

Apprendimento non supervisionato

Introduzione alla Data Science

Nicoletta Noceti

Oggi

- Parliamo oggi di metodi della famiglia del Machine Learning non supervisionato
- Abbiamo già osservato che il tipo di problemi che possiamo risolvere cambiano

Definizioni

Una definizione

Con il termine Machine Learning ci riferiamo ad una classe di metodi in grado di imparare da esempi invece di essere esplicitamente programmati a fare qualcosa

Due macro-tipologie

- Machine Learning supervisionato (simula l'imparare con un insegnante)
- Machine Learning non supervisionato (simula l'imparare senza un insegnante)

Oggi parliamo di Machine Learning NON supervisionato

Il setting cambia

$$S = \{(x_1, \cancel{y_1}), (x_2, \cancel{y_2}), \dots, (x_n, \cancel{y_n})\}$$

è il training set, ossia l'insieme di dati da cui il metodo imparerà

In quali casi potrebbe essere appropriato/utile?

Noi ne vedremo due:

- **Clustering**
- **Riduzione della dimensionalità dei dati**

Definizione

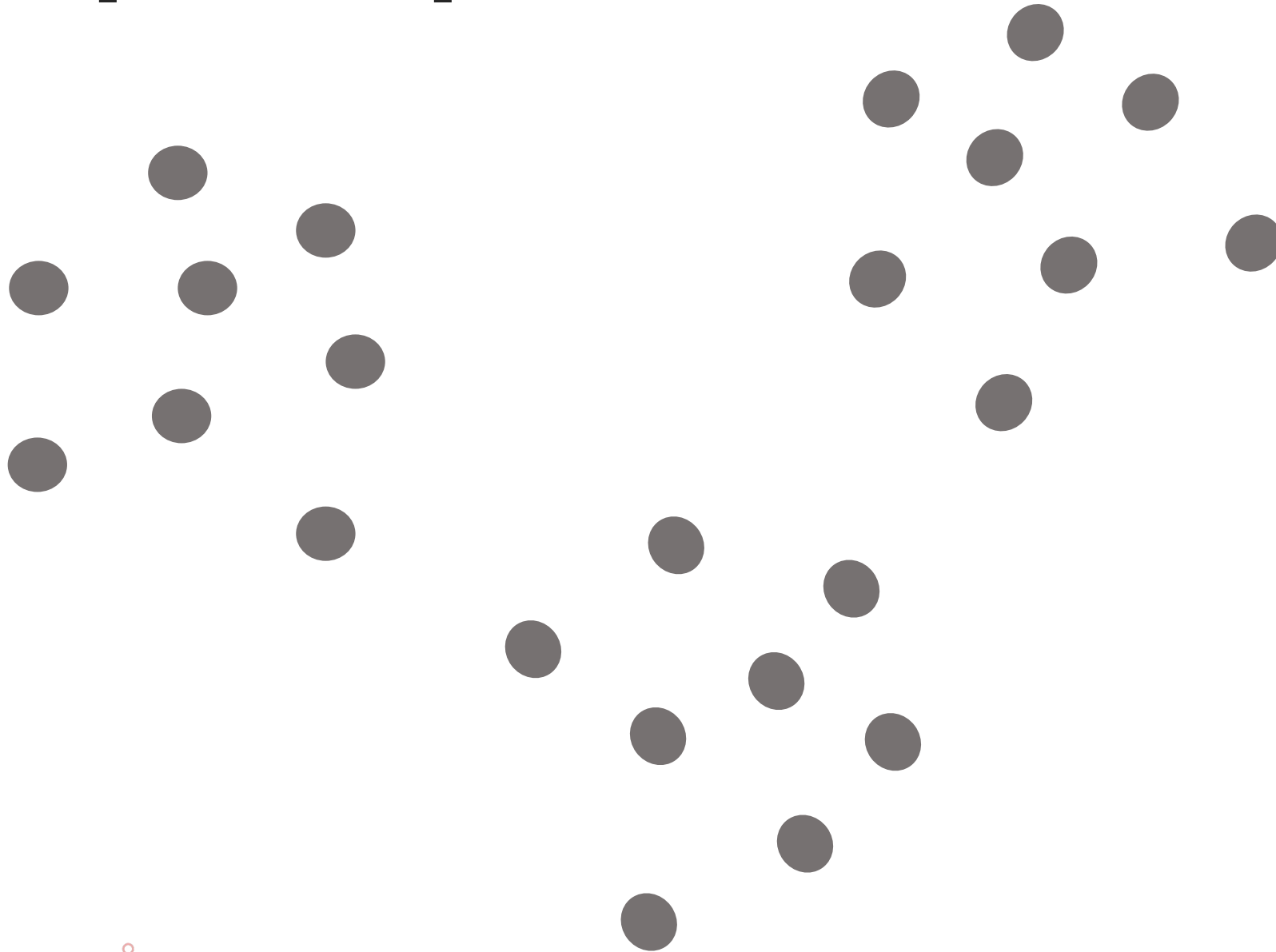
L'obiettivo degli algoritmi di clustering è di individuare strutture (gruppi di dati «coerenti» rispetto ad una qualche misura) all'interno dei dati

Due elementi importanti

- Cluster: è un gruppo di dati che si «comporta in modo analogo»
- Centroide: è il «centro» del cluster (ad esempio il punto medio, o centroide)

Clustering: K-Means

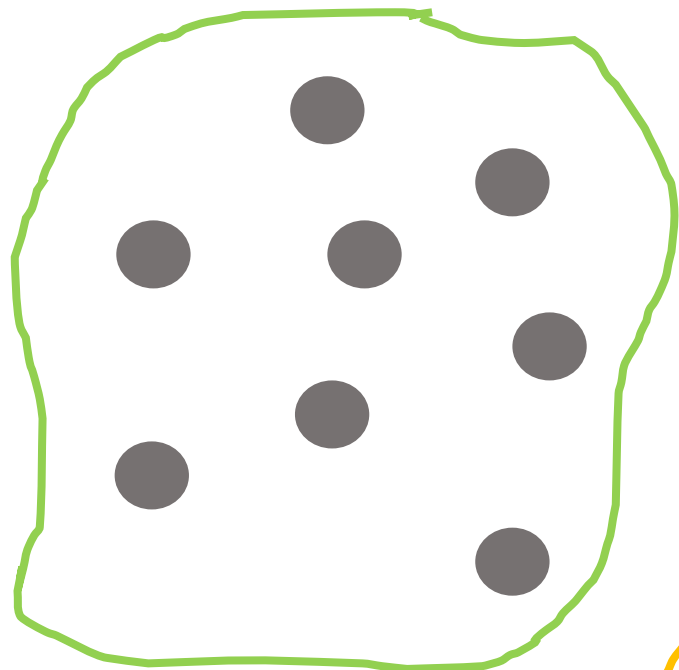
Un primo esempio



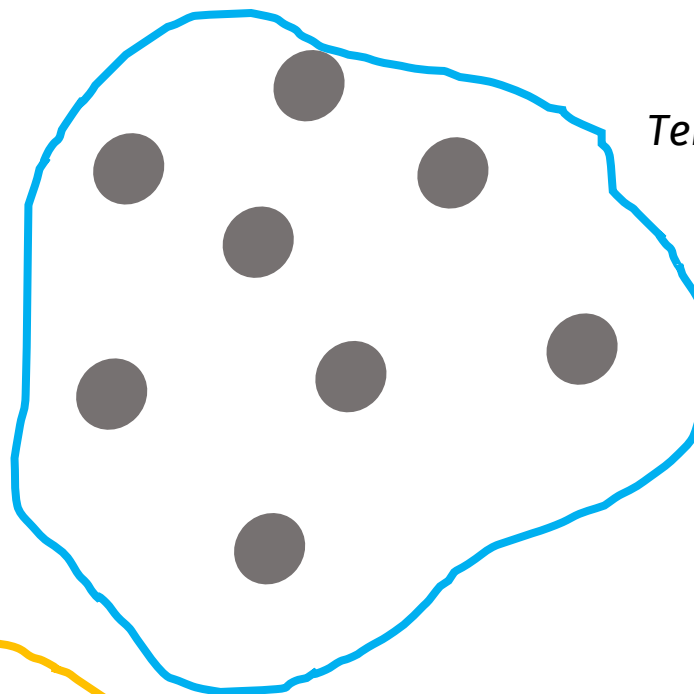
$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

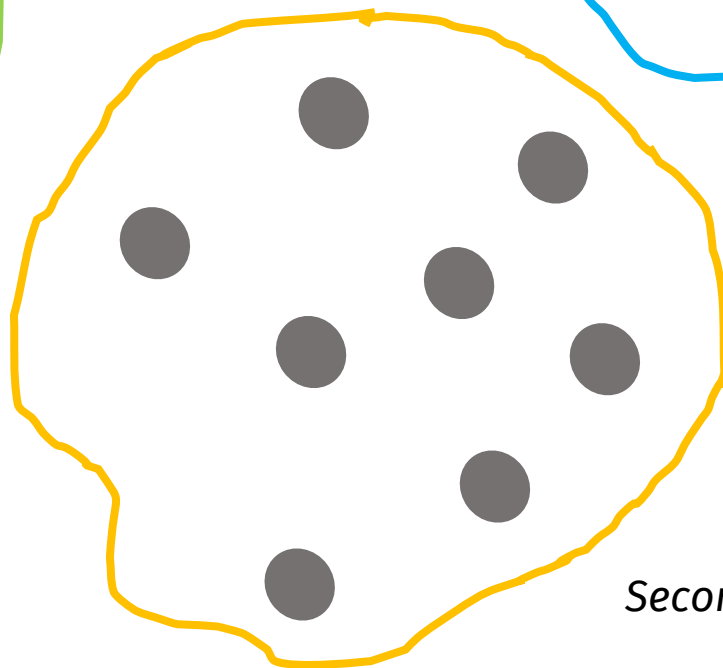
Un primo esempio



Primo cluster



Terzo cluster

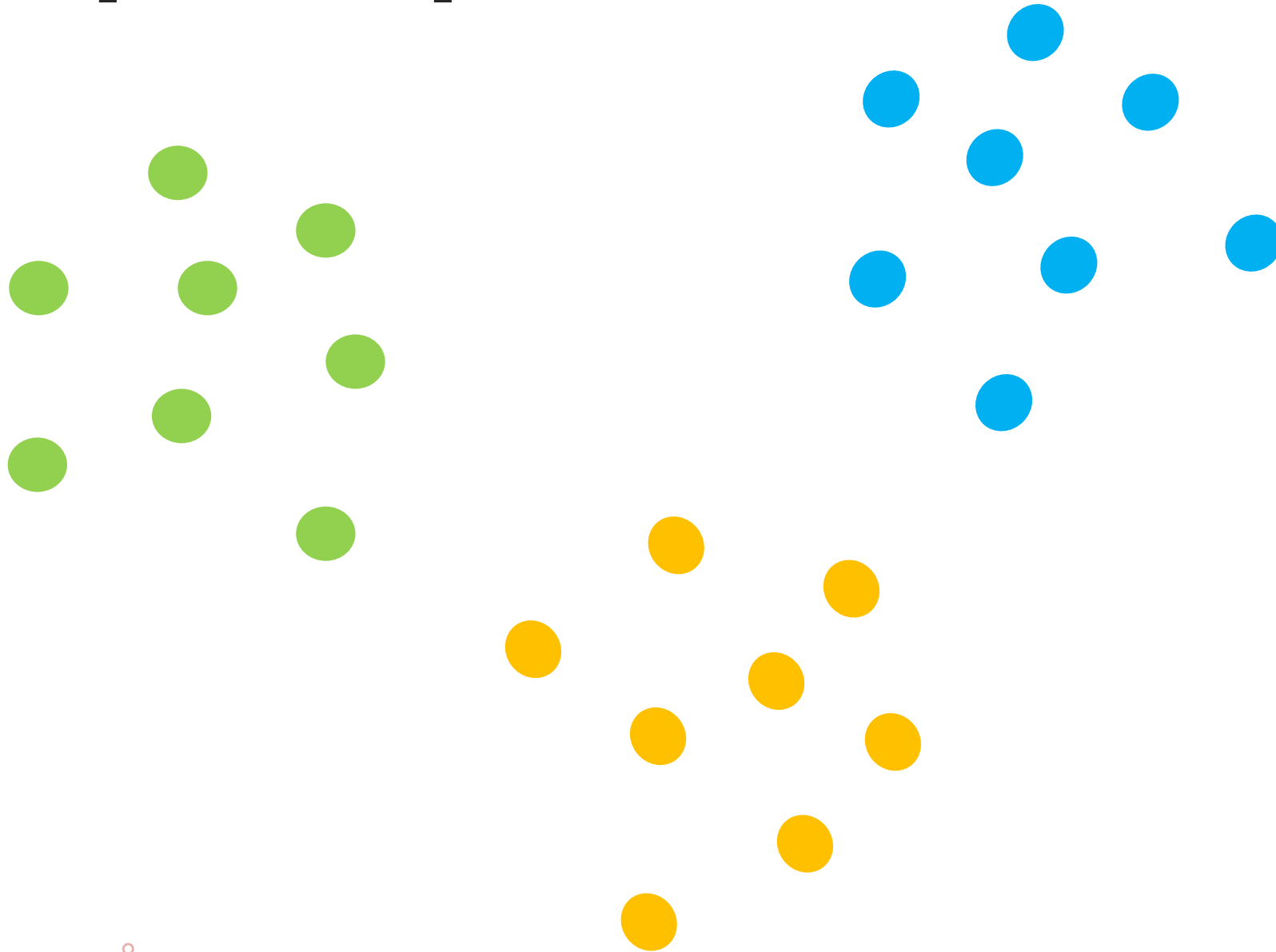


Secondo cluster

$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

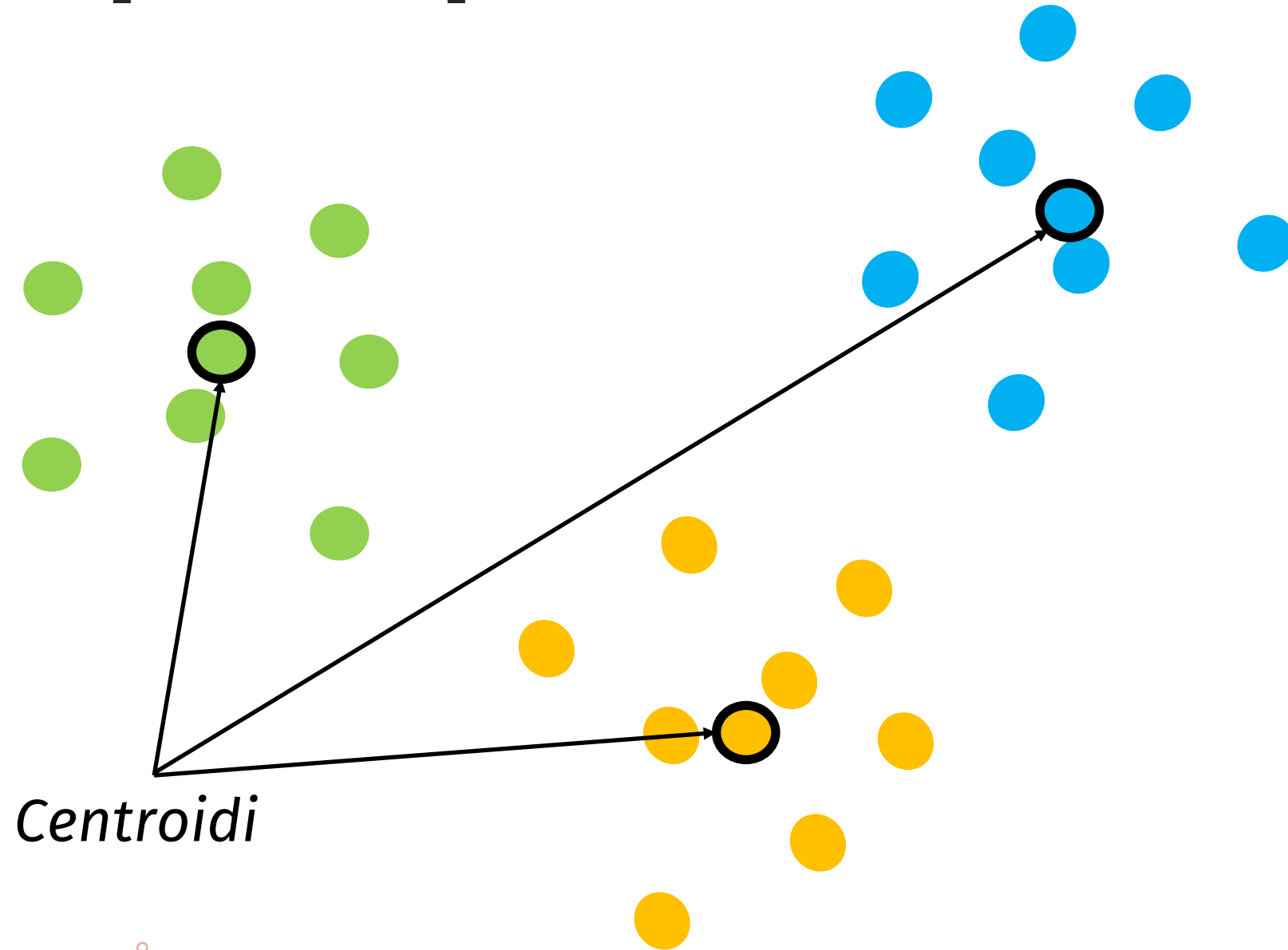
Un primo esempio



$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

Un primo esempio



$$S = \{x_i\}_{i=1}^n$$

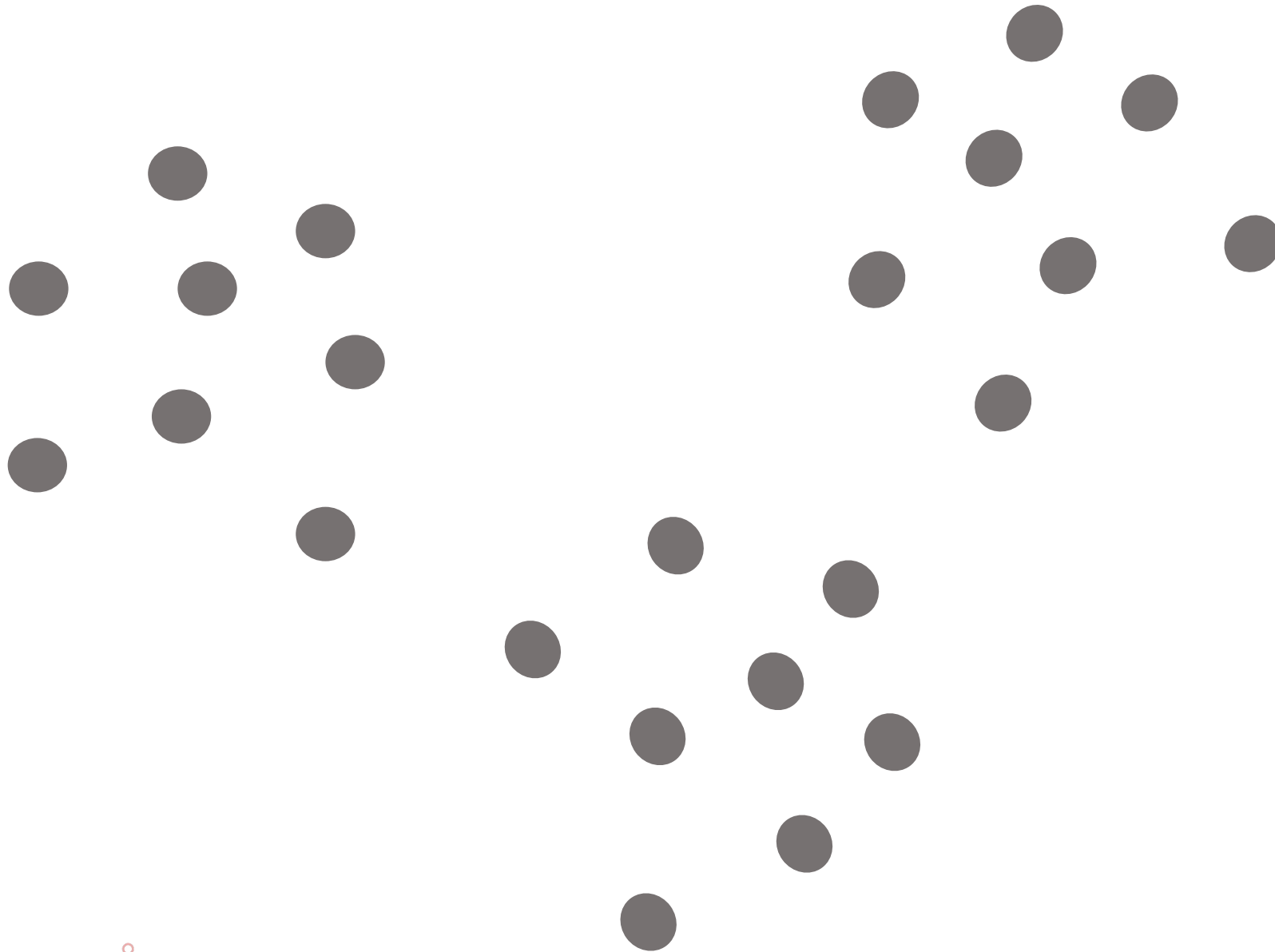
$$x_i \in \mathbb{R}^2$$

Come si procede?

I passi principali dell'algoritmo K-Means:

1. Scegliere k centroidi iniziali (k è un input!)
2. Per ogni punto:
 - Assegnare il punto al centroide più vicino
3. Per ogni centroide:
 - Aggiornare la posizione del centroide
4. Ripetere i passi 2 e 3 fino a raggiungere un criterio di arresto

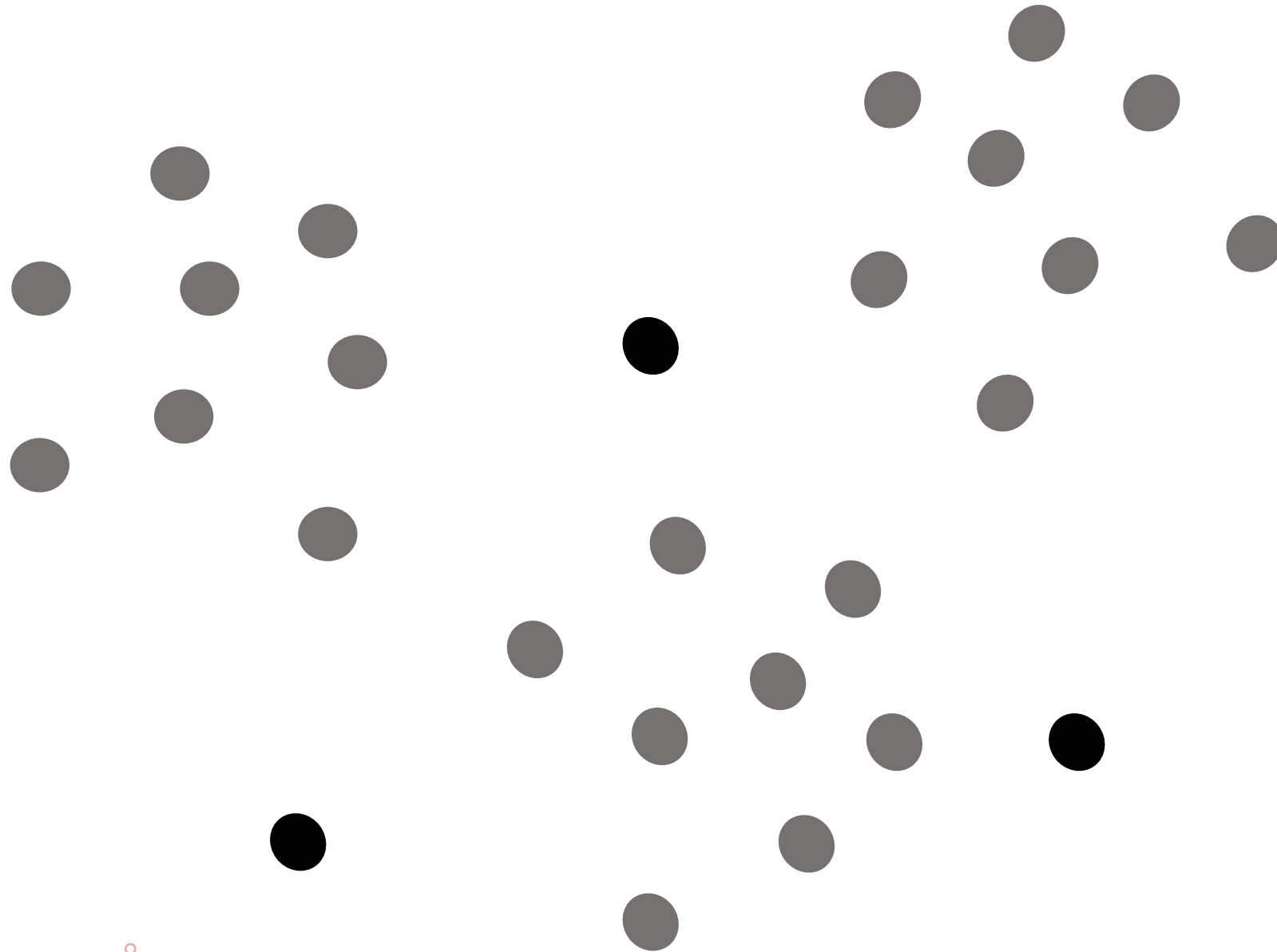
1 - Scegliere k centroidi (fissiamo k=3 per semplicità)



$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

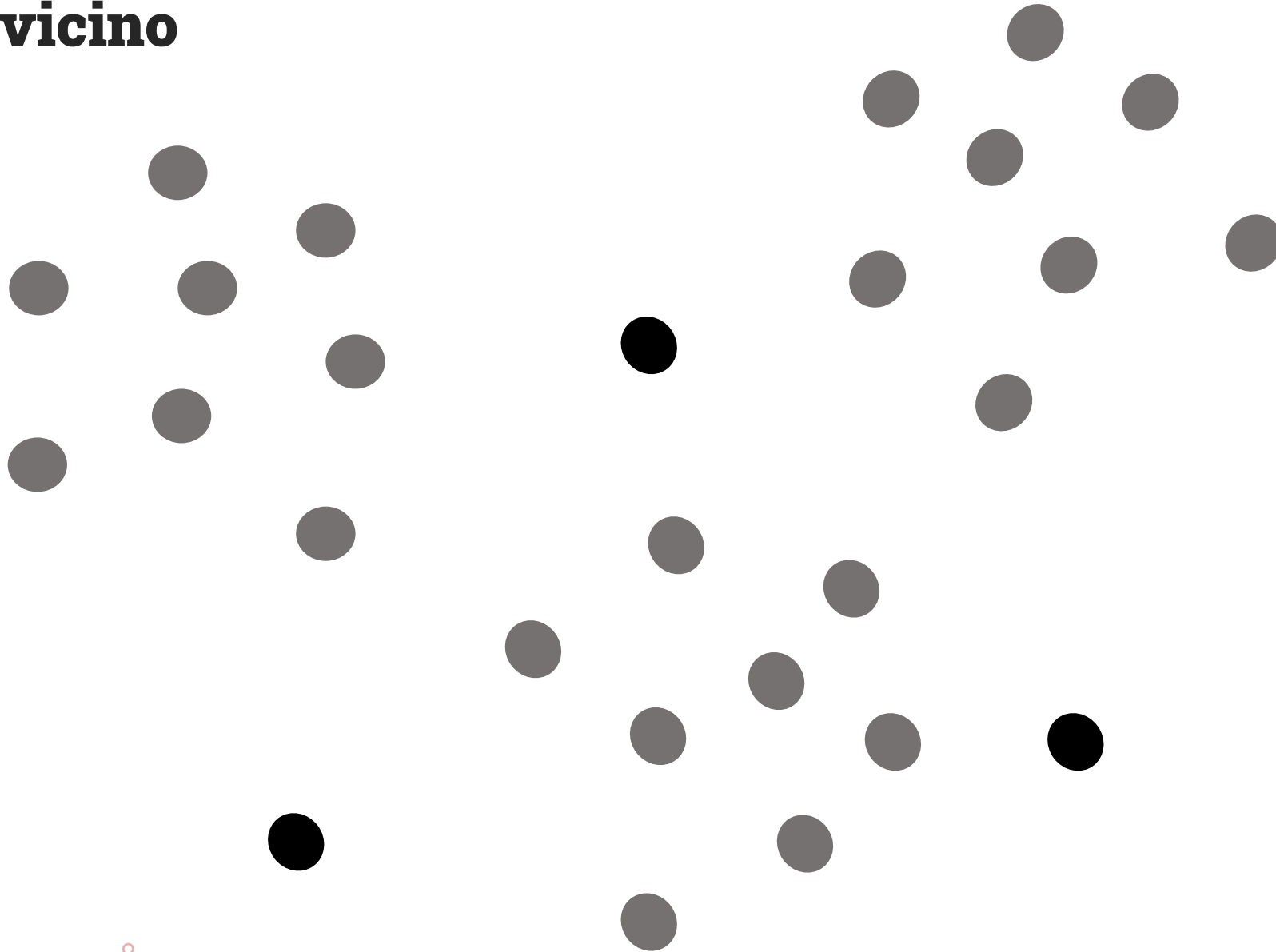
1 - Scegliere k centroidi (fissiamo k=3 per semplicità)



$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

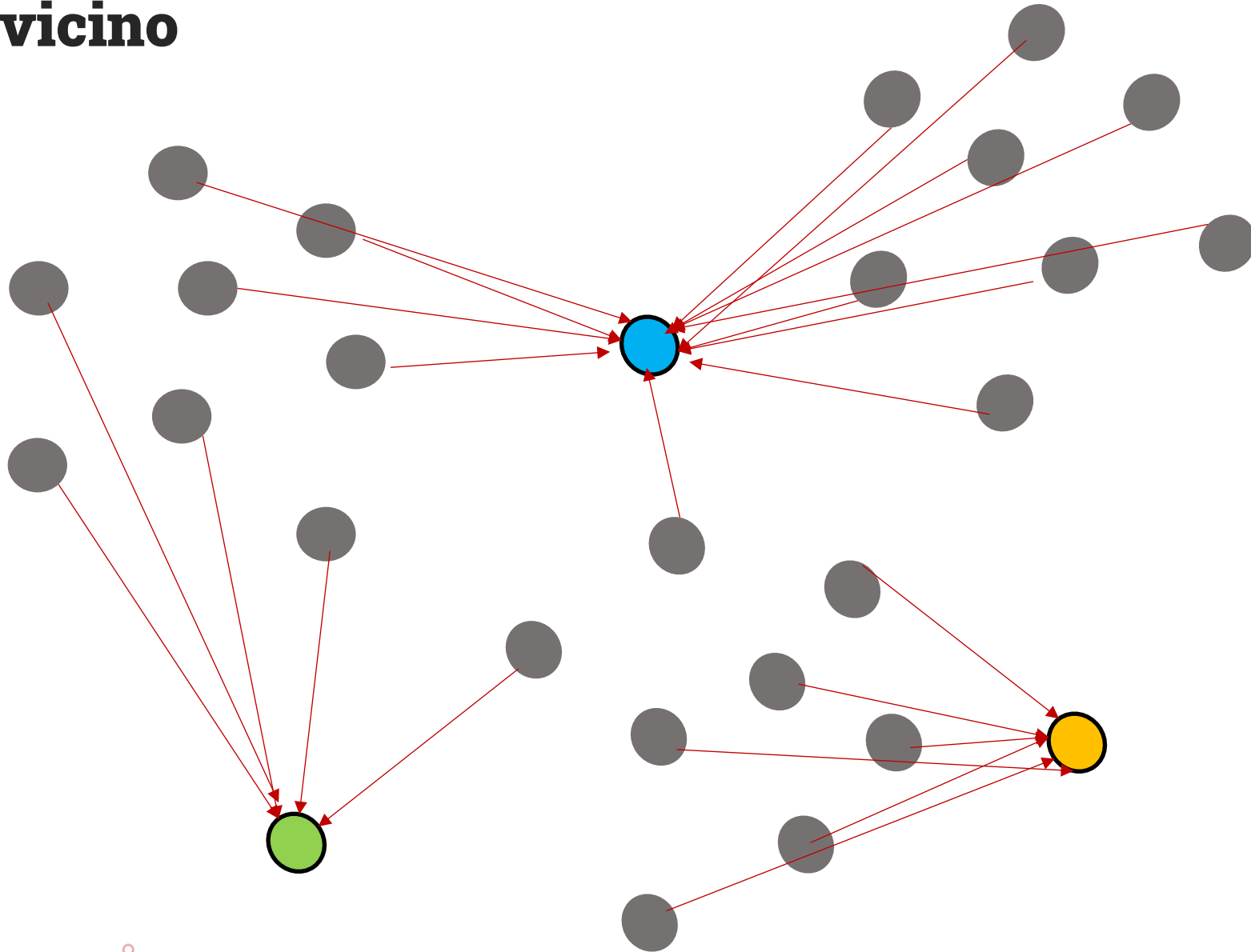
2 - Per ogni punto: Assegnare il punto al centroide più vicino



$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

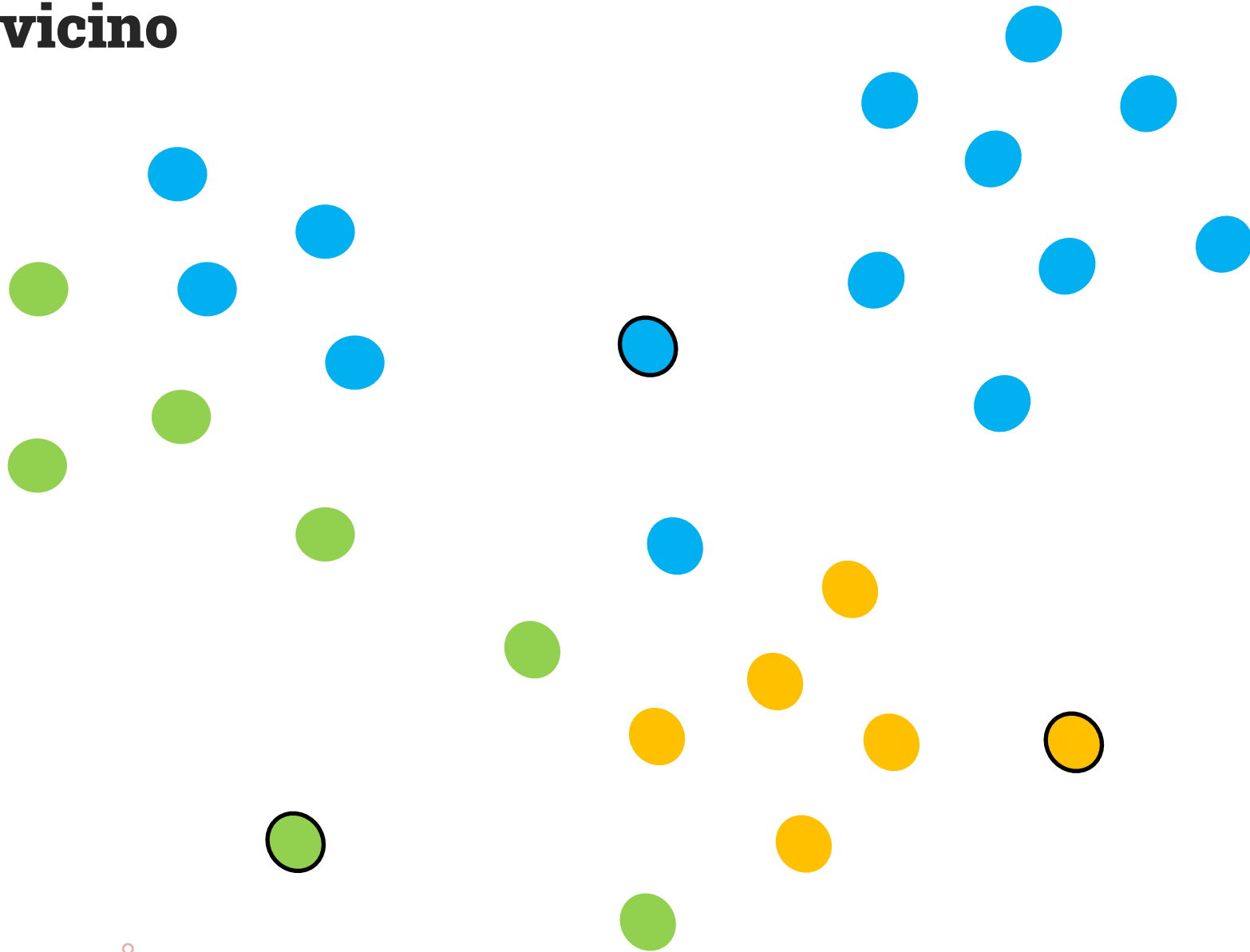
2 - Per ogni punto: Assegnare il punto al centroide più vicino



$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

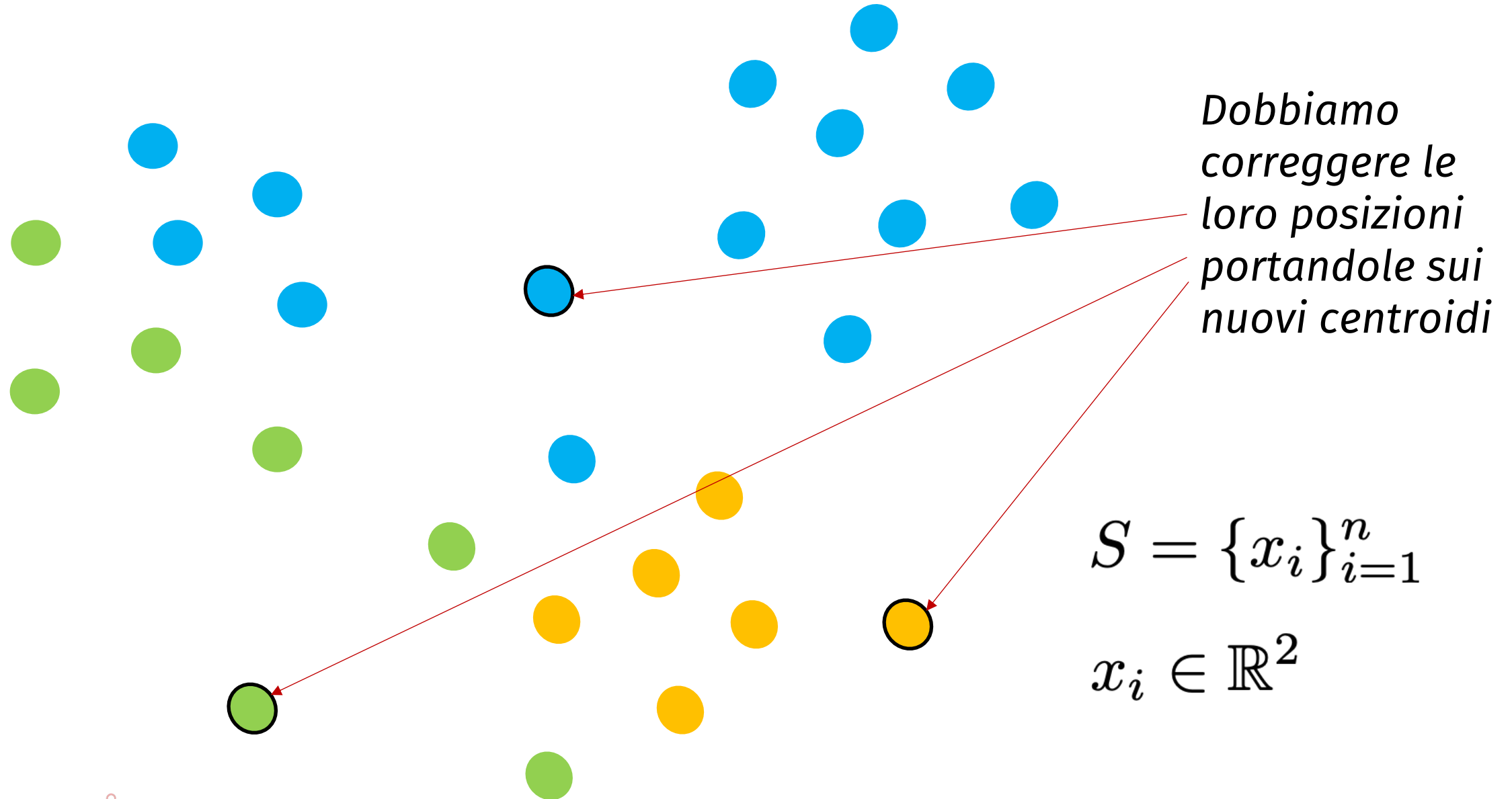
2 - Per ogni punto: Assegnare il punto al centroide più vicino



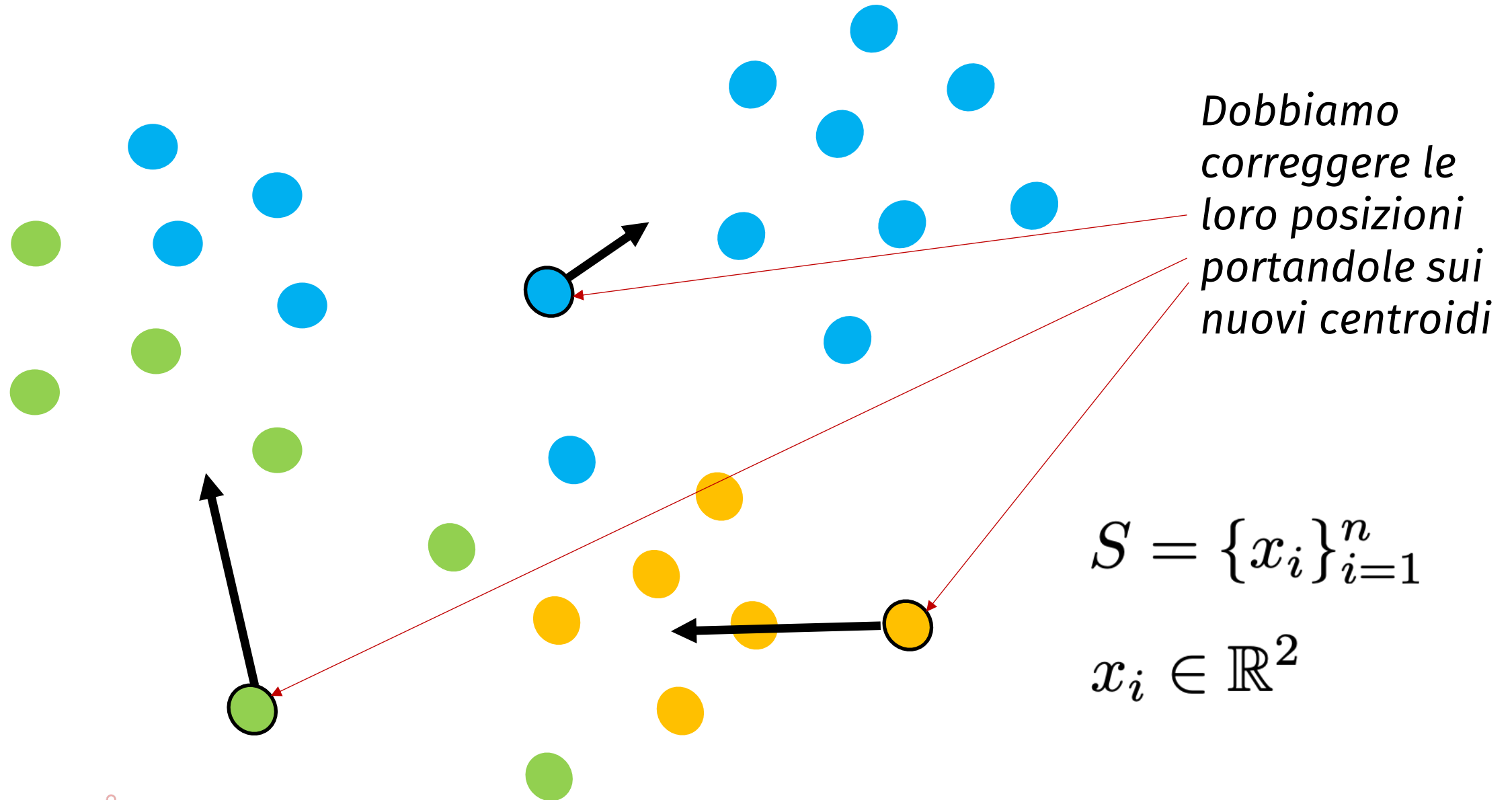
$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

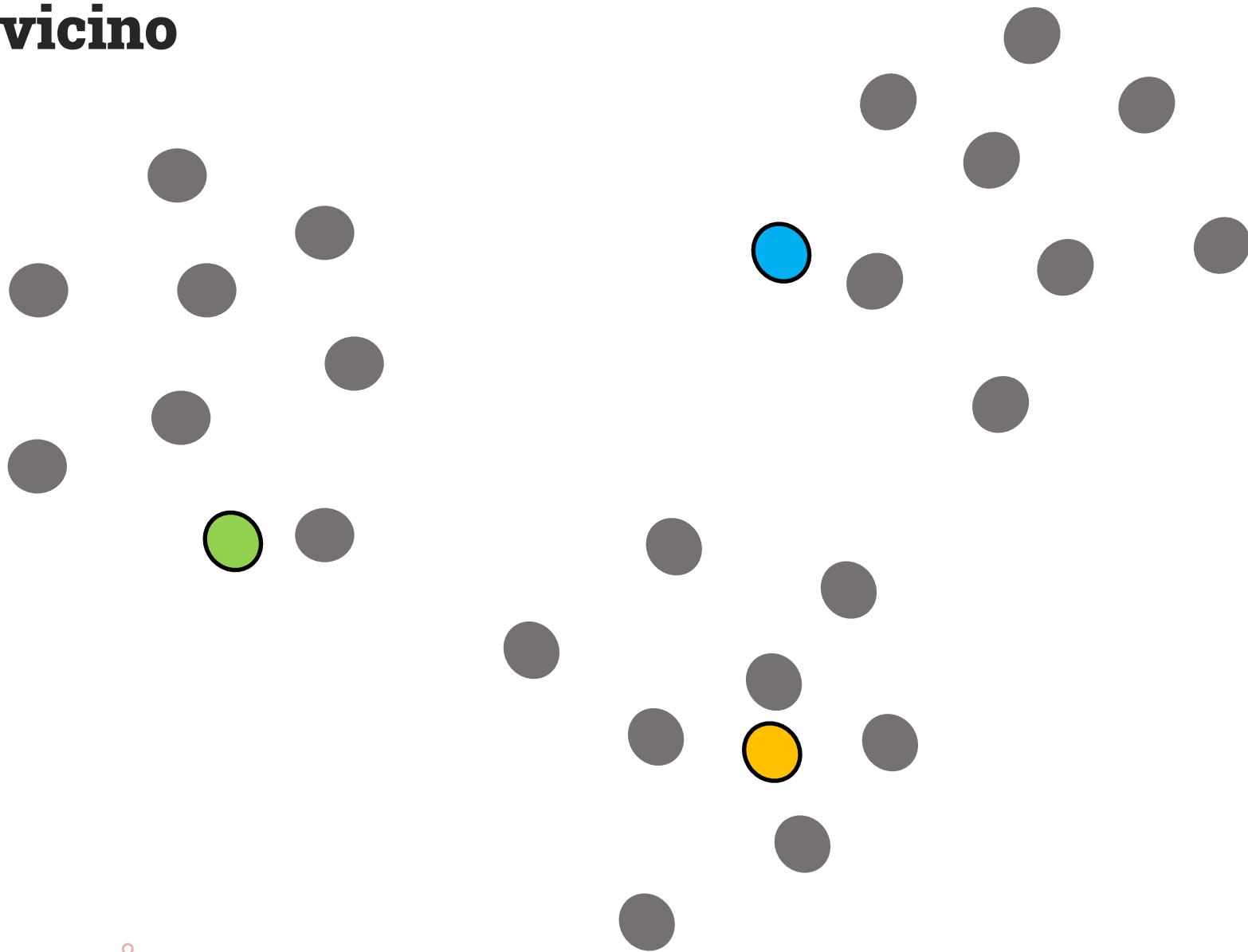
3 - Per ogni centroide: Aggiornare la posizione del centroide



3 - Per ogni centroide: Aggiornare la posizione del centroide



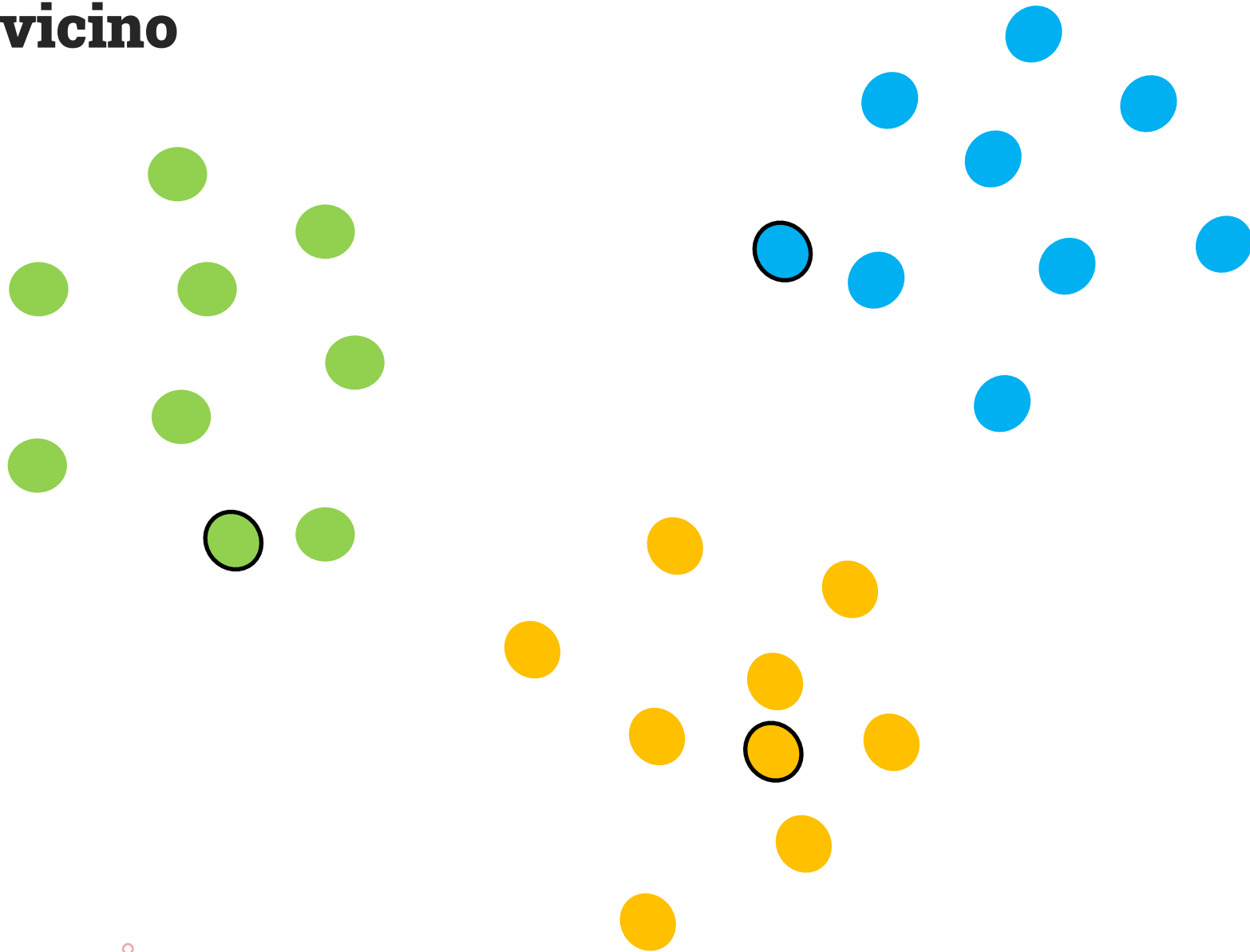
2 - Per ogni punto: Assegnare il punto al centroide più vicino



$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

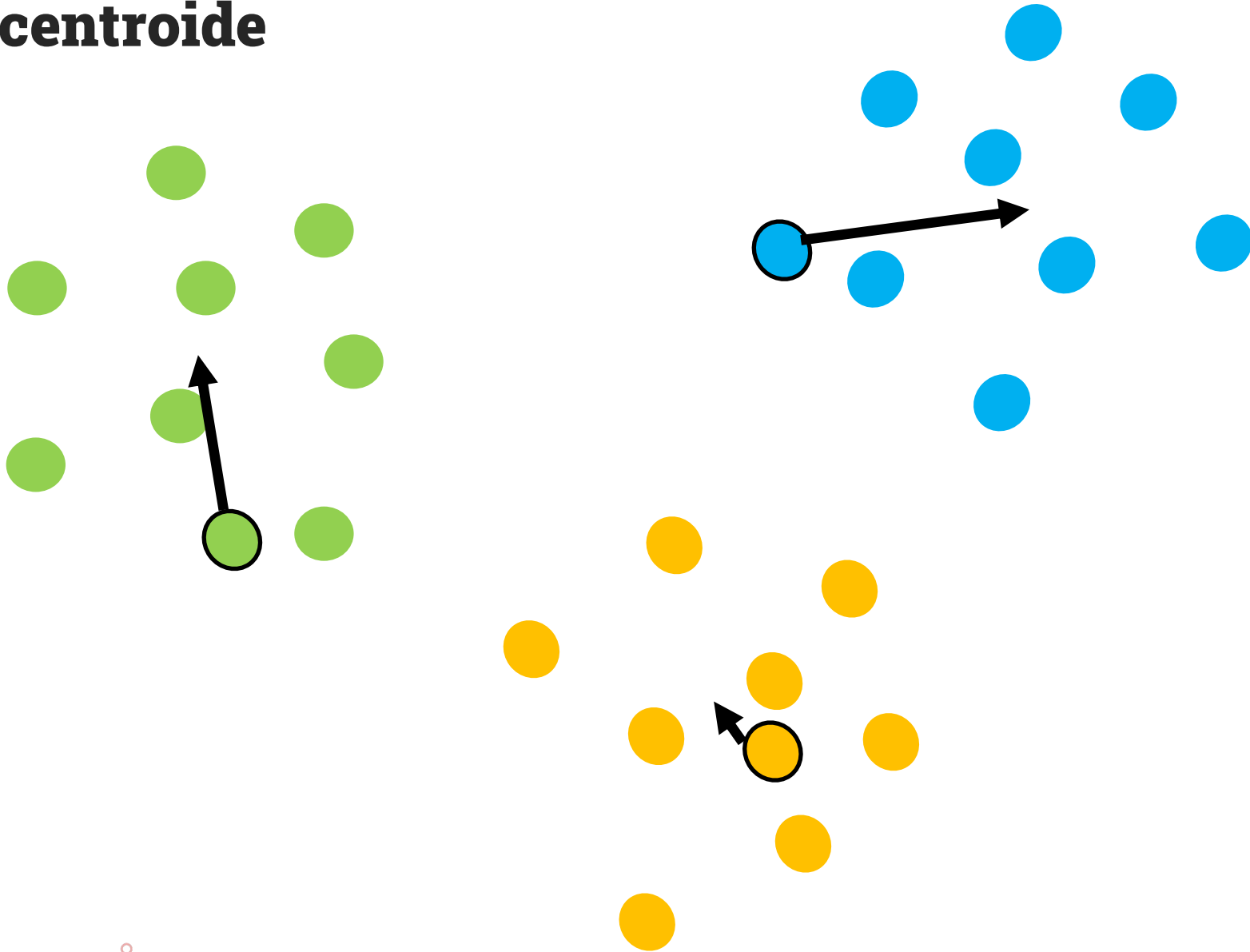
2 - Per ogni punto: Assegnare il punto al centroide più vicino



$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

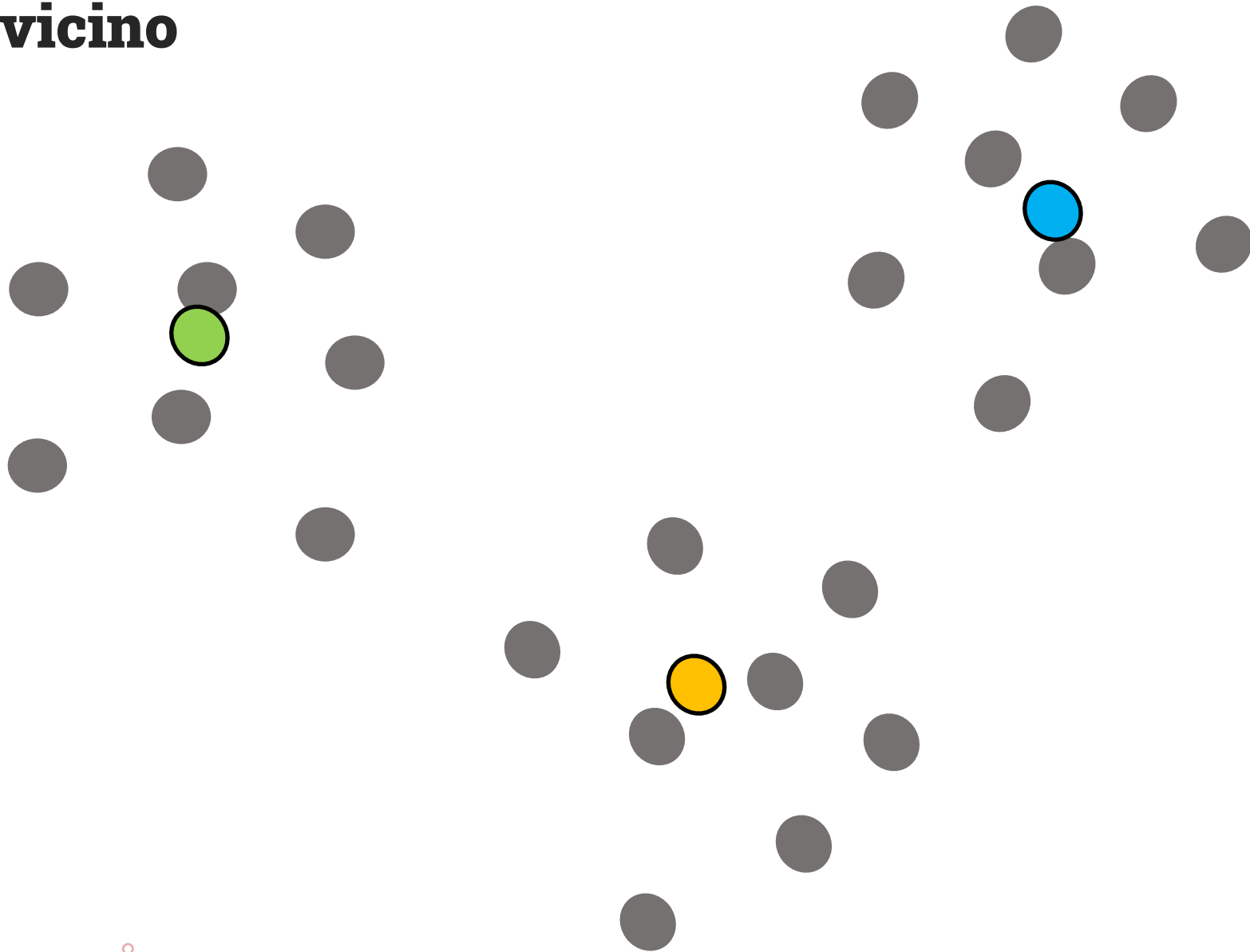
3 - Per ogni centroide: Aggiornare la posizione del centroide



$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

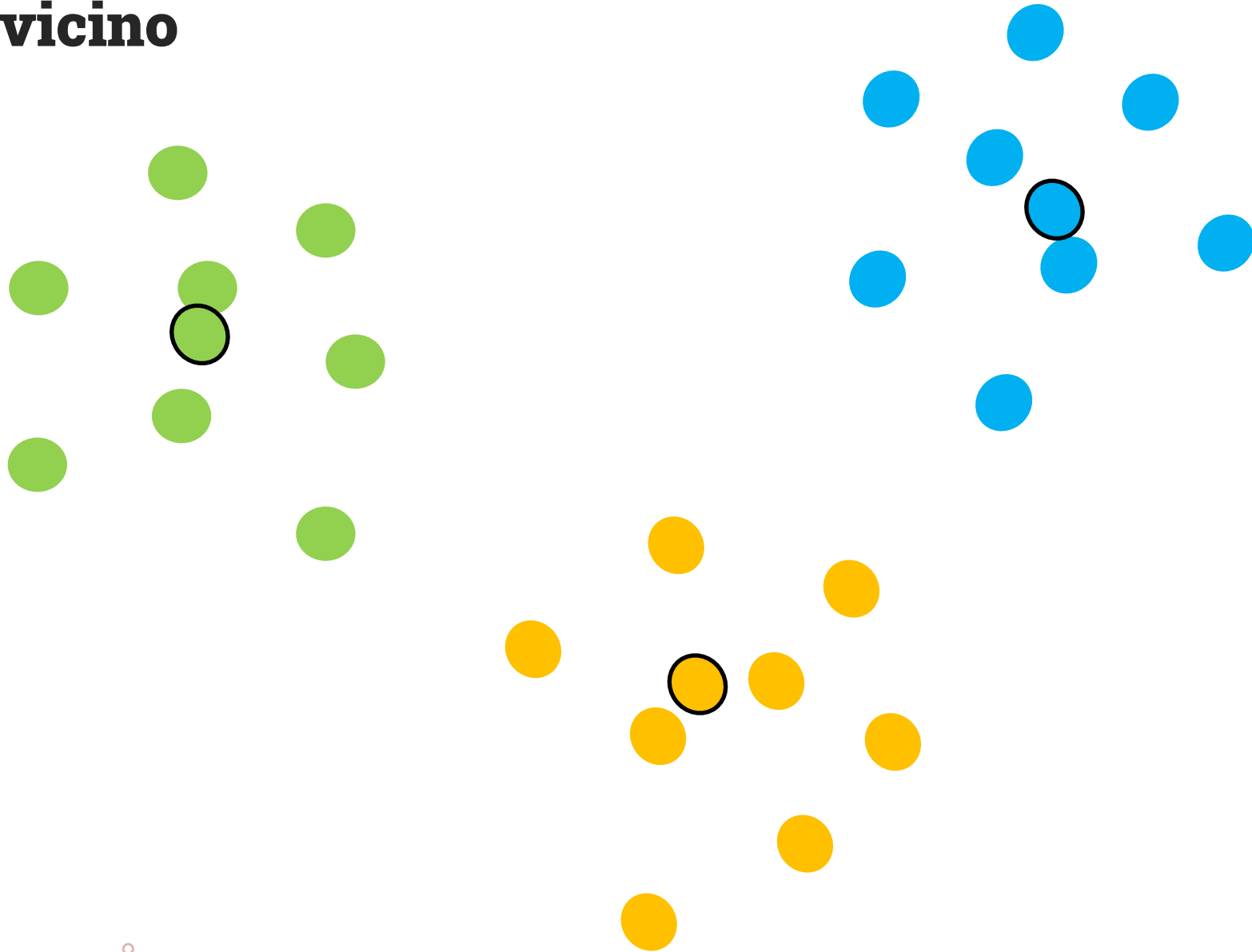
2 - Per ogni punto: Assegnare il punto al centroide più vicino



$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

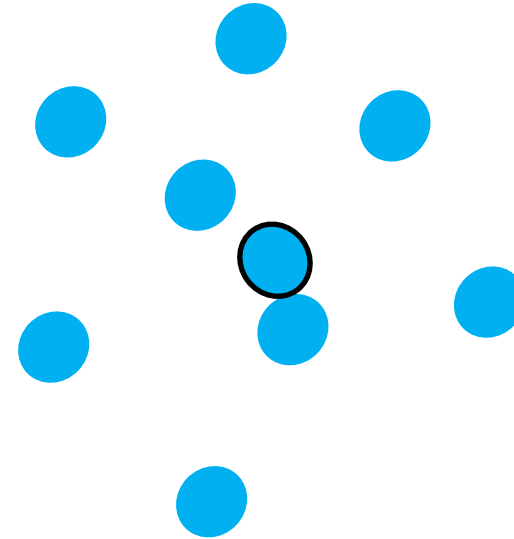
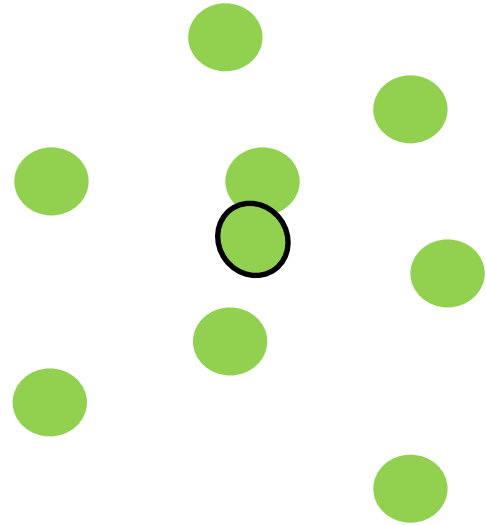
2 - Per ogni punto: Assegnare il punto al centroide più vicino



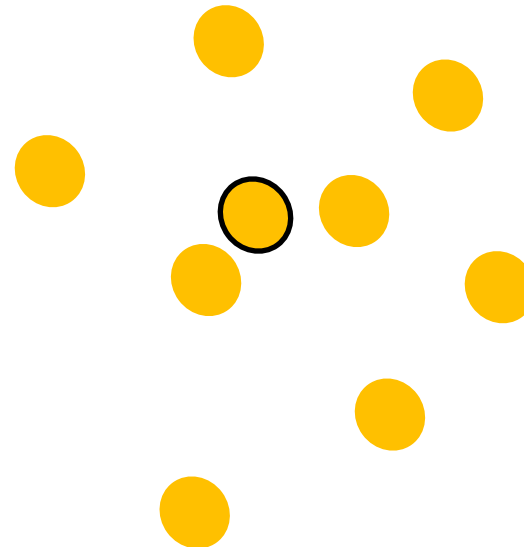
$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

3 - Per ogni centroide: Aggiornare la posizione del centroide



Quando ci fermiamo?



$$S = \{x_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^2$$

Algoritmo: implementazione (1/2)

```
def K-Means(X, centers, maxiter):
```

```
    # X: n x d
```

```
    # centers : k x d
```

```
    n, d = X.shape
```

```
    k = centers.shape[0]
```

```
    for i in range(maxiter):
```

```
        # Compute Squared Euclidean distance (i.e. the squared
```

```
        # distance) between each cluster centre and each observation
```

```
        dist = all_distances(X, centers)
```

Centers: sono k punti estratti a caso (alternativa: k viene passato in input e l'estrazione dei k centroidi iniziali viene fatta nella funzione stessa)

Alternativa a **maxiter**?

Algoritmo: implementazione (2/2)

Assign data to clusters:

for each point, find the closest center in terms of euclidean

distance

`c_ass = np.argmin(dist, axis=1)`

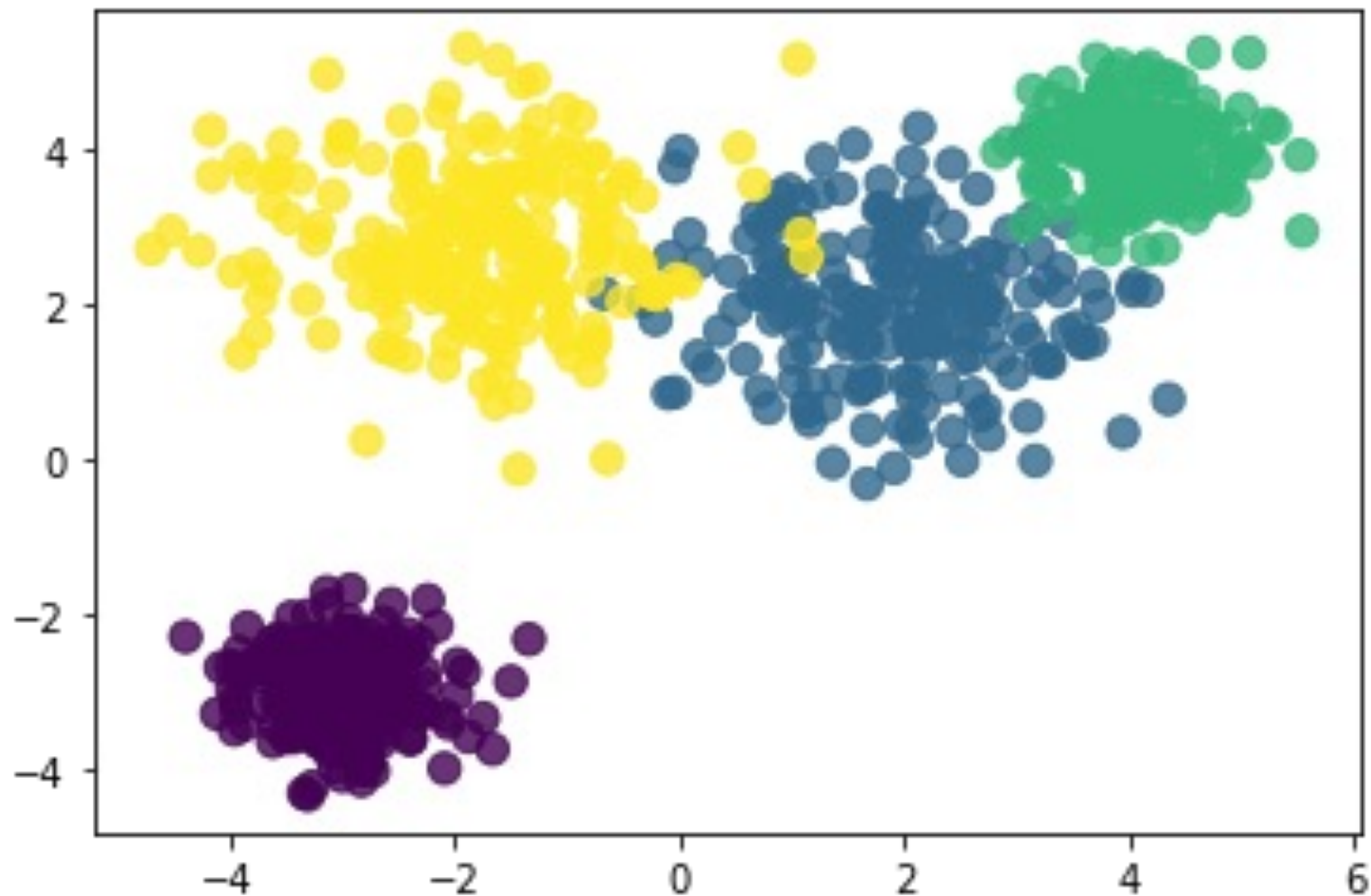
Update cluster center

for `c` in `range(k)`:

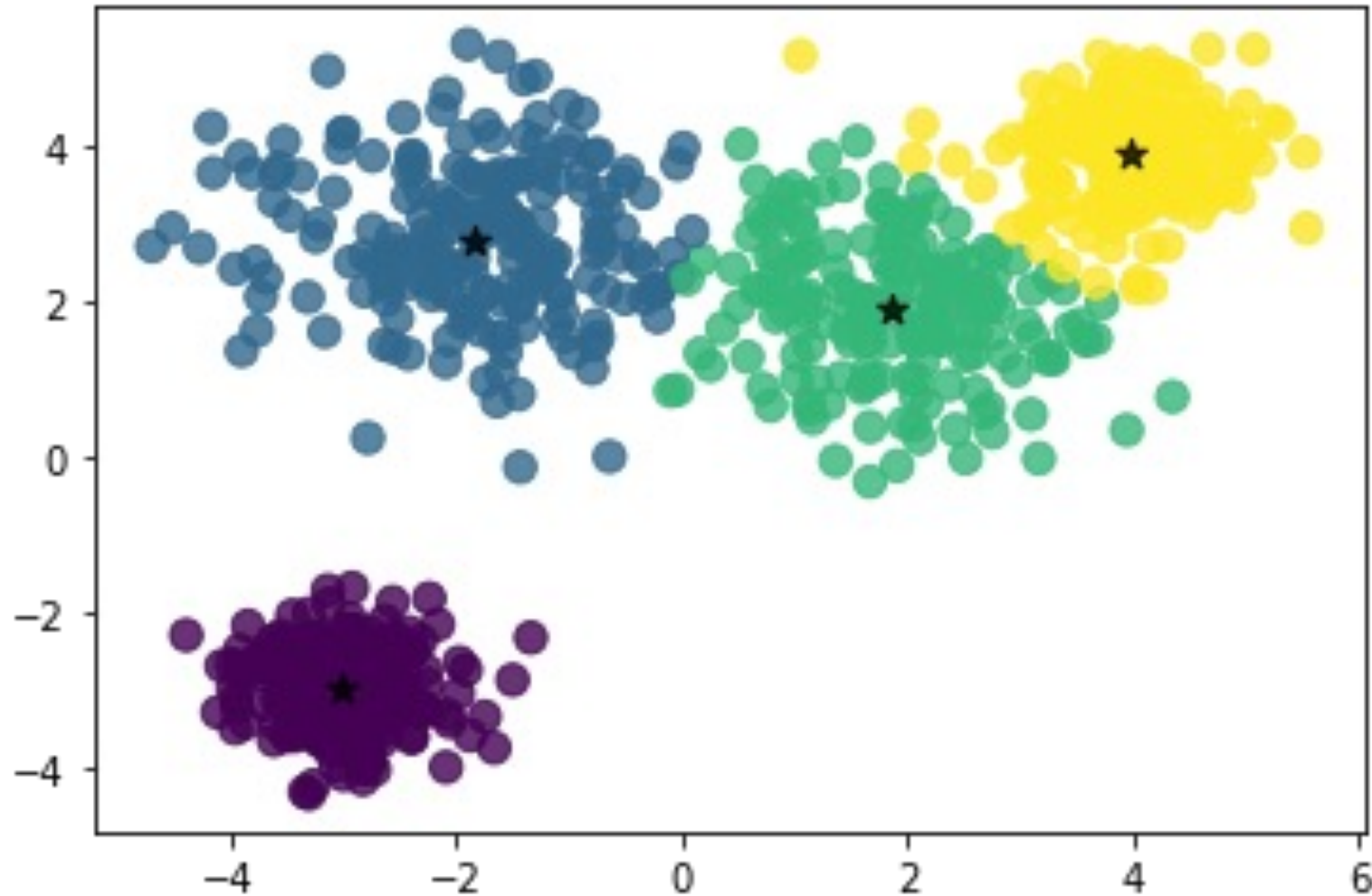
`centers[c] = np.mean(X[c_ass == c], axis=0)`

return `c_ass`, `centers`

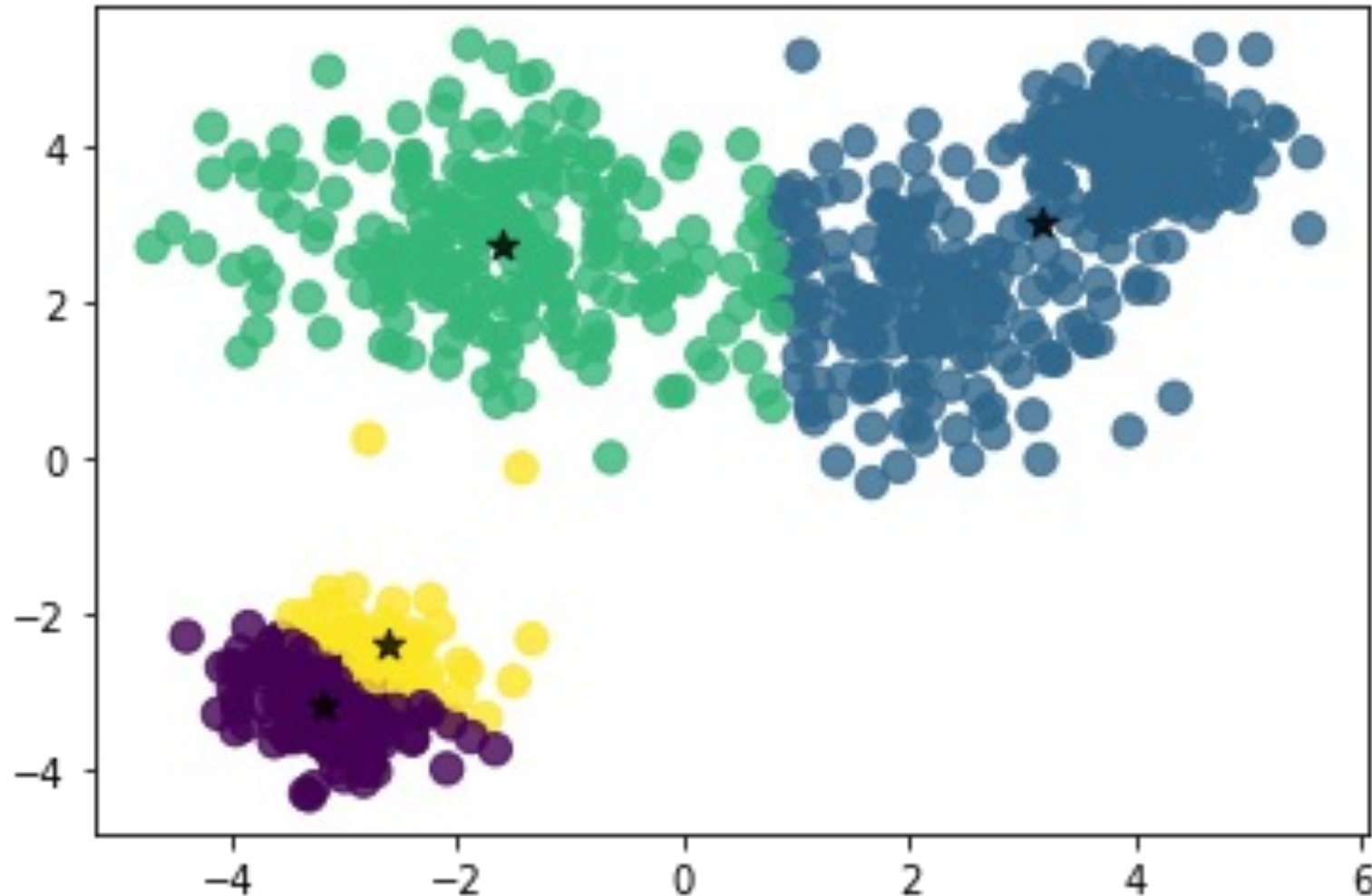
Cerchiamo di capire meglio



Cerchiamo di capire meglio: $K=4$

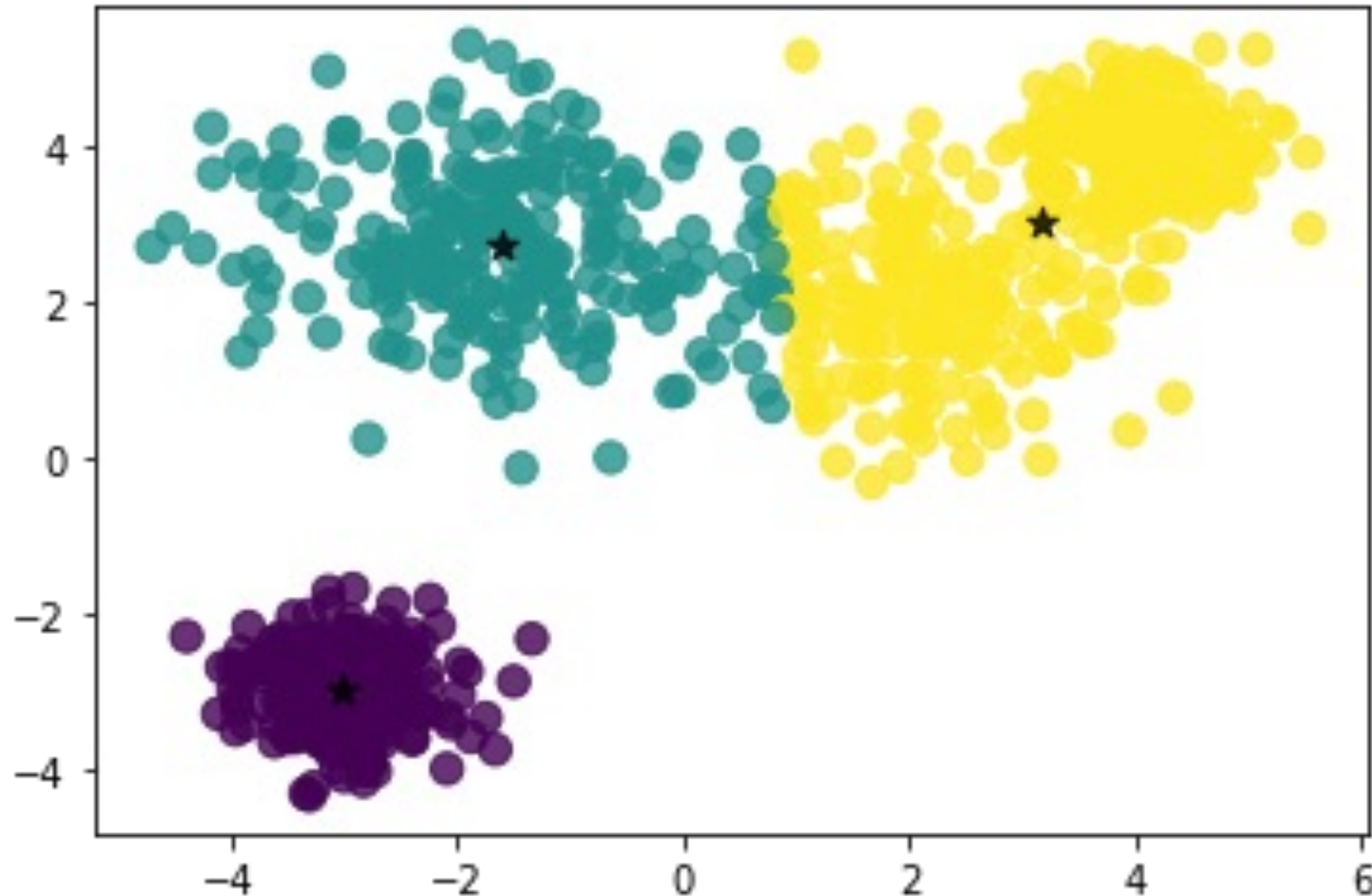


Cerchiamo di capire meglio: $K=4$



L'inizializzazione random dei centroidi può portare a risultati non soddisfacenti

Cerchiamo di capire meglio: la scelta di K



La scelta del numero di cluster è critica!! In generale potremmo non sapere quanti siano e potrebbe non essere possibile visualizzare i dati...

Valutare la qualità del risultato

Il coefficiente di silhouette

Agendo in un contesto non supervisionato non abbiamo a disposizione un insegnante che ci aiuti a valutare la bontà del risultato ottenuto

Utilizziamo delle misure per capire quanto sono belli i cluster che abbiamo individuato

Uno di questi è l'indice di Silhouette

Valutare la qualità del risultato


Il coefficiente di silhouette

*Distanza media
EXTRA-CLUSTER*

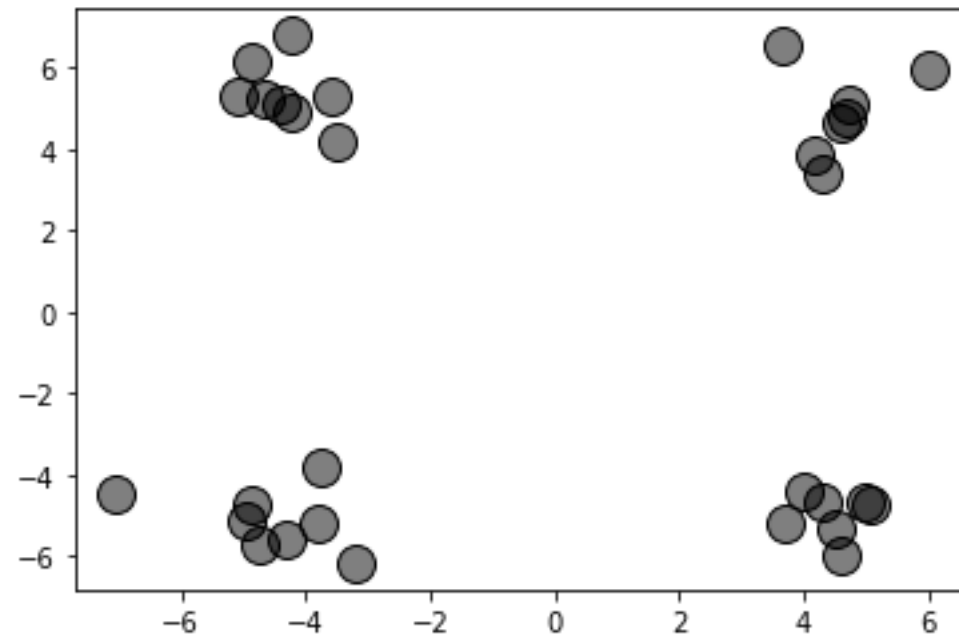
*Siamo contenti
se è alta*

*Distanza media
INTRA-CLUSTER*

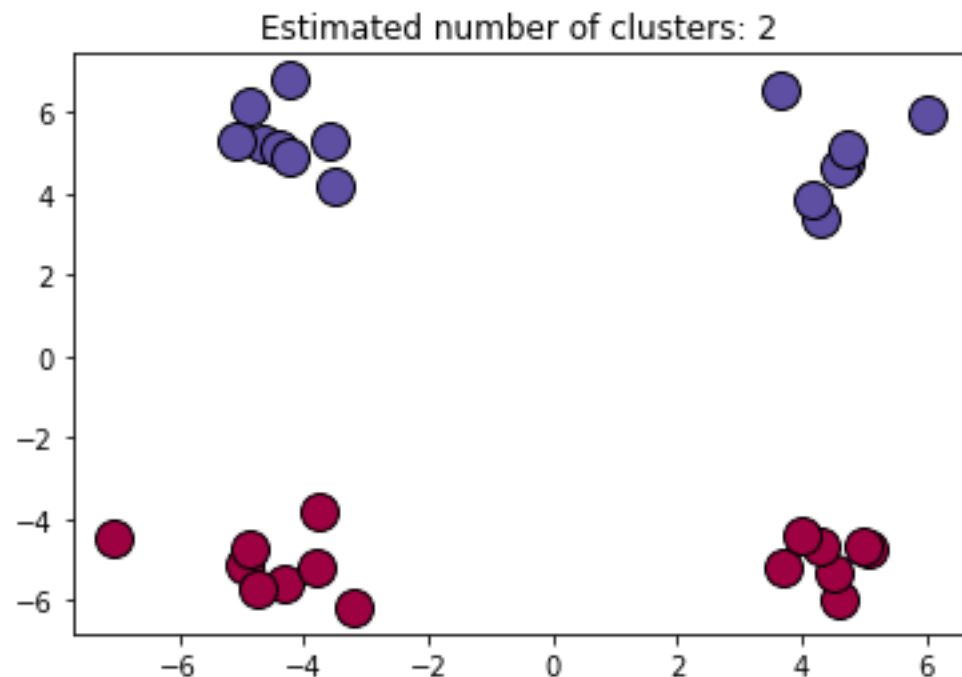
Siamo contenti se è bassa


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

Esempio



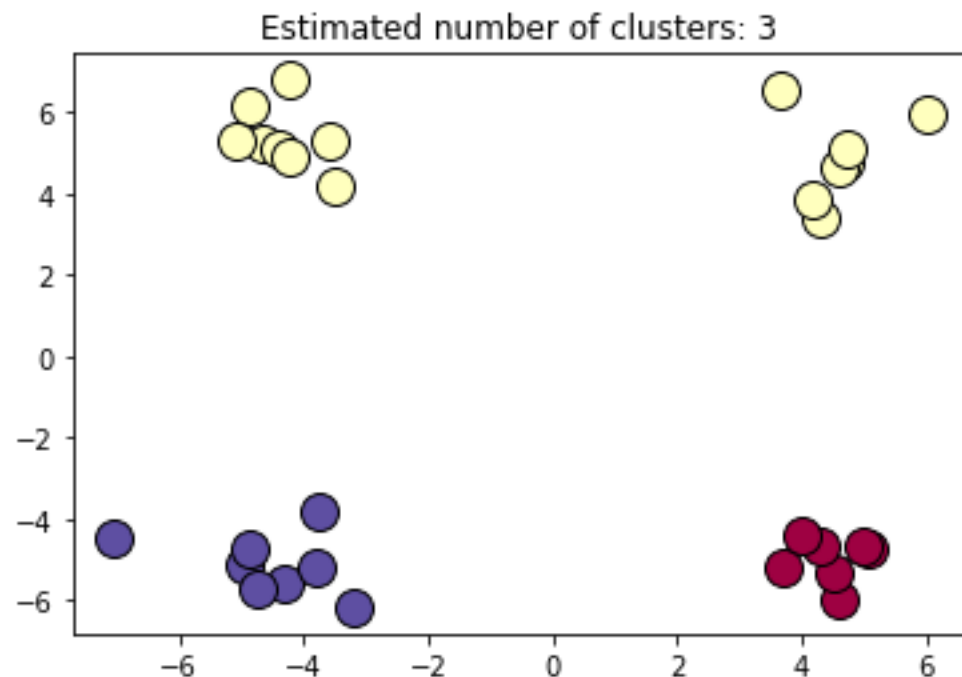
Esempio: K=2



$$SC = 0.54$$

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

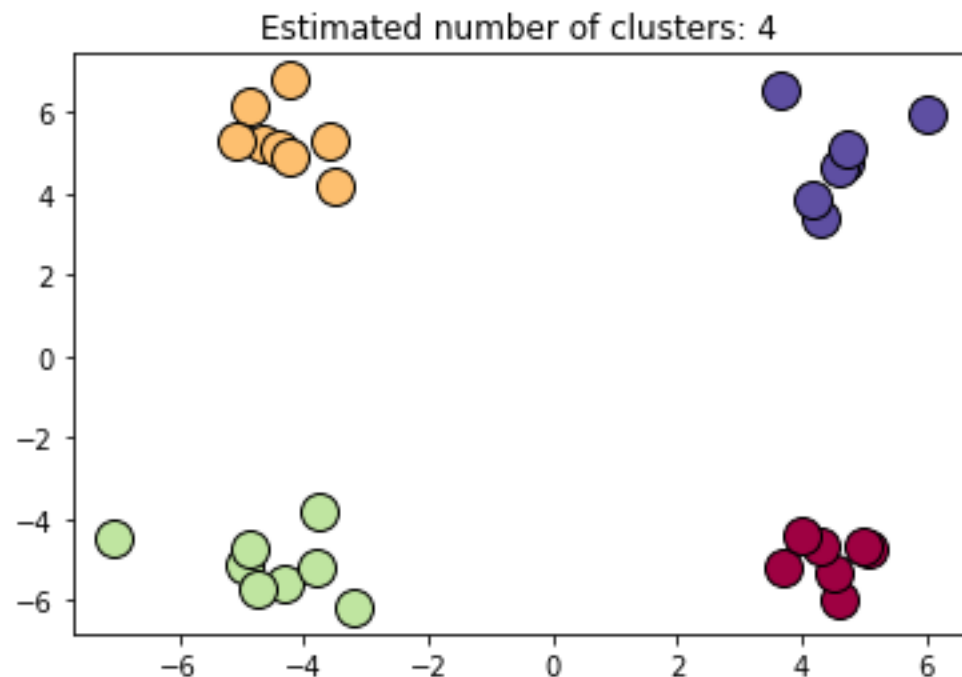
Esempio: K=3



$$SC = 0.66$$

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

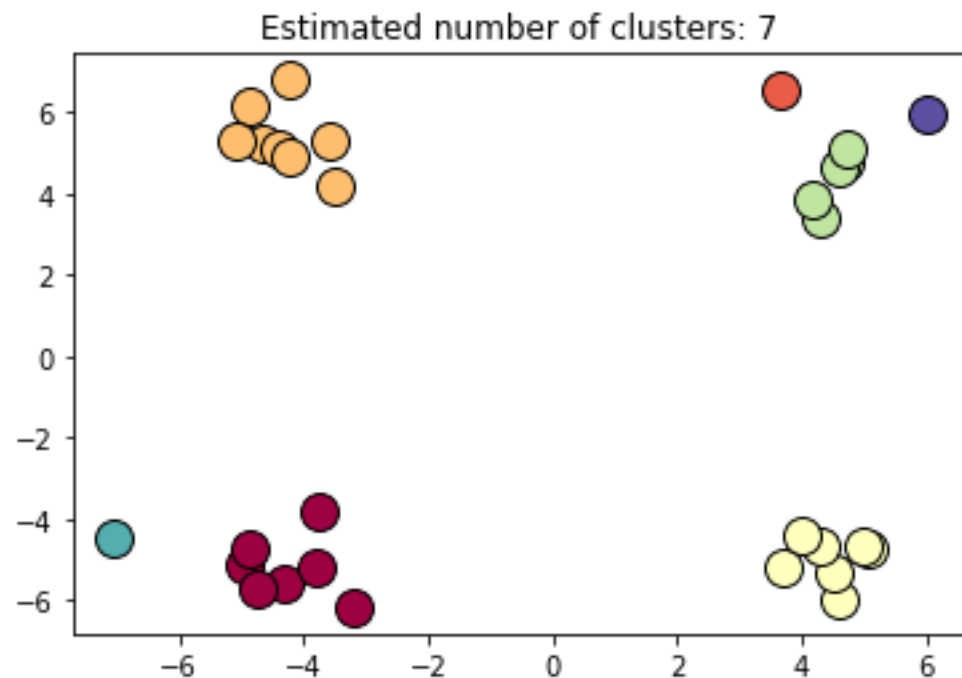
Esempio: K=4



$$SC = 0.84$$

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

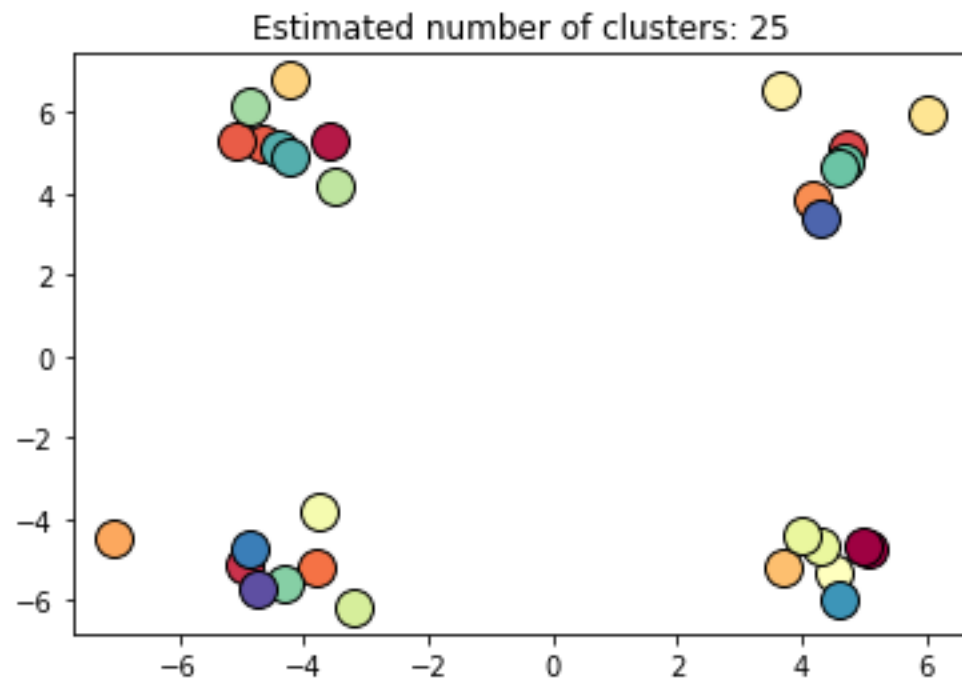
Esempio: K=7



$$SC = 0.65$$

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

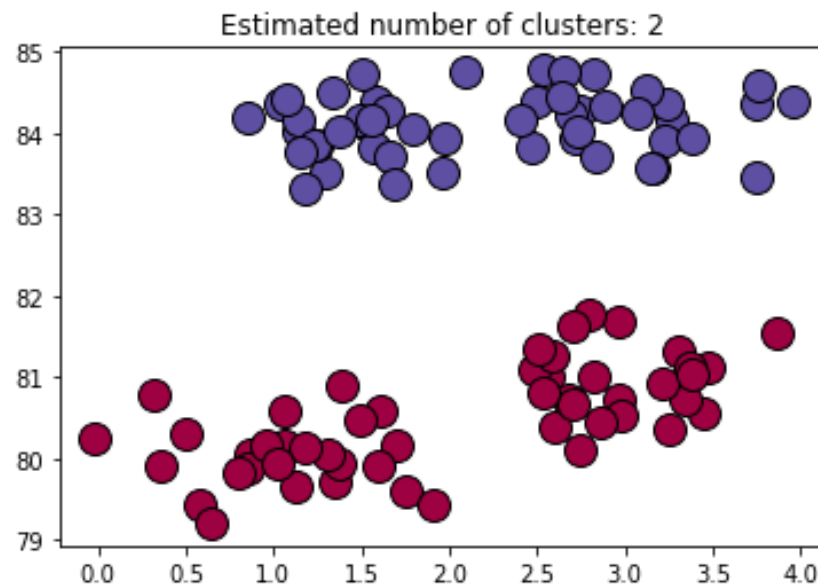
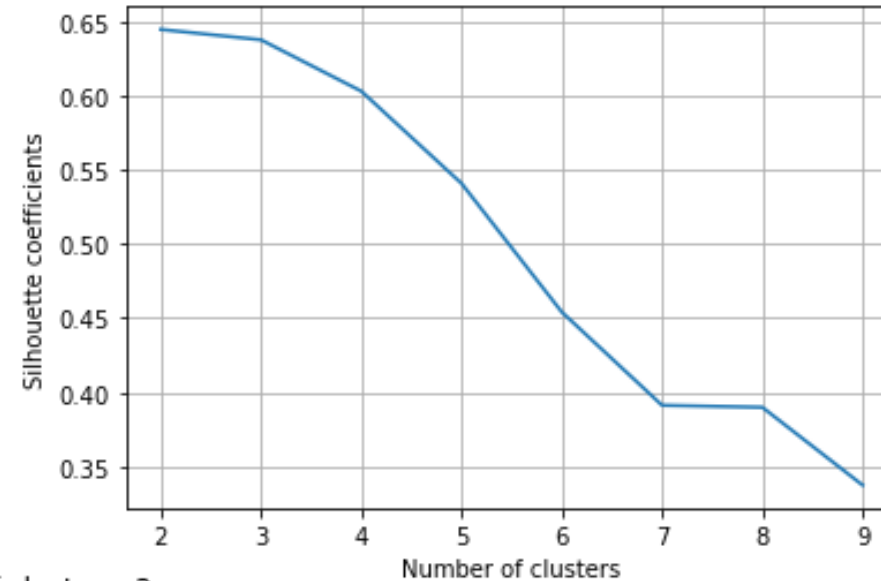
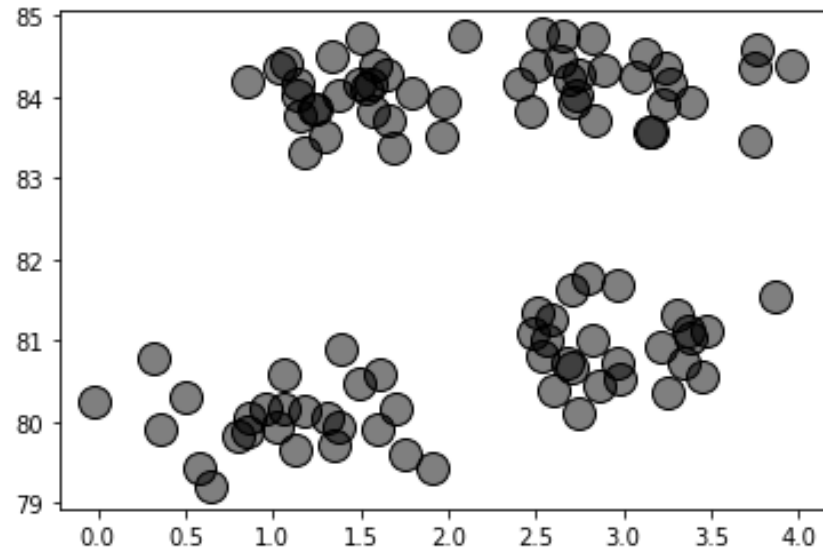
Esempio: K=25



$$SC = 0.18$$

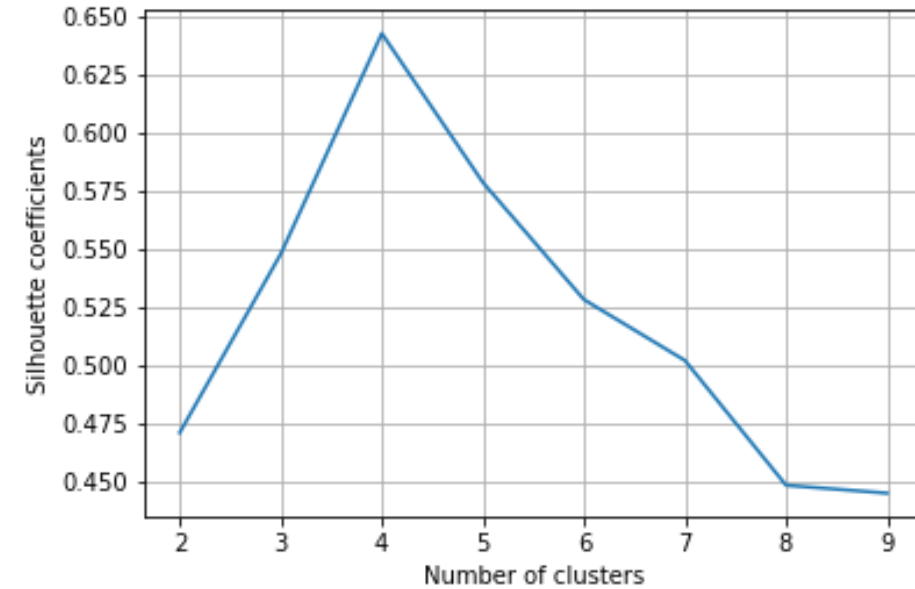
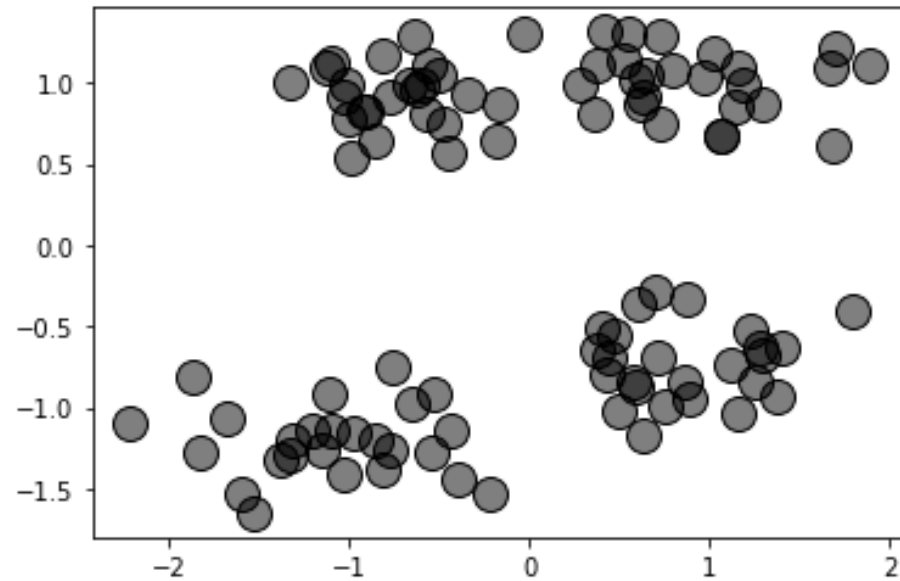
$$SC = \frac{b - a}{\max(a, b)}$$

Valutare la qualità del risultato

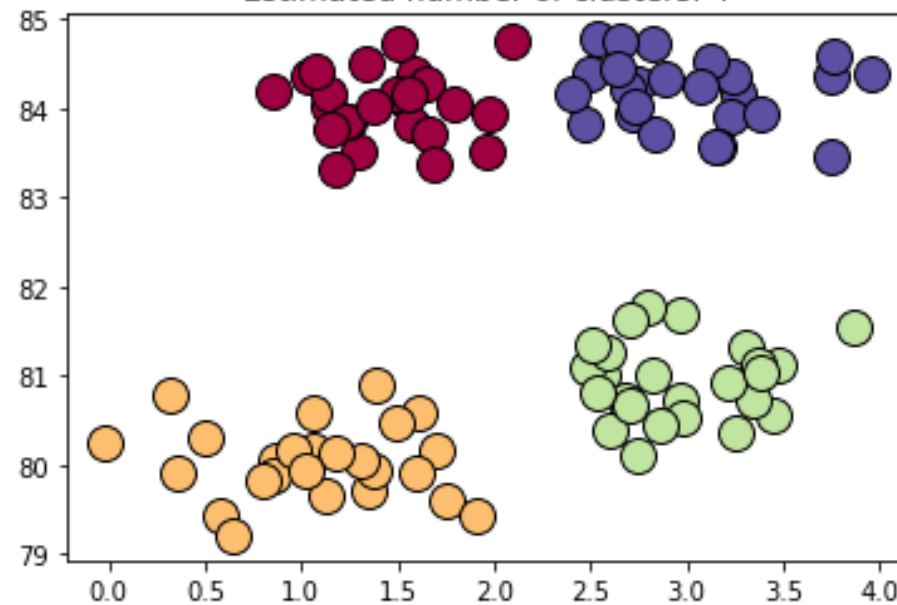


La standardizzazione dei dati è fondamentale!!!

Valutare la qualità del risultato



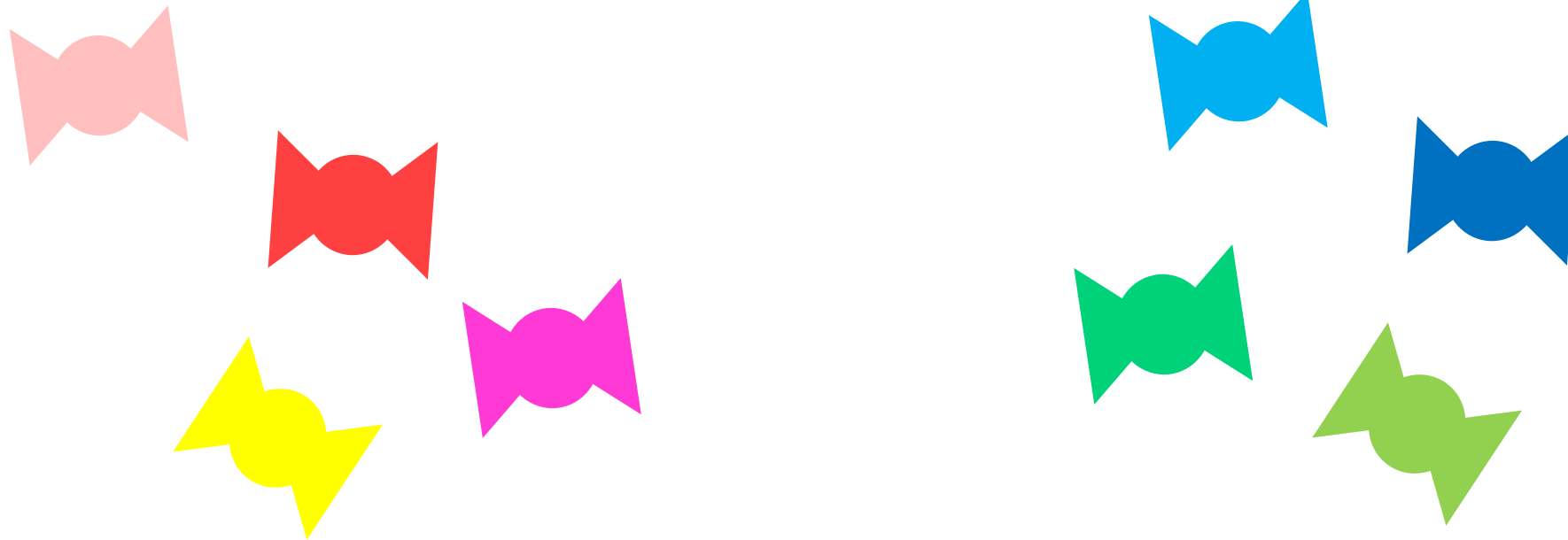
Estimated number of clusters: 4











Riduzione della dimensionalità

Un esempio

Consideriamo due insiemi di caramelle: spicy e dolci









Dati e rappresentazioni

| | Colore caldo | Colore freddo |
|-------------------------------------------------------------------------------------|-----------------|------------------|
|  | 1 | 0 |
|  | 1 | 0 |
|  | 1 | 0 |
|  | 1 | 0 |
|  | 0 | 1 |
|  | 0 | 1 |
|  | 0 | 1 |
|  | 0 | 1 |

Possiamo descriverle in base al colore della confezione...

Dati e rappresentazioni

| | Colore caldo | Colore freddo |
|-------------------------------------------------------------------------------------|--------------|---------------|
|  | 1 | 0 |
|  | 1 | 0 |
|  | 1 | 0 |
|  | 1 | 0 |
|  | 0 | 1 |
|  | 0 | 1 |
|  | 0 | 1 |
|  | 0 | 1 |









Possiamo descriverle in base al colore della confezione...

Ora immaginiamo che arrivi un dato di test



Come lo classifichereste?

Dati e rappresentazioni

| | Colore caldo | Colore freddo |
|-------------------------------------------------------------------------------------|--------------|---------------|
|  | 1 | 0 |
|  | 1 | 0 |
|  | 1 | 0 |
|  | 1 | 0 |
|  | 0 | 1 |
|  | 0 | 1 |
|  | 0 | 1 |
|  | 0 | 1 |

Cosa succede se vogliamo essere più precisi nella descrizione del colore?

Dati e rappresentazioni

| | Rosa | Rosso | Giallo | Magenta | Azzurro | Blu | Verde chiaro | Verde acqua |
|-------------------------------------------------------------------------------------|------|-------|--------|---------|---------|-----|--------------|-------------|
|  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Dati e rappresentazioni

| | Rosa | Rosso | Giallo | Magenta | Azzurro | Blu | Verde chiaro | Verde acqua |
|-------------------------------------------------------------------------------------|------|-------|--------|---------|---------|-----|--------------|-------------|
|  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Cosa succede con il dato di test di prima?



Dati e rappresentazioni

| | Rosa | Rosso | Giallo | Magenta | Azzurro | Blu | Verde chiaro | Verde acqua |
|-------------------------------------------------------------------------------------|------|-------|--------|---------|---------|-----|--------------|-------------|
|  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Cosa succede con il dato di test di prima?



Come potremmo risolvere il problema?

Dati e rappresentazioni

Quando i dati vengono rappresentati con un numero di caratteristiche troppo alto rispetto al numero di dati potremmo incontrare la cosiddetta *curse of dimensionality*

Maggiore è il numero di caratteristiche usate per la rappresentazione di un dato, più i dati stessi risultano essere distanti gli uni dagli altri

Quando le caratteristiche sono troppe rispetto alle reali necessità rischiamo di non migliorare la capacità descrittiva della rappresentazione, e anzi peggiorare i risultati

Curse of dimensionality

Quando la dimensionalità dei dati (D) cresce, il volume dello spazio dei dati cresce velocemente ed i dati diventano presto sparsi... Perchè i risultati siano affidabili il numero di dati (N) deve crescere esponenzialmente con la loro dimensionalità.

Ridurre la dimensionalità dei dati

Quando sospettiamo che la dimensione dei dati sia troppo alta rispetto alla loro quantità possiamo affidarci a tecniche di riduzione della dimensionalità, che hanno ulteriori effetti positivi

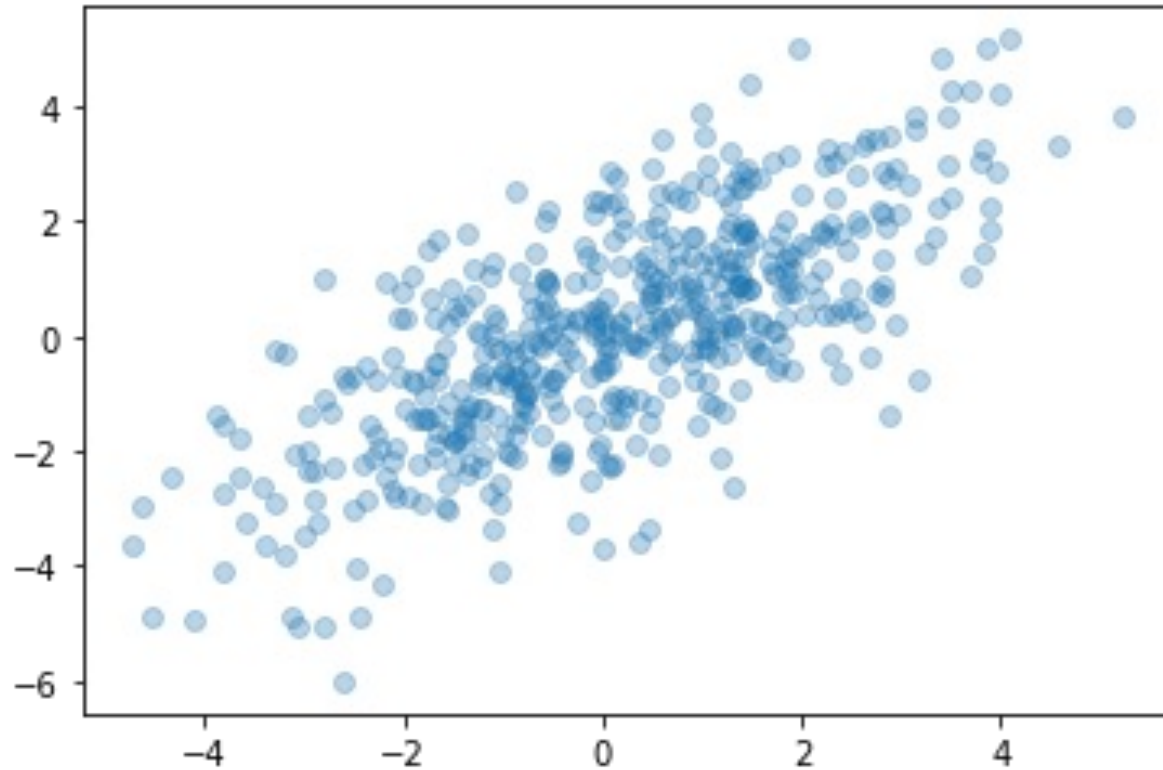
- Ci permettono di poter visualizzare i dati
- Ci permettono di interpretare meglio i dati

Proprietà desiderabili per buone rappresentazioni

Se avessimo la possibilità di progettare una rappresentazione quali sono le proprietà che ci piacerebbe avesse?

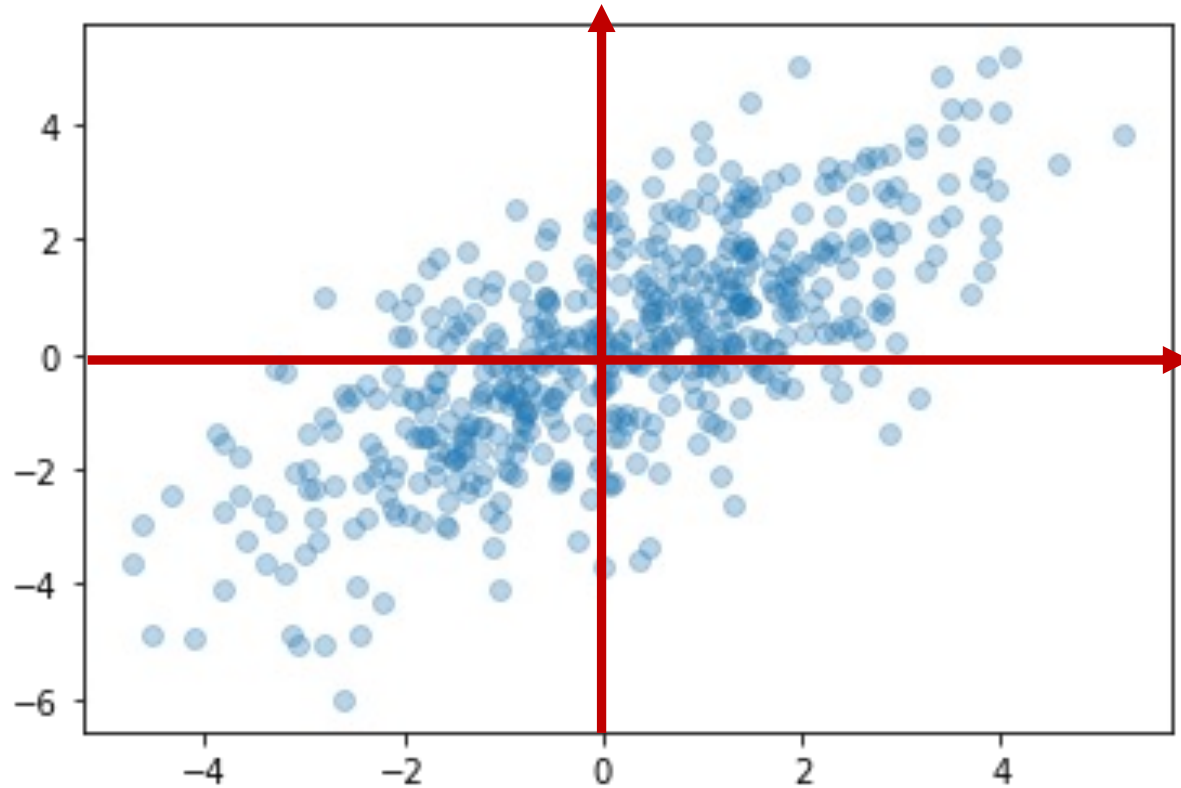
- Varianza alta: caratteristiche con varianza alta contengono «molto segnale»
- Non correlazione: caratteristiche correlate le une alle altre sono ridondanti e poco informative
- Non troppe: deve essere sempre un buon bilanciamento tra numero di dati e dimensionalità

Analisi delle componenti principali (PCA)



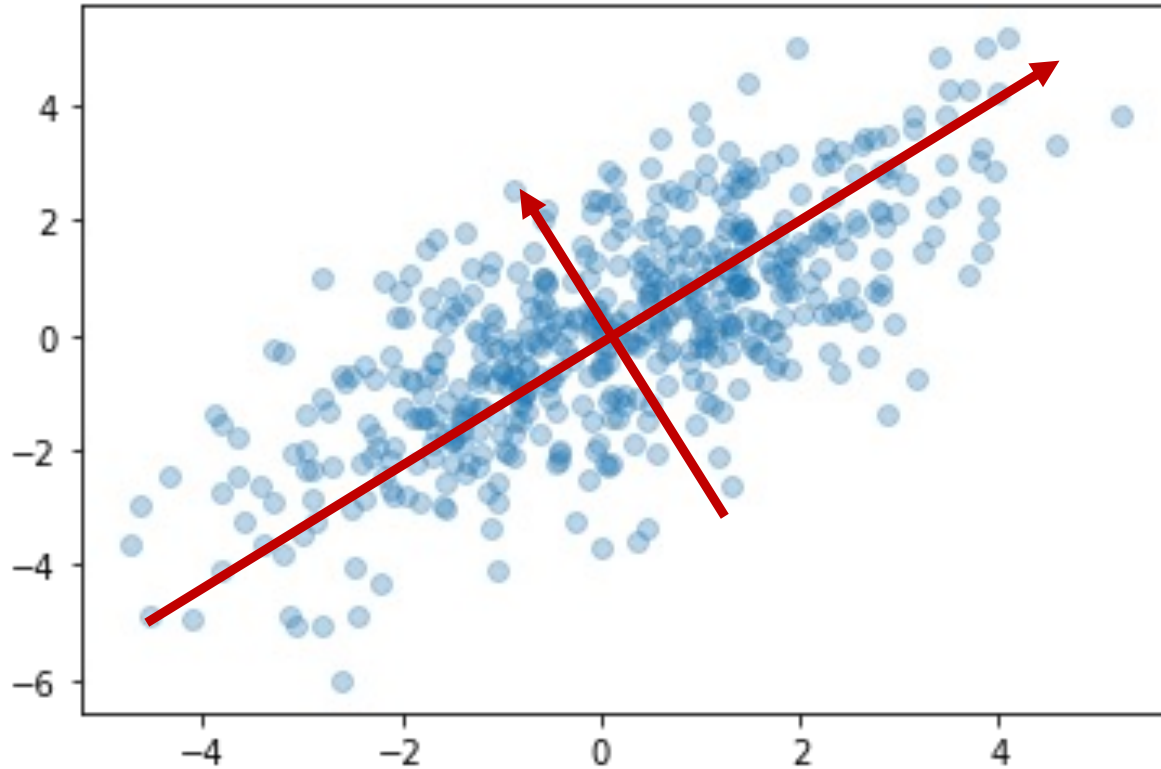
Esiste un modo per «guardare meglio» questi dati?

Analisi delle componenti principali (PCA)



Esiste un modo per «guardare meglio» questi dati?

Analisi delle componenti principali (PCA)



Esiste un modo per «guardare meglio» questi dati?

Cerchiamo un nuovo sistema di riferimento in cui sia più facile capire qualcosa sui nostri dati...

Analisi delle componenti principali (PCA)

- Si tratta di un algoritmo che possiamo applicare a matrici di dati di qualunque dimensione $N \times D$
- Consiste nell'identificare una nuova base le cui componenti catturano quanta più varianza possibile dai dati originali
- Un ingrediente chiave è la decomposizione in valori singolari

Parentesi: SVD

Data una matrice X la sua decomposizione in valori singolari è

$$X = U\Sigma V^T$$

dove

- U è una matrice ortogonale (se X è una matrice reale)
- Σ è una matrice diagonale, con elementi non negativi in ordine decrescente
- V è una matrice ortogonale (se X è una matrice reale)

Parentesi: SVD

- U e V sono matrici aventi come colonne vettori ortonormali, detti, rispettivamente, vettori singolari sinistri e destri di X
- Gli elementi diagonali di Σ sono detti valori singolari di X

Inoltre

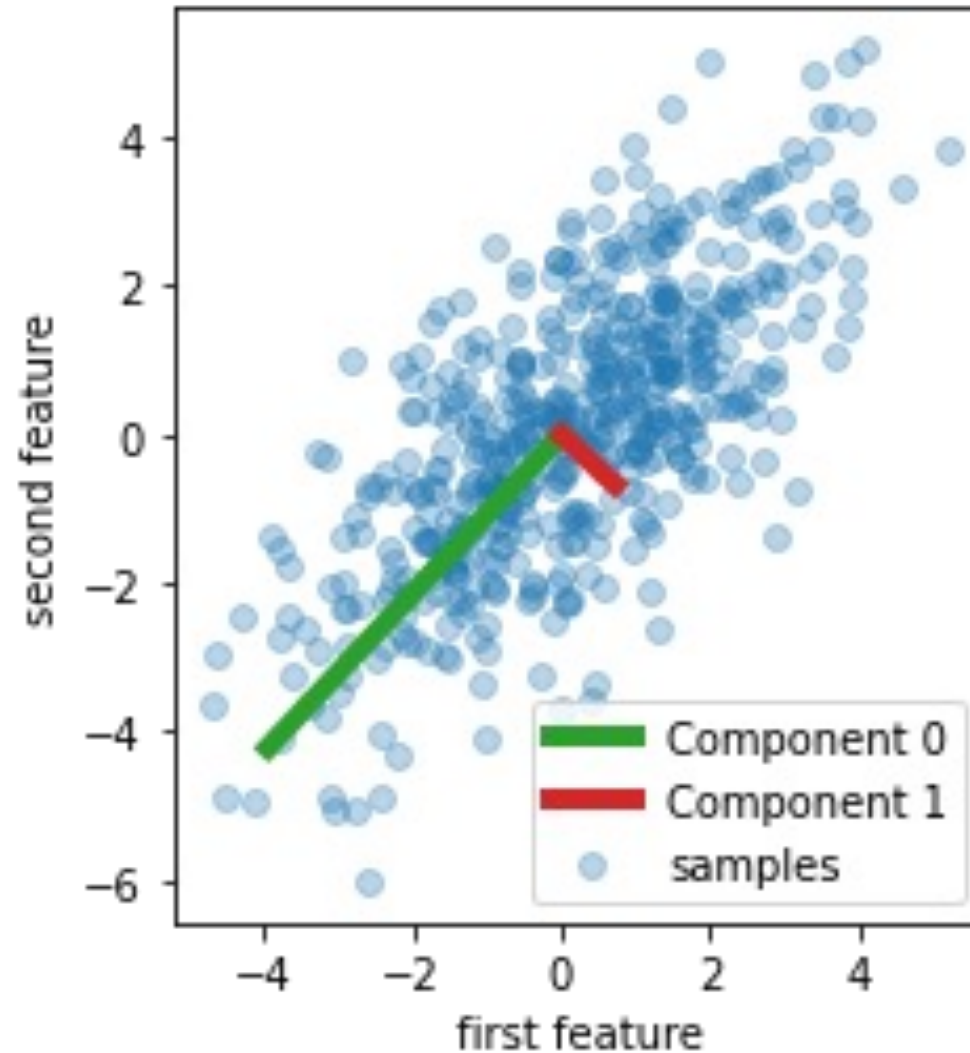
- Se X è reale e quadrata, i valori singolari di X sono le radici quadrate positive degli autovalori di $X^T X$ e XX^T

PCA: algoritmo

- Costruiamo la matrice di covarianza dei dati $C = X^T X$ *Ricordate cos'è la matrice di covarianza?*
- Calcoliamo la sua SVD $C = U \Sigma V^T$
- Identifichiamo la nuova base: le prime K colonne di V, se vogliamo utilizzare K componenti principali

Analisi delle componenti principali (PCA)

2-dimensional dataset with principal components



«Varianza spiegata»

```
def explainedVariance(eig_vals):
```

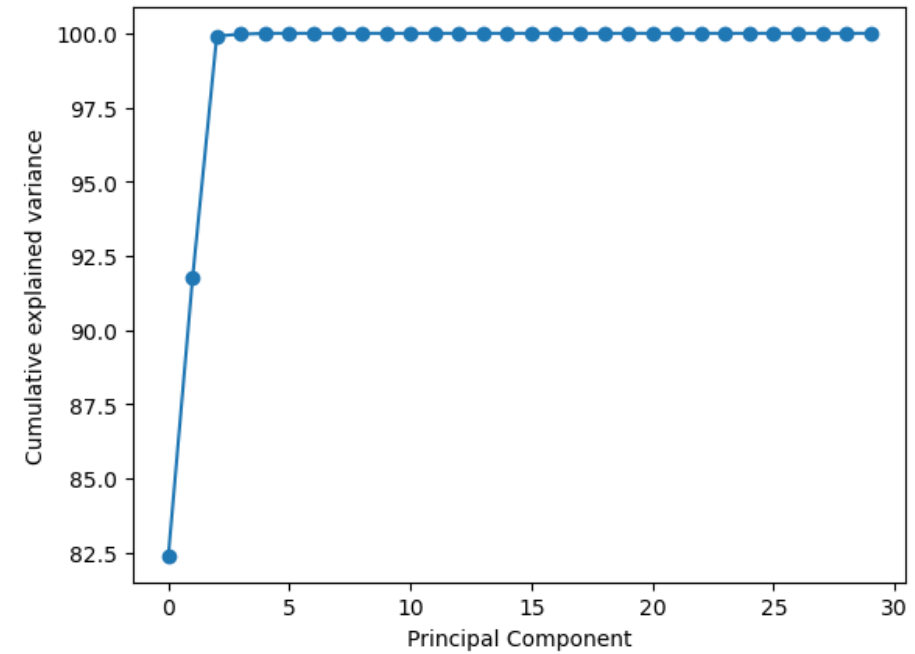
```
    tot = sum(eig_vals)
```

```
    var_exp = [(i / tot)*100 for i in eig_vals]
```

```
    cum_var_exp = np.cumsum(var_exp)
```

```
    return cum_var_exp
```

Esempio con vettori di 100 dimensioni, di cui solo 3 rilevanti



UniGe

