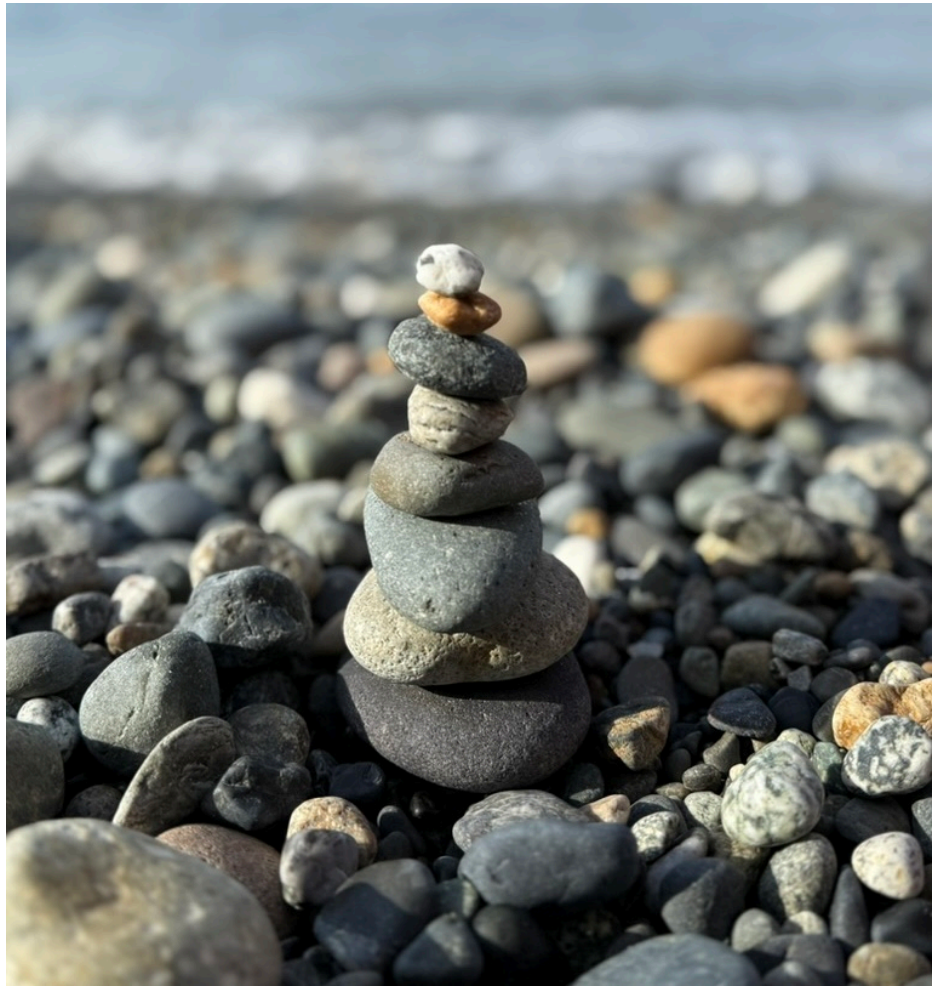
CPSC 330 Lecture 14: K-Means

Varada Kolhatkar

Focus on the breath!



Today's big questions

1. What can we learn when there is **no target to predict**?
2. How can we turn **similarity into groups**?
3. How do we choose a useful **number of clusters**?
4. What if our clusters miss the **similarity we care about**?

What can we learn without targets?

Organizing an online store

Problem: Help customers browse furniture that has no group labels.



We know how to train a classifier but **what would its target be?**

Unsupervised learning

Our supervised models learn from **features X and targets y**.

Here we have X**, without a target column**. We can instead look for useful structure:

- **Clustering:** which examples form similar groups? For the furniture store, discover groups without supplied category labels.
- **Representation learning:** can we describe the data using useful new features?

Dimensionality reduction seeks a representation with fewer features.

Representation learning: clicks to interests

Problem: Thousands of article-click features are difficult to interpret; total clicks hide interests.

Instead learn a smaller set of interest dimensions from patterns of articles read together.

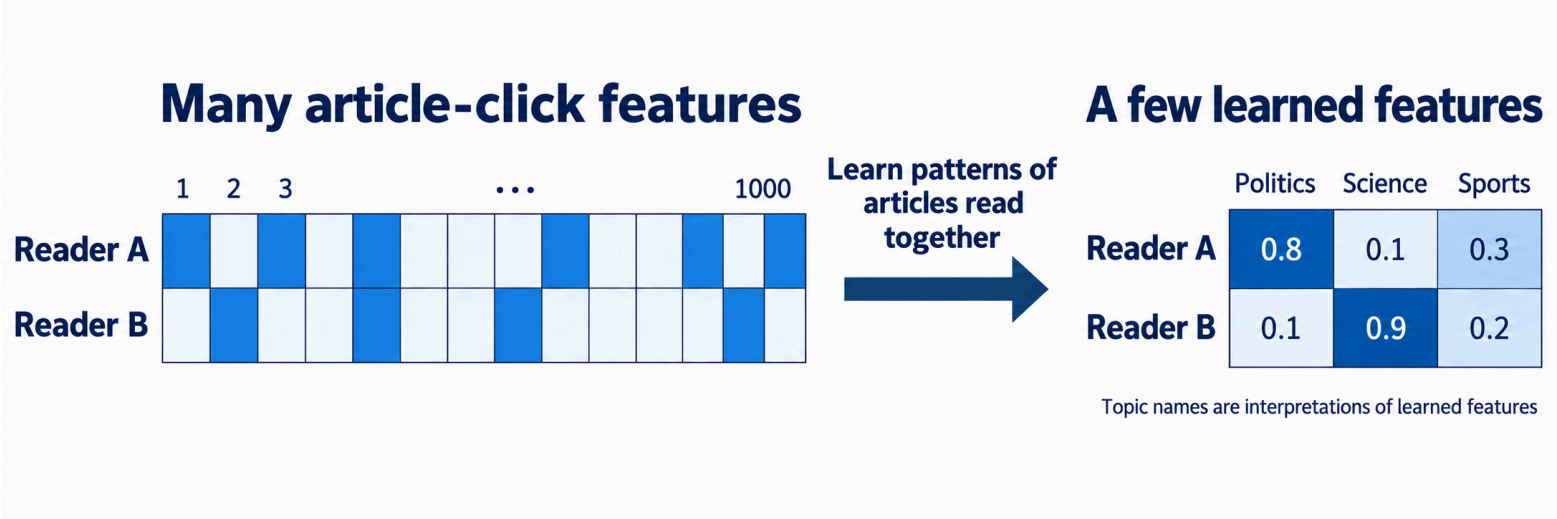
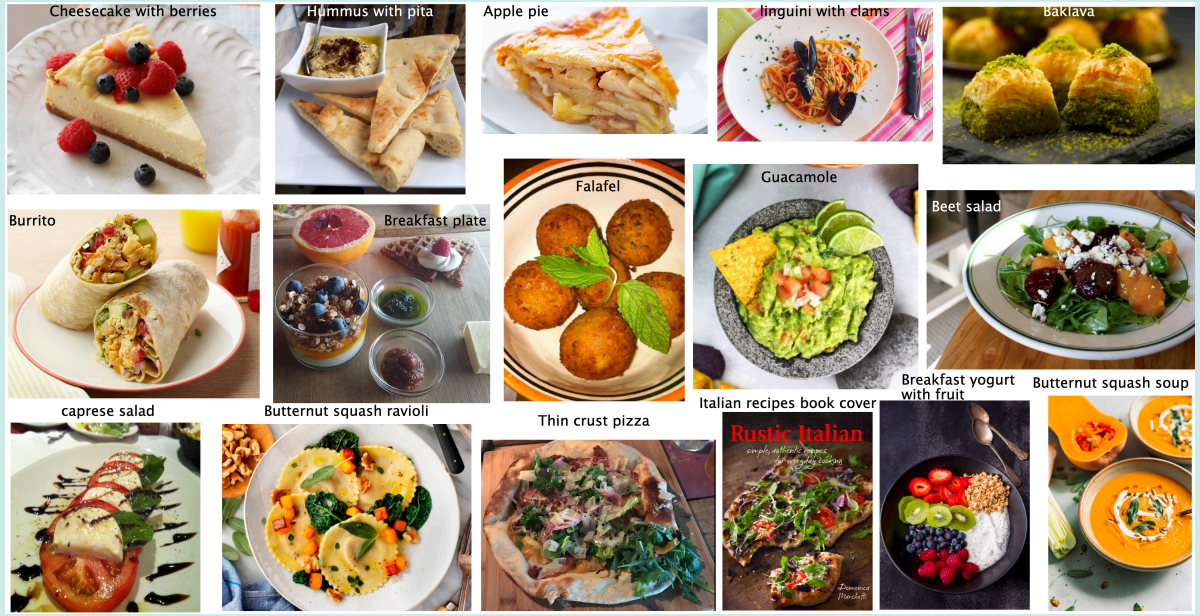


Image created by GPT-6.1 Sol Light

How can we group similar examples?

Your turn: group the foods



With a neighbour (2 minutes): Propose two useful groupings. For each, explain its purpose and the features you would use. Would raw pixels capture what you mean by “similar”?

The goal of clustering

Group similar examples without predefined group labels.

- **Within a cluster:** examples should be similar.
- **Between clusters:** examples should differ.

What counts as “similar” depends on the **features, distance, and purpose**, as in your food groupings.

The algorithm returns memberships; **we interpret what the groups mean**.

Using the distance tools we already know

Represent each food by numeric features and choose k groups.

- **KNN** taught us to compare points using Euclidean distance.
- Represent each group by a **centre**; assign points to the nearest one.
- Use the group's **mean** as its centre.

But **we need groups to calculate centres, and centres to assign groups**.

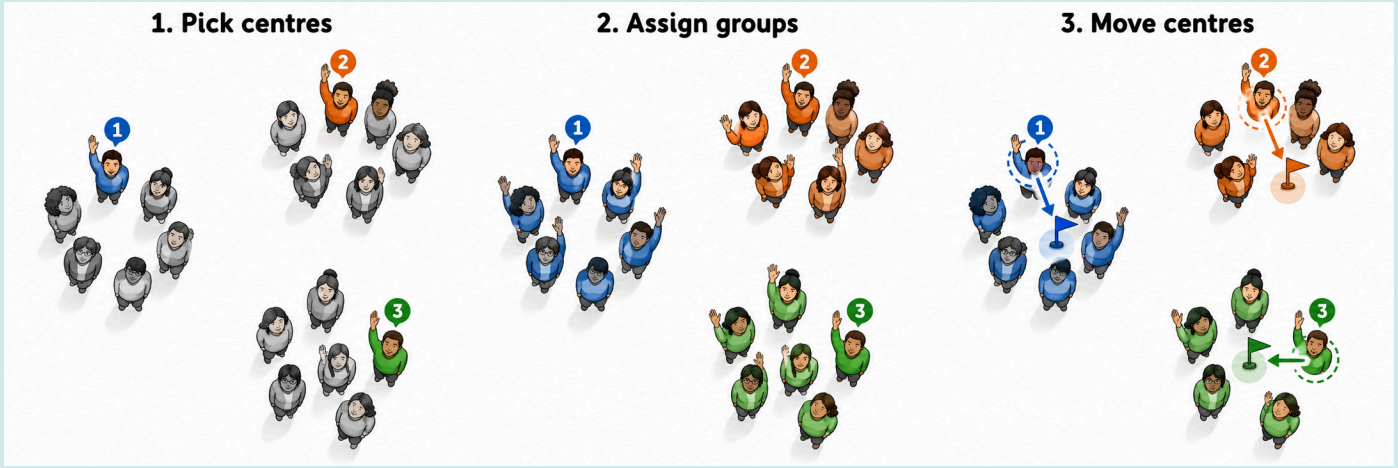
The K-Means algorithm

Input: features X and k. **Output:** k centroids and one label per example.

1. Choose k initial centres.
2. **Assign:** send each point to its nearest centre.
3. **Update:** move each centre to the mean of its assigned points.
4. Repeat assignment and update until the centres change very little, or the iteration limit is reached.

The number of clusters stays fixed during a run.

Your turn: act out K-Means



1. **Pick centres:** three volunteers mark the initial centre positions.
2. **Assign:** raise your hand when your nearest centre is called.
3. **Update:** move each centre to its group's average row/column position.
4. **Repeat:** reassign, then update.

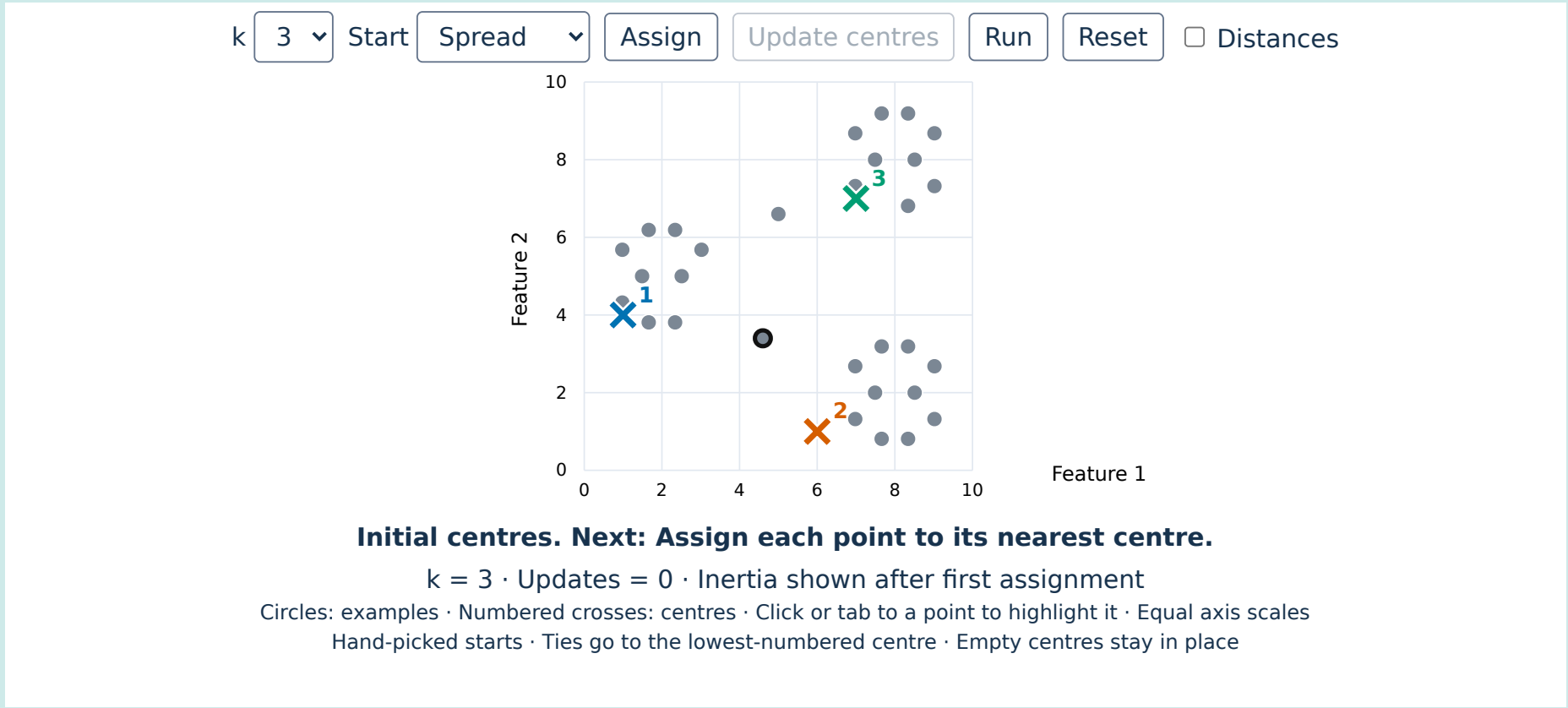
Keep everyone else's position fixed. A centre can lie **between seats**.

Discuss: what happened during K-Means?

1. Did anyone **switch clusters** in the second round? Why?
2. When do you think **K-Means should stop**?
3. Could we get a **different result** with different initial centres?

Interactive check: assign, average, repeat

Before Assign: Which centre will the highlighted point join? **Before Update:** Where will its centre move?

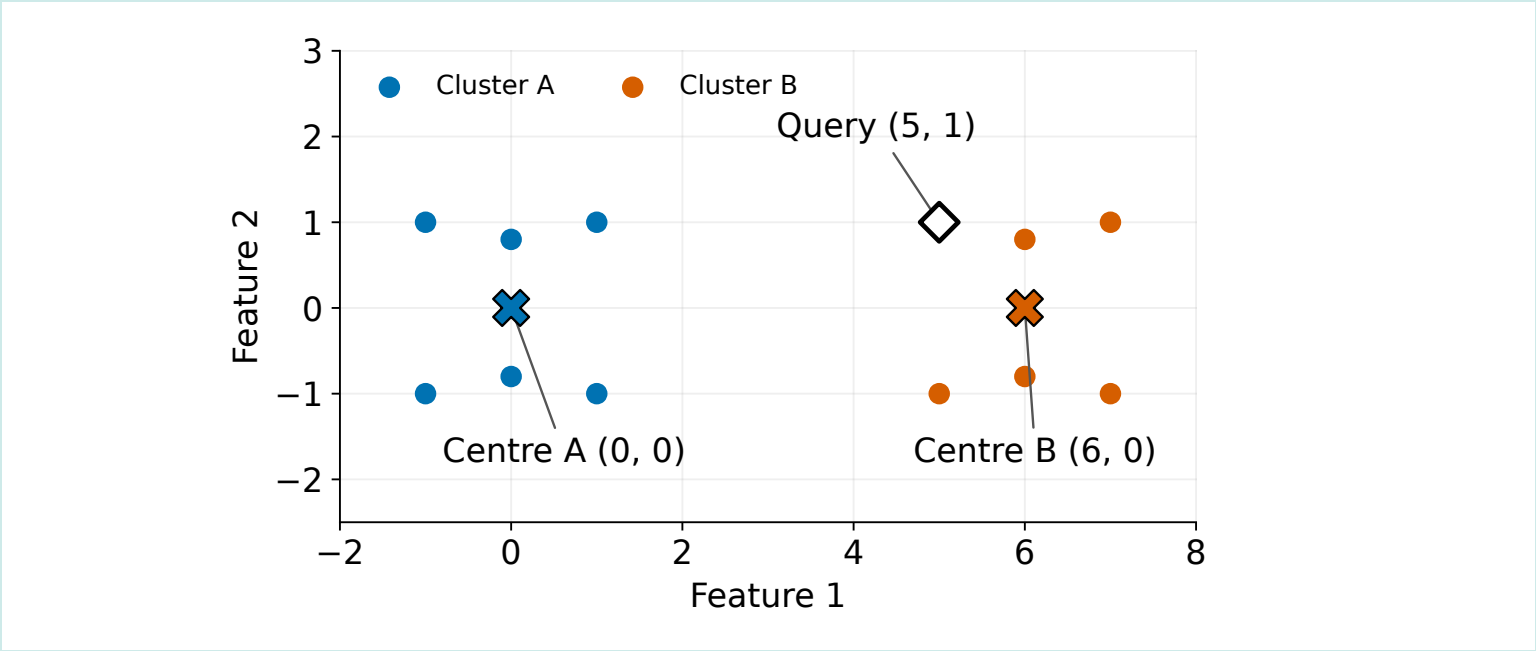


K-Means in scikit-learn

```
1 from sklearn.cluster import KMeans
2
3 km = KMeans(n_clusters=3, random_state=42)
4 labels = km.fit_predict(X)  # no target y
5 centres = km.cluster_centers_
6 new_labels = km.predict(X_new)
```

`predict` assigns new examples to their nearest **learned centre**.

Your turn: predict a new example



Which cluster gets the query point? Explain using distance.

Does `predict` move either centre? What if we instead refit on the expanded dataset?

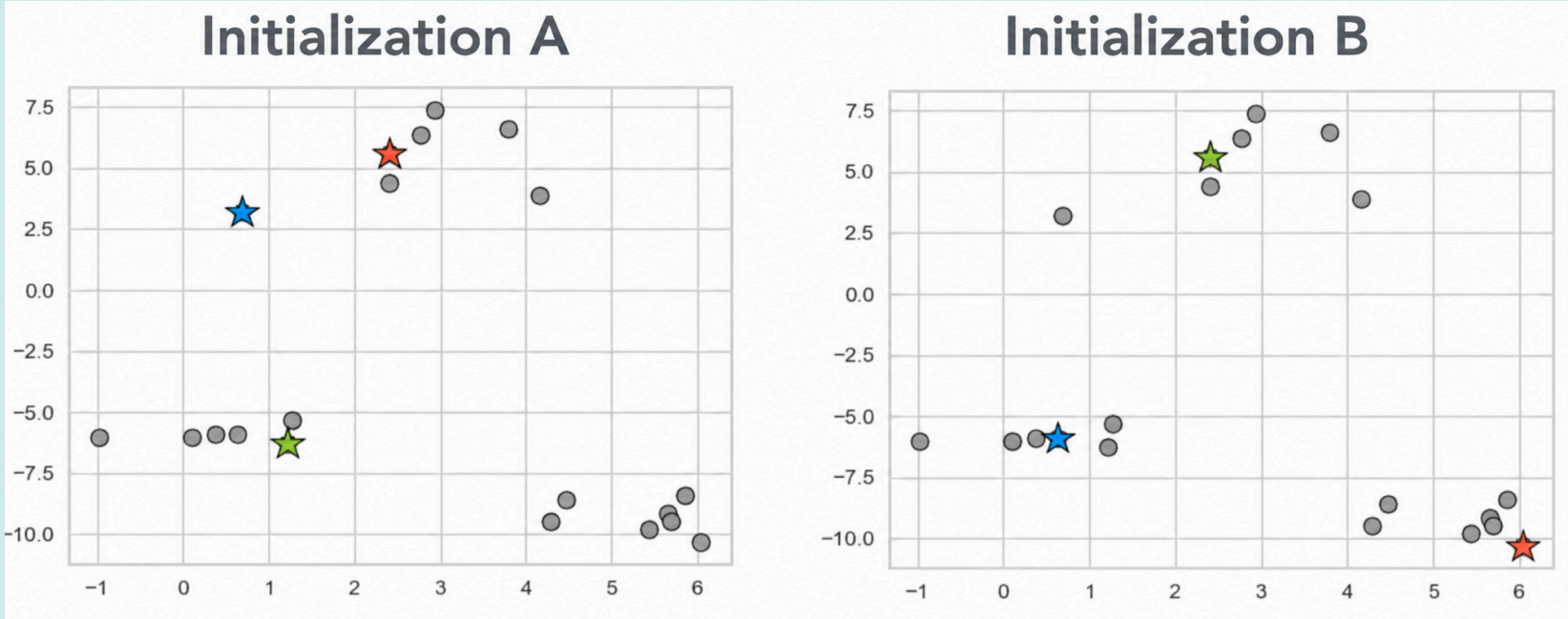
iClicker: assignments and prediction

Select all true statements.

- a. A centroid must coincide with an observed example.
- b. A point may change clusters during fitting.
- c. k increases automatically as fitting proceeds.
- d. `predict` assigns a new point without moving the learned centres.
- e. Renaming cluster IDs leaves the grouping unchanged.

Initialization: same data, different groups

We have fixed X and k. Does that determine a single final result? Which initilization is better here?



Getting a better starting point

Problem: one initialization can produce a poor solution.

- **K-Means++** spreads out the initial centres.
- **Multiple starts** try different initializations and retain the lowest within-cluster sum of squares.
- A fixed `random_state` makes the result reproducible.

If you use `n_init=10`: try ten starts, then retain the best run.

iClicker: has K-Means found the best solution?

The centres have stopped moving. **Which statement follows?**

- a. This is the globally smallest possible sum of squared distances.
- b. Every other initialization must produce the same grouping.
- c. Another assignment/update cycle will leave this solution unchanged, using the same tie rules.
- d. The chosen k must be appropriate for the application.

Scaling: should changing units change the groups?

- We cluster customers using **age in years** and **spending in dollars**.
- Changing spending to **thousands of dollars** changes its contribution to Euclidean distance even though the customers are unchanged.
- **Use the preprocessing tools we know:** usually scale numeric features before K-Means.
- Fit the scaler on the fitting data and apply the same transformation to new examples.

K-Means: strengths and limitations

Strength	Limitation
Simple and usually efficient	We must choose k
Centres summarize groups compactly	Means are sensitive to outliers
Effective for compact, roughly spherical, similarly sized groups	Curved or elongated groups can be split unnaturally
Assigns every example to a group	No explicit rejection of noise or ambiguous points

Initialization and feature representation affect the result.

How many groups should we create?

How many clusters?

K-Means asks us to choose k **before fitting**.

- **Too few:** we may merge groups whose differences matter.
- **Too many:** we may split useful groups into small fragments.
- There may not be one correct k; the intended use and domain knowledge matter.

How can we compare candidate values of k?

Evidence for choosing k

Quantitative evidence

- **Elbow plot:** how much does another centre reduce inertia?
- **Silhouette:** how cohesive and separated are the groups?

Qualitative evidence

- Inspect examples: what do members of each group share?
- Are the groups useful for the intended task?

Use numerical diagnostics alongside interpretation.

Measuring fit with inertia

We can fit K-Means for several values of k. How can we compare them?

Inertia is the sum of squared distances to assigned centroids:

$$\text{inertia} = \sum_{i=1}^{n} ||x_i - c_{z_i}||^2.$$

For a fixed k, lower inertia means a better fit to this objective.

Does the smallest inertia necessarily give the most useful grouping?

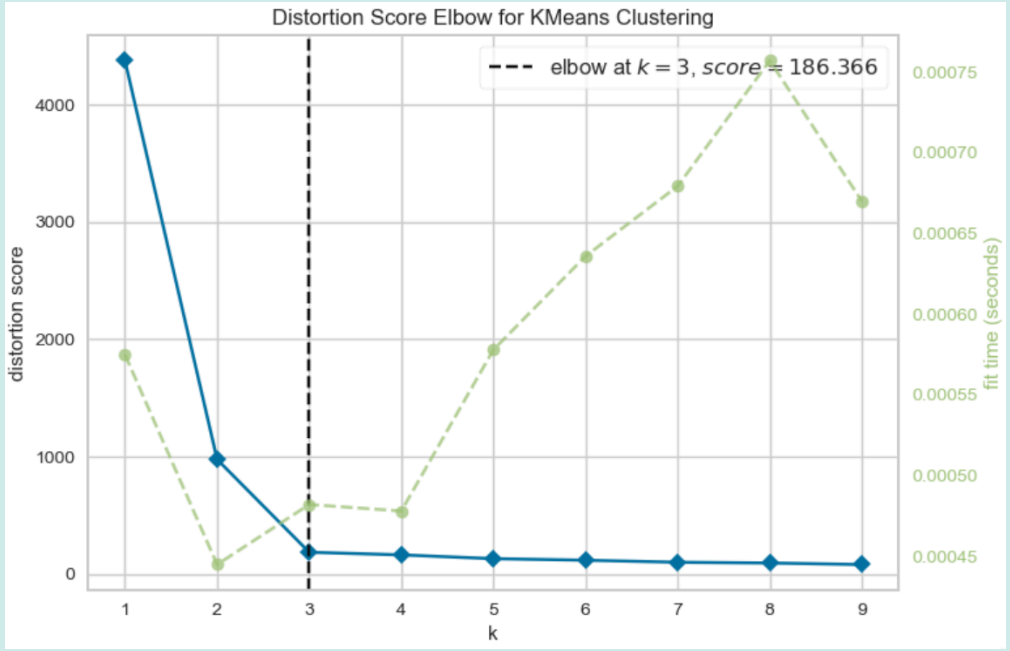
The elbow method

Adding centres can reduce inertia but how much improvement is enough?

- Fit K-Means for several values of k.
- Plot **inertia** against k.
- Look for an **elbow**, where further reductions become much smaller.

The elbow suggests a **candidate** k. It may be weak or absent.

The elbow method example



- At $k = 1$, where is the centre? What does inertia measure?
- At $k = n$, what can inertia be?
- Why don't we always choose $k = n$?

Which candidate k would you investigate from this plot?

Are individual points in suitable groups?

Inertia tells us how close points are to **their own centres**.

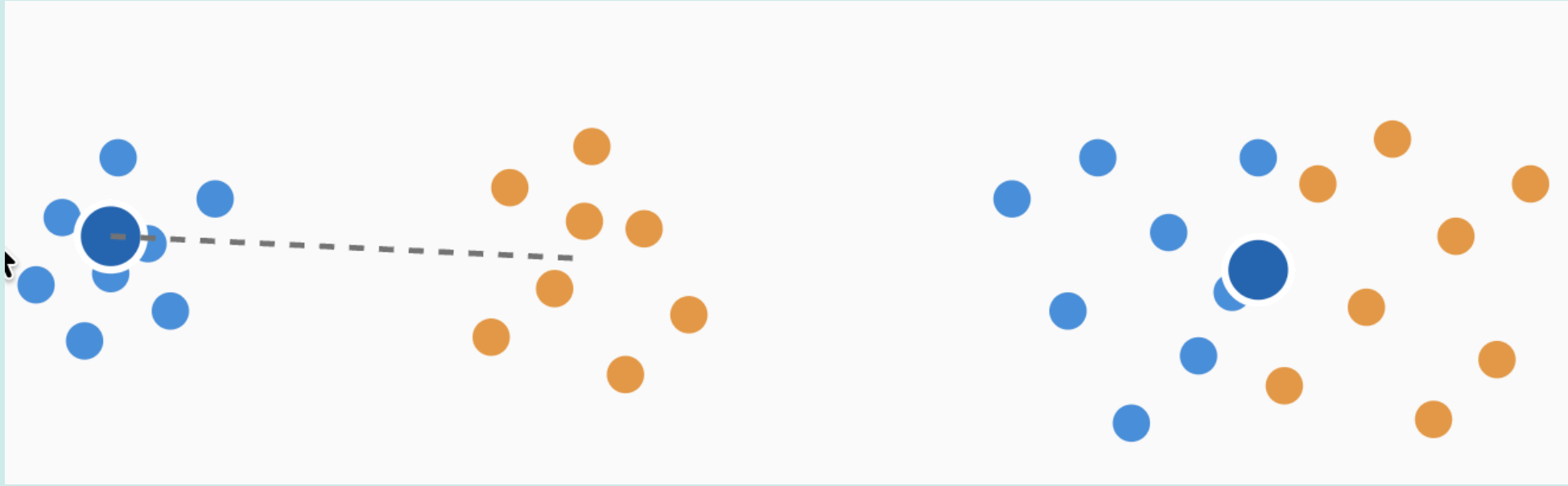
It does not directly compare a point's own group with its **nearest competing group**.

We already know how to calculate distances. Compare:

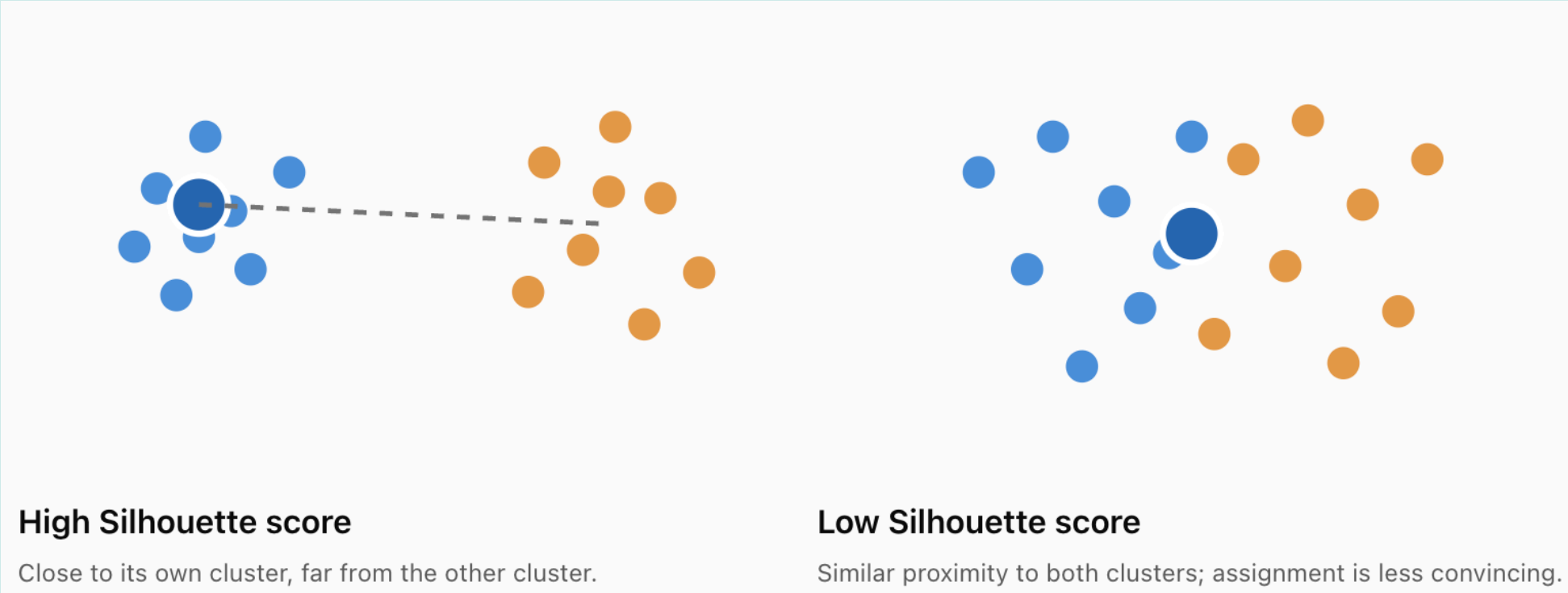
- How close is this point to others **inside its group**?
- How close is it to points in the **best alternative group**?

This motivates the **silhouette score**.

Which blue point seems more confident about belonging to the blue cluster? Why?



Which blue point seems more confident about belonging to the blue cluster? Why?



High Silhouette score

Close to its own cluster, far from the other cluster.

Low Silhouette score

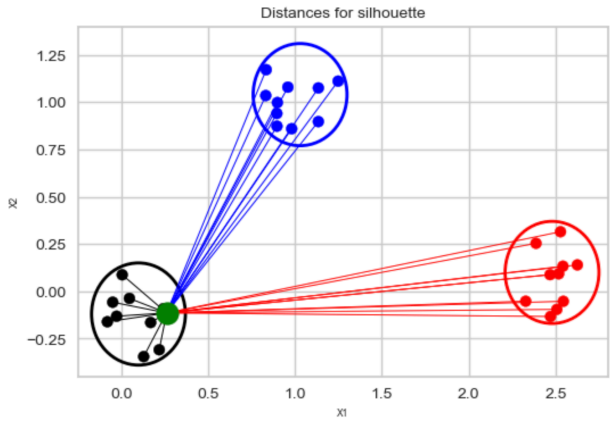
Similar proximity to both clusters; assignment is less convincing.

Silhouette: cohesion and separation

For one point i:

- $a(i)$: mean distance to **other points in its own cluster**.
- $b(i)$: smallest mean distance to points in **another cluster**.

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



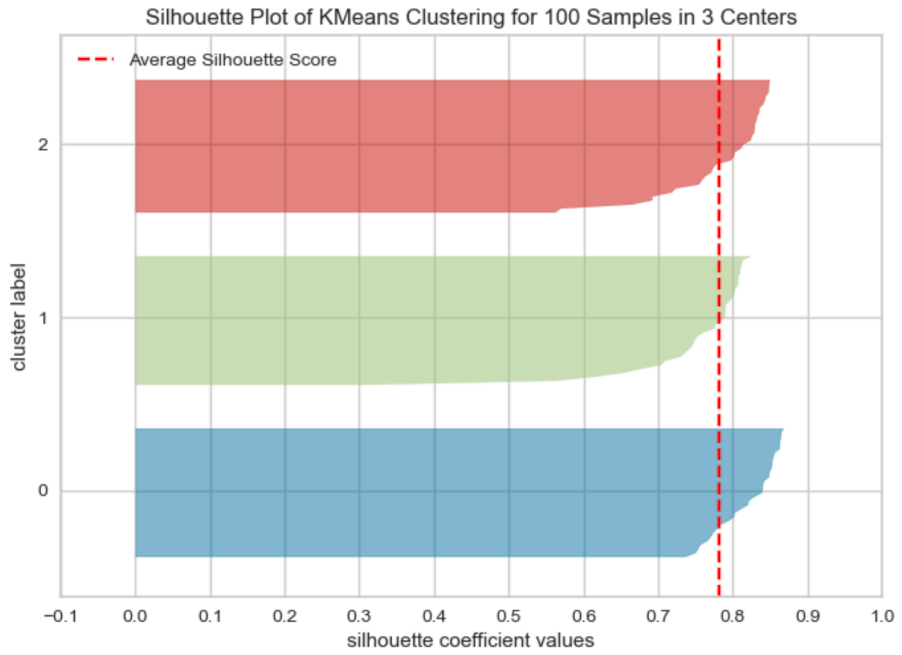
Interpreting silhouette scores

Score	What it suggests
Near **1**	Own-cluster distances are small relative to competing-cluster distances
Near **0**	Own and nearest-competing cluster distances are similar
Negative	A competing cluster is closer on average

Higher average scores suggest more cohesive, separated groups **under the chosen distance**.

They do not establish that the groups match our intended meaning.

Reading a silhouette plot



Each band is a cluster; its thickness shows its size. Horizontal position is the score; the dashed line is the overall mean.

Your turn: compare candidate clusterings

Illustrative results on the same representation:

Fit	k	Inertia	Mean silhouette
A	3	120	0.62
B	5	80	0.41

1. In B, one point has $a = 6, b = 2$. Calculate its silhouette. What does it suggest?
2. Which fit would you investigate first? Does either score settle the choice?
3. What would you inspect before recommending a grouping?

iClicker: choosing k

Which statements are true? Select all that apply.

- a. Every dataset has one obvious elbow.
- b. Silhouette compares point-to-point distances, so it can assess methods without centroids.
- c. A negative silhouette suggests a competing group is closer on average.
- d. The largest silhouette score proves the clusters have useful real-world meanings.
- e. Elbow and silhouette diagnostics should be combined with inspection.

Class demo: image clustering

K-Means demo notebook

1. Compare bird clusters from **pixels** and **pretrained features**.
2. Inspect the more varied **food images**.
3. Consider what changing k would do to these groups.

Before running: Which visual similarities can the chosen features capture?

Today's takeaways

- **Without targets:** discover groups or learn useful representations.
- **To form groups:** K-Means alternates nearest-centre assignments and mean updates.
- **To get useful results:** consider initialization, scaling, and cluster geometry.
- **To choose k:** combine elbow and silhouette diagnostics with inspection.
- **From the image demo:** changing the representation changes similarity; we interpret the resulting groups.

Exit question

Your image clusters have a **high average silhouette score**, but they group backgrounds together rather than the objects you care about.

What would you investigate or change next, and why?

Explain why changing k alone might not solve the problem.