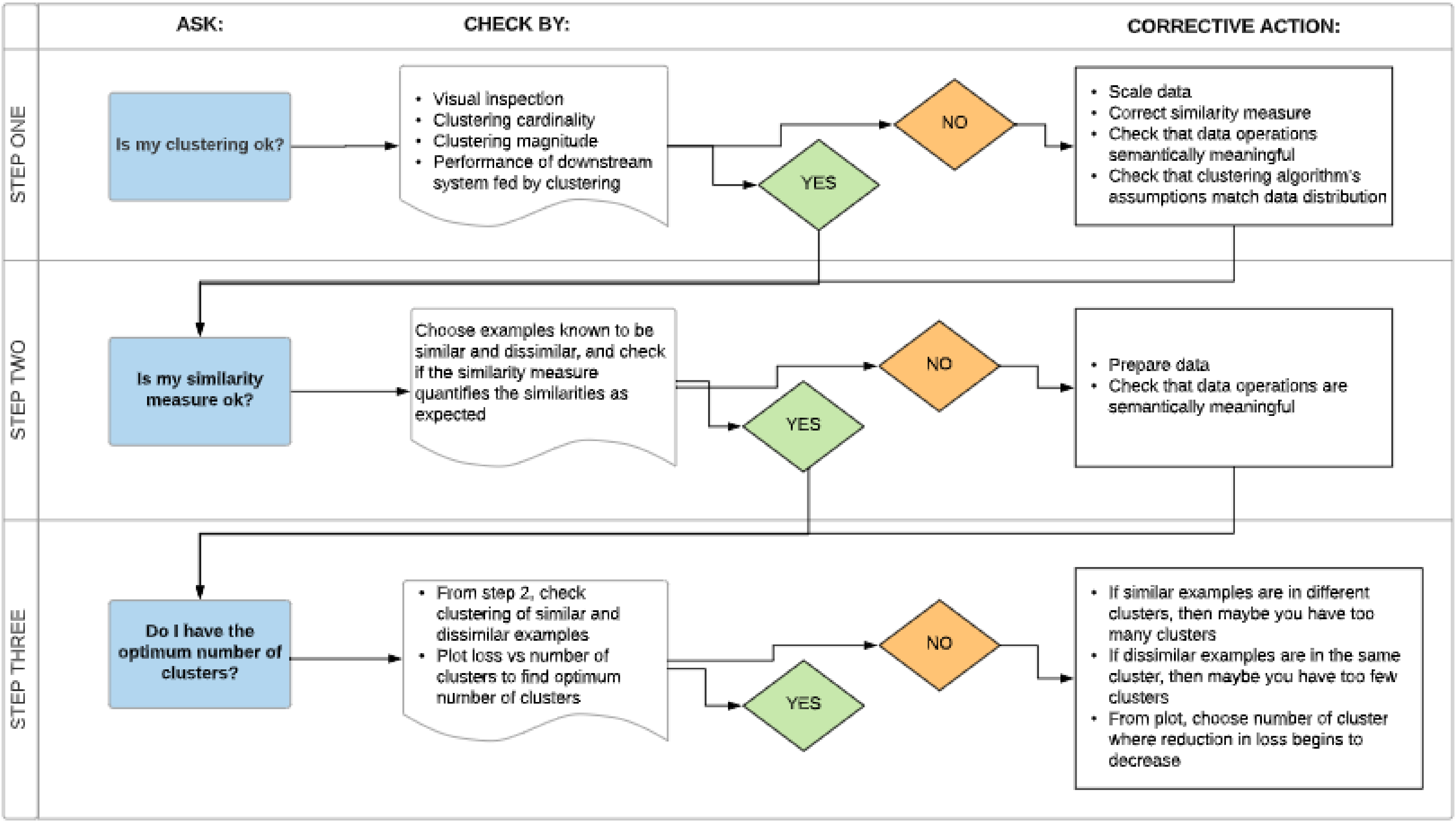


Step 4

The algorithm repeats the calculation of centroids and assignment of points until points stop changing clusters. When clustering large datasets, you stop the algorithm before reaching convergence, using other criteria instead.



Interpretazione dei risultati



	ASK:	CHECK BY:	CORRECTIVE ACTION:
STEP ONE	Is my clustering ok?	• Visual inspection • Clustering cardinality • Clustering magnitude • Performance of downstream system fed by clustering	• Scale data • Correct similarity measure • Check that data operations semantically meaningful • Check that clustering algorithm's assumptions match data distribution
STEP TWO	Is my similarity measure ok?	Choose examples known to be similar and dissimilar, and check if the similarity measure quantifies the similarities as expected	• Prepare data • Check that data operations are semantically meaningful
STEP THREE	Do I have the optimum number of clusters?	• From step 2, check clustering of similar and dissimilar examples • Plot loss vs number of clusters to find optimum number of clusters	• If similar examples are in different clusters, then maybe you have too many clusters • If dissimilar examples are in the same cluster, then maybe you have too few clusters • From plot, choose number of cluster where reduction in loss begins to decrease