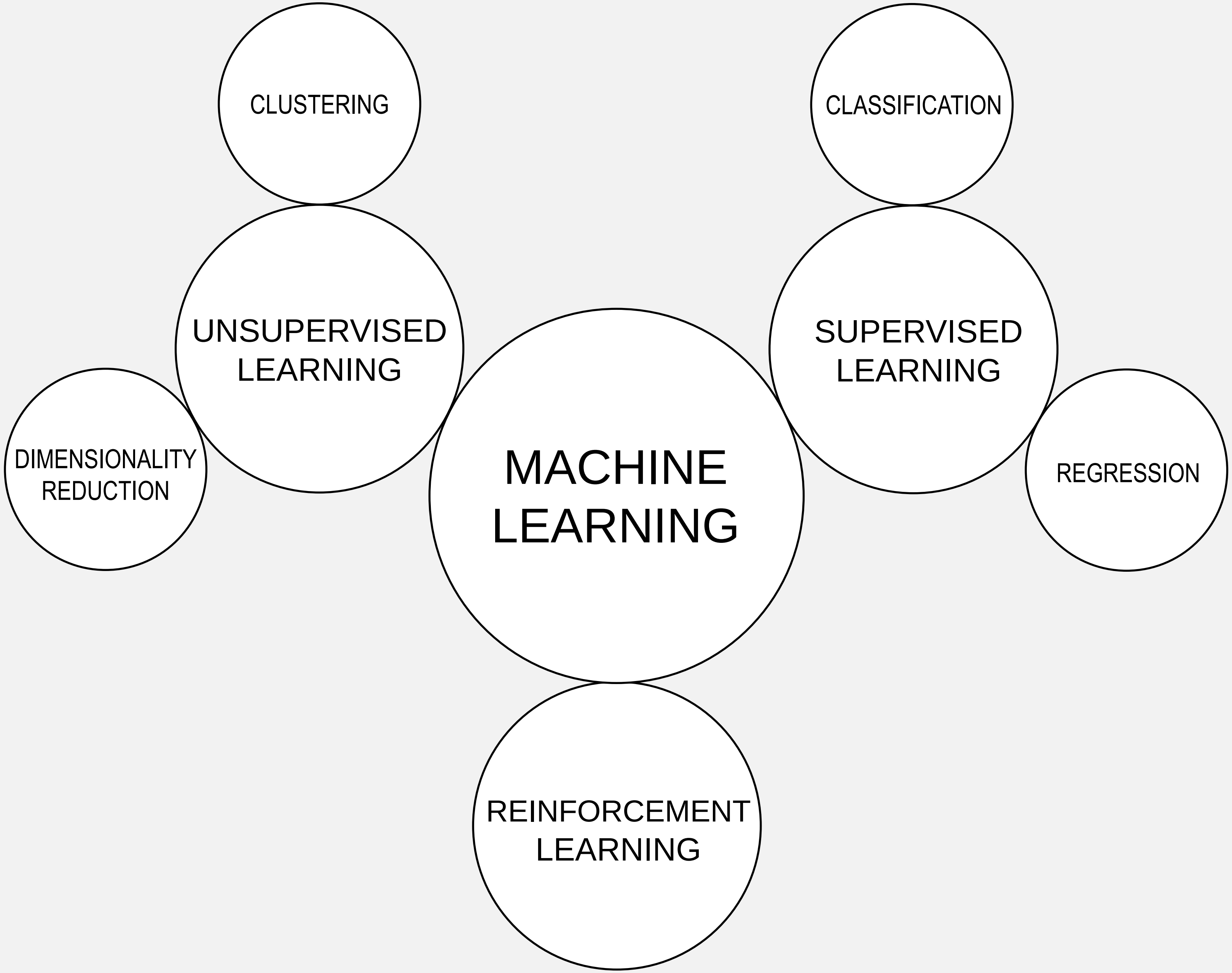
CLUSTERING

Lecture 10

MAL1, 2024


CLUSTERING
CLASSIFICATION
UNSUPERVISED LEARNING
SUPERVISED LEARNING
DIMENSIONALITY REDUCTION
MACHINE LEARNING
REGRESSION
REINFORCEMENT LEARNING

CLUSTERING

CLUSTERING

- What is clustering?
- *k*-means clustering
- Agglomerative clustering
- DBSCAN
- Application

WHAT IS CLUSTERING?

grouping data: unlabeled version of classification

most data in the world is

REVERSE IMAGE SEARCH

I want to know what this bird in my garden is

Google Lens

The corresponding websites tell me it’s a common linnet

REVERSE IMAGE SEARCH











All the images in the dataset …

REVERSE IMAGE SEARCH





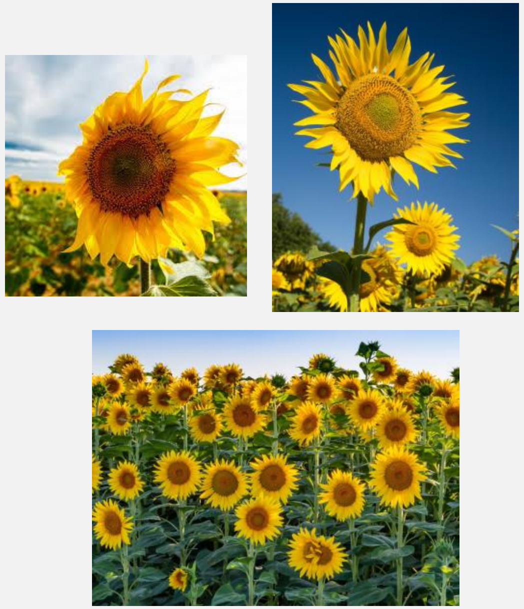


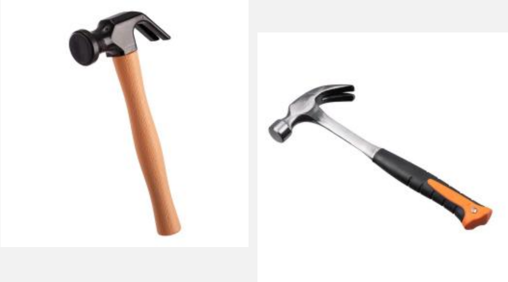




… are **clustered** into groups.

REVERSE IMAGE SEARCH





The image we search with is assigned to a cluster …

REVERSE IMAGE SEARCH

… and the other images in the cluster are returned.

DIFFERENCE FROM CLASSIFICATION?

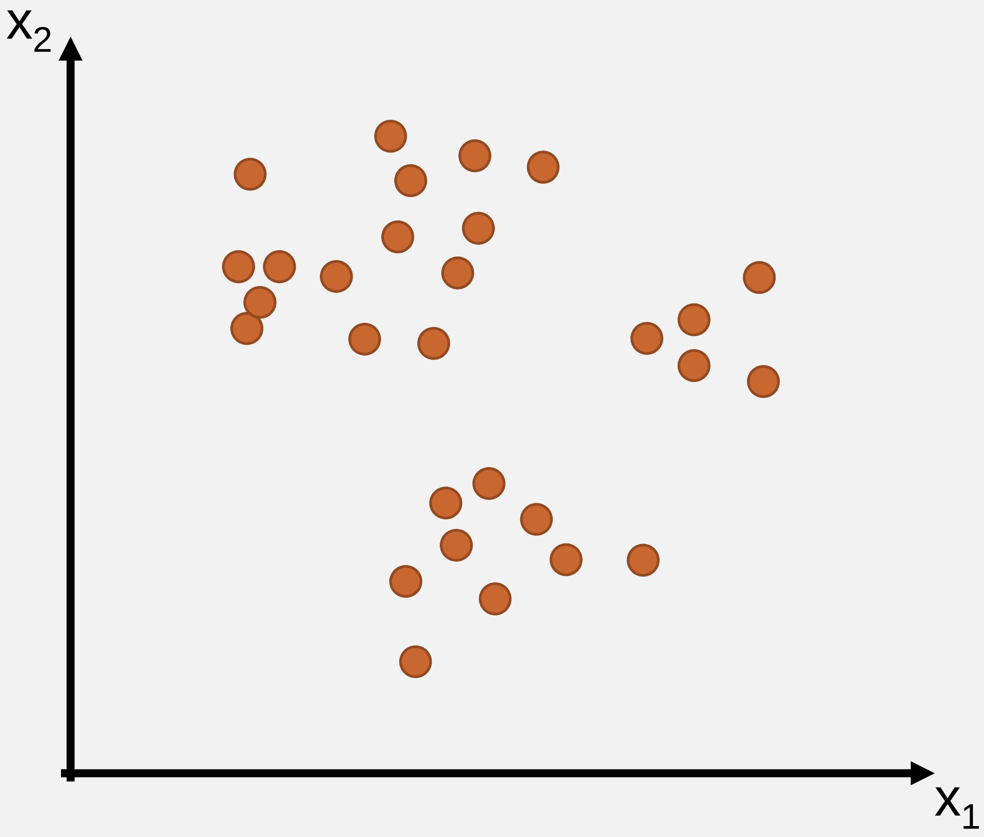
At no point did we label the images – we only care about the fact that some are similar

measure of similarity??

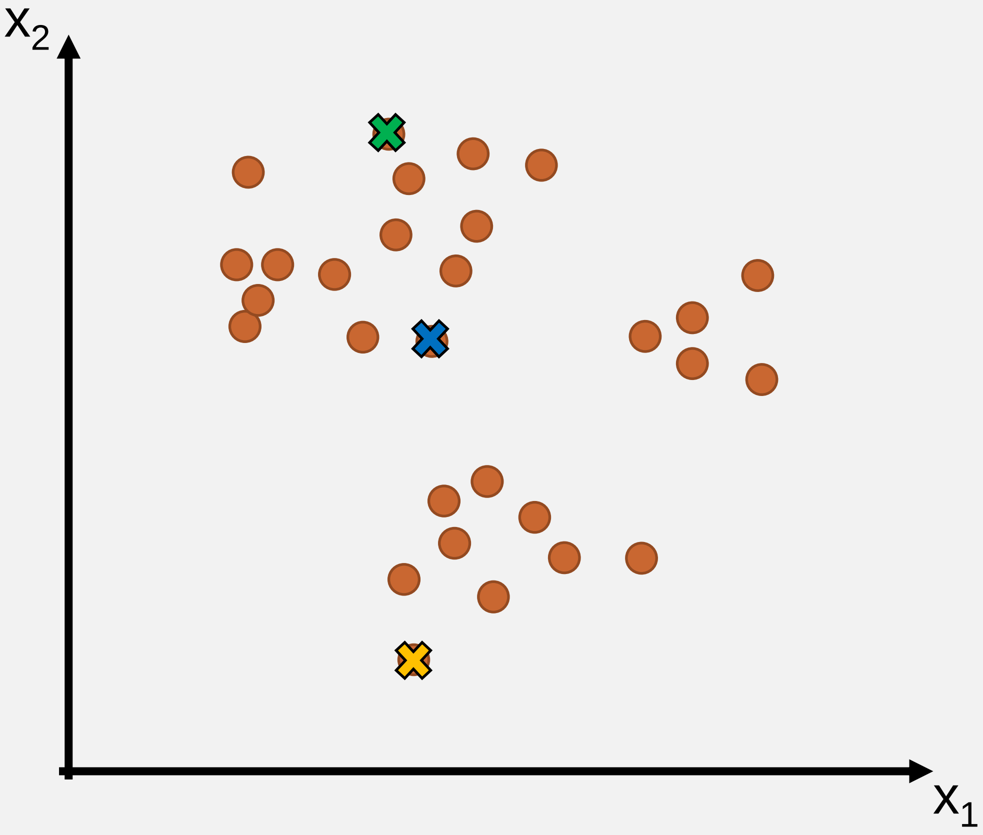
CLUSTERING

- What is clustering?
- *k*-means clustering
- Agglomerative clustering
- DBSCAN
- Application

k-MEANS CLUSTERING

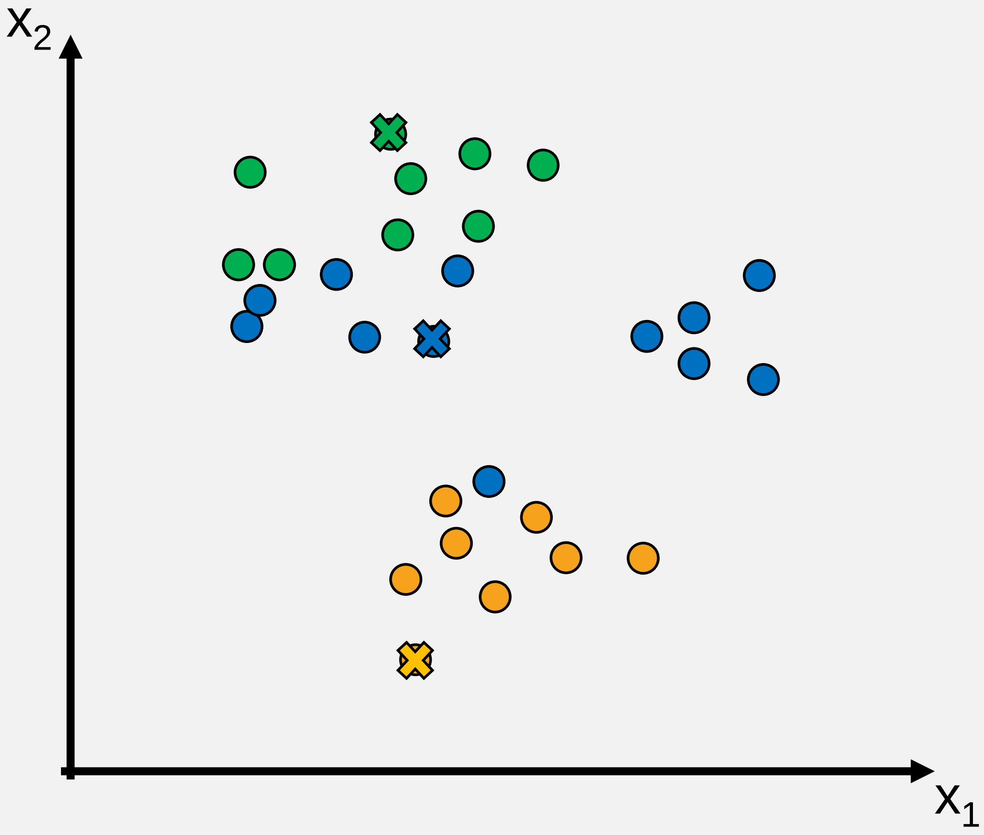


k-MEANS CLUSTERING



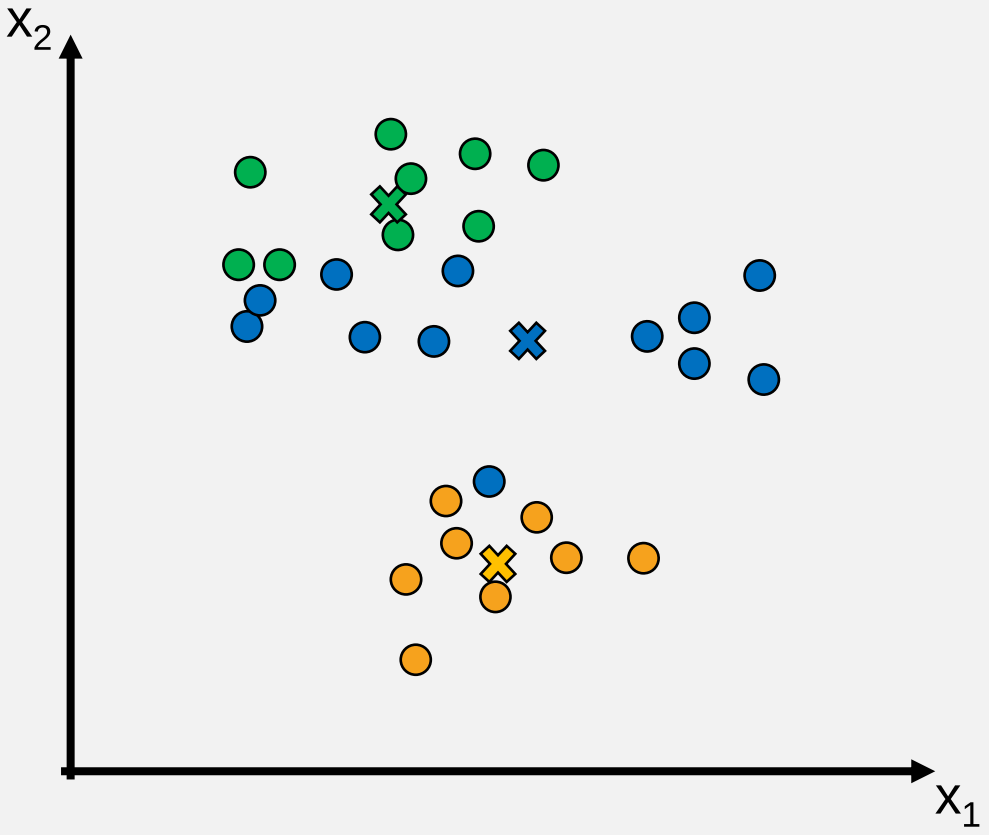
1. Assign k(=3) random points as **centroids**

k-MEANS CLUSTERING



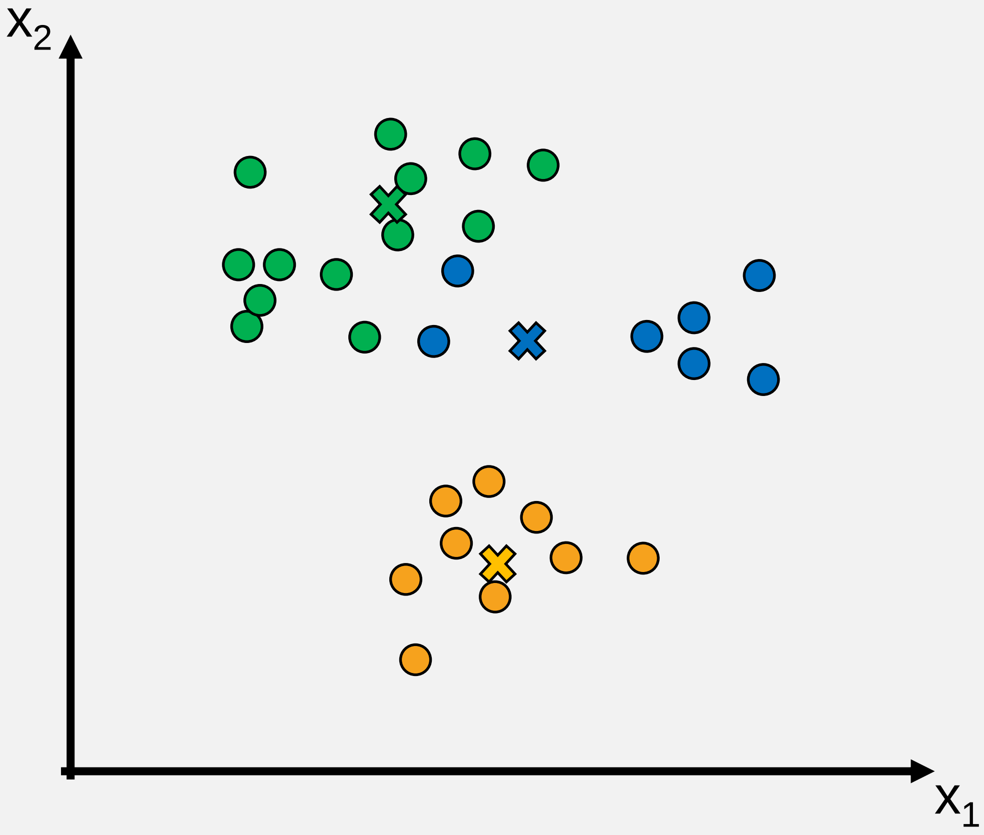
1. Assign k(=3) random points as **centroids**
2. Group the data by their distance to the centroids

k-MEANS CLUSTERING



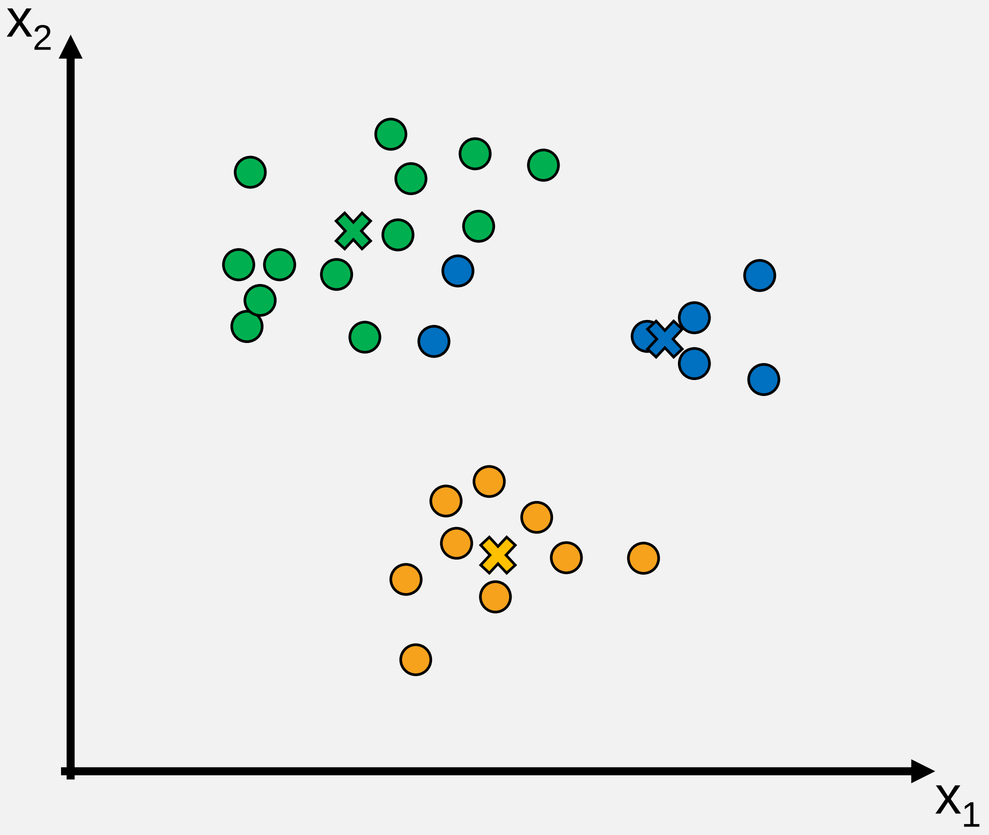
1. Assign k(=3) random points as **centroids**
2. Group the data by their distance to the centroids
3. Move the centroids to the cluster centers

k-MEANS CLUSTERING



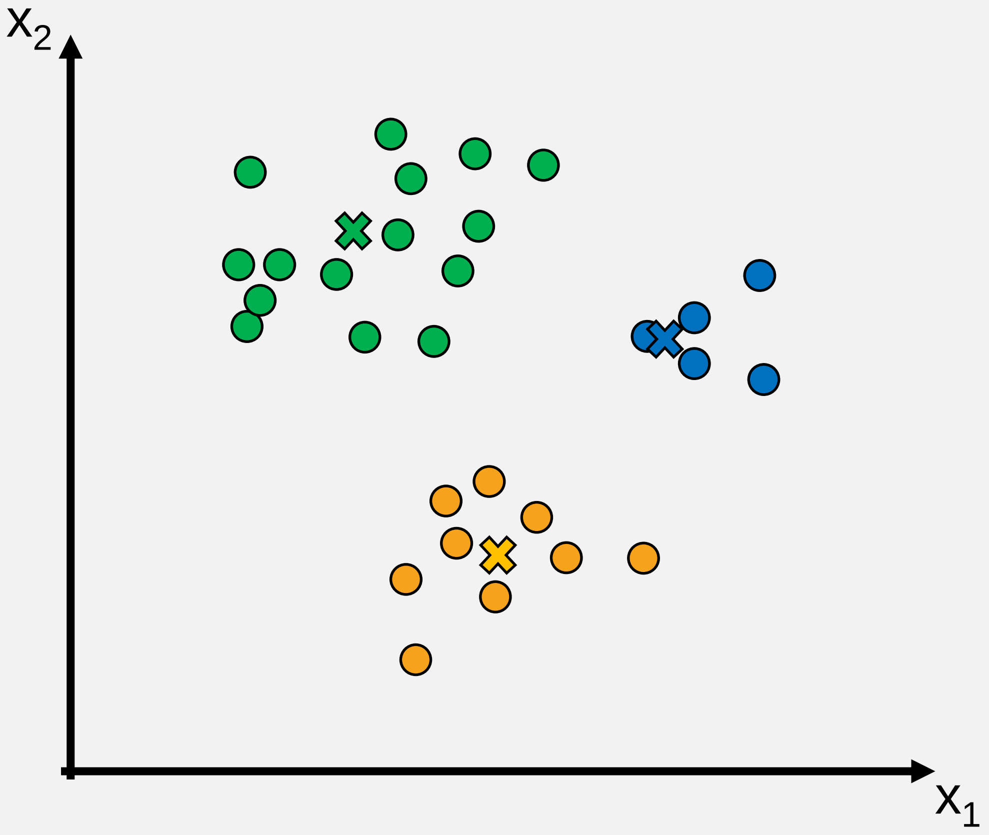
1. Assign k(=3) random points as **centroids**
2. Group the data by their distance to the centroids
3. Move the centroids to the cluster centers
4. Regroup the data

k-MEANS CLUSTERING



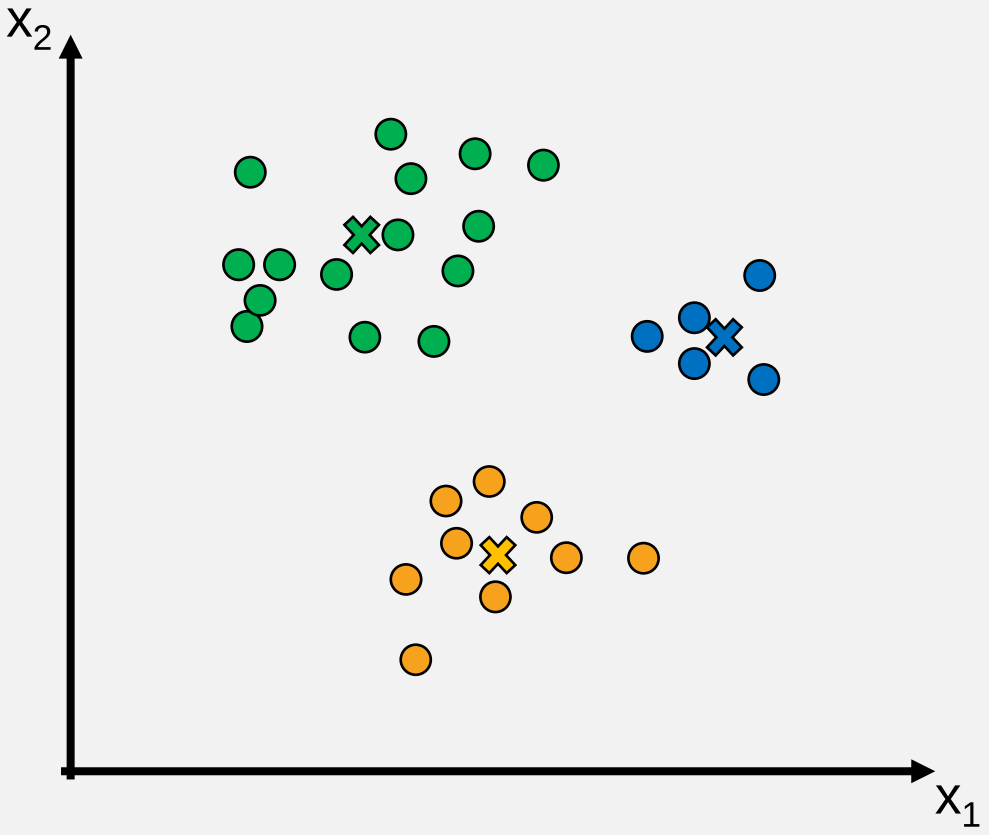
1. Assign k(=3) random points as **centroids**
2. Group the data by their distance to the centroids
3. Move the centroids to the cluster centers
4. Regroup the data
5. Repeat 3-4 until nothing changes

k-MEANS CLUSTERING



1. Assign k(=3) random points as **centroids**
2. Group the data by their distance to the centroids
3. Move the centroids to the cluster centers
4. Regroup the data
5. Repeat 3-4 until nothing changes

k-MEANS CLUSTERING



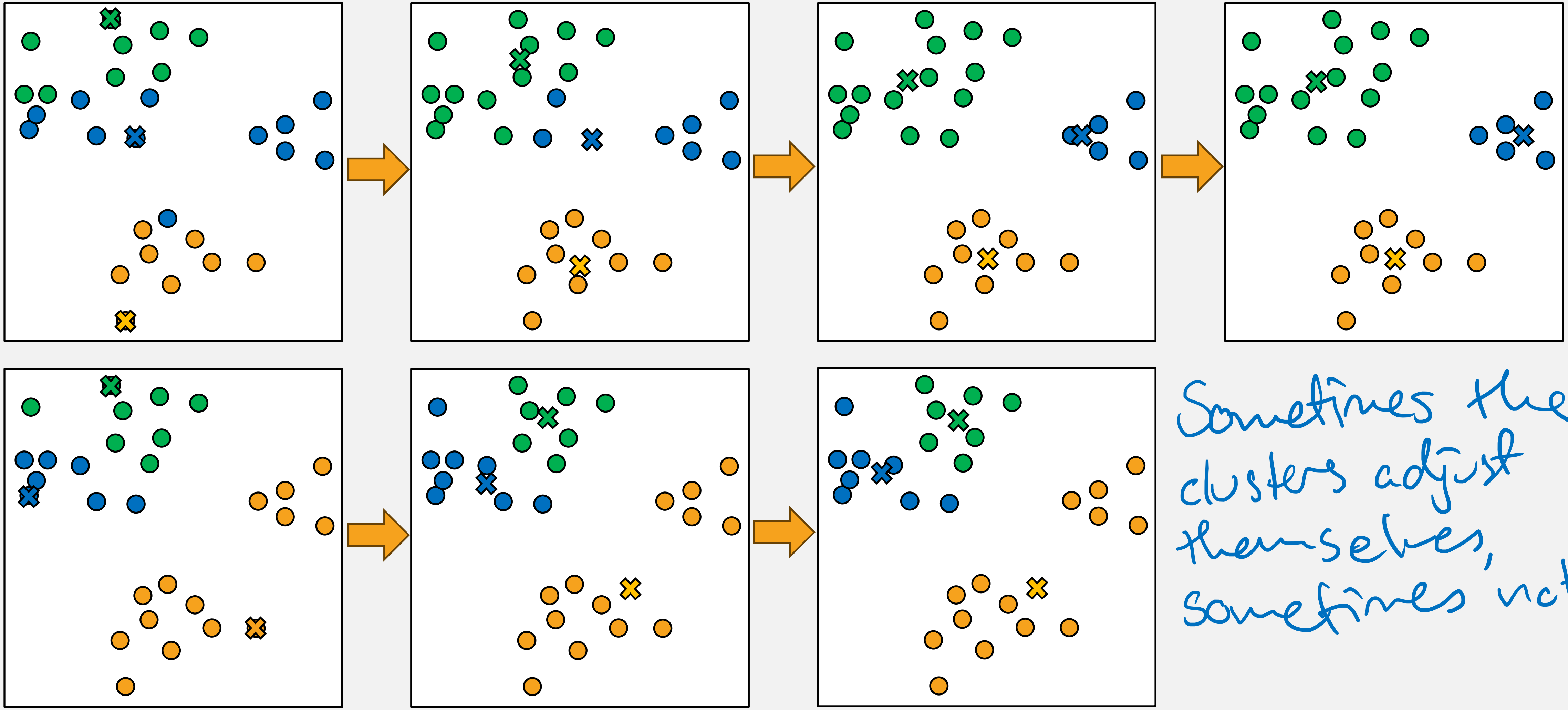
1. Assign k(=3) random points as **centroids**
2. Group the data by their distance to the centroids
3. Move the centroids to the cluster centers
4. Regroup the data
5. Repeat 3-4 until nothing changes

A FEW THINGS WE HAVE TO DEAL WITH

The value of k

The initial centroids

THE INITIAL CENTROIDS



Sometimes the clusters adjust themselves, sometimes not

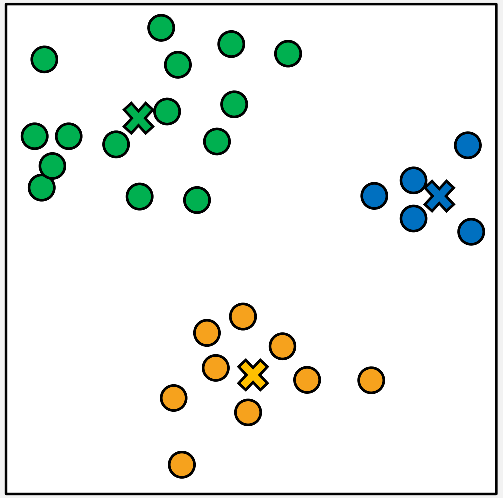
THE INITIAL CENTROIDS

Solution 1: Try different, randomized initializations and compare the **costs** of the final clusterings

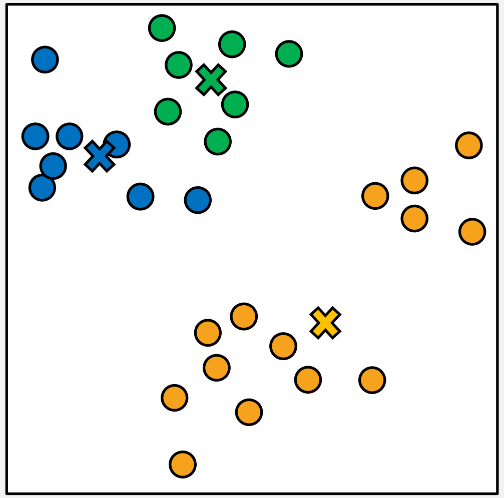
data point — associated centroid

cost function $C = \sum_i \| x_i - \mu(x_i) \|^2$

lower cost ⇒ better



$C = 157.6$



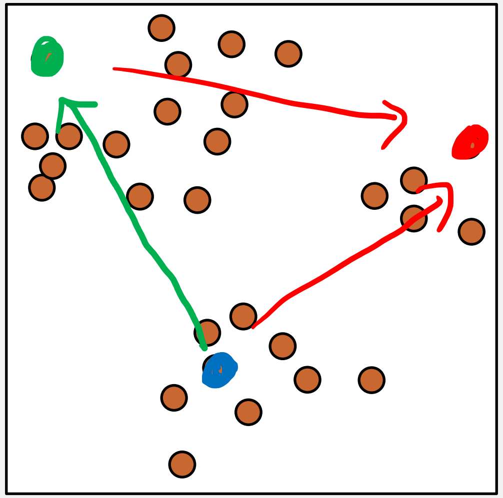
$C = 223.2$

THE INITIAL CENTROIDS

Solution 2: Choose the initial centroids based on the distance to the previous ones

Start randomly then choose point furthest away

and do the same again

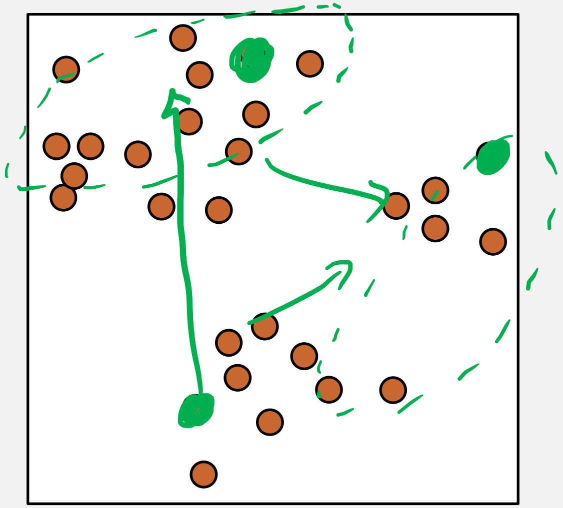


may choose outliers!

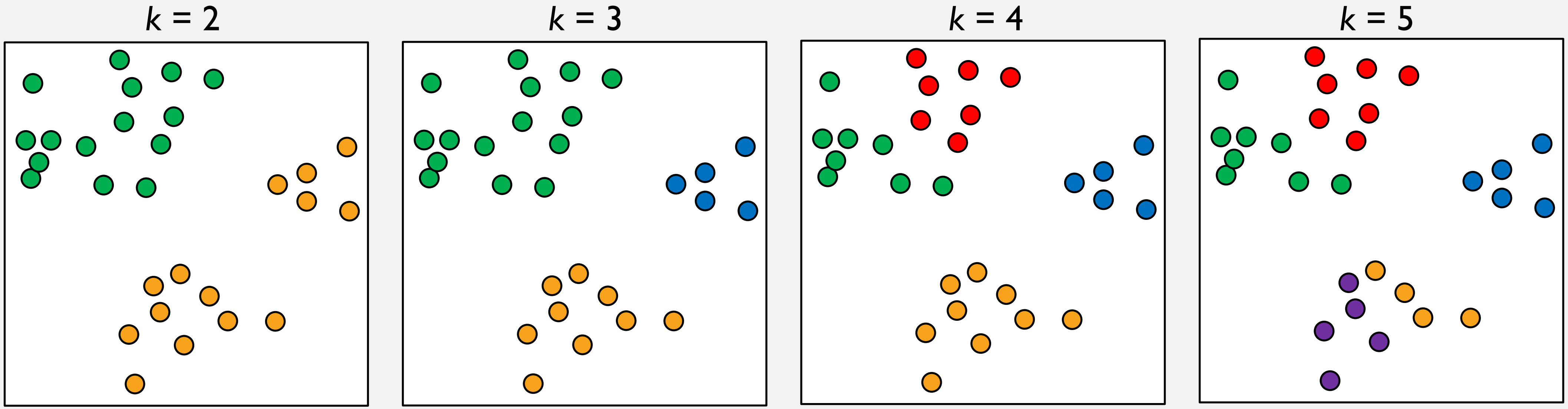
THE INITIAL CENTROIDS

Solution 3: Choose "far away but random" points ("k-means++")

probability of next point high
when far away



THE NUMBER OF CLUSTERS (k)



How do we decide appropriate k?

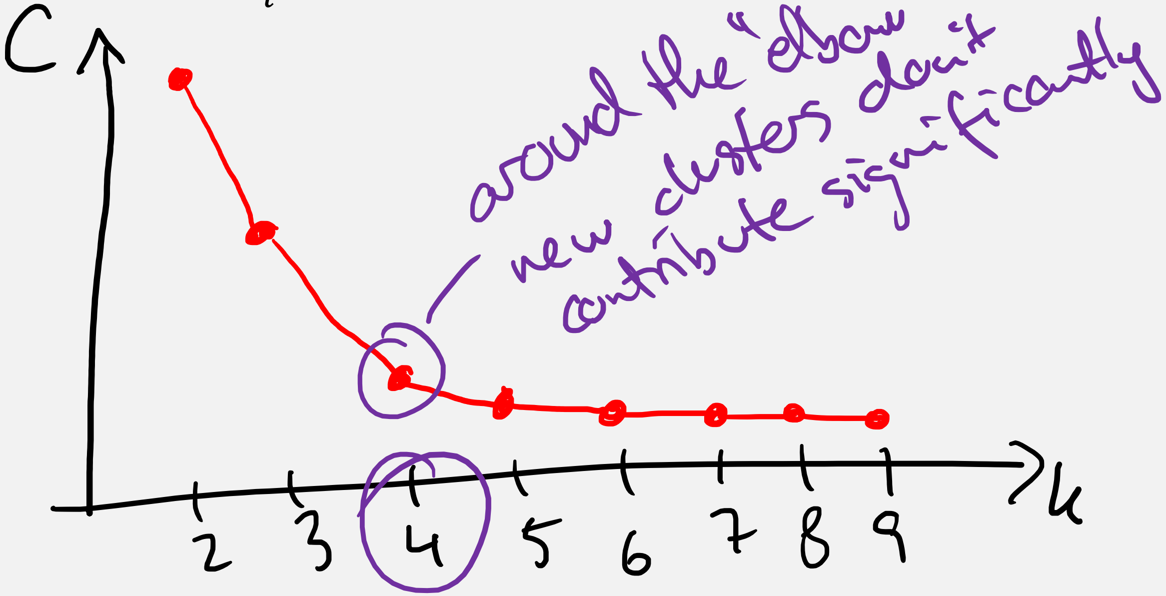
THE NUMBER OF CLUSTERS (k)

The easy way: We already know it (domain knowledge)

The hard way:

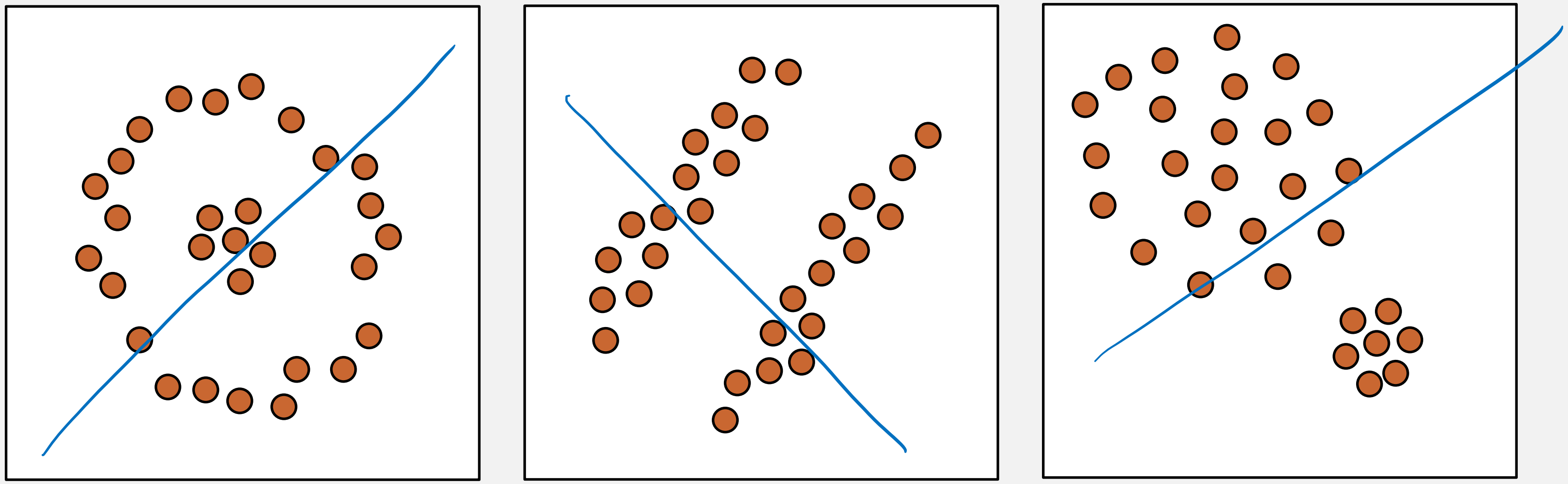
$$C = \sum_i ||x_i - \mu(x_i)||^2$$

always decreases with k



"The elbow method"

WHERE k-MEANS FAILS



CODE EXAMPLE

Jupyter Notebook **Clustering methods**

CLUSTERING

- What is clustering?
- *k*-means clustering
- Agglomerative clustering
- DBSCAN
- Application

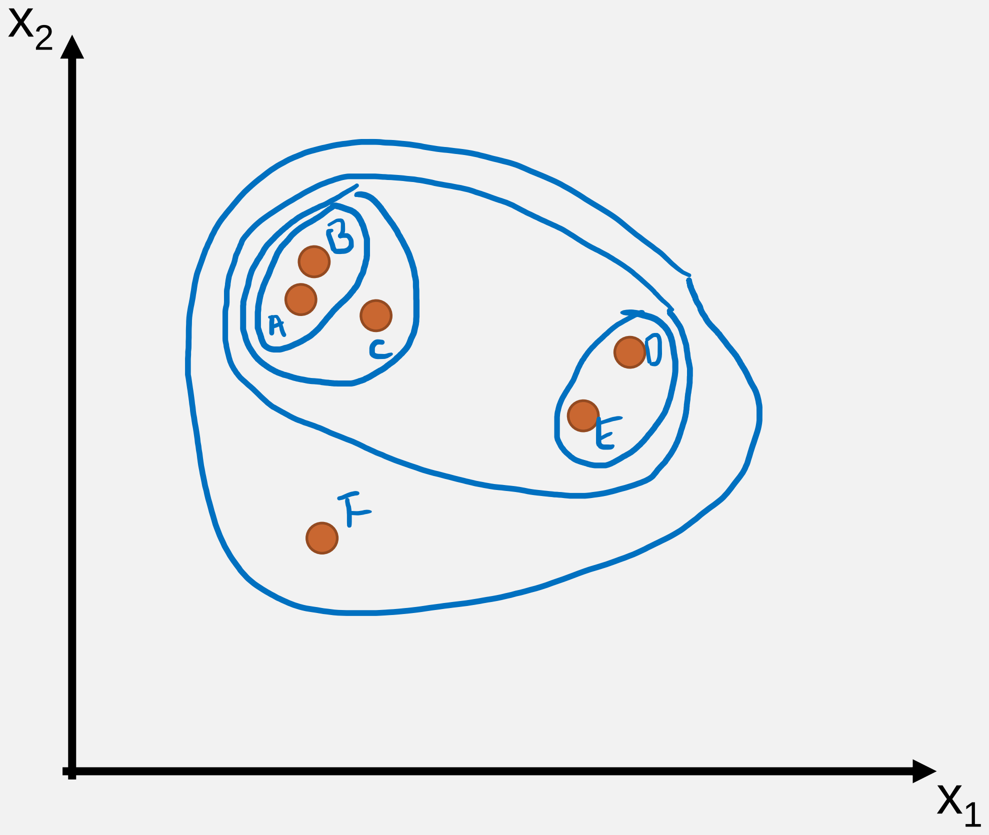
AGGLOMERATIVE CLUSTERING

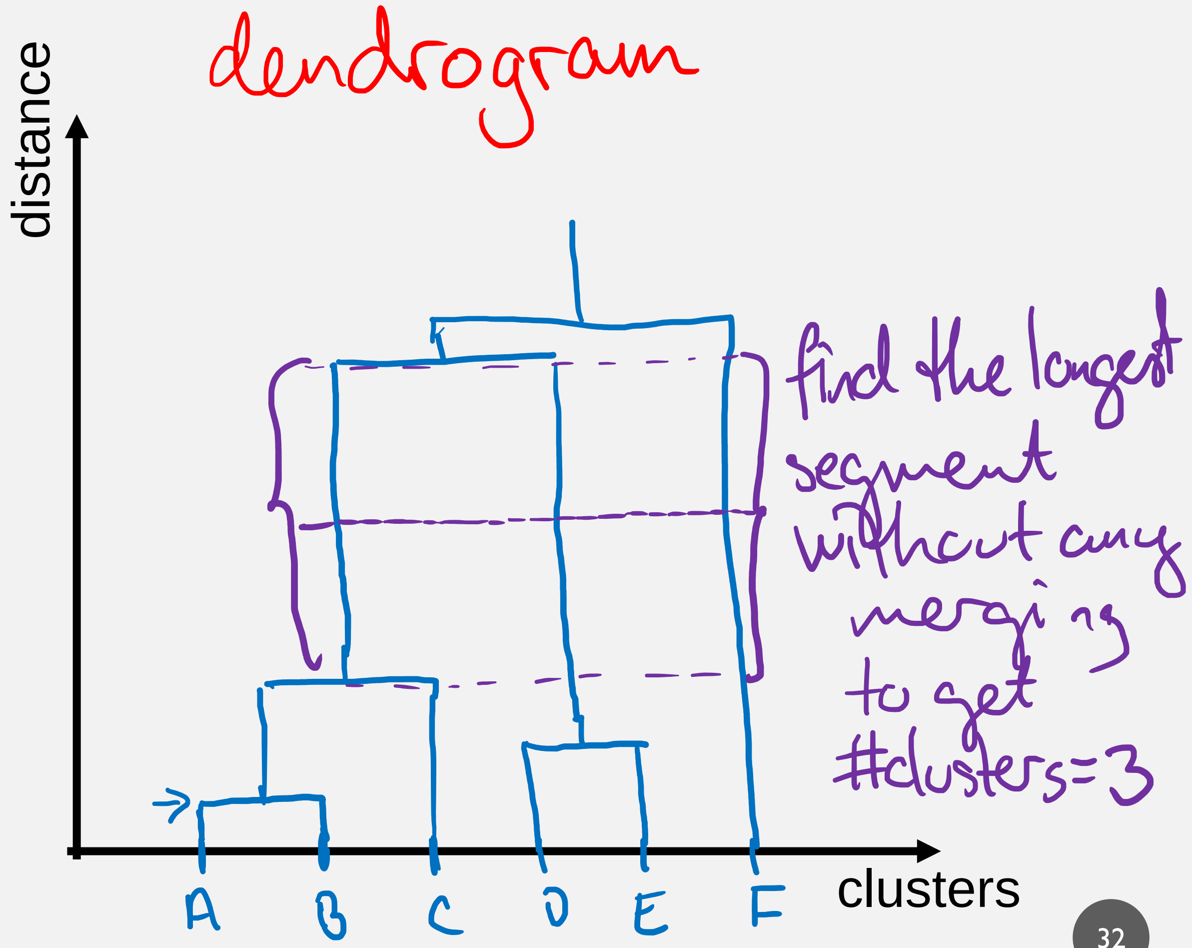
let each point be its own cluster

while there is more than 1 cluster:

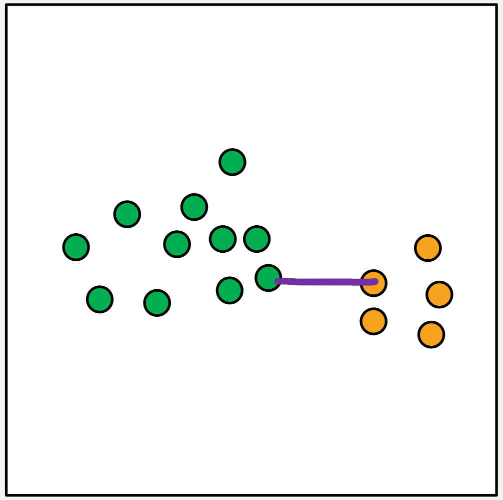
merge the two closest clusters

AGGLOMERATIVE CLUSTERING

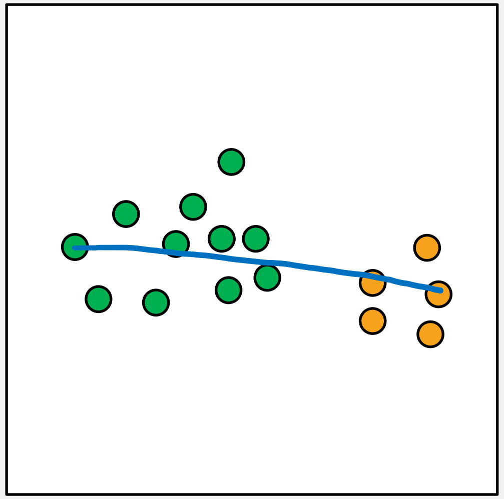




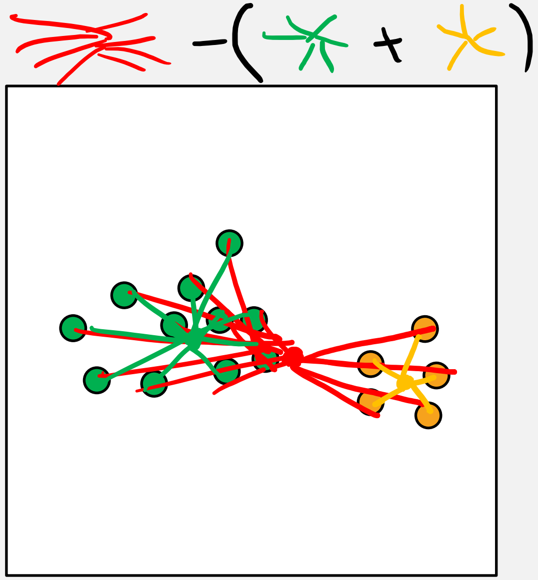
THE DISTANCE BETWEEN CLUSTERS



"single link"
(min distance)
→ sensitive to outliers/noise



"complete link"
(max distance)
→ may break large clusters



"Ward's method"
(change in cost function upon merging)
→ difficulty with odd shapes/different sizes

many more: different choices ⇒ different results

CODE EXAMPLE

Jupyter Notebook **Clustering methods**

CLUSTERING

- What is clustering?
- *k*-means clustering
- Agglomerative clustering
- DBSCAN
- Application

DBSCAN

"density-based spatial clustering of applications with noise"

Partition points into dense regions separated by not-so-dense regions

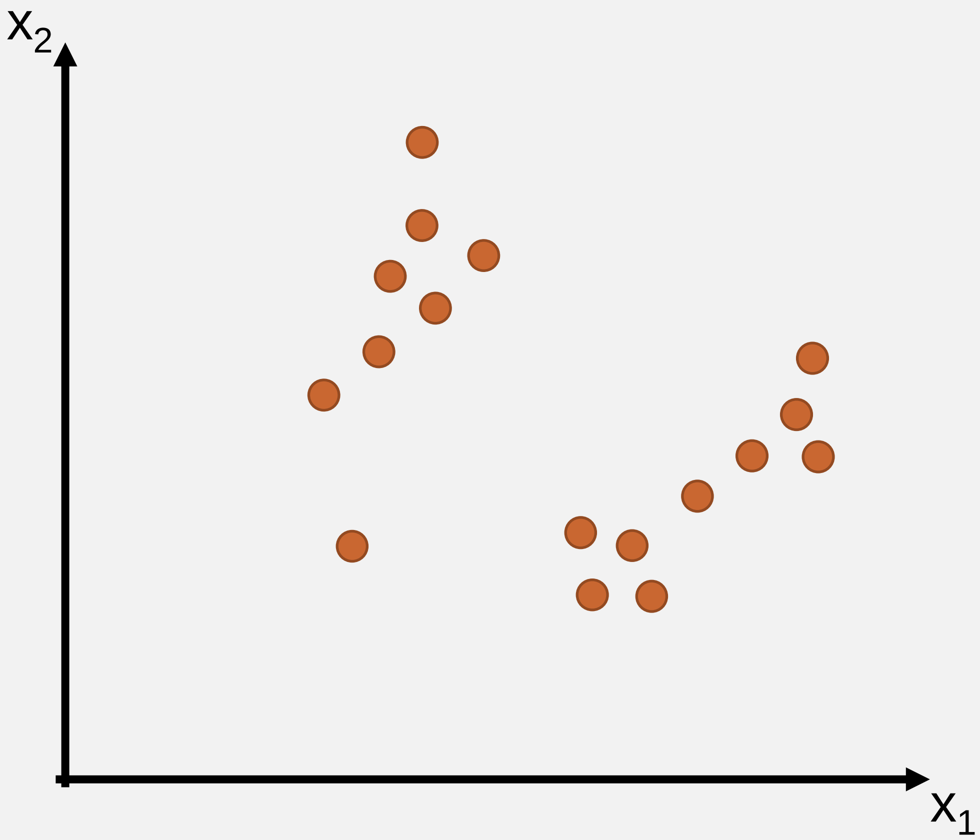
- How do we measure density?

  = number of points in a circle of radius ε

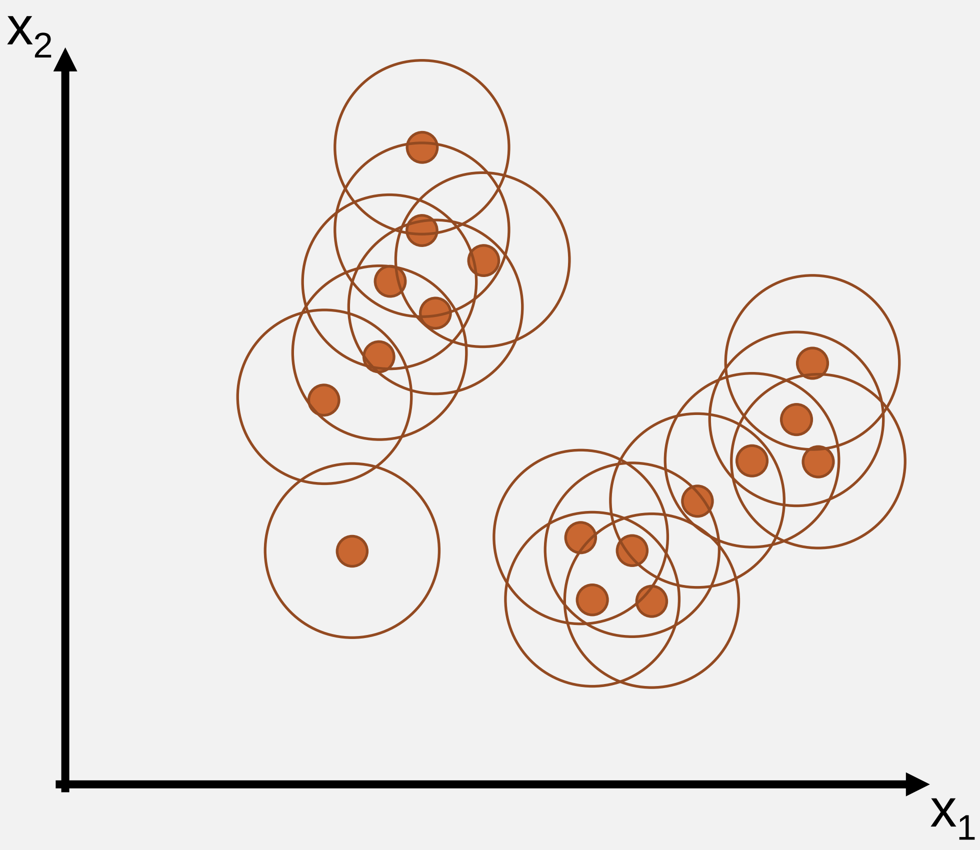
- What is a dense region?

  = density of at least n points

DBSCAN

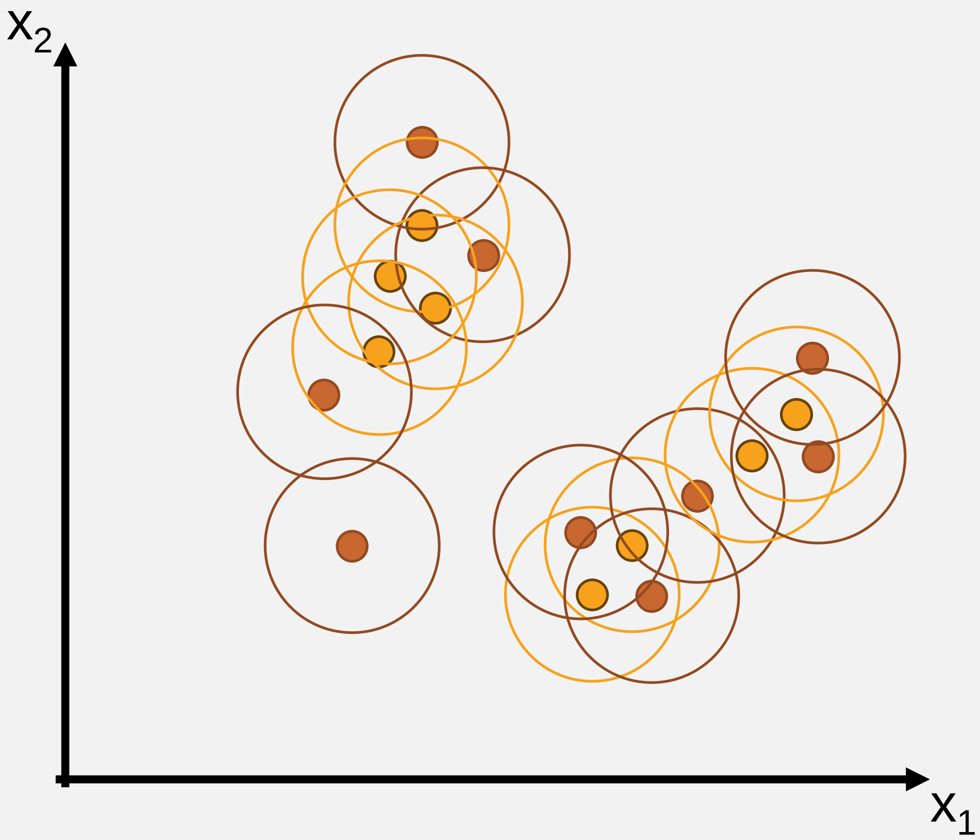


DBSCAN



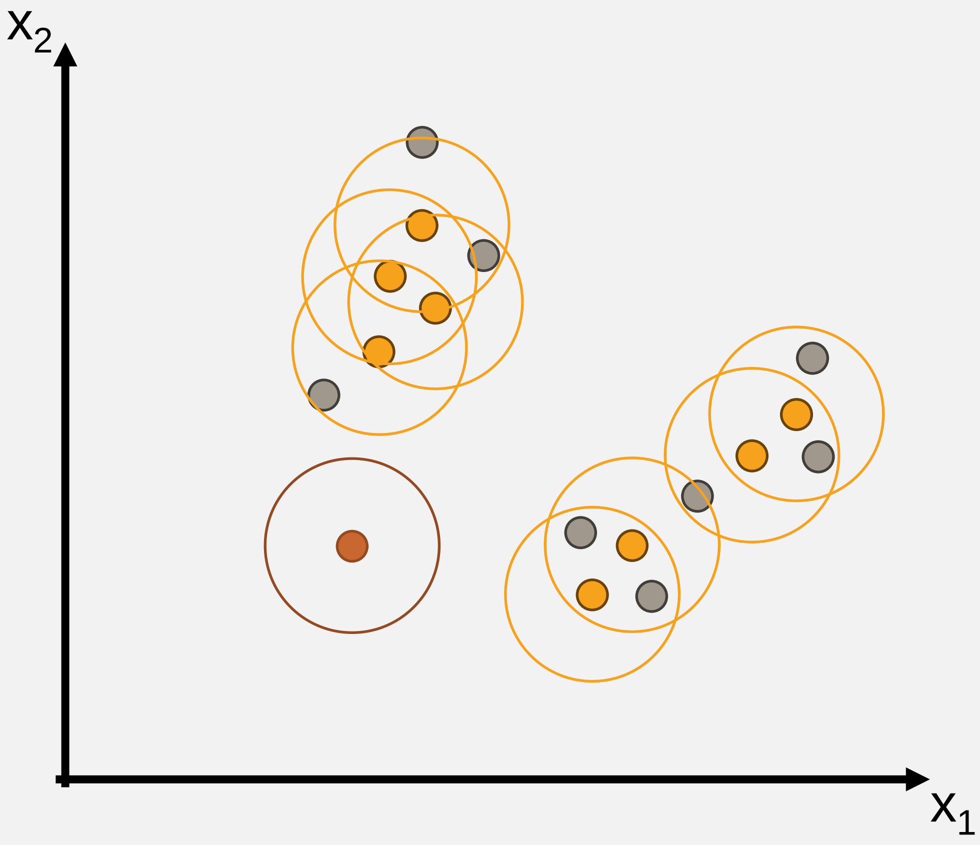
1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.

DBSCAN



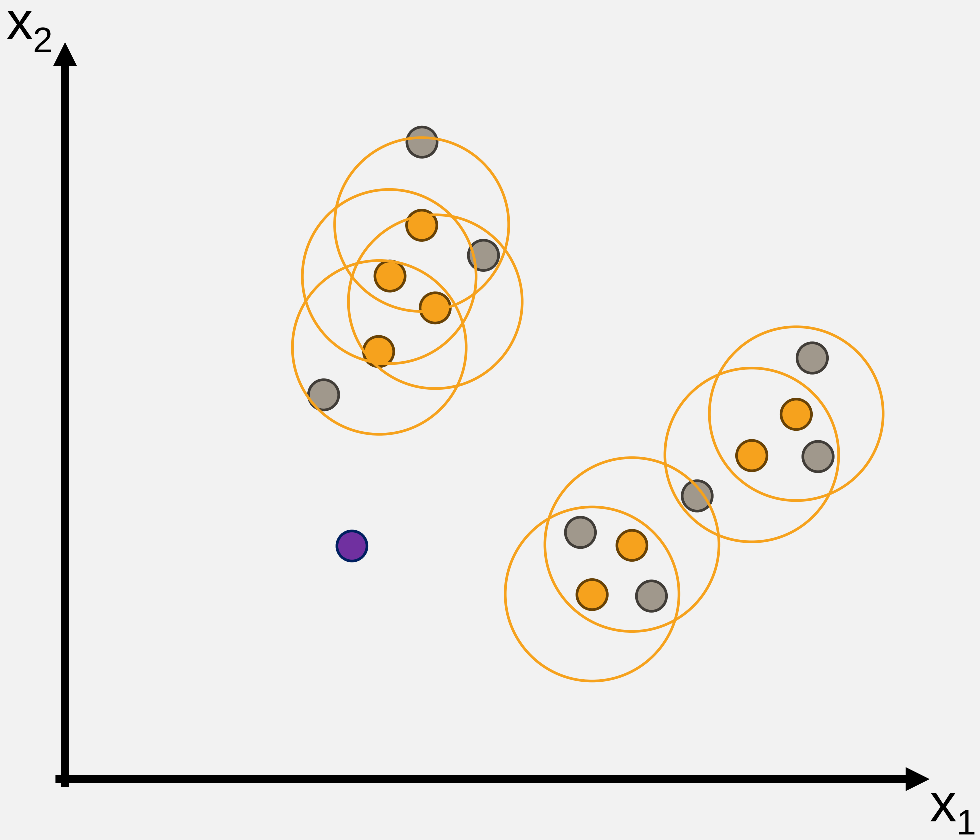
1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.
2. If the ε-neighbourhood contains at least n (=4) points, we consider the point a **core** point.

DBSCAN



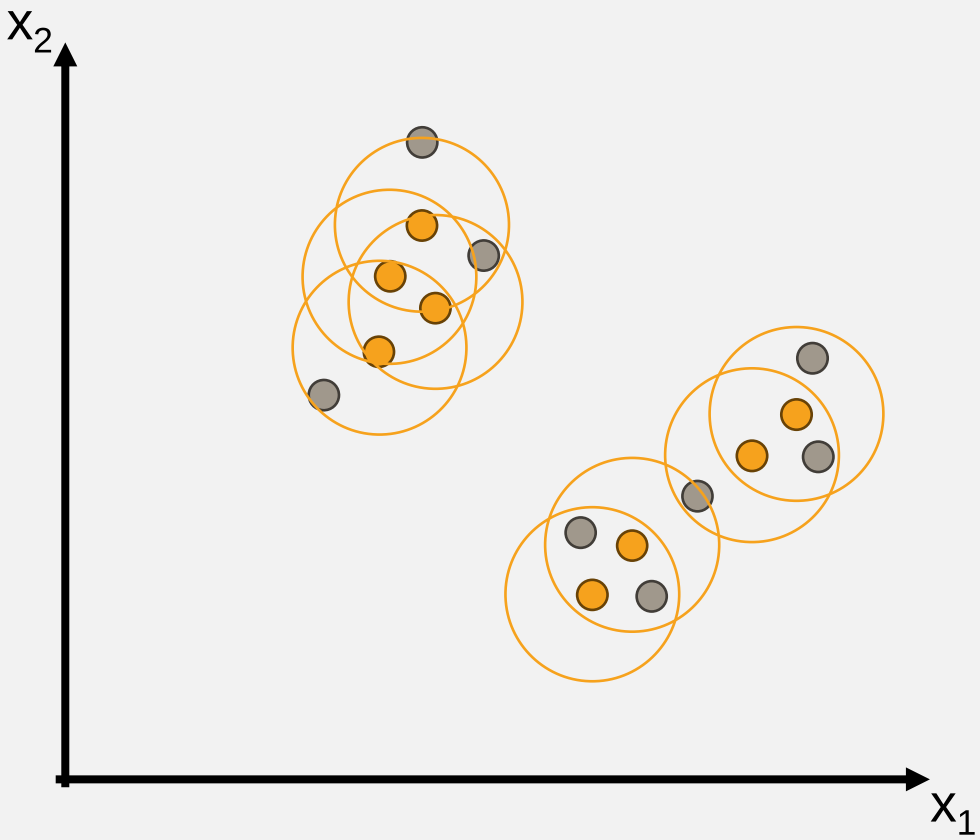
1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.
2. If the ε-neighbourhood contains at least n (=4) points, we consider the point a **core** point.
3. If the point is not a core point, but is in the ε-neighbourhood of one, it is a **border** point.

DBSCAN



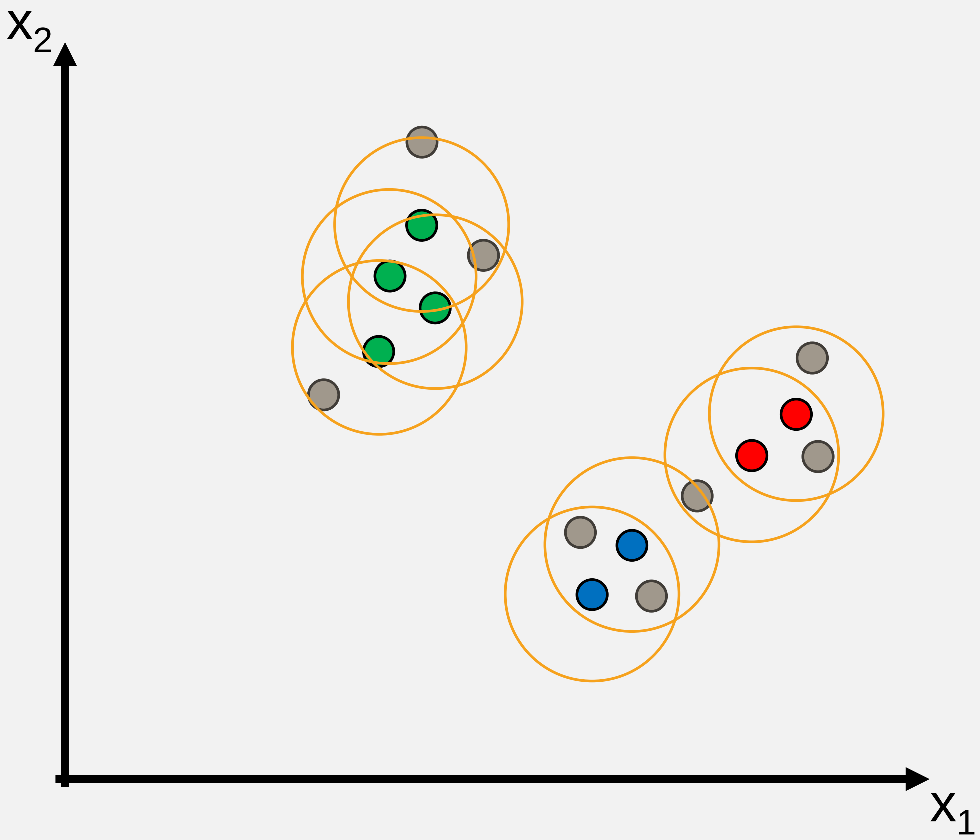
1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.
2. If the ε-neighbourhood contains at least n (=4) points, we consider the point a **core** point.
3. If the point is not a core point, but is in the ε-neighbourhood of one, it is a **border** point.
4. Otherwise, it is a **noise** point.

DBSCAN



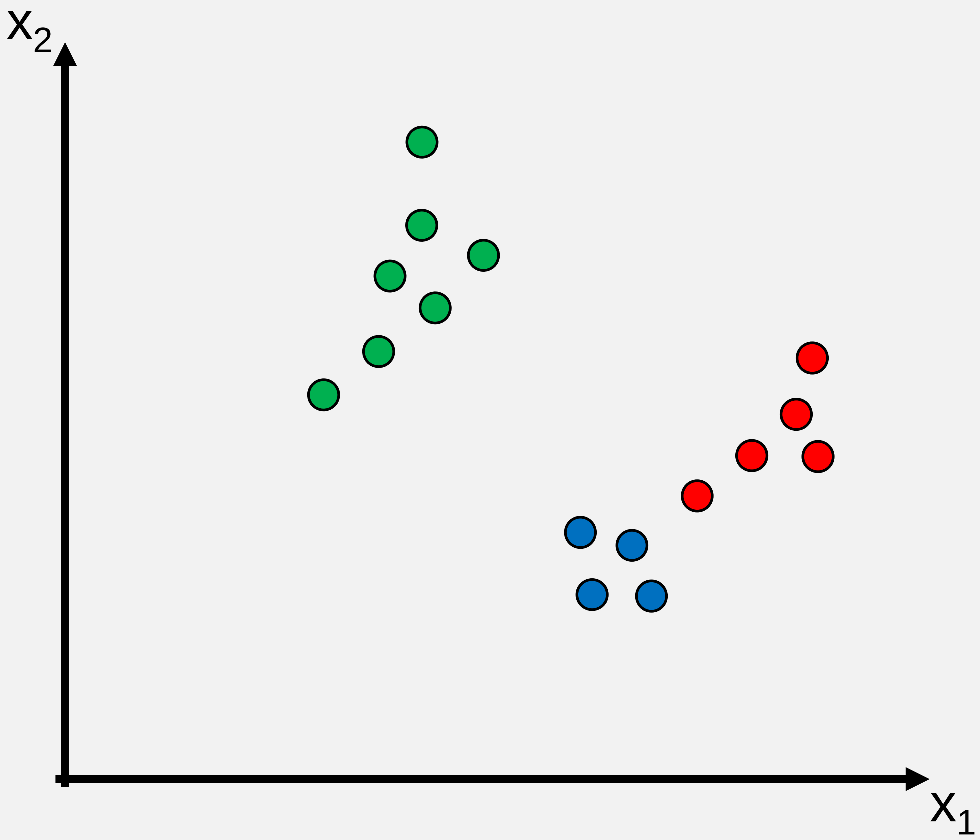
1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.
2. If the ε-neighbourhood contains at least n (=4) points, we consider the point a **core** point.
3. If the point is not a core point, but is in the ε-neighbourhood of one, it is a **border** point.
4. Otherwise, it is a **noise** point.
5. Get rid of **noise** points.

DBSCAN



1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.
2. If the ε-neighbourhood contains at least n (=4) points, we consider the point a **core** point.
3. If the point is not a core point, but is in the ε-neighbourhood of one, it is a **border** point.
4. Otherwise, it is a **noise** point.
5. Get rid of **noise** points.
6. All **core** points reachable through each other's ε-neighbourhoods belong to the same cluster.

DBSCAN



1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.
2. If the ε-neighbourhood contains at least n (=4) points, we consider the point a **core** point.
3. If the point is not a core point, but is in the ε-neighbourhood of one, it is a **border** point.
4. Otherwise, it is a **noise** point.
5. Get rid of **noise** points.
6. All **core** points reachable through each other's ε-neighbourhoods belong to the same cluster.
7. All **border** points are assigned to the cluster of closest core point.

DETERMINING ε AND n
(recommendation)

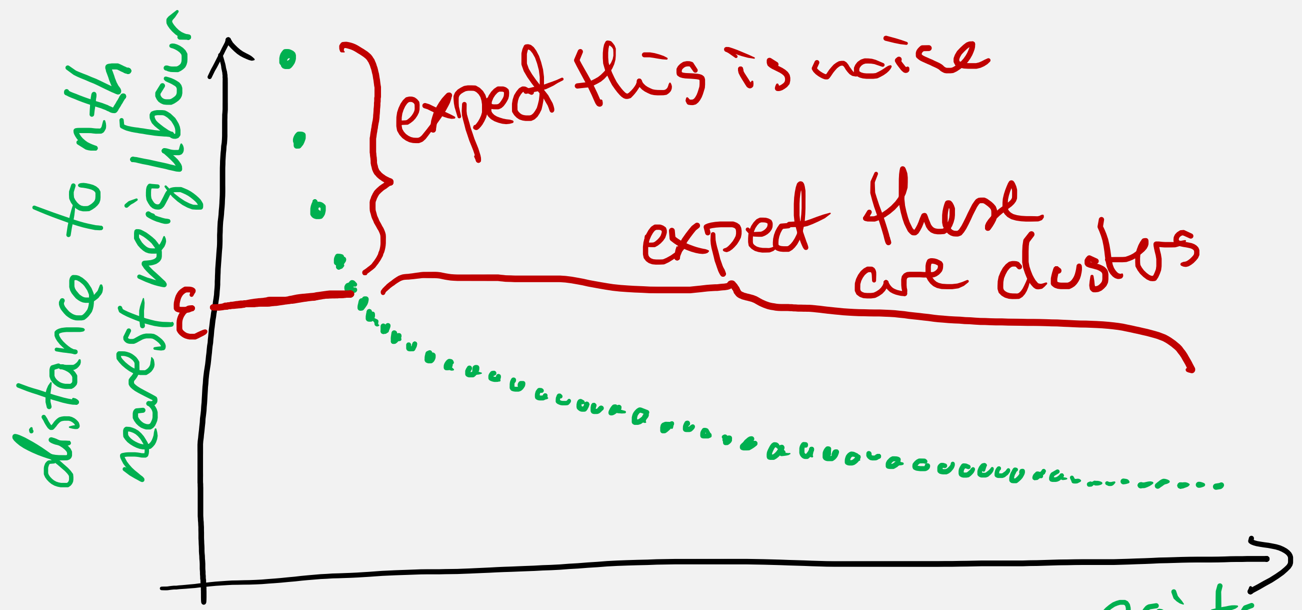
n (minPts) = 2 × d — dimensionality

2D data ⇒ n=4

3D data ⇒ n=6

- - - -

ε

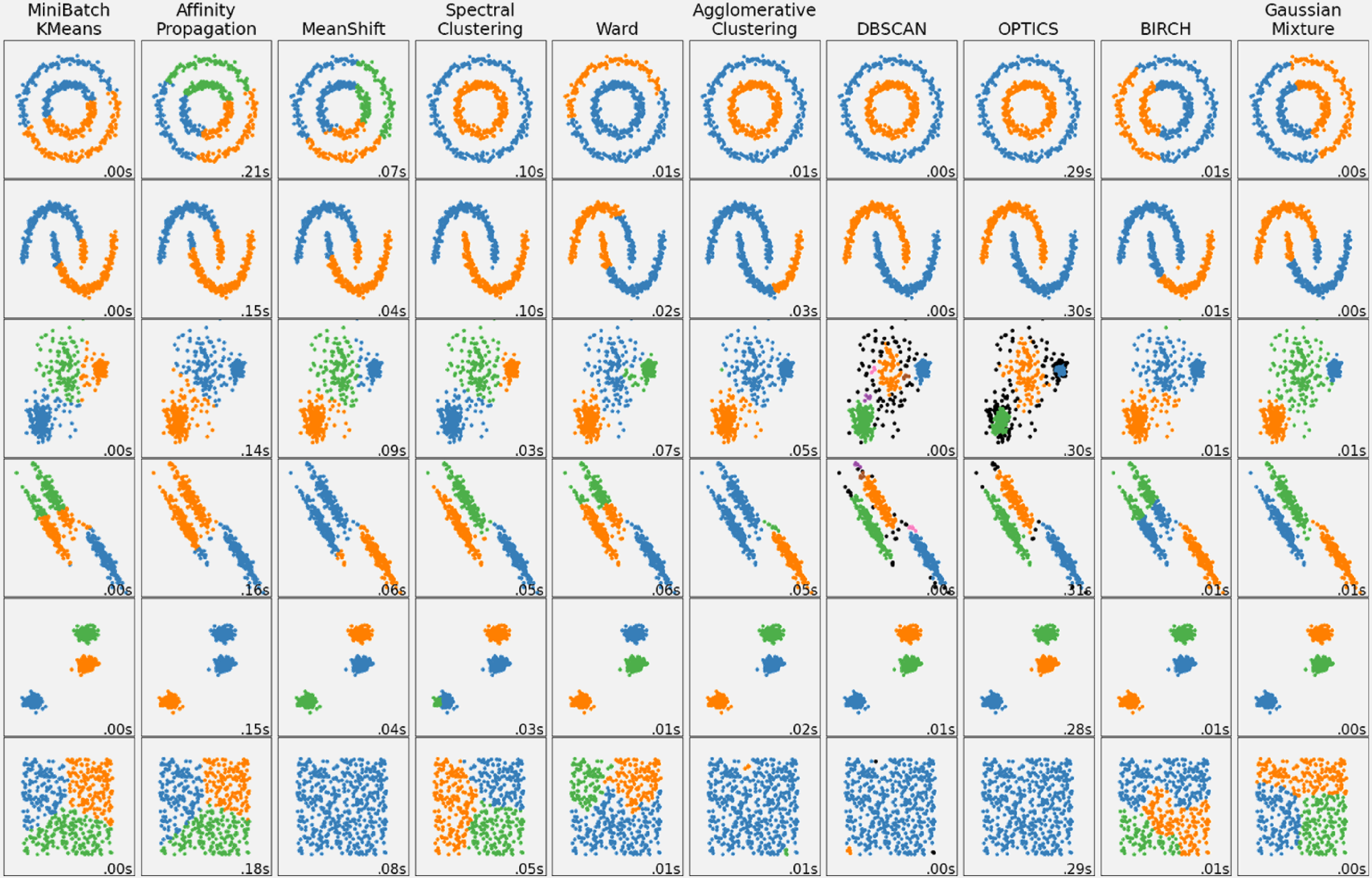


CODE EXAMPLE

Jupyter Notebook **Clustering methods**

COMPARING THE MODELS

	Pros	**Cons**
***k*-means clustering**	Efficient	Cannot handle outliers Cannot handle weird shapes User must provide k (could be ok) Initialization
Agglomerative clustering	No a priori knowledge about #clusters	Each distance metric has its own problem Computationally heavy Dendrograms can be ambiguous
DBSCAN	Arbitrary shapes Deals with outliers No a priori knowledge about #clusters	Trouble w/ diff. densities



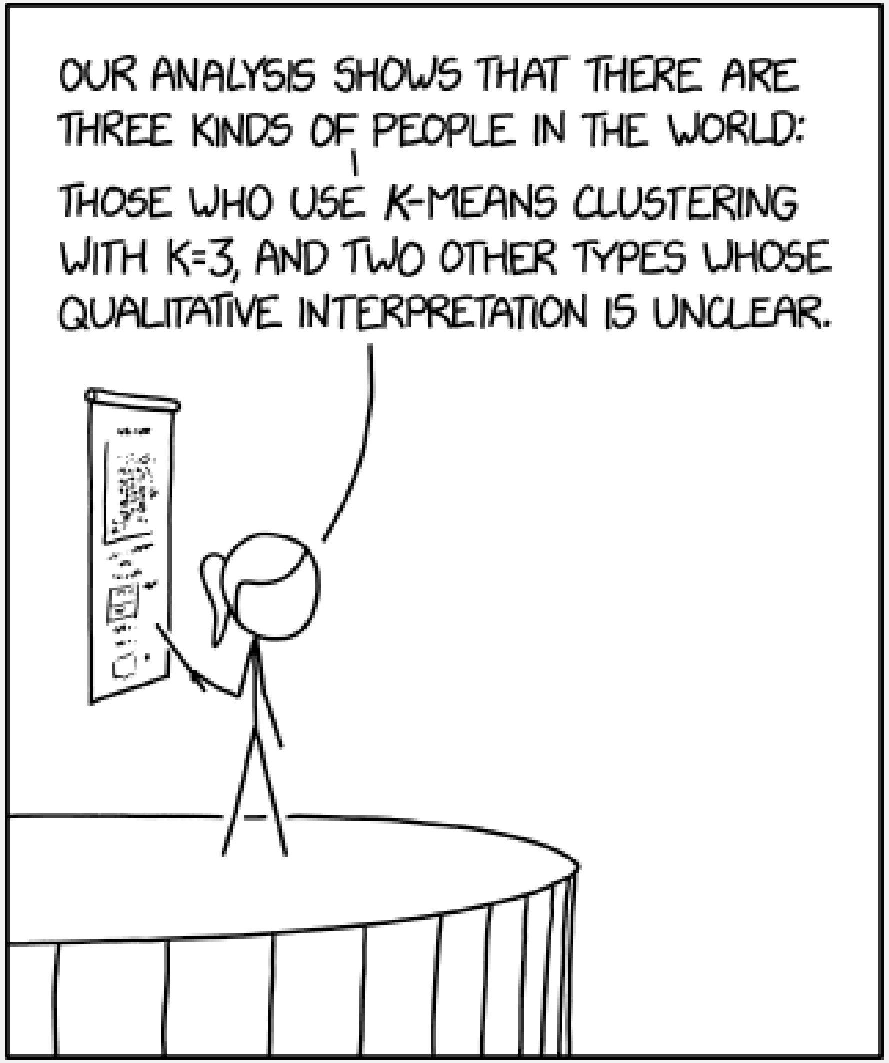
http://scikit-learn.org/stable/modules/clustering.html

CLUSTERING

- What is clustering?
- *k*-means clustering
- Agglomerative clustering
- DBSCAN
- Application

APPLICATION: IMAGE SEGMENTATION

Jupyter Notebook **Image segmentation**


OUR ANALYSIS SHOWS THAT THERE ARE THREE KINDS OF PEOPLE IN THE WORLD:
THOSE WHO USE K-MEANS CLUSTERING WITH K=3, AND TWO OTHER TYPES WHOSE QUALITATIVE INTERPRETATION IS UNCLEAR.