# K-means Clustering: Choosing k

El Kindi Rezig

# Clustering (refresher)

- In order to do clustering, we need to have two things:
  - The input dataset to be clustered $X = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^d$
  - A distance function $d(x_i, x_j) \geq 0$ that can tell us how similar two points are
  - In this class, $d(x_i, x_j) = \|x_i - x_j\|$ (Euclidean distance)

# Clustering (refresher)

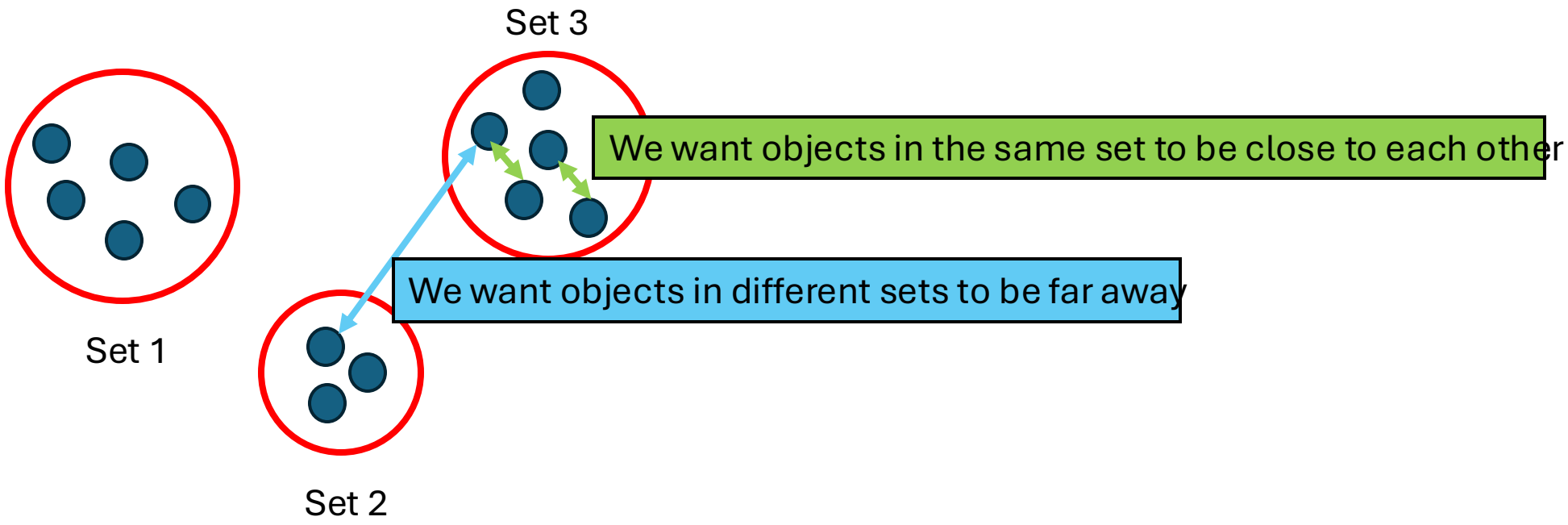
- Goal of clustering is to group objects into k sets (or clusters)
  - Objects in the same set (or cluster) are close to each other

# Clustering (refresher)

- Goal of clustering is to group objects into k sets (or clusters)
  - Objects in the same set (or cluster) are close to each other
  - Objects in different sets (or clusters) are far away from each other

# Clustering (refresher)

- Goal of clustering is to group objects into k sets (or clusters)
  - Objects in the same set (or cluster) are close to each other
  - Objects in different sets (or clusters) are far away from each other



# Clustering – Cost function (refresher)

- In mathematical terms, for a set of sites $S = \{s_1, s_2, .., s_k\} \subset \mathbb{R}^d$ and a dataset $X \subset \mathbb{R}^d$ we want to minimize:
  - $cost_2(X, S) = \sum_{i=1}^{n} \left(x_i - \phi_s(x_i)\right)^2$
  - $\phi_s(x_i) = argmin_{s_j \in S} \|x_i - s_j\|$ is the site $s_j$ that is closest to $x_i$

Can be thought of as the projection of $x_i$ onto closest site $s_j$

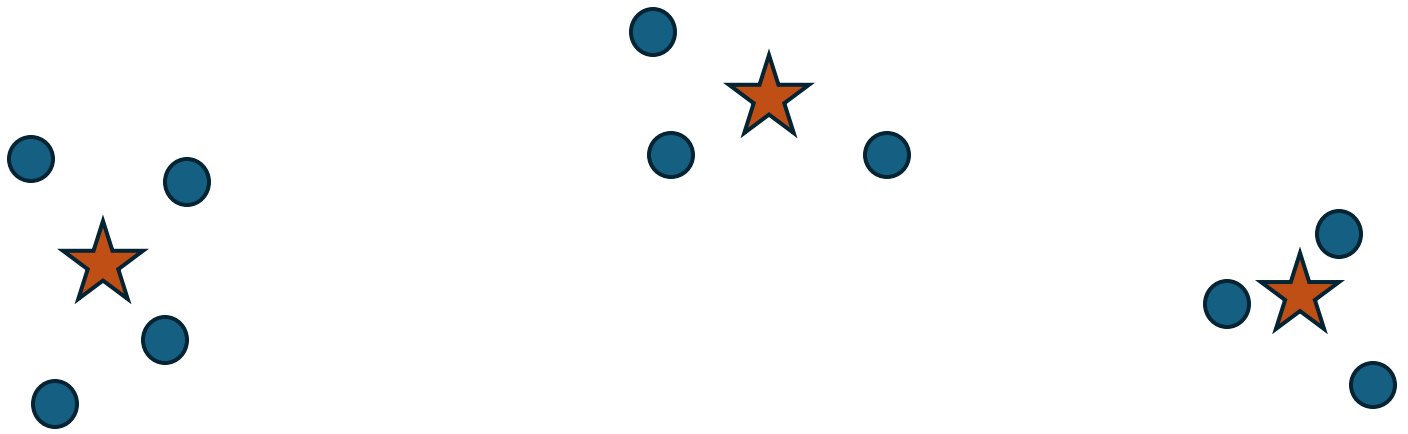
# Clustering – Cost function (refresher)

- In mathematical terms, for a set of sites $S = \{s_1, s_2, .., s_k\} \subset \mathbb{R}^d$ and a dataset $X \subset \mathbb{R}^d$ we want to minimize:
  - $cost_2(X, S) = \sum_{i=1}^{n} \left(x_i - \phi_s(x_i)\right)^2$
  - $\phi_s(x_i) = argmin_{s_j \in S} \|x_i - s_j\|$ is the site $s_j$ that is closest to $x_i$


Example. We have 3 site points (represented with stars). k = 3

# Clustering – Cost function (refresher)

- In mathematical terms, for a set of sites $S = \{s_1, s_2, .., s_k\} \subset \mathbb{R}^d$ and a dataset $X \subset \mathbb{R}^d$ we want to minimize:
  - $cost_2(X, S) = \sum_{i=1}^{n} \left(x_i - \phi_s(x_i)\right)^2$
  - $\phi_s(x_i) = argmin_{s_j \in S} \|x_i - s_j\|$ is the site $s_j$ that is closest to $x_i$



Data points (blue circles)

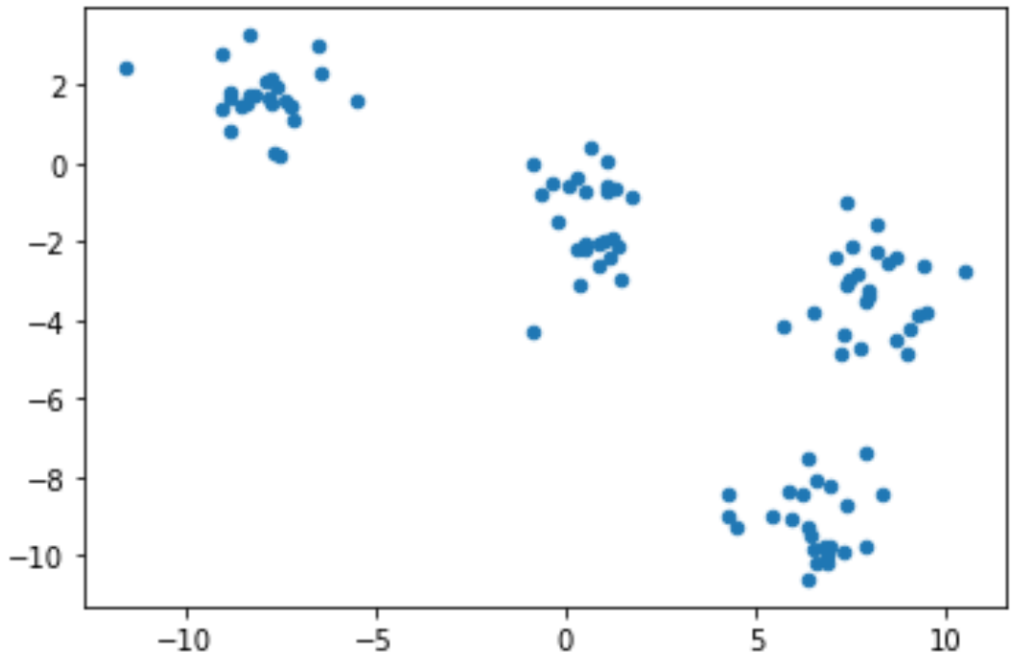
# Clustering – Cost function (refresher)

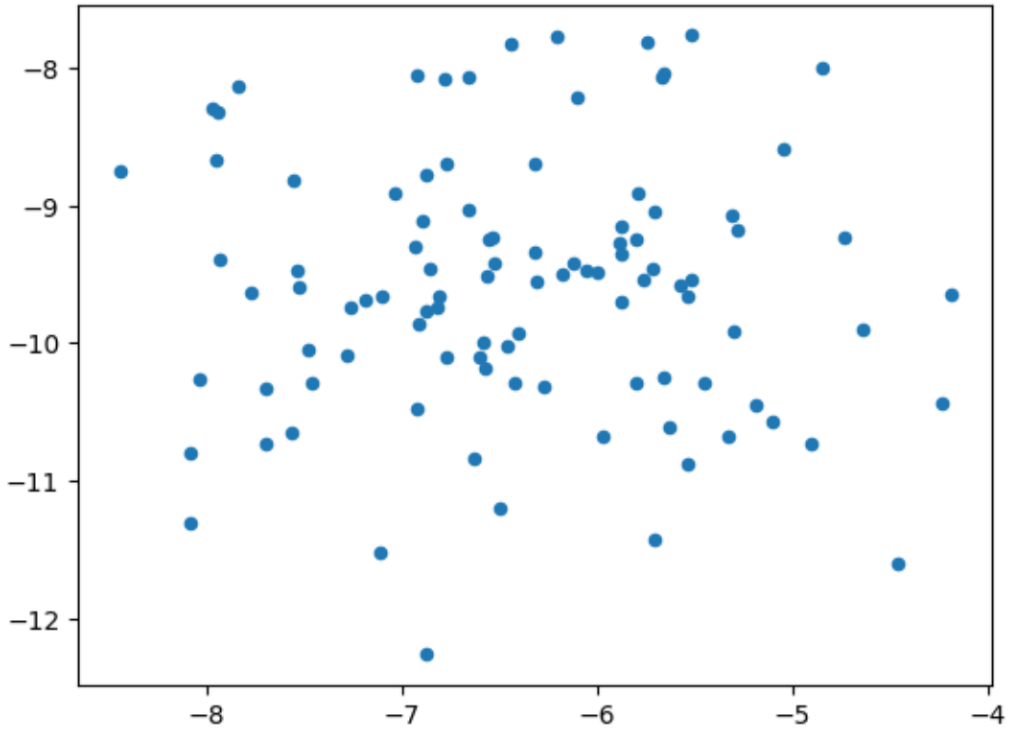
- In mathematical terms, for a set of sites $S = \{s_1, s_2, .., s_k\} \subset \mathbb{R}^d$ and a dataset $X \subset \mathbb{R}^d$ we want to minimize:
  - $cost_2(X, S) = \sum_{i=1}^{n} \left(x_i - \phi_s(x_i)\right)^2$
  - $\phi_s(x_i) = argmin_{s_j \in S} \|x_i - s_j\|$ is the site $s_j$ that is closest to $x_i$

Projecting data points onto closest site which is applying the projection functior $\phi_s(x_i)$ to every data point

# But how do we choose k?

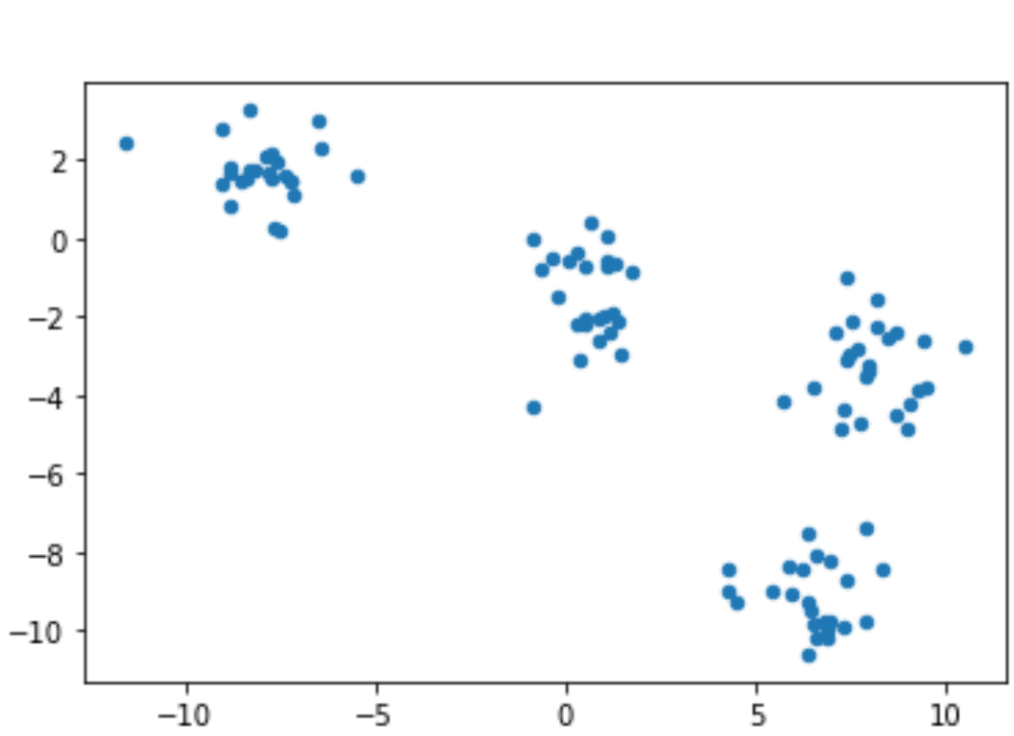
Whenever you can, especially when data $X \subset \mathbb{R}^2$, you should plot the data!
Consider the two data sets shown:

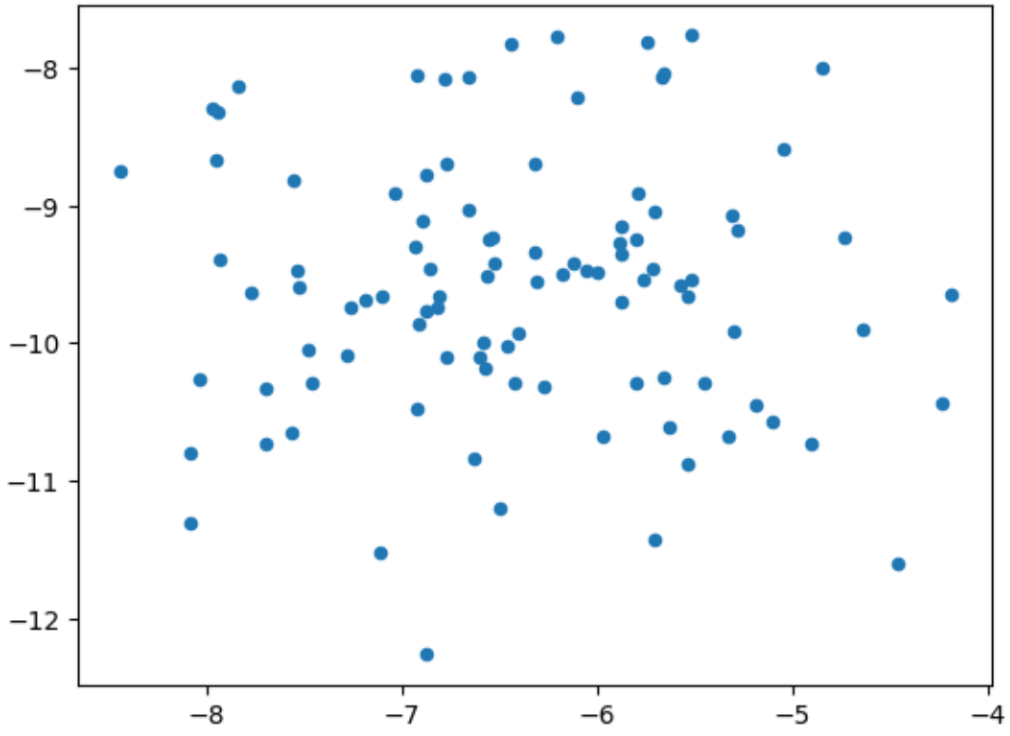


# But how do we choose k?

Whenever you can, especially when data $X \subset \mathbb{R}^2$, you should plot the data!
Consider the two data sets shown:

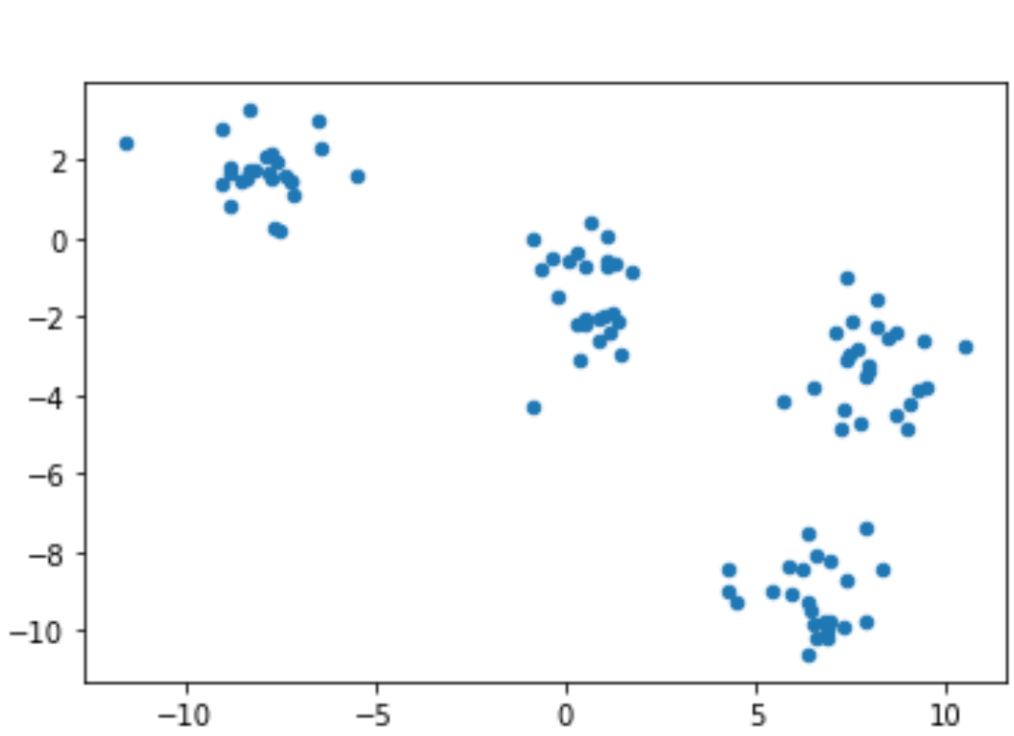


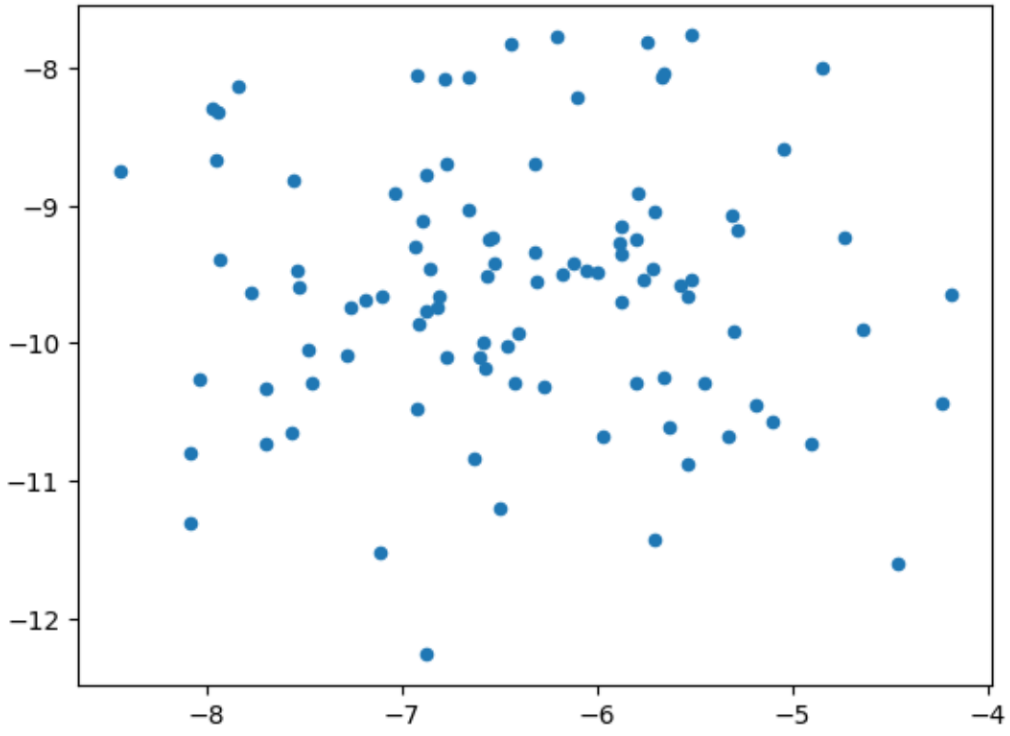




Which one shows ”better” clusters?

# But how do we choose k?

Whenever you can, especially when data $X \subset \mathbb{R}^2$, you should plot the data!
Consider the two data sets shown:

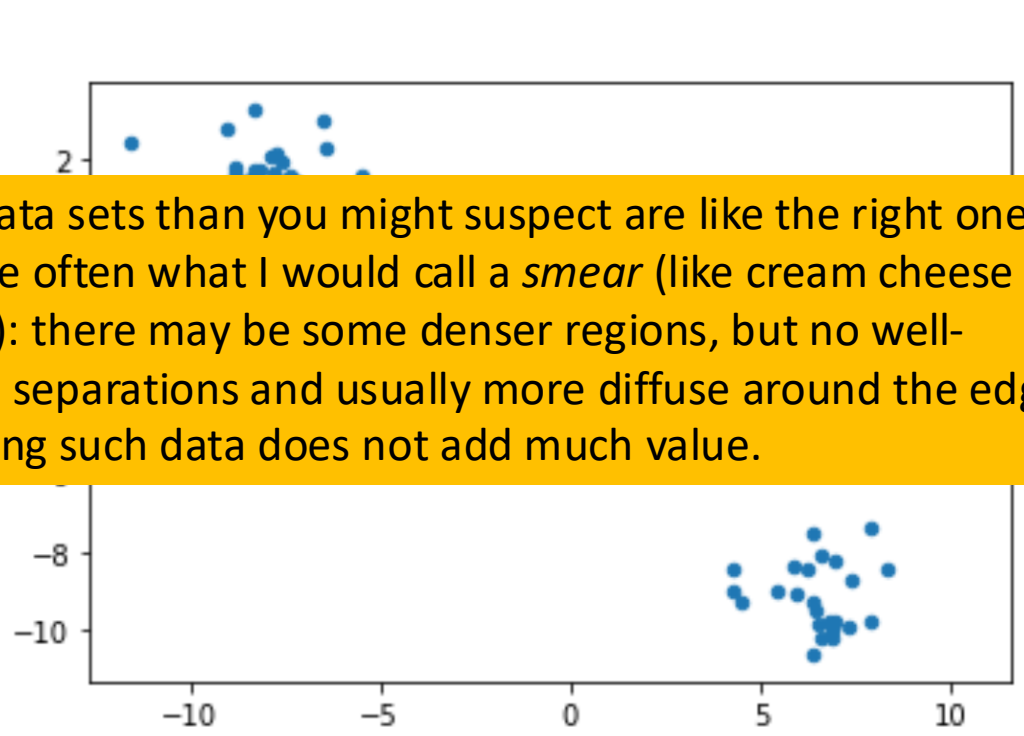


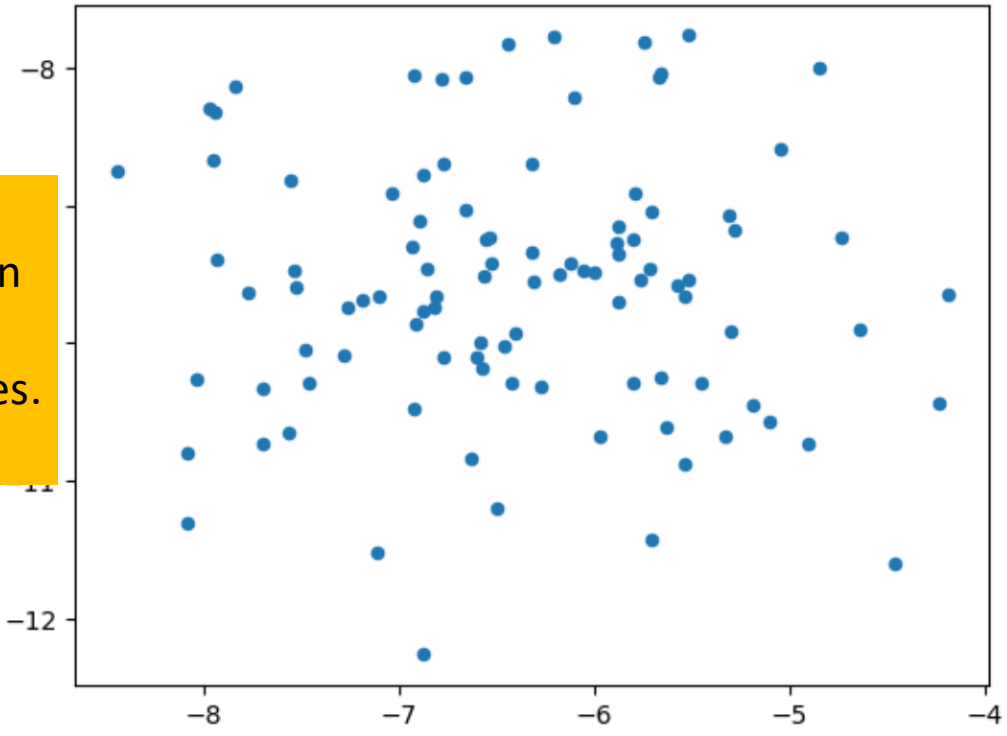




The left one has 4 well-defined blobs, the right one does not.

# But how do we choose k?

Whenever you can, especially when data $X \subset \mathbb{R}^2$, you should plot the data!
Consider the two data sets shown:




More data sets than you might suspect are like the right one. They are often what I would call a *smear* (like cream cheese on a bagel): there may be some denser regions, but no well-defined separations and usually more diffuse around the edges. Clustering such data does not add much value.

The left one has 4 well-defined blobs, the right one does not.

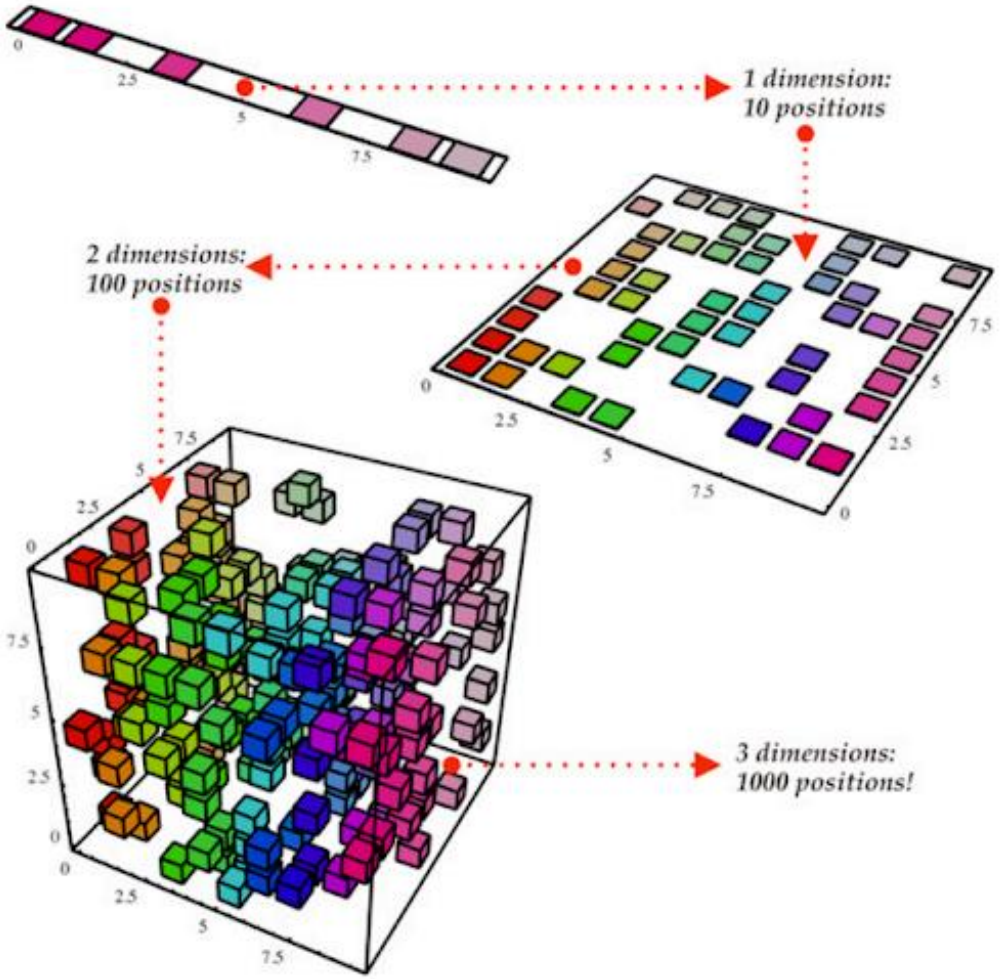
# Dimensionality reduction for clustering

- If your data is not naturally in $\mathbb{R}^2$, you may need to apply dimensionality reduction to visualize it.
- One method already seen: Laplacian Eigenmaps (covered in Lecture 10 as part of spectral clustering).

# Dimensionality reduction for clustering

- If your data is not naturally in $\mathbb{R}^2$, you may need to apply dimensionality reduction to visualize it.
- One method already seen: Laplacian Eigenmaps (covered in Lecture 10 as part of spectral clustering).
- Additional methods that will be covered in this course:
  - PCA (Principal Component Analysis)
  - MDS (Multidimensional Scaling)
  - Distance metric learning techniques
- Choosing a different distance metric can change the visualization:
  - It may transform a diffuse "smear" into more distinct clusters/blobs.

# Dimensionality reduction for clustering




Dimensionality reduction captures all the ”significant” directions In which the data is changing

Img source:
https://lh3.googleusercontent.com/proxy/FgcaiVRuuJ2fc5ndqO6BorIIstmvkHNiwzEAveT5qW8YGpre_HEjxic5JZcFn2uEv67ms1NaQujAU2J_5OEUqnrevbLu5LHC-G101JInEpZggmap4W8dVT0

# Dimensionality reduction for clustering

- Dimensionality reduction and visualization approaches are not limited to assignment-based clustering.
- They can also work for non-centrally symmetric clusters (e.g., irregular shapes).
- Regarding the two-moons example:
  - If you think it's relevant, you should demonstrate real-world data that resembles it.
  - Artificial examples generated with t-SNE or UMAP don't count as evidence.
- Regardless, you can still plot your data and circle clusters to highlight structure.

# How to choose k: The Elbow method

- In clustering methods, we define a **cost function** (Cost of a clustering).

# How to choose k: The Elbow method

- In clustering methods, we define a **cost function** (Cost of a clustering).
- The goal is to find the clustering that **minimizes this cost** (e.g., using Lloyd’s algorithm).

# How to choose k: The Elbow method

- In clustering methods, we define a **cost function** (Cost of a clustering).
- The goal is to find the clustering that **minimizes this cost** (e.g., using Lloyd's algorithm).
- One might think:
  - Evaluate the cost for each choice of **k**.
  - Return the **k** with the smallest cost.

# How to choose k: The Elbow method

- In clustering methods, we define a **cost function** (Cost of a clustering).
- The goal is to find the clustering that **minimizes this cost** (e.g., using Lloyd's algorithm).
- One might think:
  - Evaluate the cost for each choice of **k**.
  - Return the **k** with the smallest cost.
- Problem: For most formulations, the cost will **always decrease as k increases**.

# How to choose k: The Elbow method

- In clustering methods, we define a **cost function** (Cost of a clustering).
- The goal is to find the clustering that **minimizes this cost** (e.g., using Lloyd's algorithm).
- One might think:
  - Evaluate the cost for each choice of **k**.
  - Return the **k** with the smallest cost.
- Problem: For most formulations, the cost will **always decrease as k increases**.
- **Simply minimizing cost would always favor larger k, which is not meaningful.**

# How to choose k: The Elbow method

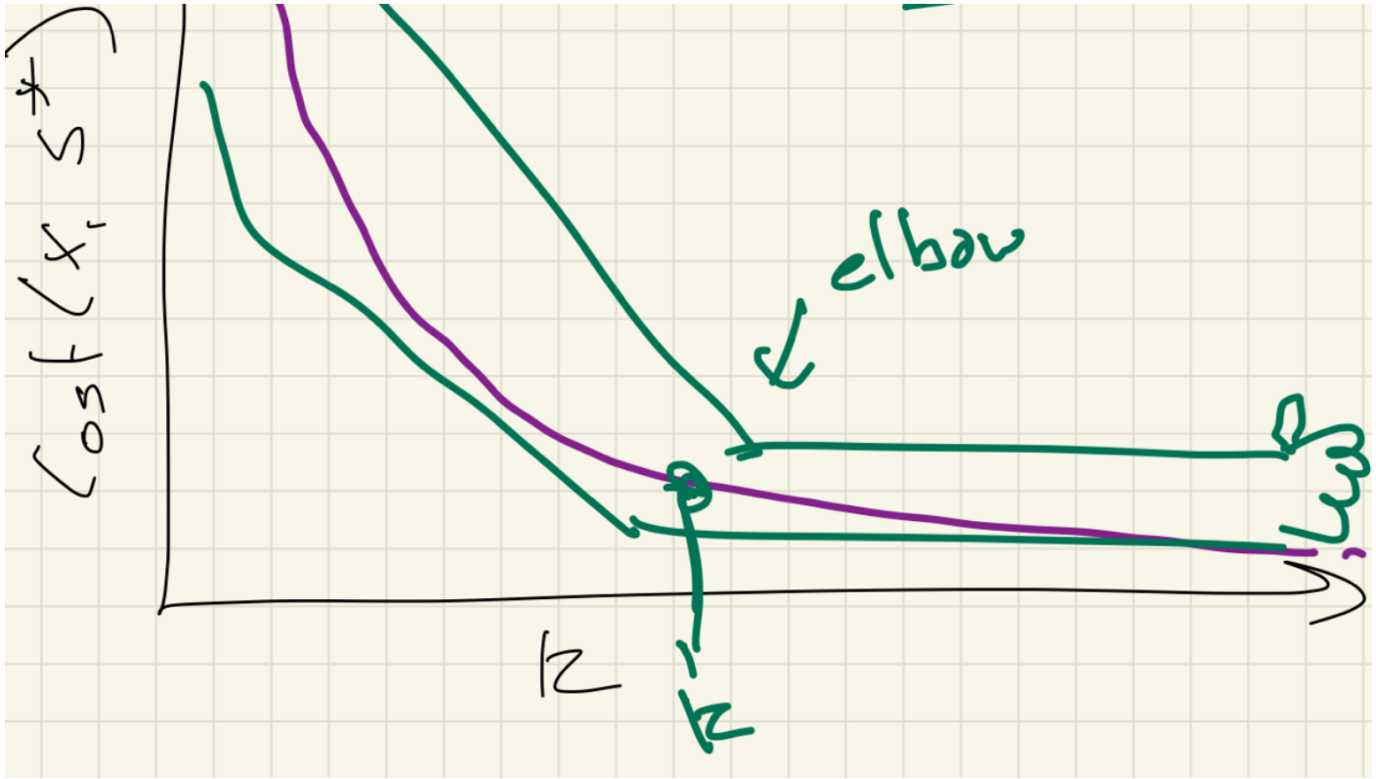
- For a clustering defined by sites s = {$s_1$, $s_2$, …, $s_k$} for a data set X, consider the cost

$$\text{Cost}_2(S, X) = \frac{1}{|X|} \sum_{x \in X} (x - \phi_S(x))^2$$

Recall that: $\phi_S(x) = \arg\min_{s_j \in S} \|x - s\|$

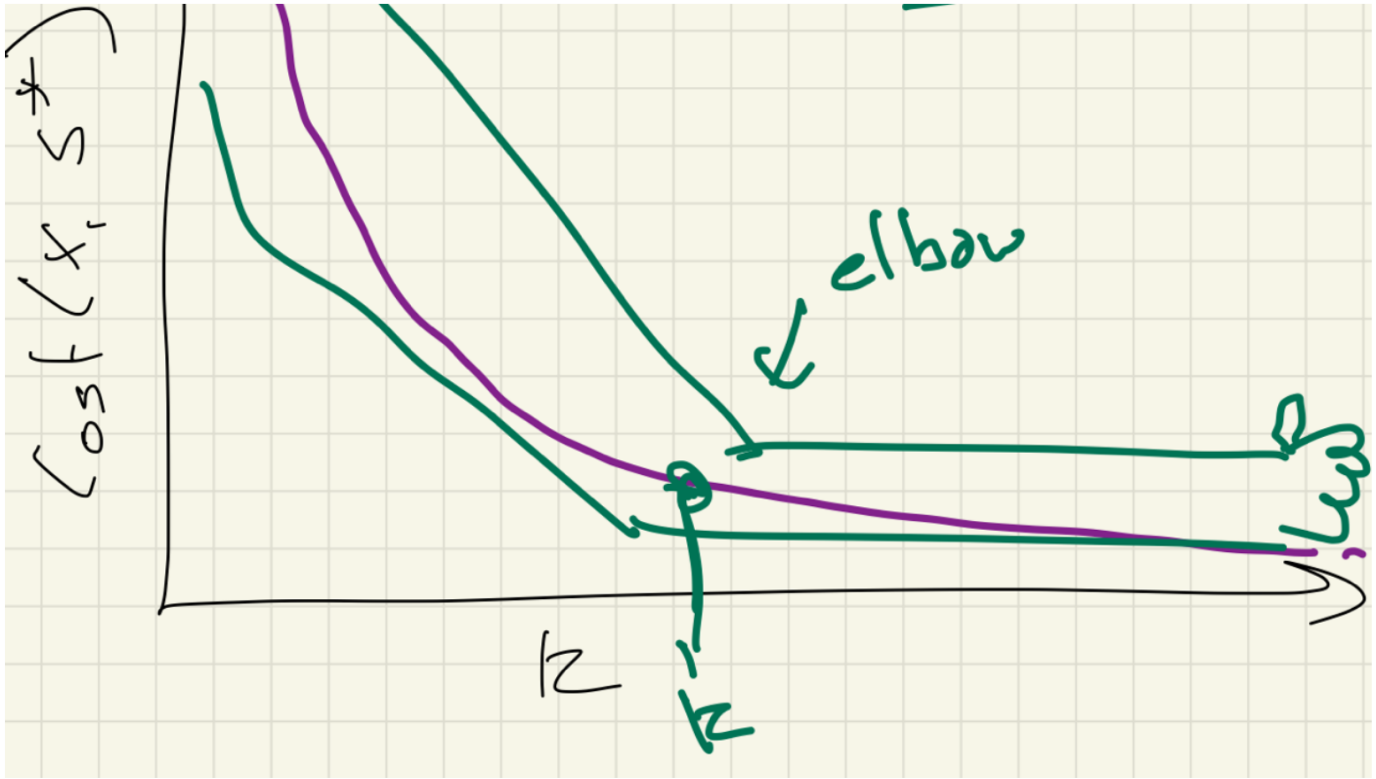
# How to choose k: The Elbow method

For some choice of $k$, if we let $S^*$ be the optimal clustering in terms of $\text{Cost}_2(S^*, X)$, then we can plot the score for each $k$. It will look something like this:



# How to choose k: The Elbow method

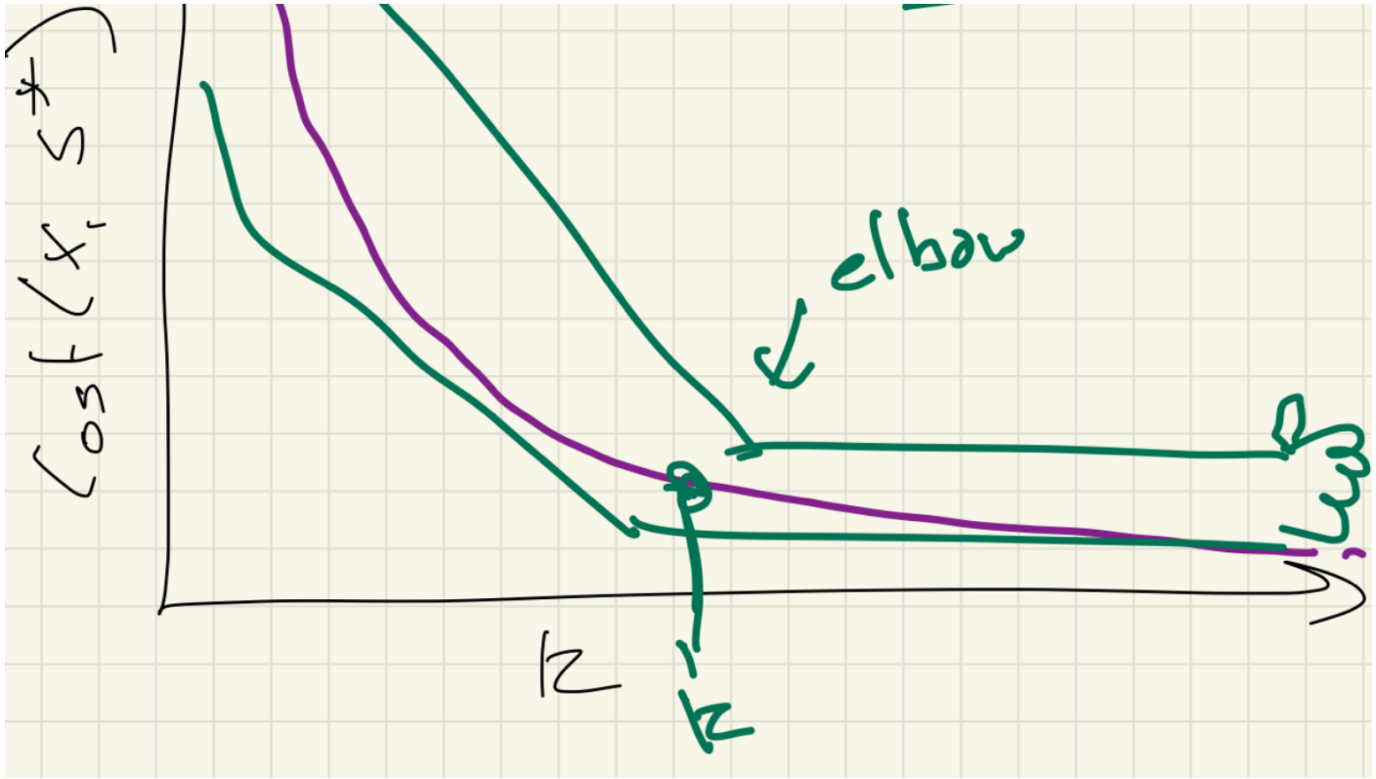
For some choice of $k$, if we let $S^*$ be the optimal clustering in terms of $\text{Cost}_2(S^*, X)$, then we can plot the score for each $k$. It will look something like this:




What happens when k = n?

# How to choose k: The Elbow method

For some choice of $k$, if we let $S^*$ be the optimal clustering in terms of $\text{Cost}_2(S^*, X)$, then we can plot the score for each $k$. It will look something like this:




Now that as k increases the curve (in purple) decreases. It should start high, and then decrease towards 0. When k = n it will be exactly 0, since each x ∈ X can be a center, and the distance to the closest one is 0.

# How to choose k: The Elbow method

- Suppose there are **well-defined clusters** for some choice of $k^*$.
- When $k < k^*$:

# How to choose k: The Elbow method

- Suppose there are **well-defined clusters** for some choice of $k^*$.
- When $k < k^*$:
  - Multiple true clusters get merged into one site $s_j$.
  - This causes the **cost to be high**.

# How to choose k: The Elbow method

- Suppose there are **well-defined clusters** for some choice of $k^*$.
- When $k < k^*$:
  - Multiple true clusters get merged into one site $s_j$.
  - This causes the **cost to be high**.
- When $k > k^*$:

# How to choose k: The Elbow method

- Suppose there are **well-defined clusters** for some choice of $k^*$.
- When $k < k^*$:
  - Multiple true clusters get merged into one site $s_j$.
  - This causes the **cost to be high**.
- When $k > k^*$:
  - True clusters get split across multiple sites.
  - The cost decreases only **slightly**, since points were already in compact clusters.

# How to choose k: The Elbow method

- Suppose there are **well-defined clusters** for some choice of $k^*$.
- When $k < k^*$:
  - Multiple true clusters get merged into one site $s_j$.
  - This causes the **cost to be high**.
- When $k > k^*$:
  - True clusters get split across multiple sites.
  - The cost decreases only **slightly**, since points were already in compact clusters.
- The **elbow point** occurs where:

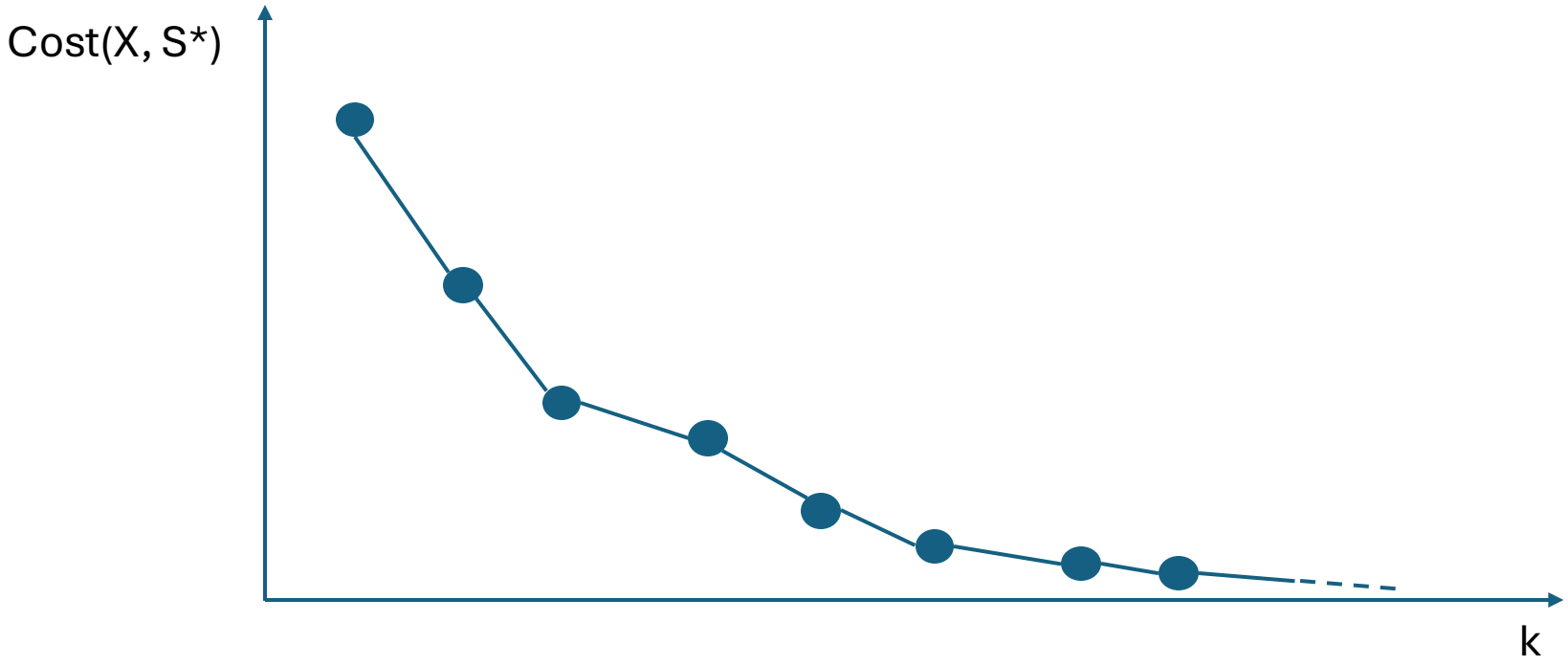
# How to choose k: The Elbow method

- Suppose there are **well-defined clusters** for some choice of $k^*$.
- When $k < k^*$:
  - Multiple true clusters get merged into one site $s_j$.
  - This causes the **cost to be high**.
- When $k > k^*$:
  - True clusters get split across multiple sites.
  - The cost decreases only **slightly**, since points were already in compact clusters.
- The **elbow point** occurs where:
  - The cost transitions from **rapidly decreasing** to **slowly decreasing**.
  - This bend in the curve (like an arm) indicates a **good choice of** $k$.

# Caveats of the Elbow method

# Caveats of the Elbow method

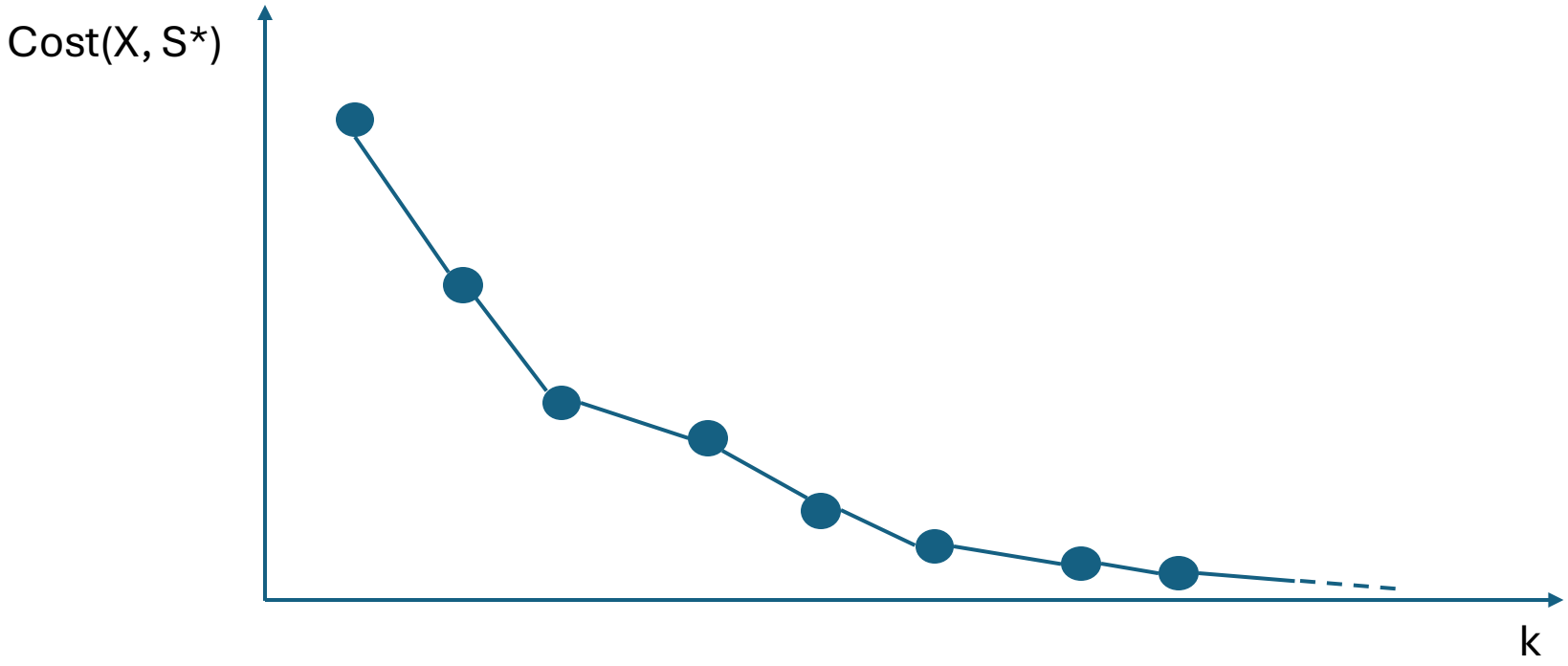
- Which elbow point is best?




https://community.ibm.com/community/user/blogs/moloy-de1/2020/07/02/points-to-ponder

# Caveats of the Elbow method

- Which elbow point is best?




Caveat: Multiple elbow points are possible

https://community.ibm.com/community/user/blogs/moloy-de1/2020/07/02/points-to-ponder

# Caveats of the Elbow method

- **Caveat: Elbow point is not always clear-cut**
  - Even in well-clustered data, some points are very well separated → splitting them lowers cost a lot.
  - Some clusters are close together → splitting has only a moderate effect on cost.
  - Some clusters are spread out → splitting them still reduces cost noticeably.
  - ⇒ Even with good clustering, the elbow point may not be obvious.

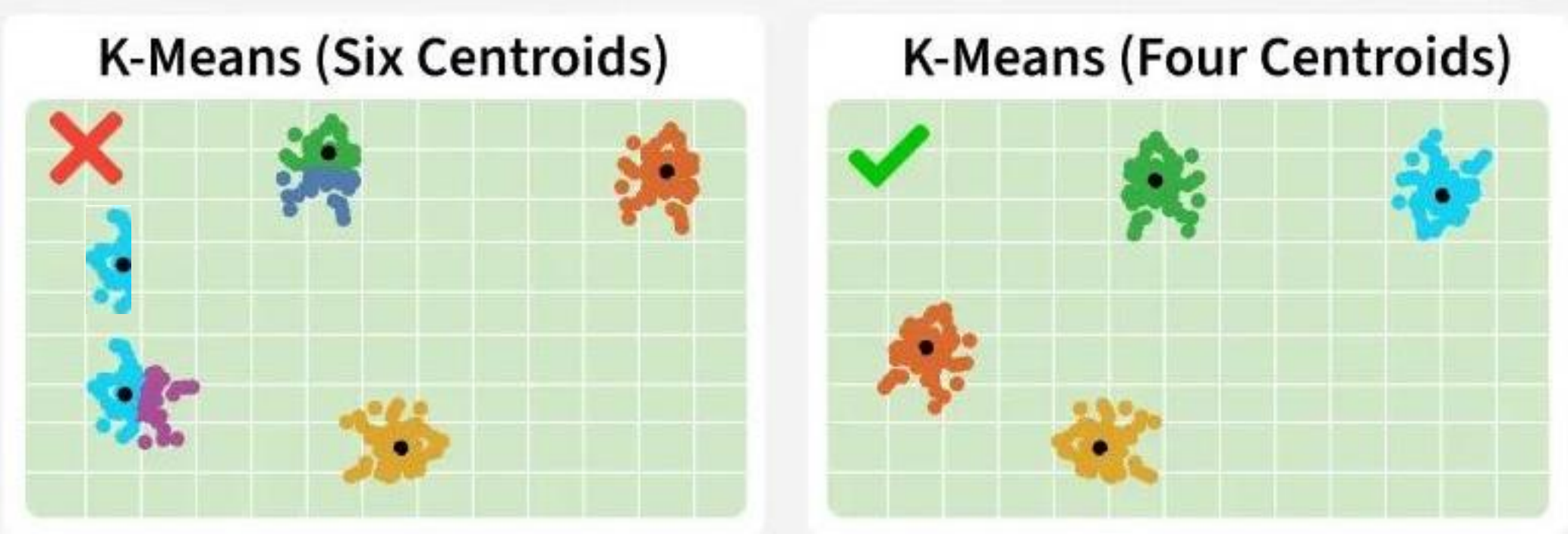
# Caveats of the Elbow method

- **Caveat: Elbow point is not always clear-cut**
  - Even in well-clustered data, some points are very well separated → splitting them lowers cost a lot.
  - Some clusters are close together → splitting has only a moderate effect on cost.
  - Some clusters are spread out → splitting them still reduces cost noticeably.
  - ⇒ Even with good clustering, the elbow point may not be obvious.
- **Positive side**
  - Getting $k$ slightly wrong may still be acceptable.

# Caveats of the Elbow method

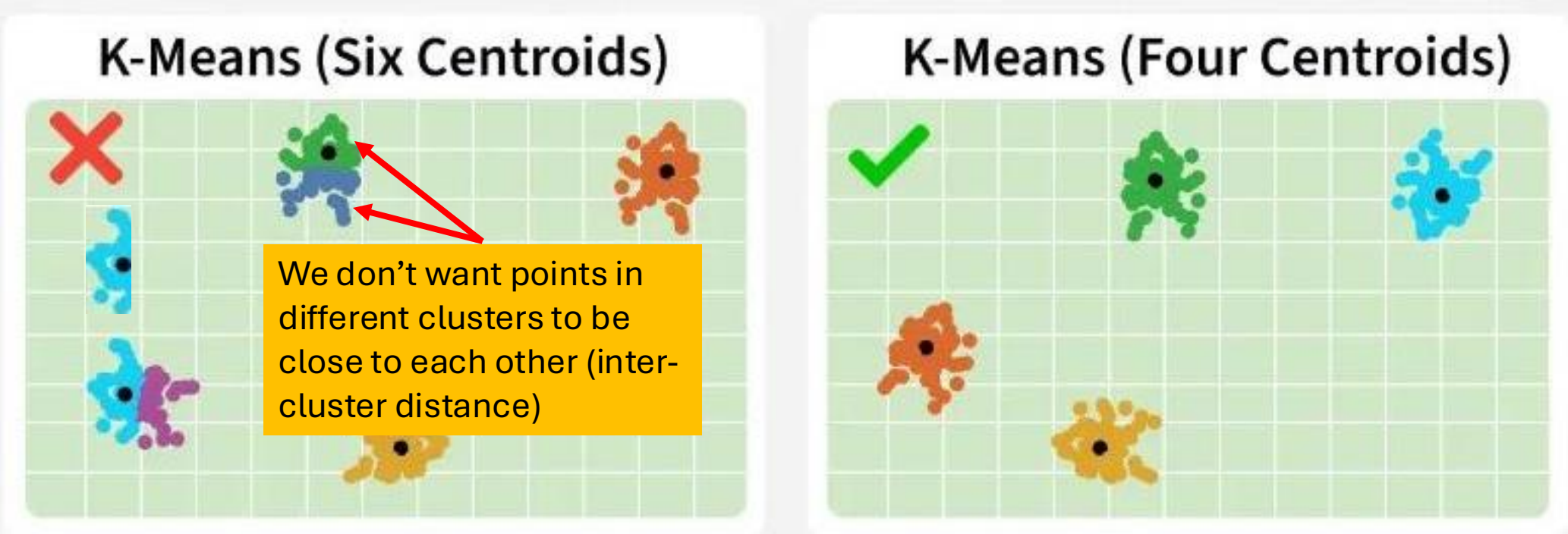
- **Caveat: Elbow point is not always clear-cut**
  - Even in well-clustered data, some points are very well separated → splitting them lowers cost a lot.
  - Some clusters are close together → splitting has only a moderate effect on cost.
  - Some clusters are spread out → splitting them still reduces cost noticeably.
  - ⇒ Even with good clustering, the elbow point may not be obvious.
- **Positive side**
  - Getting $k$ slightly wrong may still be acceptable.
- **Additional complication: Hierarchical structure**
  - Data may have multiple levels of clusters (clusters within clusters).
  - This can produce **two elbow points**.
  - Either choice may be reasonable—pick the one that best fits the analysis scale.

# What makes a clustering model good/bad?



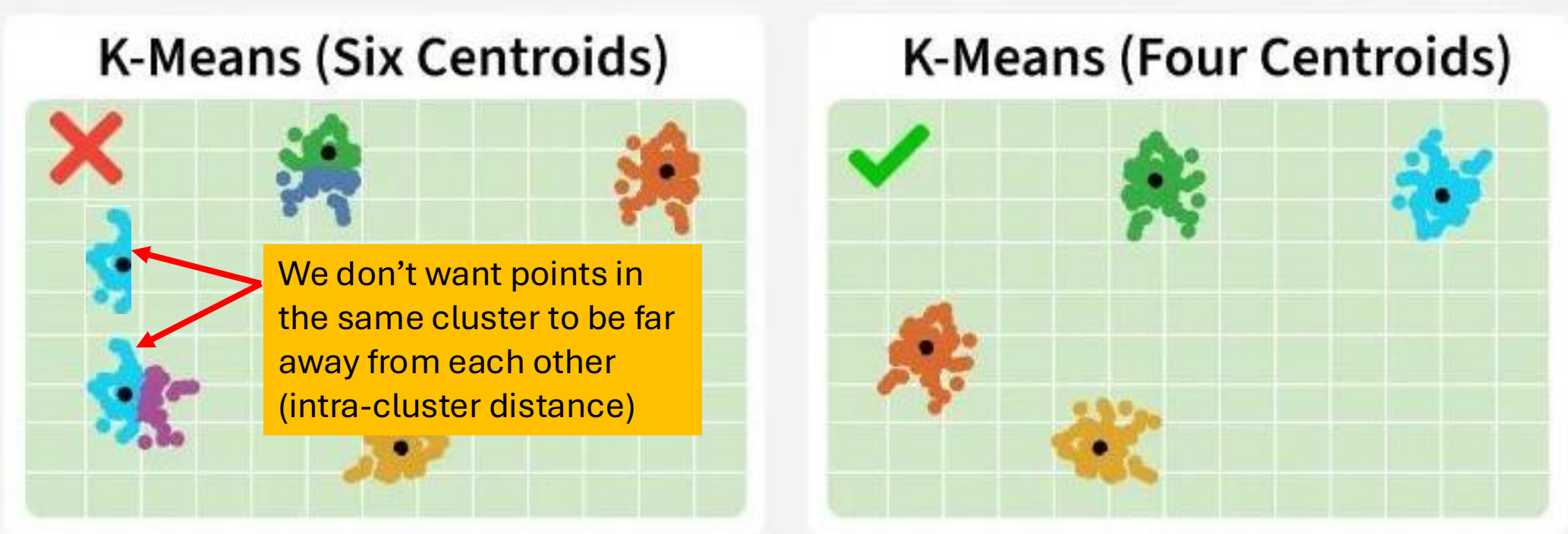

Img source: https://www.geeksforgeeks.org/machine-learning/what-is-silhouette-score/ (modified)

# What makes a clustering model good/bad?




Img source: https://www.geeksforgeeks.org/machine-learning/what-is-silhouette-score/

# What makes a clustering model good/bad?




Img source: https://www.geeksforgeeks.org/machine-learning/what-is-silhouette-score/

# The Silhouette score

- The elbow method started to quantify how good a given “k” is, but the ultimately resorted to drawing a plot, and using human judgement. The Silhouette score helps quantify this choice

# The Silhouette score

- The elbow method started to quantify how good a given “k” is, but the ultimately resorted to drawing a plot, and using human judgement. The Silhouette score helps quantify this choice
- The silhouette score measures **how well a point fits within its assigned cluster compared to other clusters**.

# The Silhouette score

- The elbow method started to quantify how good a given “k” is, but the ultimately resorted to drawing a plot, and using human judgement. The Silhouette score helps quantify this choice
- The silhouette score measures **how well a point fits within its assigned cluster compared to other clusters**.
- For each point $x_i$:
- **Cohesion (within-cluster distance)**
  - Compute $a(i)$ :the average distance from $x_i$ to all other points in its own cluster.
  - What does a small $a(i)$ mean?

# The Silhouette score

- The elbow method started to quantify how good a given “k” is, but the ultimately resorted to drawing a plot, and using human judgement. The Silhouette score helps quantify this choice
- The silhouette score measures **how well a point fits within its assigned cluster compared to other clusters**.
- For each point $x_i$:
- **Cohesion (within-cluster distance)**
  - Compute $a(i)$ :the average distance from $x_i$to all other points in its own cluster.
  - Small $a(i)$means $x_i$is close to its cluster buddies → good.

# The Silhouette score

- The elbow method started to quantify how good a given “k” is, but the ultimately resorted to drawing a plot, and using human judgement. The Silhouette score helps quantify this choice
- The silhouette score measures **how well a point fits within its assigned cluster compared to other clusters**.
- For each point $x_i$:
- **Cohesion (within-cluster distance)**
  - Compute $a(i)$ :the average distance from $x_i$ to all other points in its own cluster.
  - Small $a(i)$ means $x_i$ is close to its cluster mates → good.

- **Separation (distance to other clusters)**
  - Compute $b(i)$ :the smallest average distance from $x_i$ to all points in another cluster (the “next-best” cluster).
  - What does a large $b(i)$ means?

# The Silhouette score

- The elbow method started to quantify how good a given “k” is, but the ultimately resorted to drawing a plot, and using human judgement. The Silhouette score helps quantify this choice
- The silhouette score measures **how well a point fits within its assigned cluster compared to other clusters**.
- For each point $x_i$:
- **Cohesion (within-cluster distance)**
  - Compute $a(i)$ :the average distance from $x_i$ to all other points in its own cluster.
  - Small $a(i)$ means $x_i$ is close to its cluster mates → good.

- **Separation (distance to other clusters)**
  - Compute $b(i)$ :the smallest average distance from $x_i$ to all points in another cluster (the “next-best” cluster).
  - Large $b(i)$ means $x_i$ is far from other clusters → also good.

# The Silhouette score

For each point $x_i$:

- **Cohesion (within-cluster distance)**
  - Compute $a(i)$ :the average distance from $x_i$to all other points in its own cluster.
  - Small $a(i)$means $x_i$is close to its cluster mates → good.

# The Silhouette score

For each point $x_i$:

- **Cohesion (within-cluster distance)**
  - Compute $a(i)$ :the average distance from $x_i$to all other points in its own cluster.
  - Small $a(i)$means $x_i$is close to its cluster mates → good.

- This assumes some model similar to k-means, k-medioid, or mean-link HAC. We want to quantify a disjoint clustering $S_1$, $S_2$, . . . , $S_k$. We then quantify the average *inter-cluster distance* for a point $x_i \in X$ in cluster j as:

$$a(i) = \frac{1}{|S_j| - 1} \sum_{x \in S_j; x \neq x_i} \mathbf{d}(x_i, x)$$

# The Silhouette score

For each point $x_i$:

- **Separation (distance to other clusters)**
  - Compute $b(i)$ :the smallest average distance from $x_i$to all points in another cluster (the “next-best” cluster).
  - Large $b(i)$means $x_i$is far from other clusters → also good.

# The Silhouette score

For each point $x_i$:

- **Separation (distance to other clusters)**
  - Compute $b(i)$ :the smallest average distance from $x_i$ to all points in another cluster (the "next-best" cluster).
  - Large $b(i)$ means $x_i$ is far from other clusters → also good.

- We also consider the average *replacement cluster score* again for a point $x_i \in X$ in cluster j where

$$b(i) = \min_{j' \neq j} \frac{1}{|S_{j'}|} \sum_{x \in S_{j'}} \mathbf{d}(x_i, x)$$

This is what the score a(i) would be for $x_i$ if it could not use cluster $S_j$ and instead had to use the next best option (which would be cluster $S_{j'}$)

# The Silhouette score

With these values, we can define the Silhouette score for a point $x_i \in X$ as

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

Note that $a(i)$ is not defined if the cluster for $x_i$ is of size 1. In this case, we define $s(i) = 0$.

# The Silhouette score

With these values, we can define the Silhouette score for a point $x_i \in X$ as

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

Note that $a(i)$ is not defined if the cluster for $x_i$ is of size 1. In this case, we define $s(i) = 0$.

How to interpret s(i) > 0

# The Silhouette score

With these values, we can define the Silhouette score for a point $x_i \in X$ as

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

Note that $a(i)$ is not defined if the cluster for $x_i$ is of size 1. In this case, we define $s(i) = 0$.

Note that $s(i) \in [-1, 1]$, and that if $s(i) > 0$, then its "better off" in its cluster than the replacement one scored by $b(i)$.

The *average silhouette score* for the entire clustering is reported as

$$\mathsf{sil}(S, X) = \frac{1}{|X|} \sum_{x_i \in X} s(i)$$

# The Silhouette score

Finally, **we can choose k as the value which results in the clustering with highest average silhouette score**. This is a popular way to fully automate the decision on k, but note that it assumes a specific model of what constitutes a good cluster. For instance, this may not be meaningful for density based clustering or single-link HAC.

# Bayesian Information Criteria (BIC)

- The **BIC** is a model selection criterion that tries to balance two things:
- **Goodness of fit**
  - How well the model explains the data (e.g., likelihood).
  - A more complex model (higher $k$) will usually fit better.
- **Model simplicity (penalty for complexity)**
  - Adding parameters always risks overfitting.
  - BIC penalizes models with more parameters, especially when the dataset is large.

# Bayesian Information Criteria (BIC)

- We want to fit each cluster $S_j$ with a *generative likelihood model* fj . Given a data element x, we can measure $f_j(x)$ which evaluates how likely a point is to come from the model. It is a positive probability density function, so it is normalized so its integral is 1.
- Let's make this concrete by fitting each cluster $S_j \subset X$ with an isotropic Gaussian (multi-dimensional normal) distribution. The *isotropic* term means we consider the same variance σ in each direction; for simplicity, assume σ is fixed. To define this we need a center parameter $s_j \in R^d$. Then

$$f_j(x) = \frac{1}{(\sqrt{2\pi}\sigma)^d} \exp\left(-\frac{\|x - s_j\|^2}{2\sigma^2}\right)$$

# Bayesian Information Criteria (BIC)

- And the the likelihood for a cluster, assuming the data is iid from $f_j$ is

$$f_j(S_j) = \prod_{x \in S_j} f_j(x) = \prod_{x \in S_j} \frac{1}{(\sqrt{2\pi}\sigma)^d} \exp\left(-\frac{\|x - s_j\|^2}{2\sigma^2}\right)$$

Indeed if we were to maximize this for $S_j$ over the choice of center $s_j$, the mean $\frac{1}{|S_j|}\sum_{x \in S_j}$ is the optimal choice: the *maximum likelihood estimator*.

# Bayesian Information Criteria (BIC)

We can assume we have defined a likelihood for a set $f(S_k)$ for the best clustering $S_k = \{S_1, S_2, \ldots, S_k\}$.

It is typically more numerically stable to work with the *negative log-likelihood*:

$$\ell(\mathcal{S}_k) = -\ln(f(\mathcal{S}_k))$$

# Bayesian Information Criteria (BIC)

We can assume we have defined a likelihood for a set $f(S_k)$ for the best clustering $S_k = \{S_1, S_2, \ldots, S_k\}$.

It is typically more numerically stable to work with the *negative log-likelihood*:

$$\ell(\mathcal{S}_k) = -\ln(f(\mathcal{S}_k))$$

For one cluster $S_j$

$$\ell(S_j) = -\ln(f_j(S_j)) = -\sum_{x \in S_j} \ln(f_j) = \sum_{x \in S_j} \left( \frac{\|x - s_j\|^2}{2\sigma^2} - d \ln(\frac{1}{\sqrt{2\pi}\sigma}) \right)$$

Because we negated it, we seek to minimize $\ell(\cdot)$ when we sought to maximize the likelihood $f(\cdot)$.\

# Bayesian Information Criteria (BIC)

- **Likelihood challenge:** Increasing the number of clusters $k$ always increases the likelihood.
- As $k$ grows, the model can fit the data more closely (sometimes overfitting).
- Correspondingly, the negative log-likelihood always decreases with larger $k$.
- This makes it tricky to use likelihood or negative log-likelihood alone for choosing the “right” $k$.

# Bayesian Information Criteria (BIC)

- To avoid overfitting, we **penalize models with more parameters**.
- In $k$**-means clustering** in $\mathbb{R}^d$:
  - Each cluster center $s_j$ has $d$ parameters.
  - Total parameters = $kd$
- Using **information-theoretic arguments** and a Bayesian perspective, we derive the **Bayesian Information Criterion (BIC)**.

# Bayesian Information Criteria (BIC)

- For a model $M$ with $m$ parameters and $n$ observations, the BIC is:

$$\mathrm{BIC}(M) = -2\ln(f(M)) + m\ln(n)$$

# Bayesian Information Criteria (BIC)

- For a model $M$ with $m$ parameters and $n$ observations, the BIC is:

$$\mathrm{BIC}(M) = -2\ln(f(M)) + m\ln(n)$$

- Since our k-means algorithm has kd parameters, for a best-fit clustering $S_k$ of size k we have

$$\mathrm{BIC}(\mathcal{S}_k) = -2\ln(f(\mathcal{S}_k)) + kd\ln(|X|)$$

- Now the first term is twice the negative log-likelihood, and so decreases with k increasing. On the other hand, the second term kd ln(|X|) has a fixed quantity dln(|X|) and so increases linearly with k.

- The value k which minimizes BIC(Sk) provides a choice for k.

# Bayesian Information Criteria (BIC)

- The BIC method applies broadly to any model with a well-defined likelihood function.
- It is well-defined for:
  - Mixture of Gaussians
  - k-means
- For Hierarchical Agglomerative Clustering (HAC):
  - Application of BIC is possible but debated.
- For Spectral clustering or DBScan:
  - Applying BIC is difficult and not straightforward.