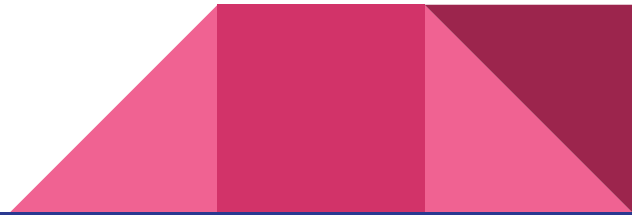
18. Clustering

Corso di Python per il Calcolo Scientifico

Angelo Cardellicchio
angelo.cardellicchio@stiima.cnr.it

Outline

- Il clustering
- Tipi di clustering
- Workflow del clustering
- L’algortimo k-means
- Valutazione della bontà del clustering
- DBSCAN
- Metriche di valutazione



Il clustering

- Prevede la suddivisione dei campioni nei dataset **senza che questi abbiano un'etichetta a priori**.
- Ha varie applicazioni, come ad esempio:
  - segmentazione del mercato;
  - individuazione di aree coerenti all'interno di un'immagine;
  - suddivisione delle stelle sulla base delle caratteristiche di magnitudine;
  - definizione delle feature mancanti in un dataset (anche supervisionato);
  - raggruppamento dei film proposti da Netflix.
- Ogni cluster è contraddistinto da un **identificativo**.
  - Può essere usato come ingresso ad un altro algoritmo di machine learning (magari supervisionato).

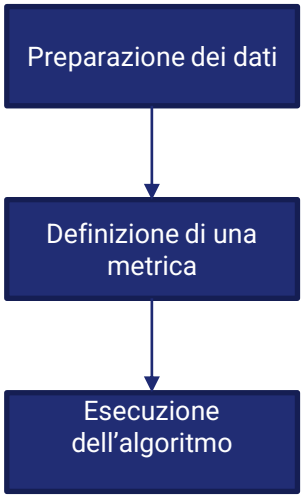
Tipi di clustering

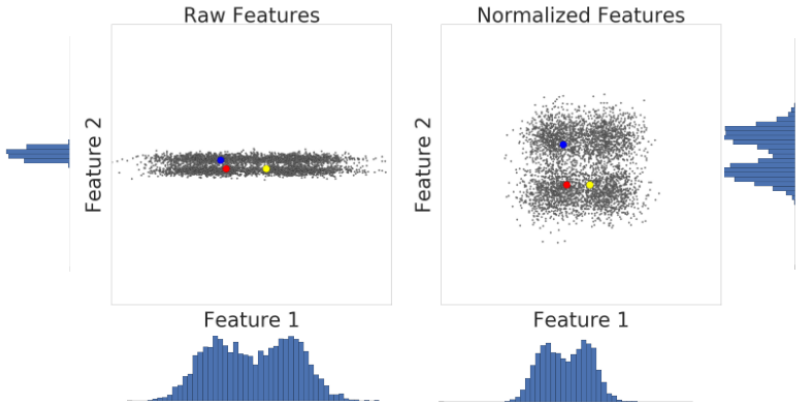
- Ogni tipo di algoritmo di clustering ha una diversa applicazione e complessità computazionale.

Tipo di clustering	Descrizione
Centroid – based	Dati organizzati in base alla distanza da un centroide. Efficienti, ma sensibili a condizioni iniziali e presenza di outliers.
Density – based	Dati organizzati in base alla densità. Efficaci nel caso di cluster ad alta densità e per l'outlier detection.
Distribution – based	Dati organizzati secondo la distribuzione, supposta gaussiana. Efficaci soltanto se la tesi di distribuzione gaussiana risulta essere corretta.
Hierarchical	Dati organizzati secondo un albero gerarchico, che può essere tagliato per ridurre il numero complessivo di cluster. Efficaci nel caso di dati di un certo tipo, come le tassonomie.

Workflow del clustering (1)

- Gli algoritmi di clustering prevedono un workflow, esattamente come quelli di machine learning visti finora.



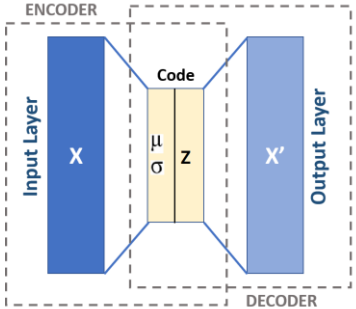


Workflow del clustering (2)

- Per definire una metrica ci sono due possibilità
- Nel **primo caso**, ci possiamo affidare ad una semplice combinazione di due/tre feature del nostro dato

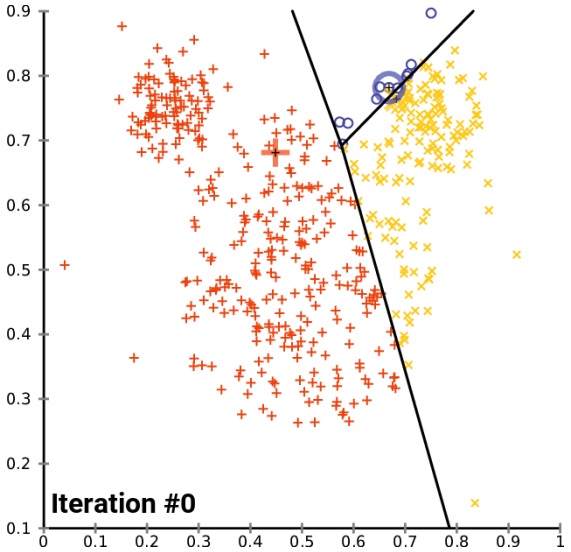
$$d = \sqrt{(x_1 - x_2)}^2$$

- Nel secondo caso, dobbiamo usare un **embedding**, ovvero una rappresentazione ridotta di un dato ad alta dimensionalità



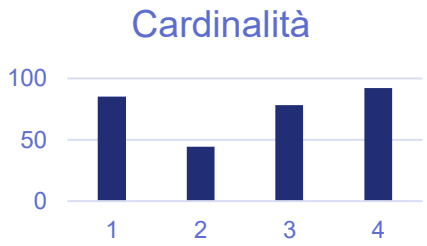
L’algoritmo K-means

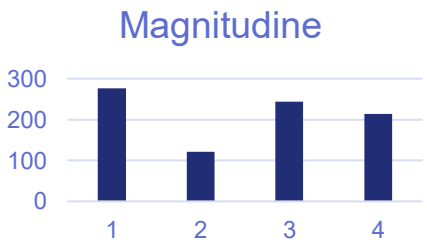
- Il **k-means** è un algoritmo iterativo
- Prevede l’assegnazione a priori del numero di cluster (il valore ***k***)
- **Primo step**: determinare i centroidi
- **Secondo step**: calcolare la distanza dai centroidi
- **Terzo step**: aggiornare i centroidi, e ripetere dal secondo step fino a che non si arriva a convergenza
- Implementato in Scikit Learn grazie alla classe `KMeans()`

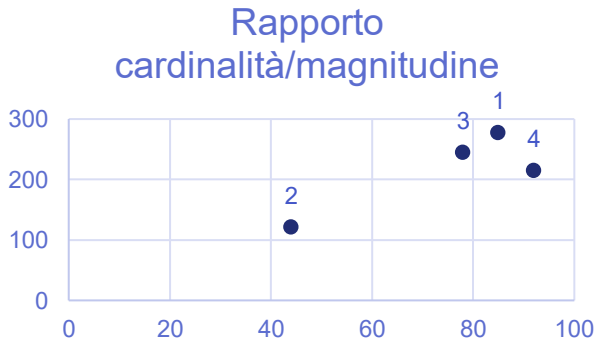


Valutazione della bontà del clustering (1)

- Valutiamo il rapporto tra **cardinalità** e **magnitudine**
  - **Cardinalità:** numero di campioni in ogni cluster
  - **Magnitudine:** somma delle distanze tra i campioni in ciascun cluster
- Il rapporto dovrebbe essere quanto più possibile lineare.
  - Di conseguenza, il quarto cluster ha dei problemi!

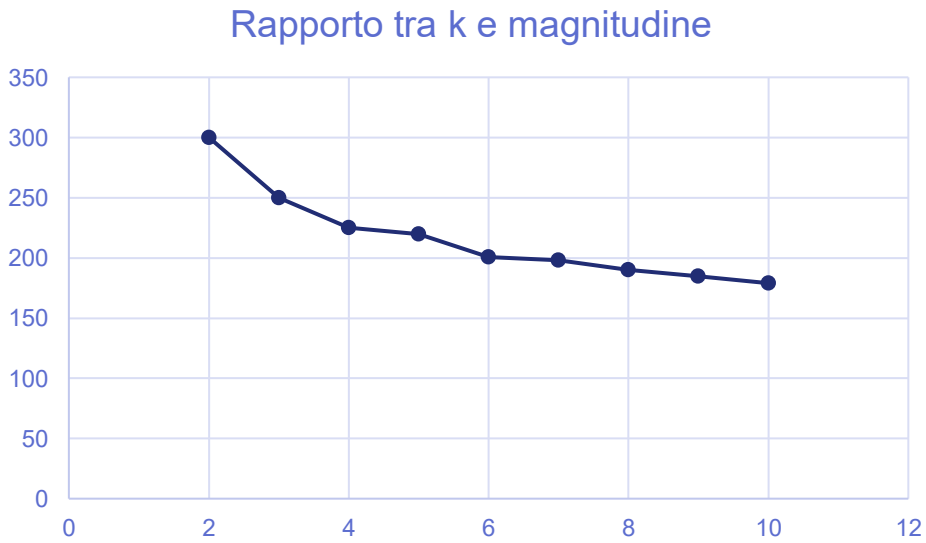






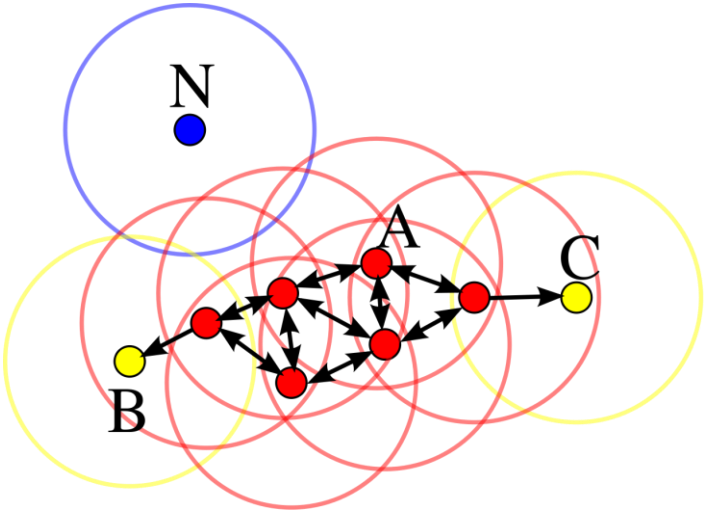
Valutazione della bontà del clustering (2)

- Valutiamo il rapporto tra **k** e **magnitudine** per stabilire un numero *ottimo* di cluster



DBSCAN

- Il **DBSCAN** è un algoritmo di tipo agglomerativo basato sulla densità.
- Utilizza il concetto di **distanza minima** tra nodi (ϵ) e **numero minimo di campioni** per la definizione di un cluster.
- In pratica, i punti in rosso vanno a definire un **core point**, quelli in giallo sono **density reachable** (e quindi appartengono al cluster definito dal core point), mentre quello in blu è un outlier.
- In Scikit Learn è implementato mediante l'uso della classe `DBSCAN()`.



Metriche di valutazione

- L'approccio usato in precedenza per la valutazione del numero ottimale di cluster è chiaramente subottimo.
- In tal senso, è possibile affidarsi ad opportune metriche (proprio come per gli algoritmi supervisionati).
- Due di queste metriche sono l'**indice di Rand** ed il **silhouette score**.
- L'indice di Rand è definito come:

$$RI = \frac{a + b}{C_2^n}$$

- dove a (e b) è l'insieme di coppie di campioni che appartengono (o *non* appartengono) allo stesso cluster sia nell'assegnazione 'vera' sia in quella data dall'algoritmo di clustering, mentre C_2^n è il numero totale di coppie di campioni possibili.

Metriche di valutazione

- L'indice di Rand assume valore tra 0 e 1. Tuttavia, **non è garantito che sia 0 per un'assegnazione completamente casuale delle label**.
- In tal senso, si usa l'indice di Rand modificato, che tiene in conto un'assegnazione completamente casuale delle label:

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]}$$

- L'indice di Rand presuppone la conoscenza del ground truth; se non lo si conosce, si può usare il **silhouette score**, definito come:

$$s = \frac{b - a}{\max(a, b)}$$

- con a e b rispettivamente distanze medie tra ogni campione e gli altri campioni appartenenti allo stesso cluster o al cluster più vicino.
- Le metriche sono al solito implementate mediante le funzioni `silhouette_score()` ed `adjusted_rand_score()`.

Domande?

42