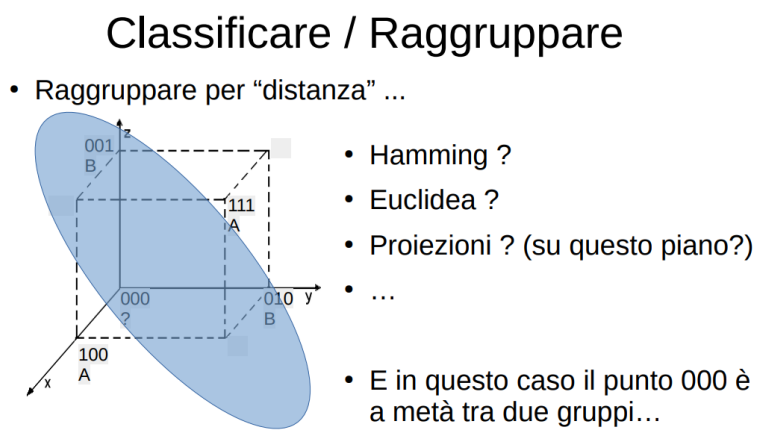
Lezione 8 1/12/2023

Apprendimento non supervisionato

Questo è un'approccio completamente diverso da quello visto fin'ora. Mentre di solito c'è qualcosa che insegno (istanze etichettate econ target) (apprendimento supervisionato), se vogliamo che il sistema impari qualcosa ma non gli diamo le etichette e i target, ci aspettiamo che il sistema separi (categorizzi) le istanze in autonomia. Se non ho target, ciascuno di noi potrebbe avere una sua idea di come raggruppare le istanze per categoria.



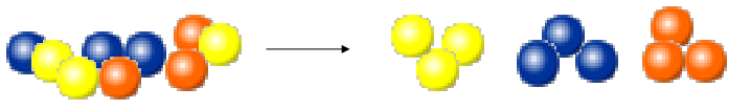
Non ho etichette, devo decidere come metterli insieme, posso farlo per distanza e ci sono molti modi per farlo.

L'apprendimento non supervisionato viene chiamato clustering (formazione di gruppi).

Può essere che il metodo non supervisionato funzioni meglio perchè evita i possibili errori date dalle etichette inserite manualmente. è più facile usare questa tecnica quando ci sono delle feature geometriche (numeriche). In alcuni ambiti si applica bene, in altri no.

Il clustering è un procedimento che si pone come obiettivo la suddivisione di un insieme di elementi in sottoinsiemi

Gli elementi di ogni sottoinsieme sono accomunati da caratteristiche simili



Insieme di elementi da classificare

- *Ogni elemento è specificato da un vettore caratteristico*

Misura di similarità (o dissimilarità) tra gli elementi

Criteri da rispettare:

- *OMOGENEITA': elementi dello stesso cluster hanno alto livello di similarità*
- *SEPARAZIONE: elementi di cluster diversi hanno basso livello di similarità*

Sia $\mathcal{N} = \{e_1, \ldots, e_n\}$ un insieme di n elementi, e sia $C = \{C_1, \ldots, C_k\}$ una partizione di $\mathcal{N}$ in sottoinsiemi. Ogni sottoinsieme è chiamato cluster e C è detto clustering di $\mathcal{N}$

Due elementi e_1 e e_2 sono chiamati mates rispetto a C se sono membri dello stesso cluster in C

Un elemento può essere rappresentato da un vettore di numeri reali, ciascuno dei quali misura una specifica caratteristica (feature)

Misura di similarità distanza tra vettori

- *Distanza euclidea* $$d(\mathbf{x},\mathbf{y}) = \left[\sum_i (x_i - y_i)^2\right]^{\frac{1}{2}}$$
- *Distanza di Manhattan* $$d(\mathbf{x},\mathbf{y}) = \sum_i |x_i - y_i|$$
- *Distanza di Minkowski* $$d(\mathbf{x},\mathbf{y}) = \left[\sum_i |x_i - y_i|^k\right]^{\frac{1}{k}}$$

Gli elementi sono vettori (di numeri) senza etichette.

Il sistema potrà raggruppare come vuole, ma vogliamo che rispetti questi due criteri:

- Quelli che metto insieme devono avere una vicinanza
- Se prendo un elemento da un gruppo e uno da un'altro questi non devono essere simili.

(metto insieme chi è simile e separo chi è diverso)

Se partiziono N voglio che due sottoinsiemi siano distinti, e che l'unione di tutti i sottoinsiemi sia uguale ad N.

Se due elementi appartengono allo stesso sottoinsieme sono compagni.

Un'ingrediente fondamentale è quello della misura quantitativa di similarità. Similarità e distanza sono la stessa cosa qui, la geometria diventa un numero.

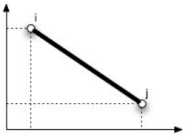
Qui abbiamo 3 alternative.

$$d(i,j)=\sqrt{(|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+...+|x_{ip}-x_{jp}|^2)}$$

$i=(x_{i1},x_{i2},...,x_{ip})$

$j=(x_{j1},x_{j2},...,x_{jp})$

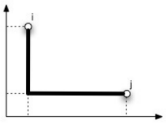
$i,j=1,2,...,n$



✓ *Invariante rispetto a traslazioni e rotazioni degli assi*

La distanza **eulidea** si sviluppa per un caso a p dimensioni in quella radice, in un caso a 2 dimensioni corrisponde alla diagonale.

$$d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+...+|x_{ip}-x_{jp}|$$



✓ *Non è invariante rispetto a traslazioni o rotazioni degli assi e pone meno enfasi sulle variabili con distanze maggiori, non elevando al quadrato le differenze*

Manhattan assomiglia ad Hamming, la formula è comunque la somma delle distanze lungo le dimensioni.

Non ci preoccupiamo di vedere le differenze tra un metodo e l'altro perchè passeremo ad esempi geometrici su piani a due dimensioni.

$$d(i,j)=\sqrt[q]{(|x_{i1}-x_{j1}|^q+|x_{i2}-x_{j2}|^q+...+|x_{ip}-x_{jp}|^q)}$$

✓ *Dove q è un intero positivo:*
- *q = 1 ➪ Distanza di Manhattan*
- *q = 2 ➪ Distanza euclidea*
- *q = ∞ ➪ Distanza di Lagrange-Tchebychev*

✓ *Clustering gerarchico*
- *Neighbor joining*
- *Metodo del centroide*

✓ *Clustering non gerarchico*
- *K-means*

✓ *Basati sulla teoria dei grafi:*
- *Highly Connected Subgraph (HCS)*
- *CLustering Identification via Connectivity Kernels (CLICK)*

✓ *Euristica per un algoritmo polinomiale:*
- *Clustering Affinity Search Technique (CAST)*
- *Self-Organizing Maps (SOM)*

Esistono tecniche diverse, noi useremo l'algoritmo K-means.

L'algoritmo da usare dipende dalla situazione in cui ci si trova, in base ai dati.

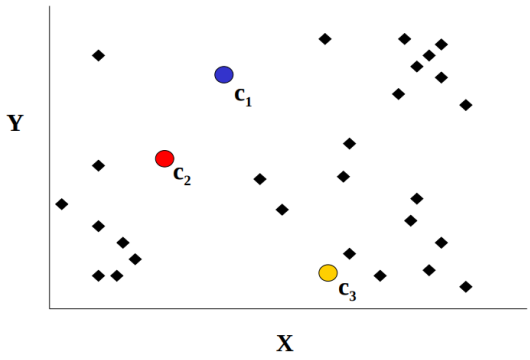
Quindi noi siamo nella sezione non gerarchica, con partizione netta dei punti, senza sotto sotto sotto insiemi.

K-means

Non è gerarchico, è una parzizione dei punti. é divisivo, cioè all'inizio i punti sono insieme e poi li separa.
Questo algoritmo parte con un dato in ingresso che dovremo decidere quasi arbitrariamente. Decideremo noi un k, che sarà il numero di gruppi da creare. Alla fine si potrà vedere se k andava bene ma non si sà a priori.

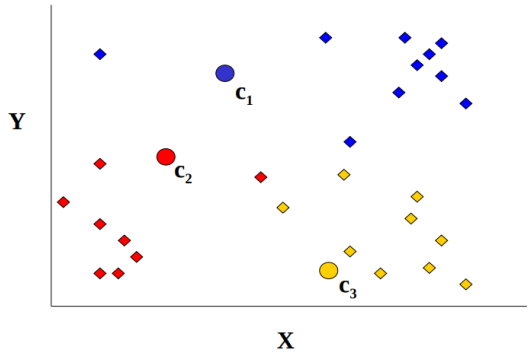
Esempio k=3

1. *Scelta casuale di 3 centroidi*



I punti colorate non sono istanze, sono l'identità del cluster.

2. *Assegnazione di ciascun punto al centroide più vicino*



Ogni punto va al centroide più vicino.

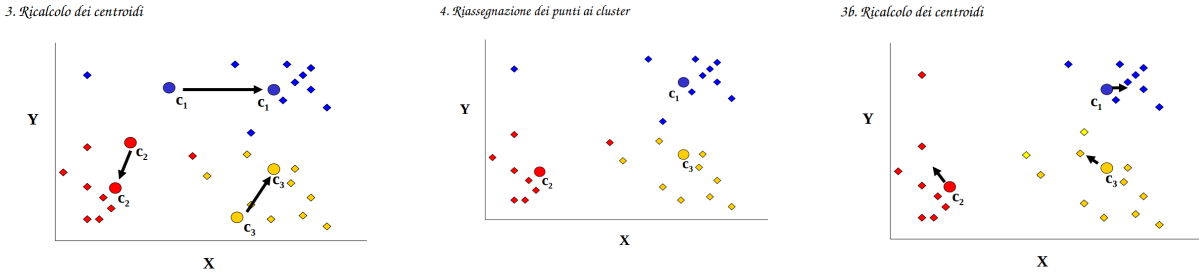
Algoritmo

- ✓ *Algoritmo (lavora solo con dati numerici)*
  1. *Si fissano a caso k centroidi iniziali di altrettanti cluster*
  2. *Per ogni individuo si calcola la distanza da ciascun centroide e lo si assegna al più vicino*
  3. *Per la partizione provvisoria così ottenuta si ricalcolano i centroidi di ogni cluster (media aritmetica)*
  4. *Per ogni individuo si ricalcola la distanza dai centroidi e si effettuano gli eventuali spostamenti tra cluster*
  5. *Si ripetono le operazioni 3 e 4 finché si raggiunge il numero massimo di iterazioni impostate o non si verificano altri spostamenti*

Scegliamo il valore di k, poi dal 2 al 5 è un loop. Iterazione dopo iterazione andrò a spostare i centroidi, calcolando la distanza tra tutti i punti e tutti i centroidi.

Dato un centroide ho un insieme di gruppi appartenenti a quel cluster. Facendo una media delle posizioni degli elementi di quel gruppo trovo un nuovo centroide, che quindi si sposta. A quel punto riparto. Anche gli aggiustamenti li faccio un numero prefissato di volte. Può capitare che i cluster non cambino dopo il ricalcolo dei centroidi e quindi l'iterazione si ferma.

Tornando all'esempio



Qui poi facendo i ricalcoli i clussters non cambiano e quindi l'iterazione si ferma.

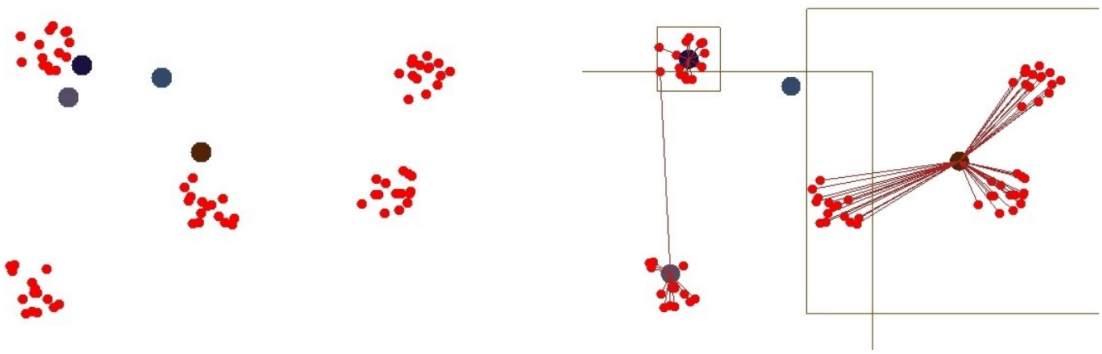
Vantaggi:

- *Semplice implementazione;*
- *tempo di calcolo* $O(tkn)$ *in cui* n *è la cardinalità dell'insieme dei dati,* k *il numero di clusters e* t *il numero di iterazioni del ciclo (*k*,* $t << n$*)*

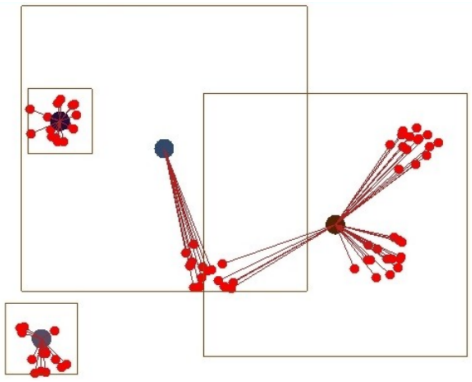
Svantaggi:

- *sensibilità rispetto alla scelta dei centroidi iniziali*
- *non possiamo predire il numero di cluster non conoscendo i dati a priori*
- *Non esiste un* k *ottimale e non ci sono proprietà che ce lo possano suggerire*

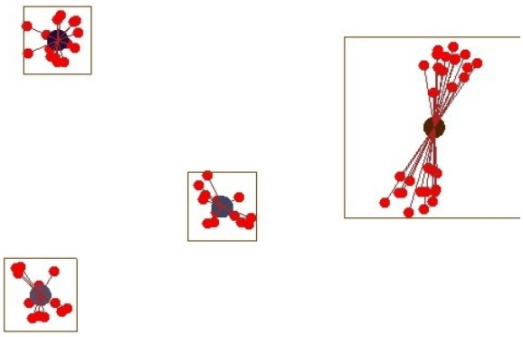
Esempio 2 (4 cluster)



Vediamo che qui avremmo bisogno di 5 clusers non 4.



I centroidi si sposteranno comunque

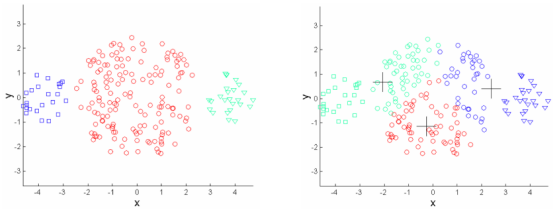


Iterando i centroidi si spostano più vicini al gruppo più forte e arriverò ad un risultato accettabile dove però avendo una quantità di centroidi insufficienti, si contenderanno i punti.

Se la scelta del k si dimostra non ideale, si può cambiare il k.

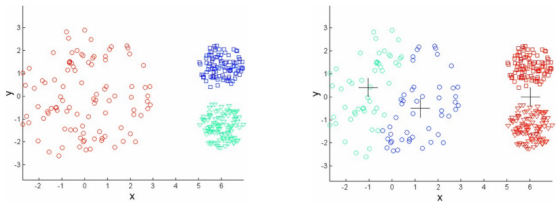
Problematiche del k-means

✓ *Cluster con differenti dimensioni*



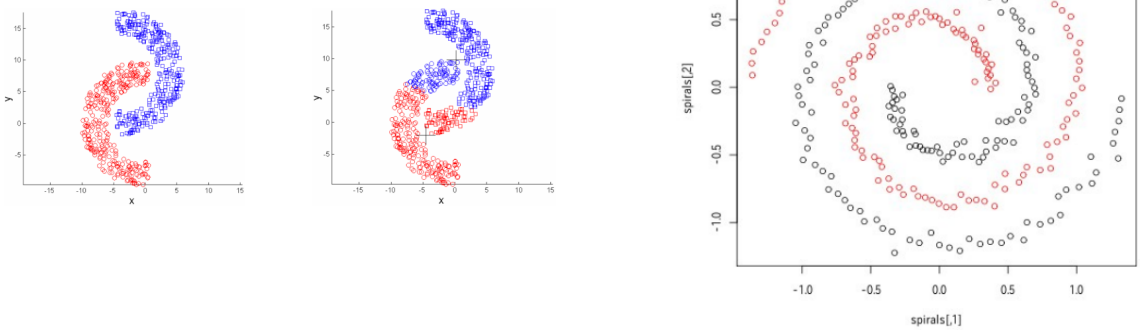
K-means metterà i centroidi nei punti medi che però sono diversi da quelli ideali. Questo accade perchè i cluster dovrebbero avere dimensioni diverse.

✓ *Cluster con differenti densità*



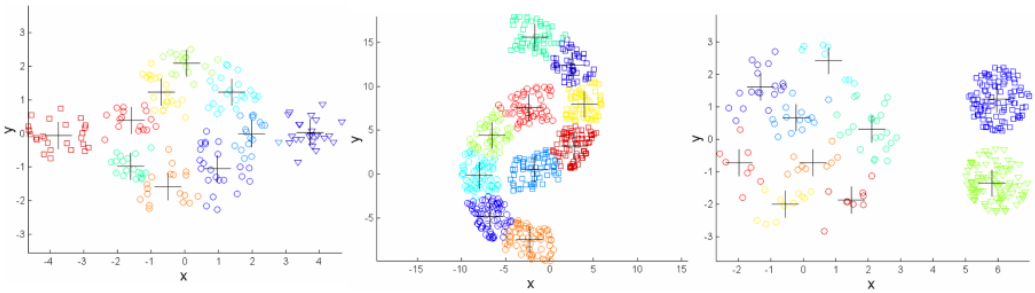
Lavodando per media dei punti, i cluster si posizionano in modo errato.

✓ *problematiche relative alle proprietà gemetriche del cluster*



Canali che separano due gruppi.

- *Utilizzo di un numero maggiore di cluster*
- *Necessaria fase di unione*



Vediamo qui come l'algoritmo è sensibile alla quantità k. Avendo più k, i gruppi di punti iniziano ad essere frazionati di più e si crea una via di mezzo perchè potrei sistemare questi clusers a valle, unendoli.

Misura di silhouette

Misura di “silhouette” (dim.cluster>1)

Data una distanza d(,), calcoliamo per un punto *i* di C_I:

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i,j)$$

- distanza media “INTRA”

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$

- d.media “INTER-cluster”

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

- Silhouette per punto *i*

Se il clustering è definito tramite distanza, allora possiamo tradurre i due criteri (separare i diversi e unire i simili) in questi termini:

siamo dopo k-means, valutiamo il cluster creato, per ogni punto sappiamo dire a quale gruppo esso appartiene. Possiamo quindi calcolare questi 3 elementi a, b, s per ogni singolo elemento.

Partiamo da un punto i fisso, vediamo come si pone rispetto agli altri.

La misura a è quella che prende tutti i j presenti al suo stesso cluster, che devono quindi avere distanza bassa.

Analogamente in b si fa per i punti che non appartengono al cluster. Qui stiamo prendendo il caso peggiore (min).

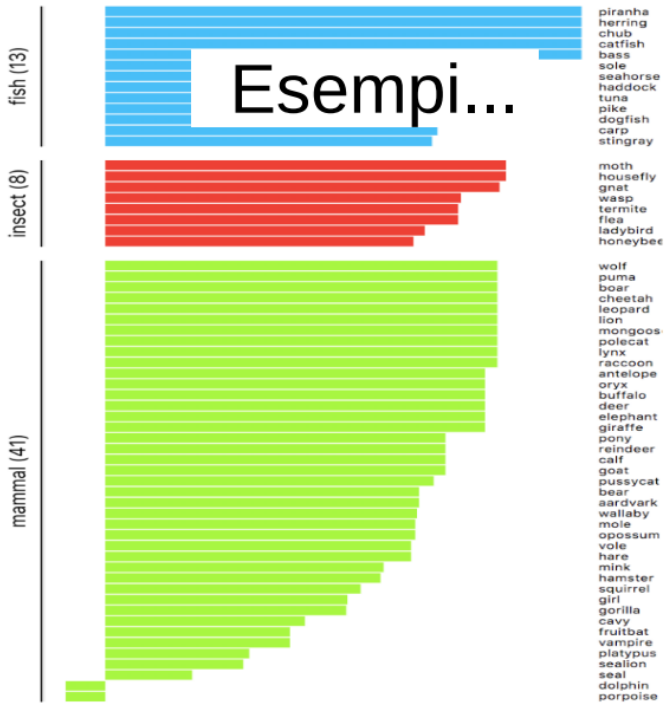
s all’aumentare di b e al diminuire di a, aumenta.

Abbiamo quindi messo in un calcolo numerico quei due criteri: vogliamo che si abbassi la distanza all’interno del cluster e che aumenti quella al di fuori del cluster.

Remarks (from Wikipedia)

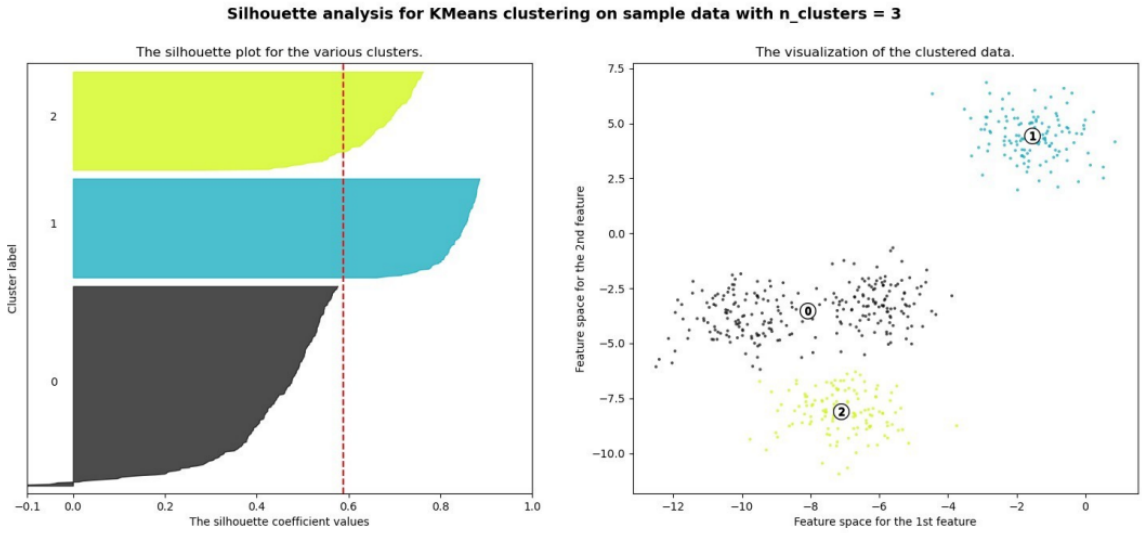
- The mean *s(i)* over all points of a cluster is a measure of how tightly grouped all the points in the cluster are.
- Thus the mean *s(i)* over all data of the entire dataset is a measure of how appropriately the data have been clustered. If there are too many or too few clusters, as may occur when a poor choice of *k* is used in the clustering algorithm (e.g.: k-means), some of the clusters will typically display much narrower silhouettes than the rest.
- Thus silhouette plots and means may be used to determine the natural number of clusters within a dataset.

L’idea è quella di fare la media degli s, che è la nostra valutazione, e possiamo cercare il k ottimale. Infine si può fare una rappresentazione grafica di queste misure.

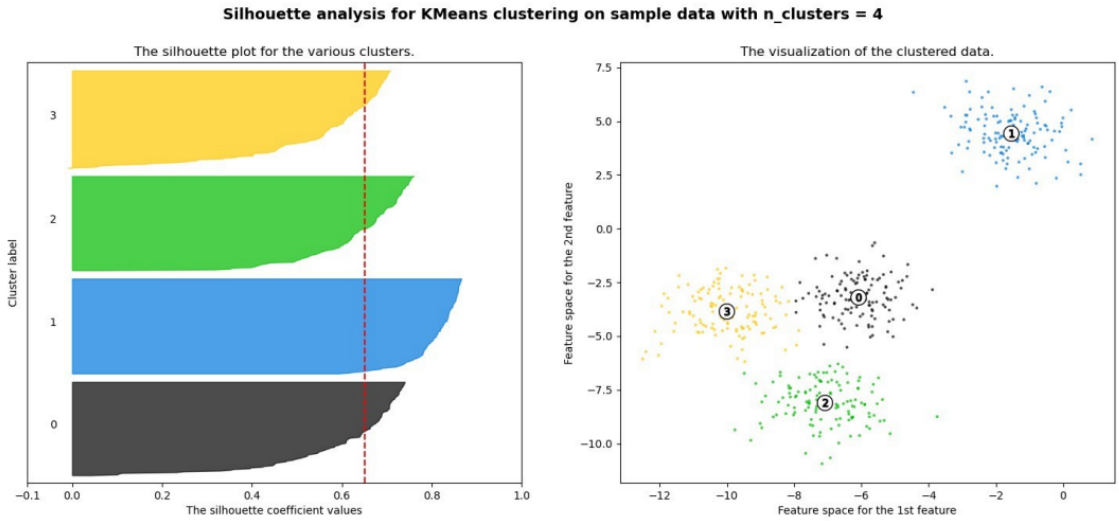


Abbiamo a destra le istanze (nomi di animali), immaginiamo che ci sia un sistema che associa ad ogni animale un vettore che ci porta nel caso geometrico. Non è detto che il clustering corrisponda alla nostra divisione degli animali “ad occhio”.

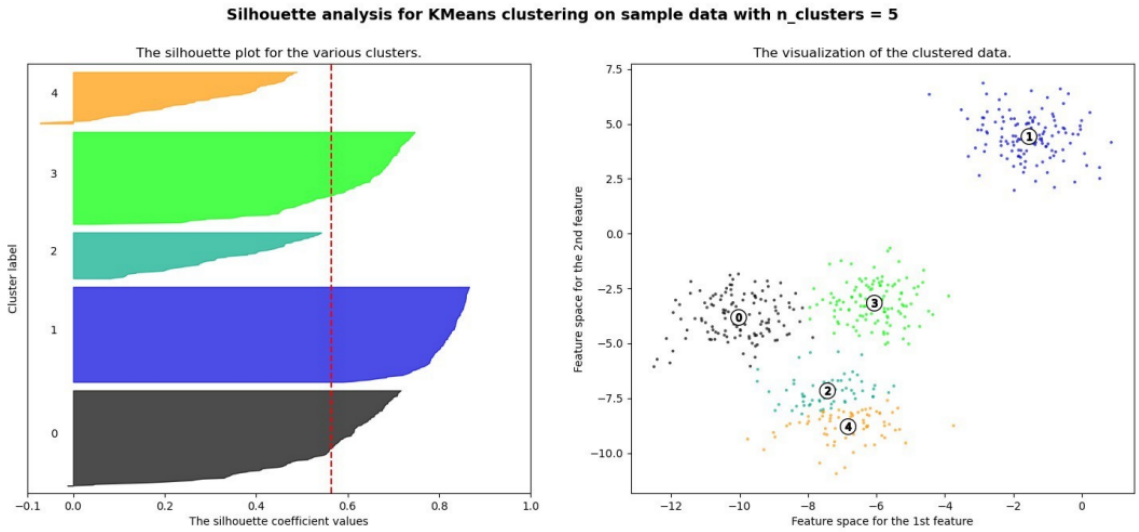
Qua vediamo che in alcune istanze il valore di s è brutto, qui con dolphin e porpoise (anomali).



Qui invece vediamo che il k ottimale doveva essere 4. Si vede come il terzo cluster (nero) ha una silhouette bassa rispetto alla s media.



Con k=4 invece vediamo che tutti i cluster hanno elementi positivi e hanno buoni valori rispetto alla media.



Se però il k è troppo grande, vedo che alcuni valori di s sono negativi.