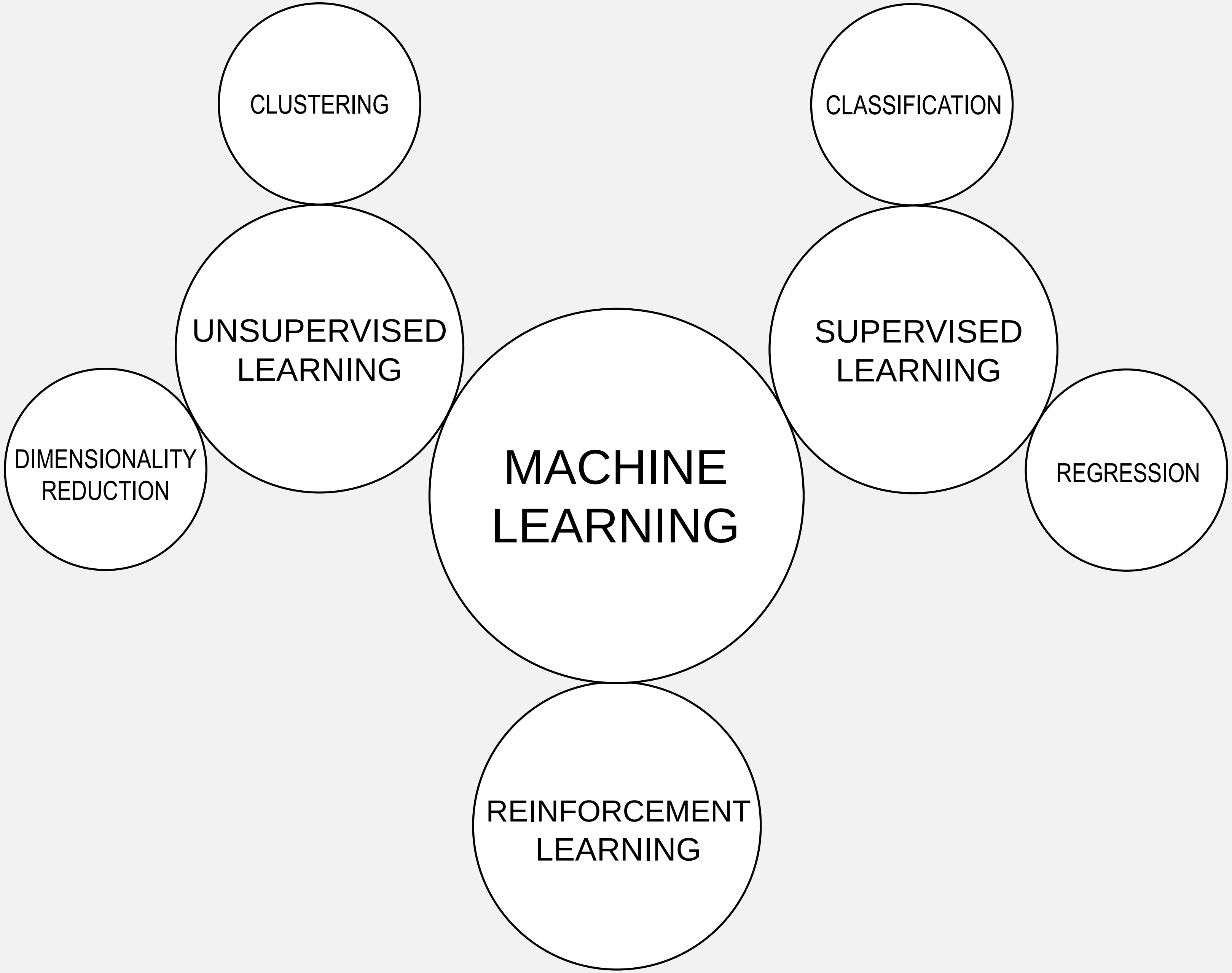
CLUSTERING

Lecture 10

MAL1, 2024


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

CLUSTERING

WHAT IS CLUSTERING?

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





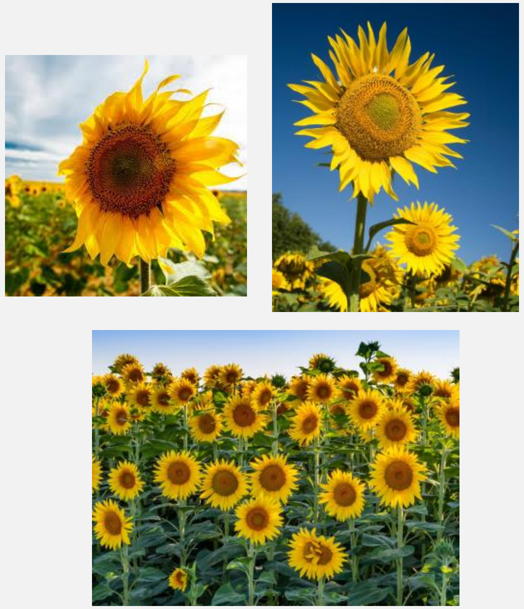


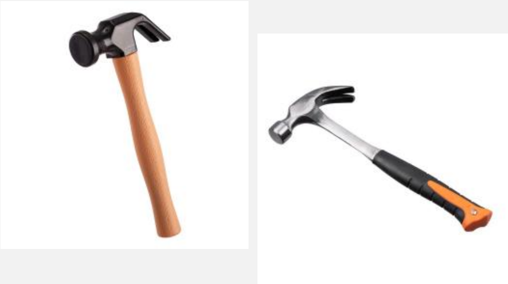




All the images in the dataset …

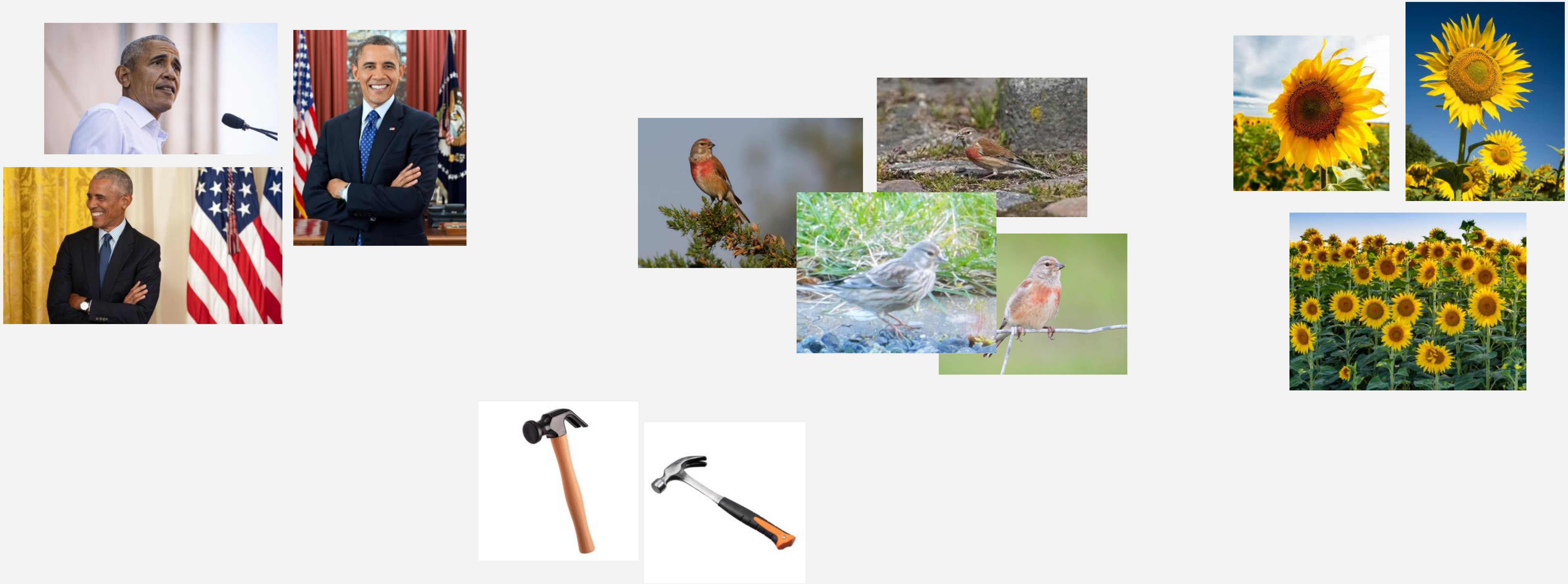
REVERSE IMAGE SEARCH





… are **clustered** into groups.

REVERSE IMAGE SEARCH



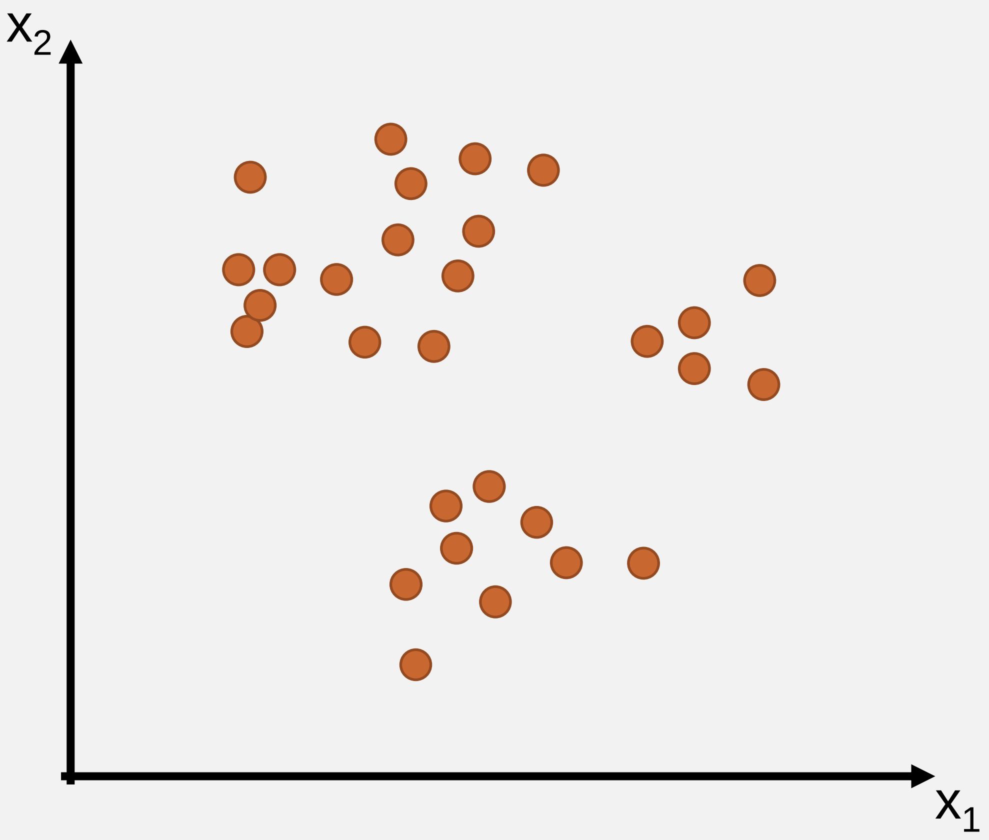
The image we search with is assigned to a cluster …

REVERSE IMAGE SEARCH

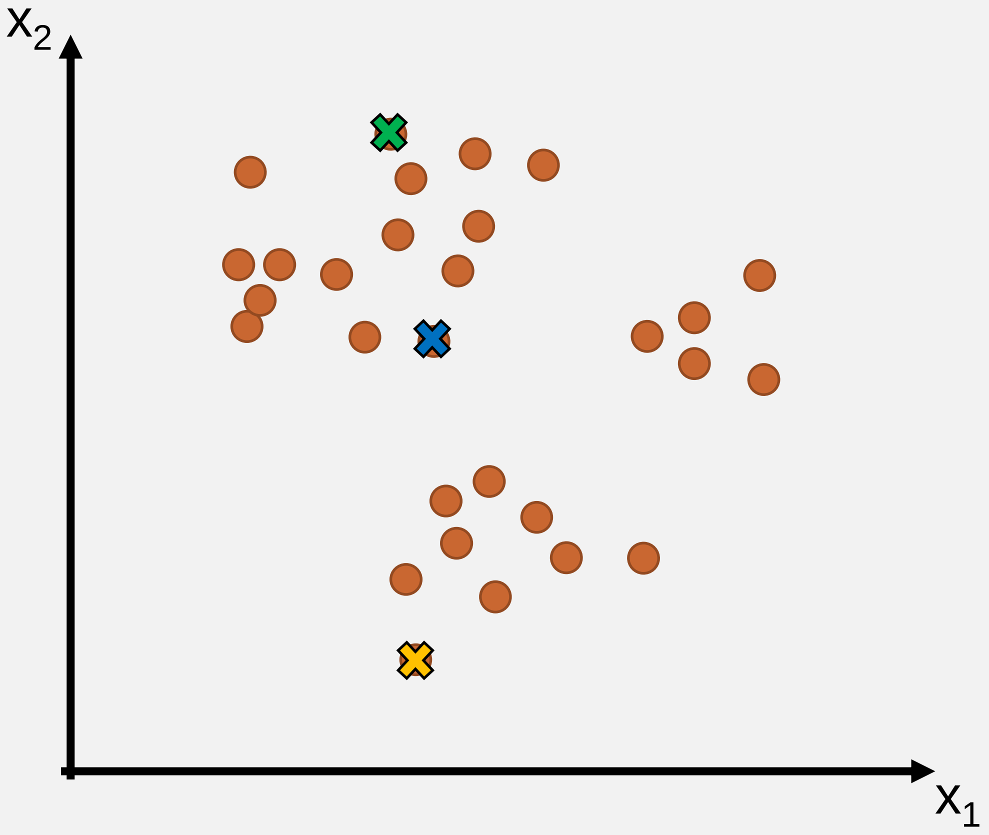
… and the other images in the cluster are returned.

DIFFERENCE FROM CLASSIFICATION?

k-MEANS CLUSTERING

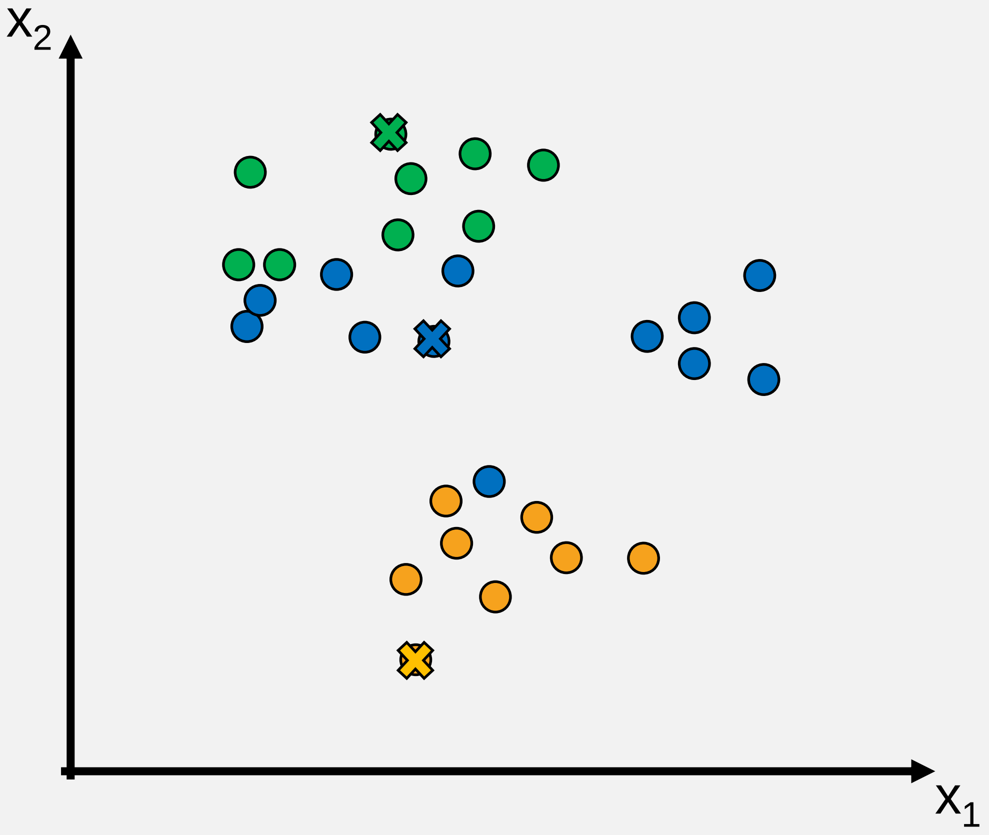


k-MEANS CLUSTERING



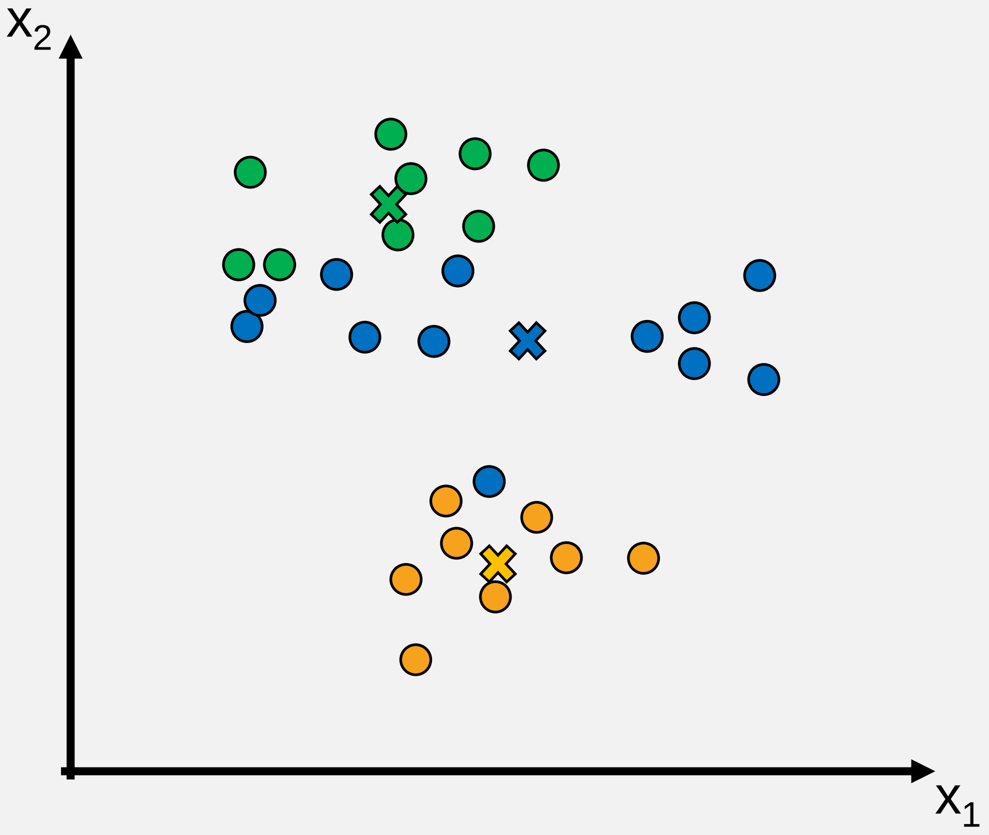
1. Assign k(=3) random points as **centroids**

k-MEANS CLUSTERING



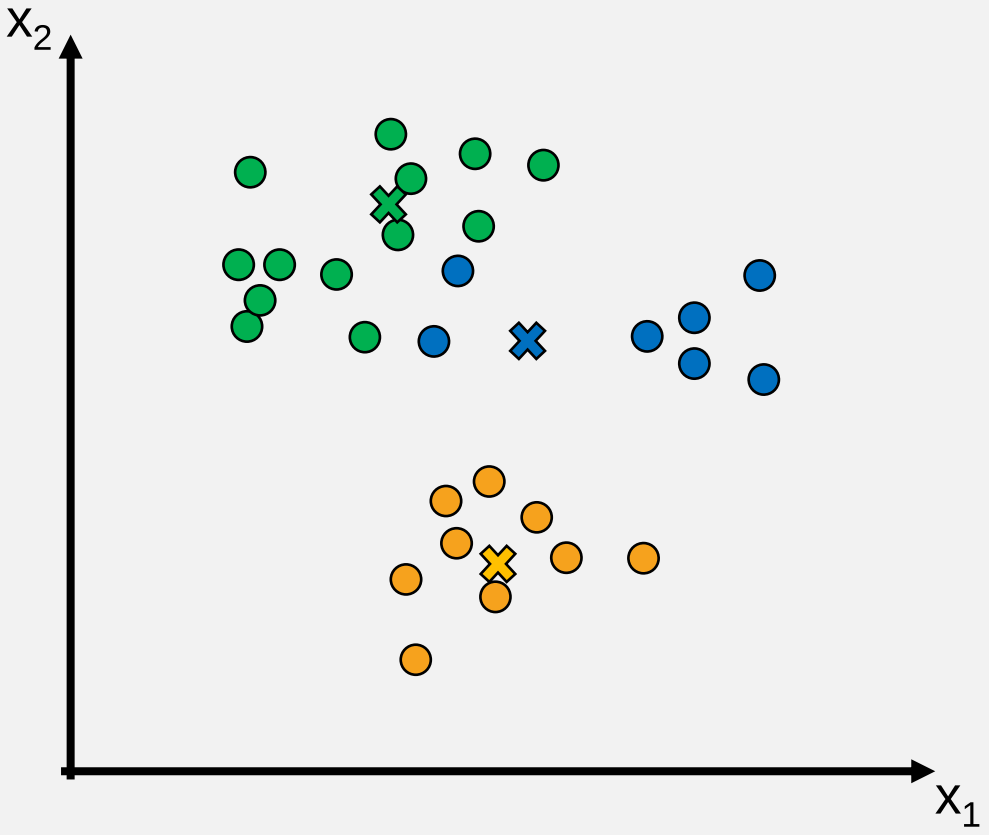
1. Assign k(=3) random points as **centroids**
2. Group the data by their distance to the centroids

k-MEANS CLUSTERING



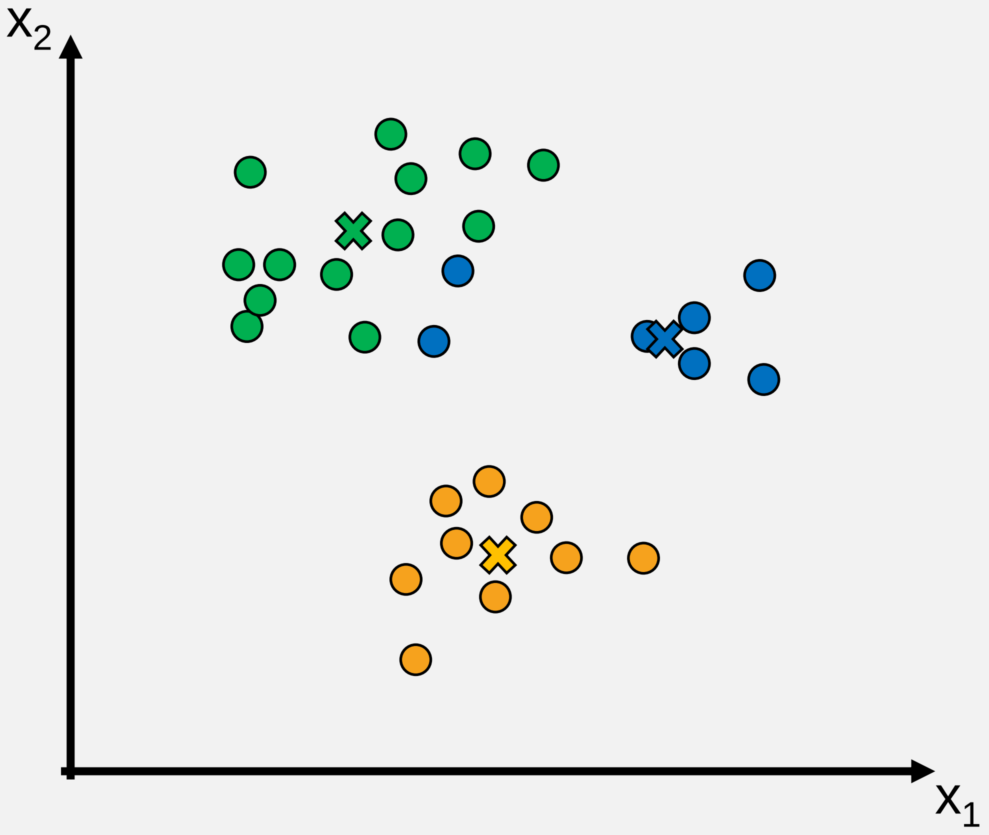
1. Assign k(=3) random points as **centroids**
2. Group the data by their distance to the centroids
3. Move the centroids to the cluster centers

k-MEANS CLUSTERING



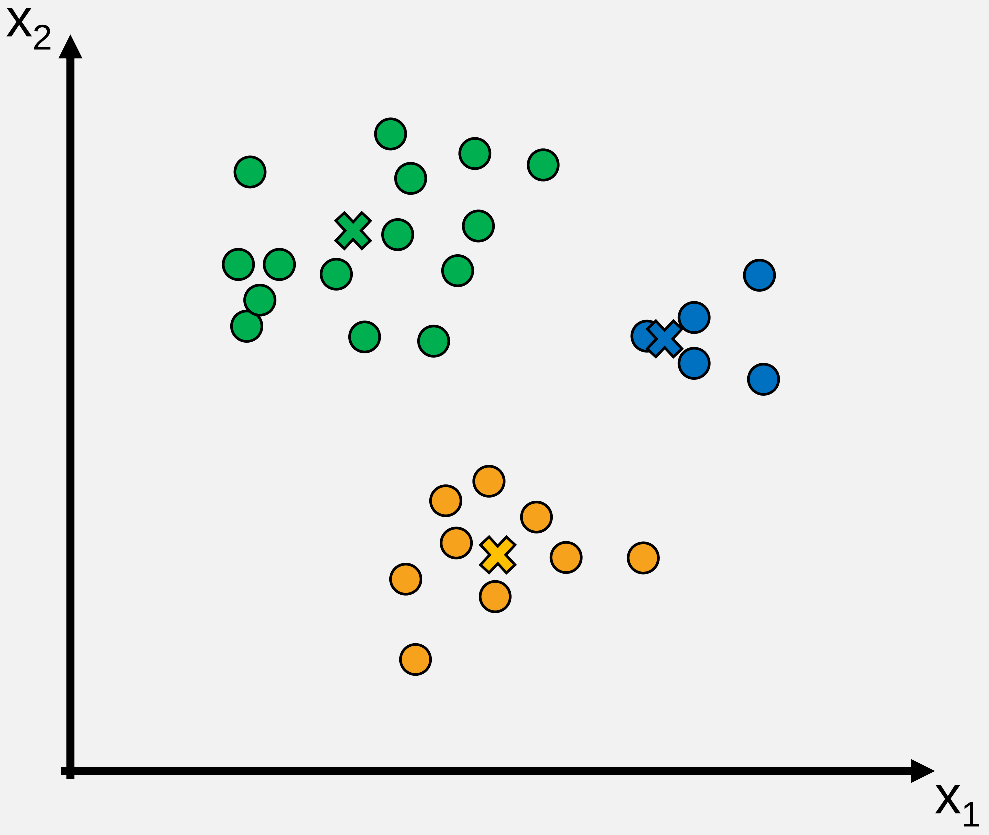
1. Assign k(=3) random points as **centroids**
2. Group the data by their distance to the centroids
3. Move the centroids to the cluster centers
4. Regroup the data

k-MEANS CLUSTERING



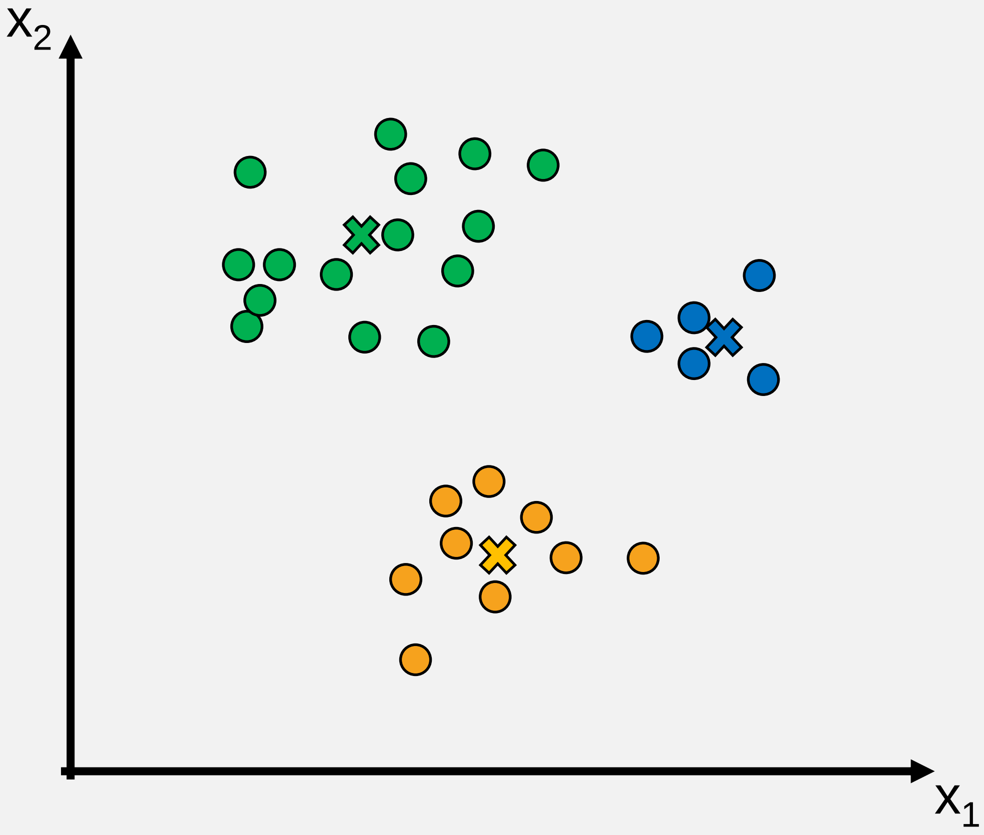
1. Assign k(=3) random points as **centroids**
2. Group the data by their distance to the centroids
3. Move the centroids to the cluster centers
4. Regroup the data
5. Repeat 3-4 until nothing changes

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

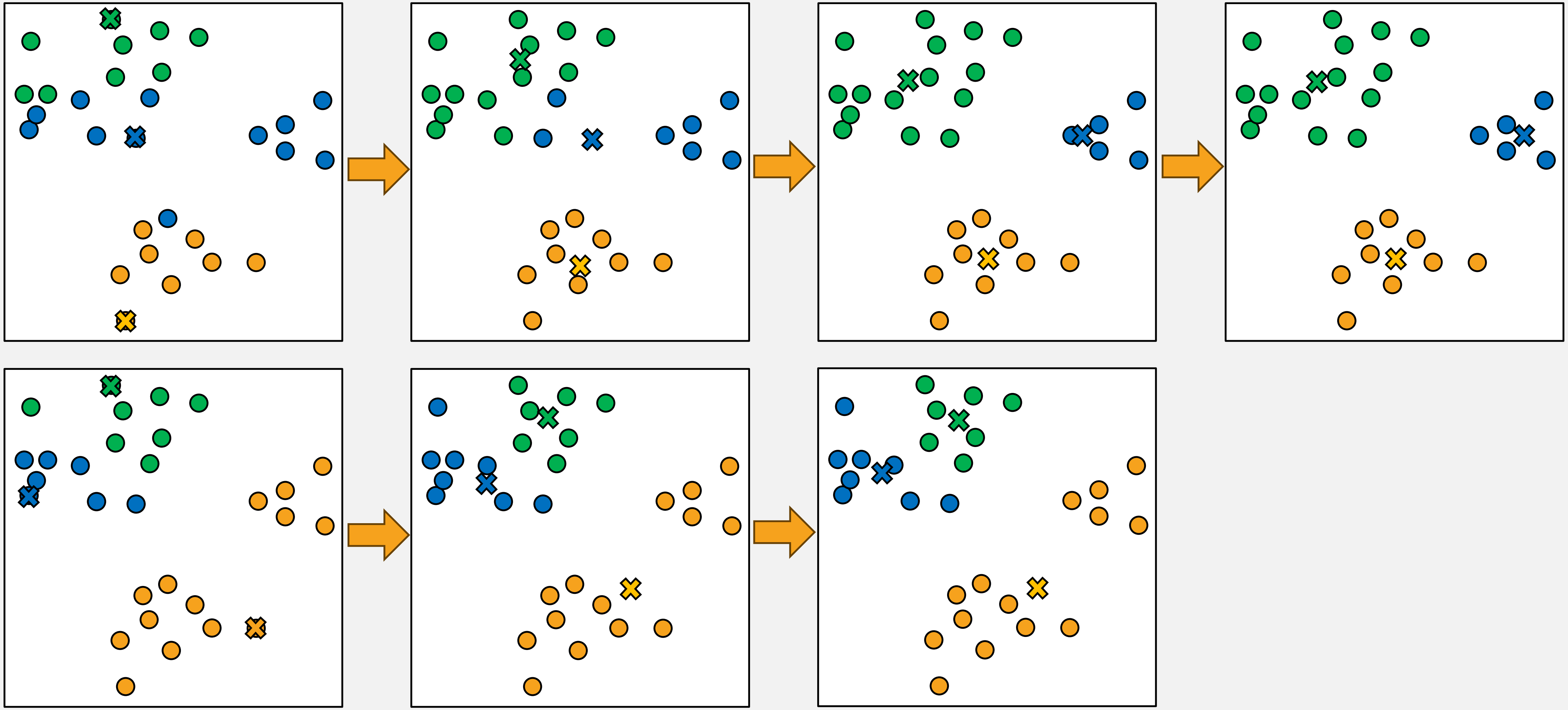
k-MEANS CLUSTERING



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

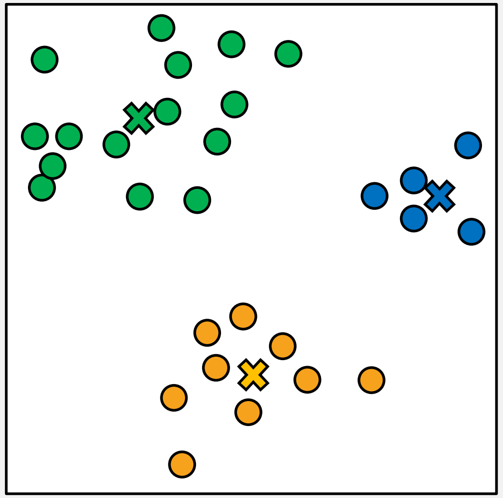
A FEW THINGS WE HAVE TO DEAL WITH

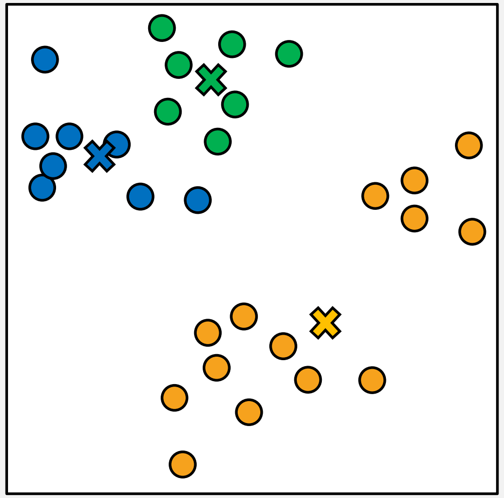
THE INITIAL CENTROIDS



THE INITIAL CENTROIDS

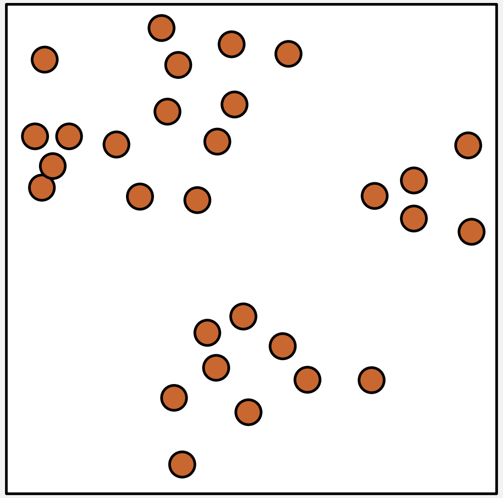
Solution 1: Try different, randomized initializations and compare the **costs** of the final clusterings





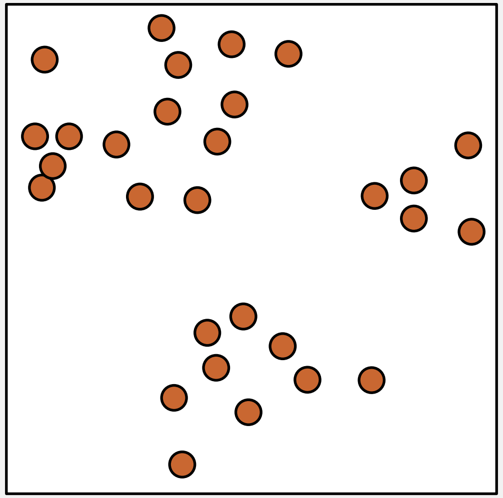
THE INITIAL CENTROIDS

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

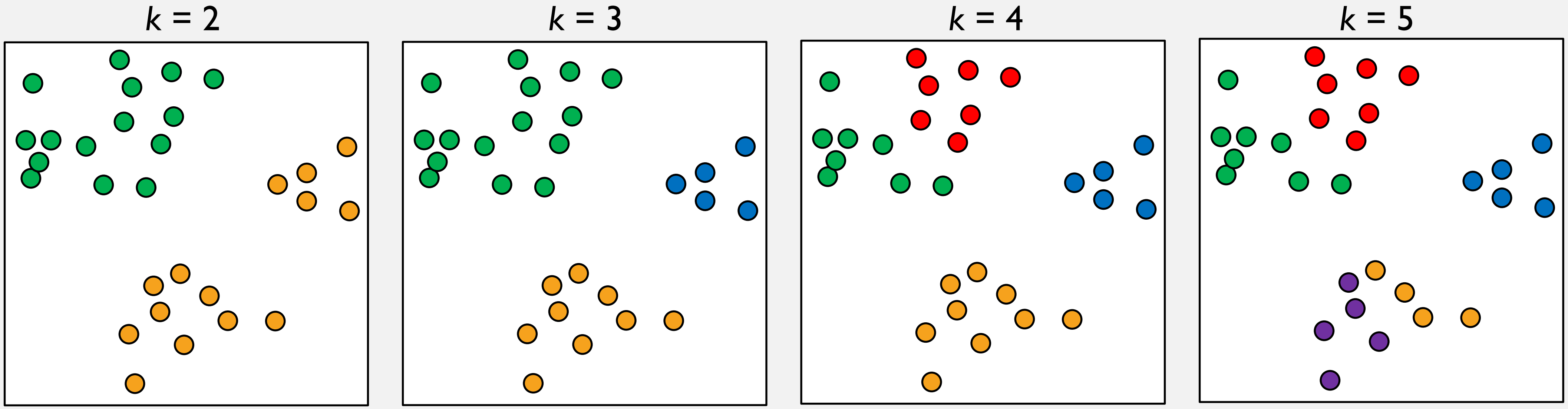


THE INITIAL CENTROIDS

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



THE NUMBER OF CLUSTERS (k)



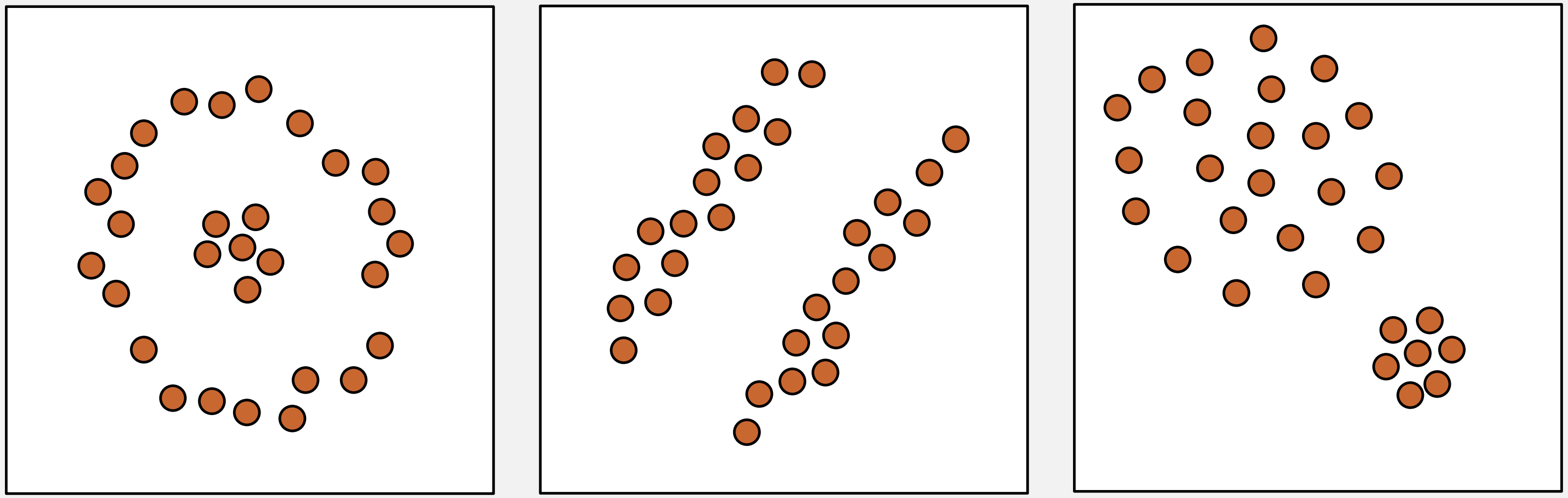
THE NUMBER OF CLUSTERS (k)

The easy way:

The hard way:

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

WHERE *k*-MEANS FAILS

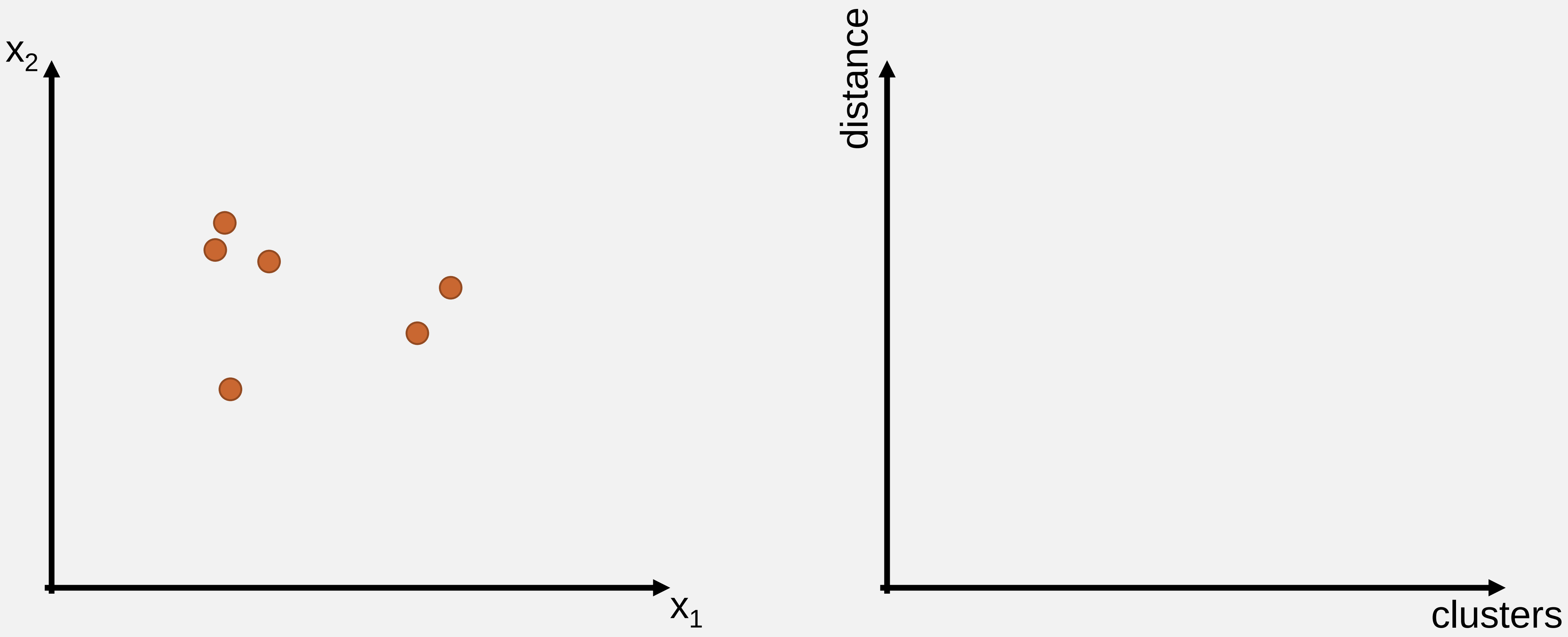


CODE EXAMPLE

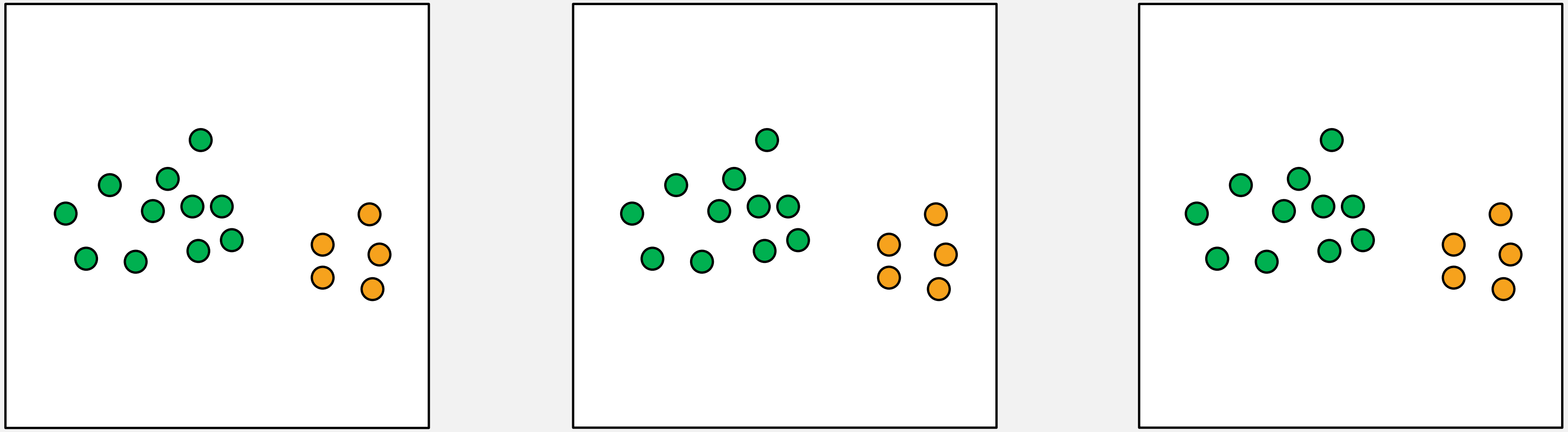
Jupyter Notebook **Clustering methods**

AGGLOMERATIVE CLUSTERING

AGGLOMERATIVE CLUSTERING



THE DISTANCE BETWEEN CLUSTERS



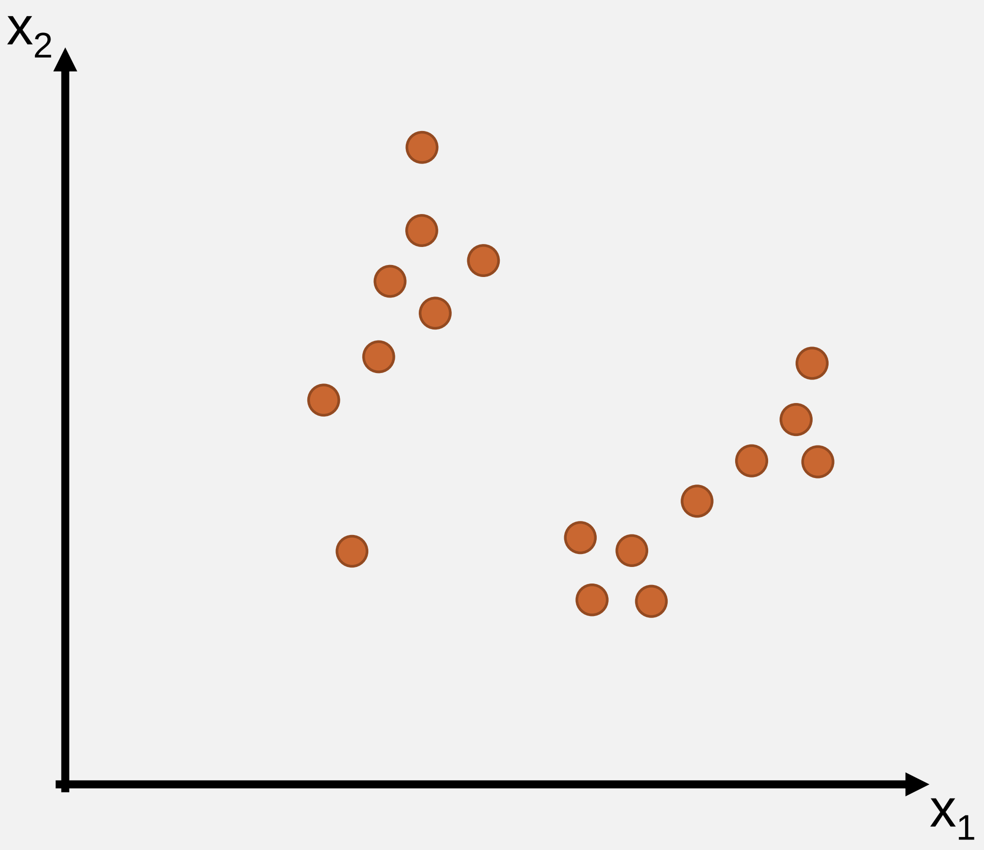
CODE EXAMPLE

Jupyter Notebook **Clustering methods**

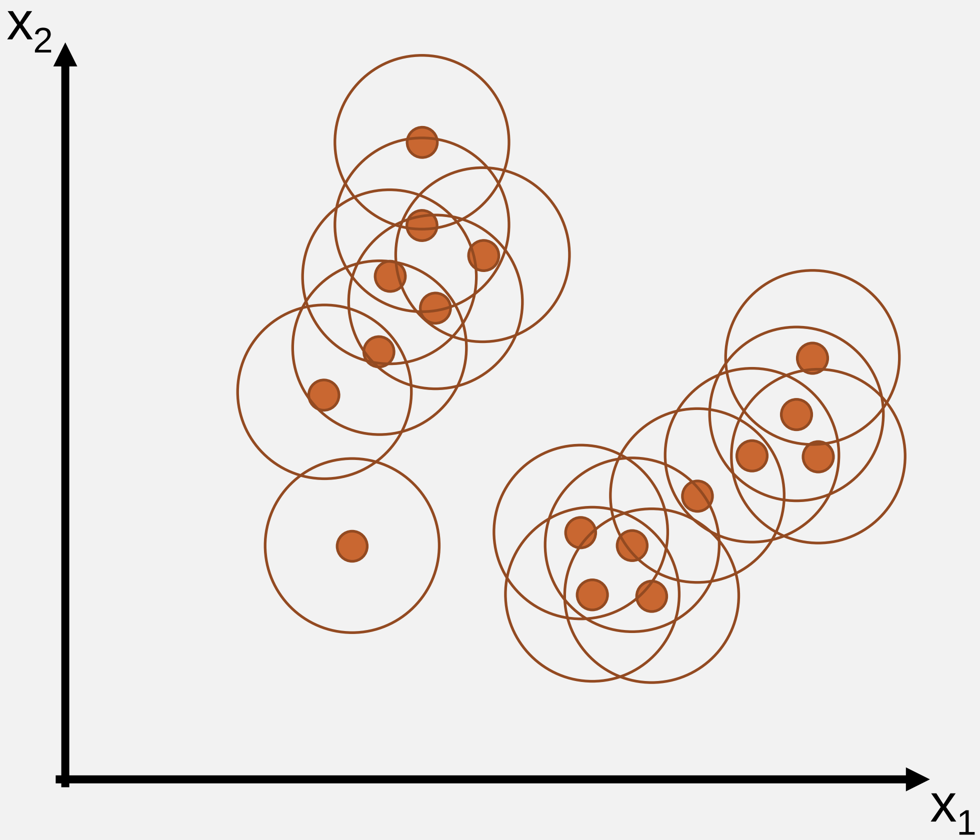
DBSCAN

- How do we measure density?
- What is a dense region?

DBSCAN

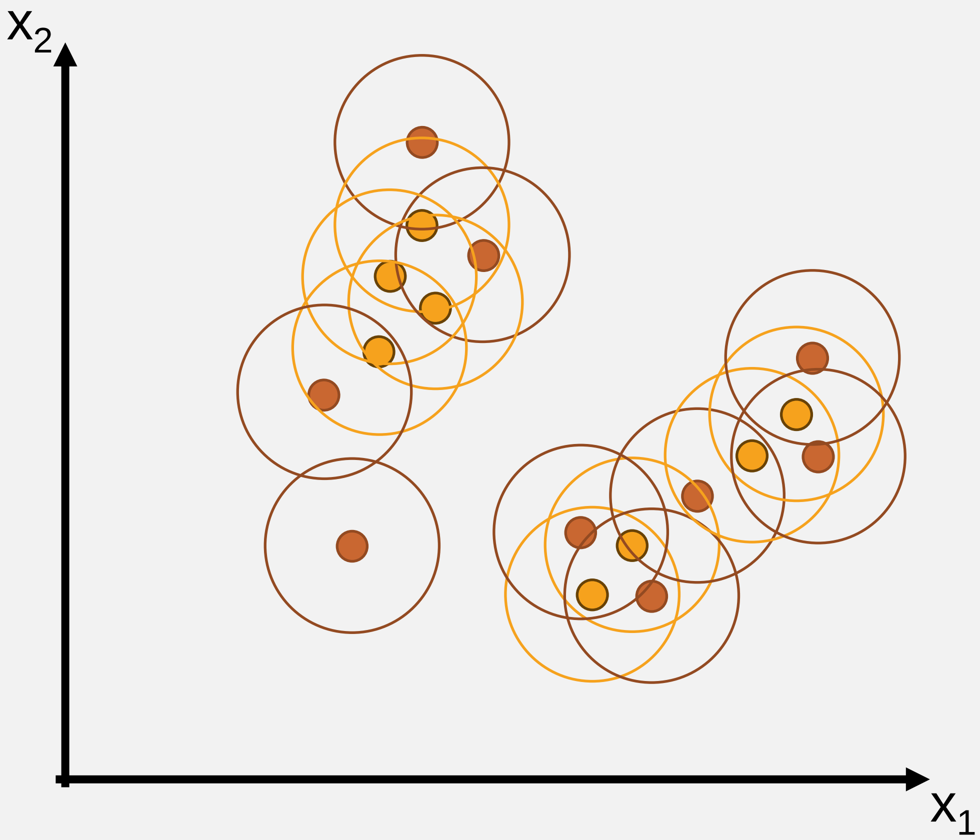


DBSCAN



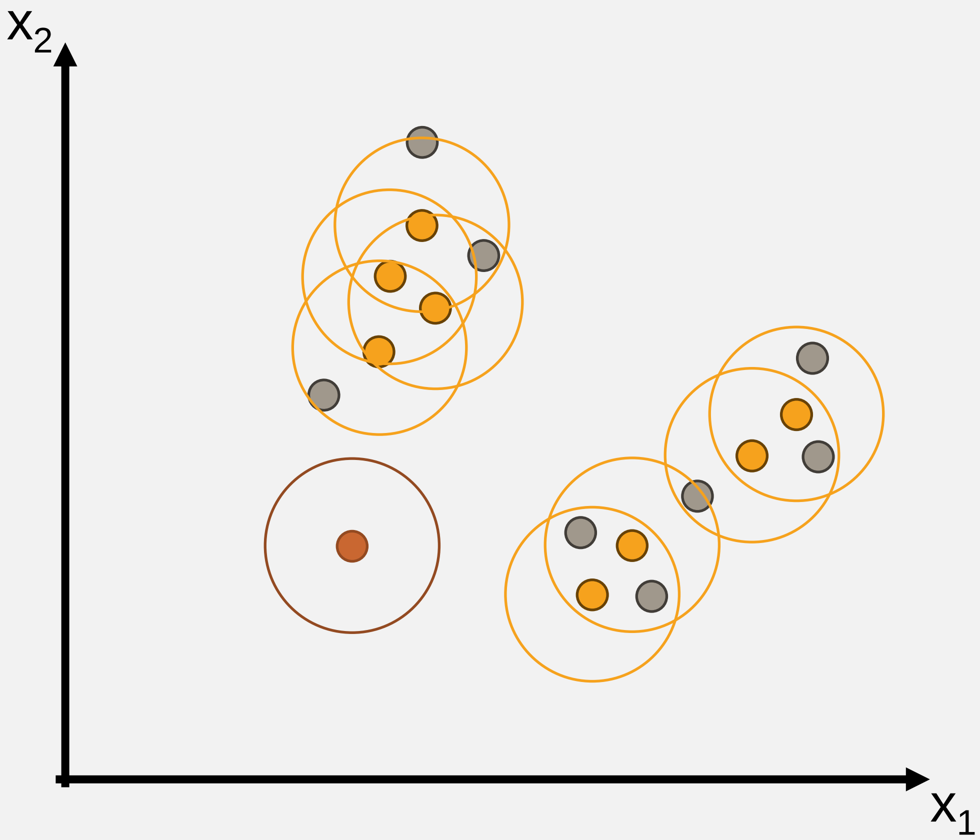
1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.

DBSCAN



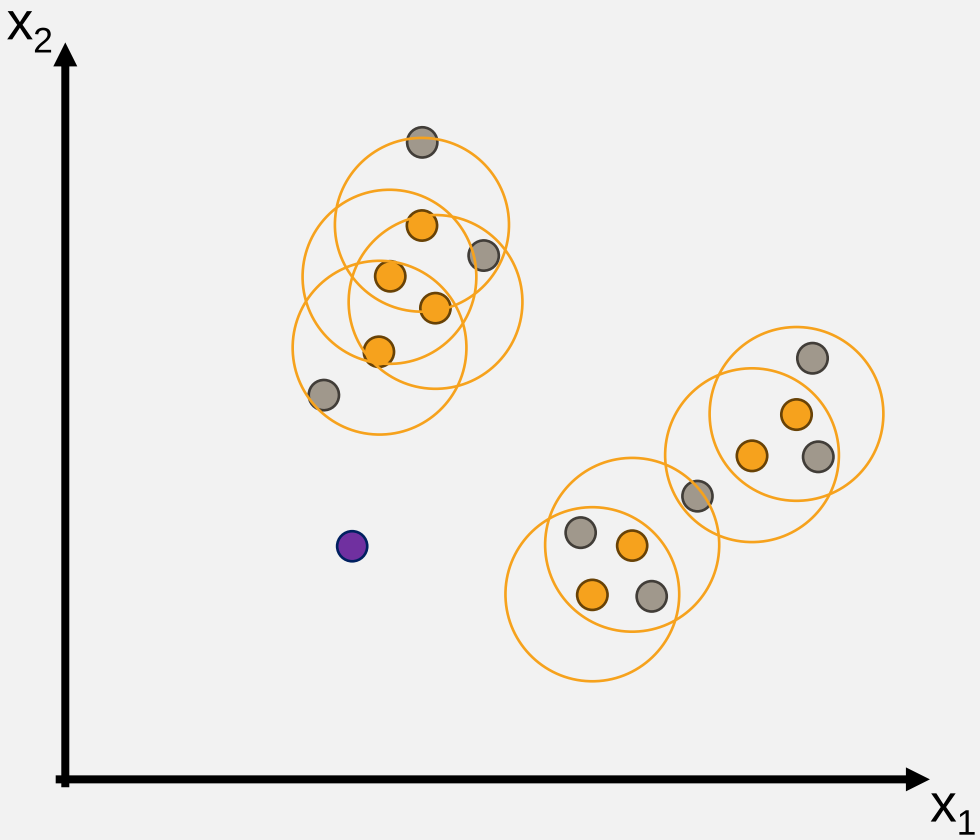
1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.
2. If the ε-neighbourhood contains at least n (=4) points, we consider the point a **core** point .

DBSCAN



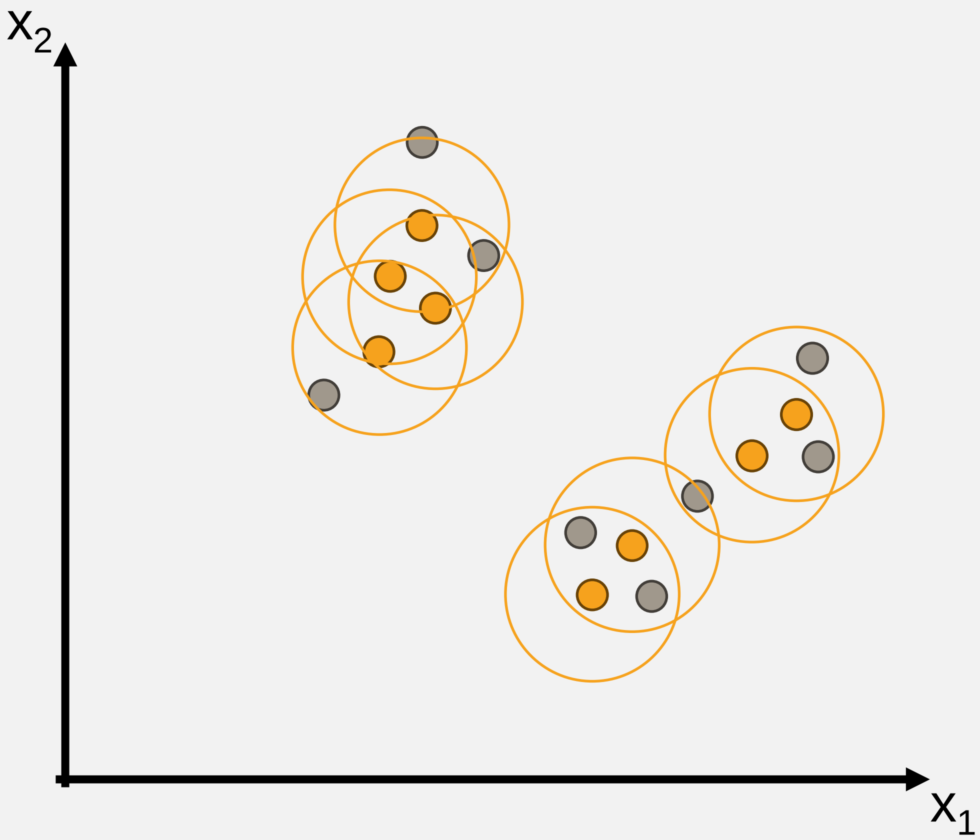
1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.
2. If the ε-neighbourhood contains at least n (=4) points, we consider the point a **core** point.
3. If the point is not a core point, but is in the ε-neighbourhood of one, it is a **border** point.

DBSCAN



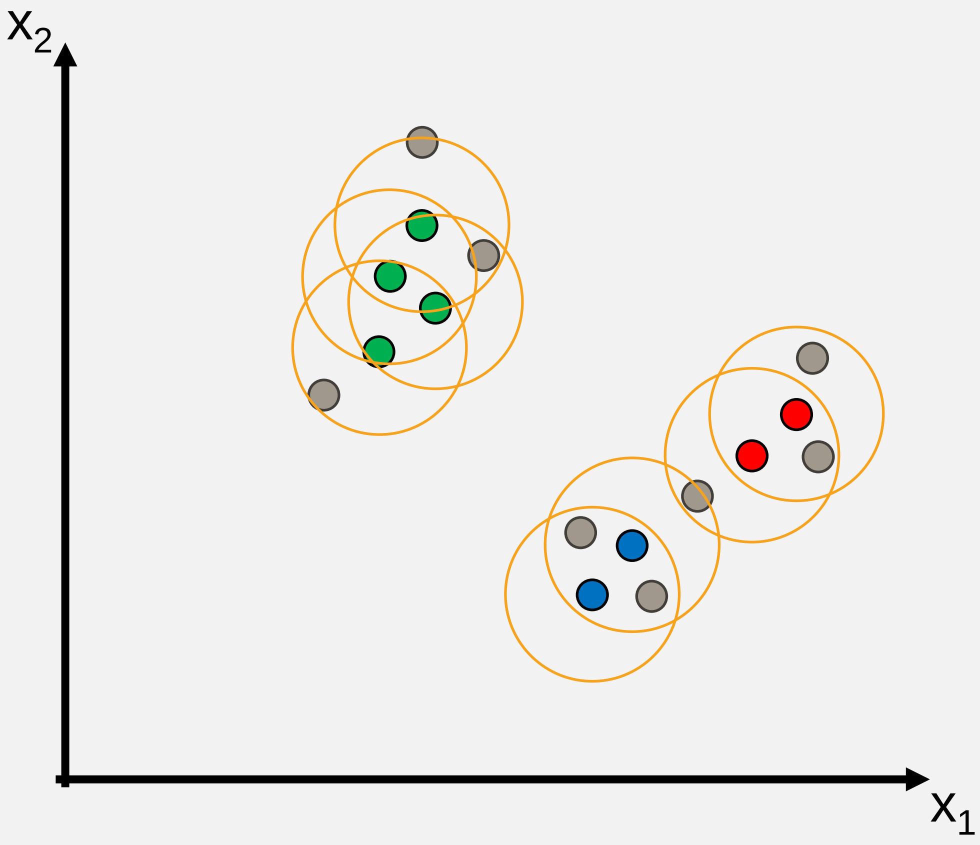
1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.
2. If the ε-neighbourhood contains at least n (=4) points, we consider the point a **core** point.
3. If the point is not a core point, but is in the ε-neighbourhood of one, it is a **border** point.
4. Otherwise, it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

DBSCAN



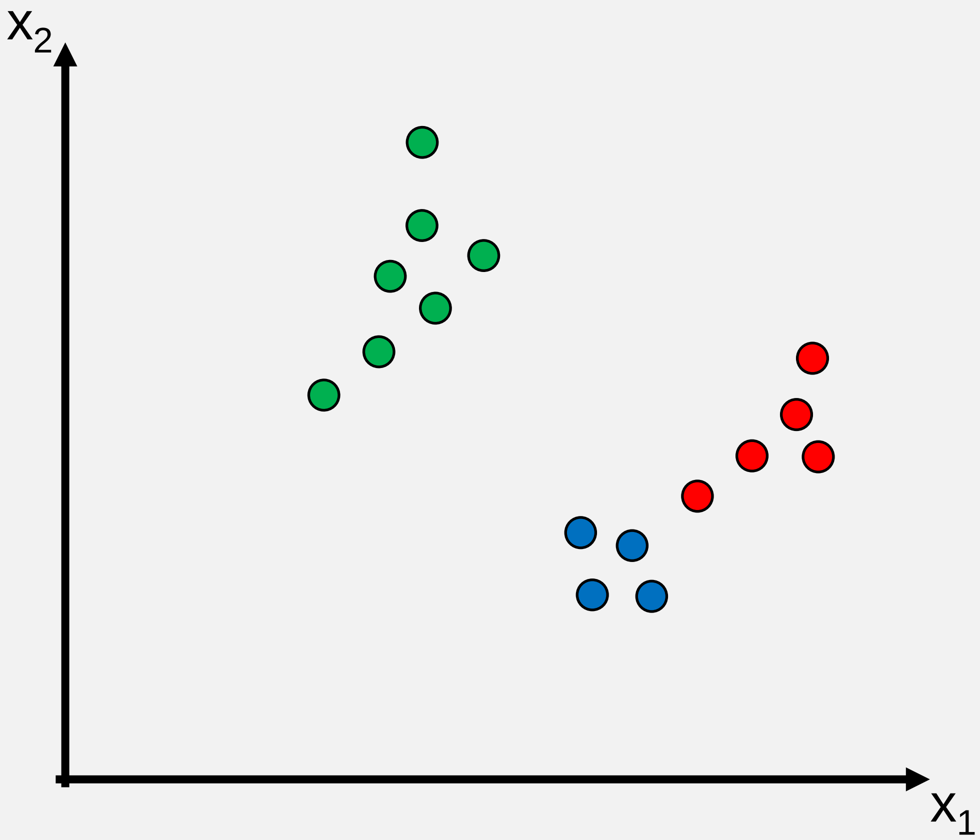
1. Draw a circle of radius ε around every point. This region is the ε-neighbourhood.
2. If the ε-neighbourhood contains at least n (=4) points, we consider the point a **core** point.
3. If the point is not a core point, but is in the ε-neighbourhood of one, it is a **border** point.
4. Otherwise, it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

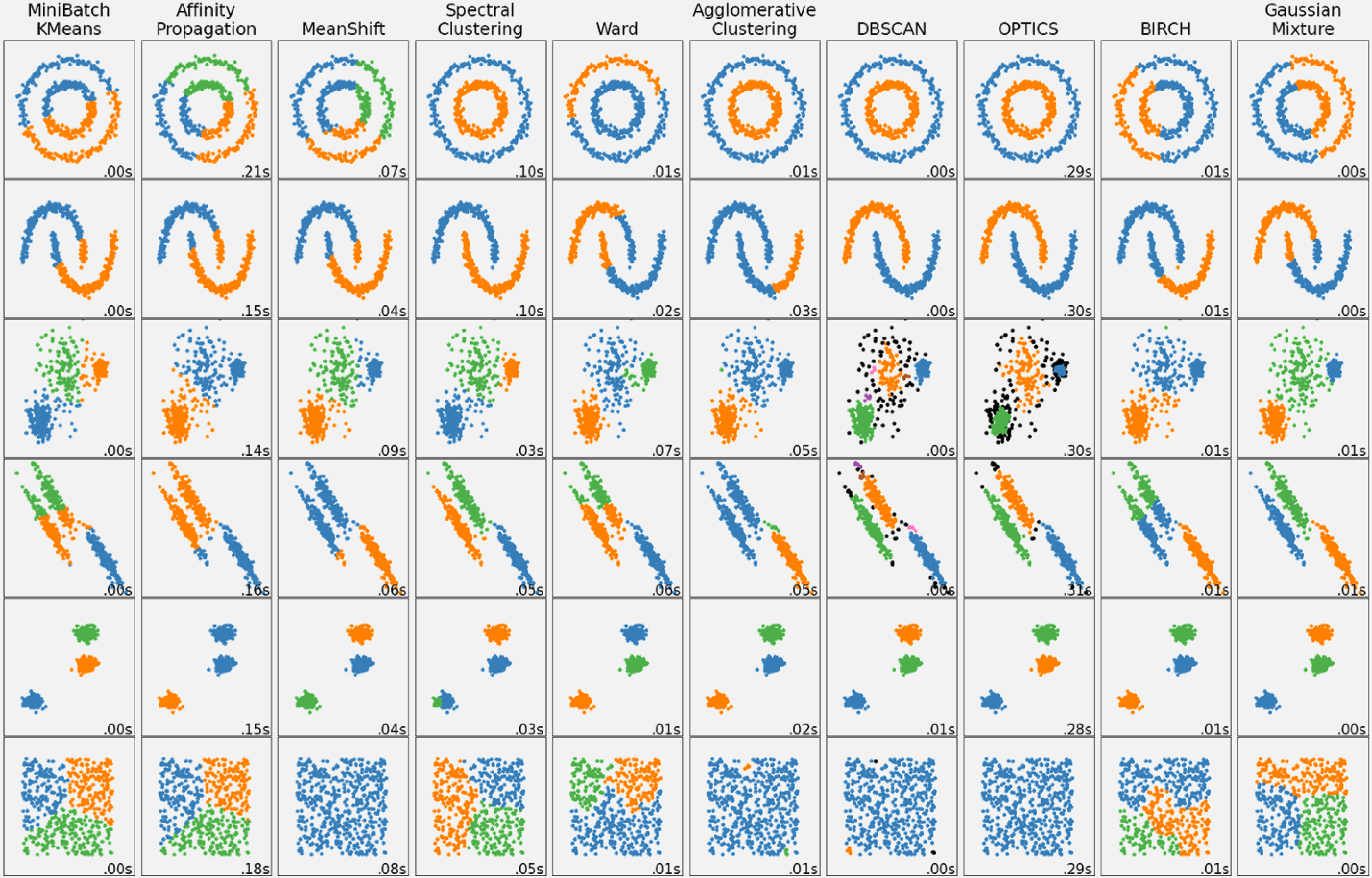
DETERMINING ε AND n

CODE EXAMPLE

Jupyter Notebook **Clustering methods**

COMPARING THE MODELS

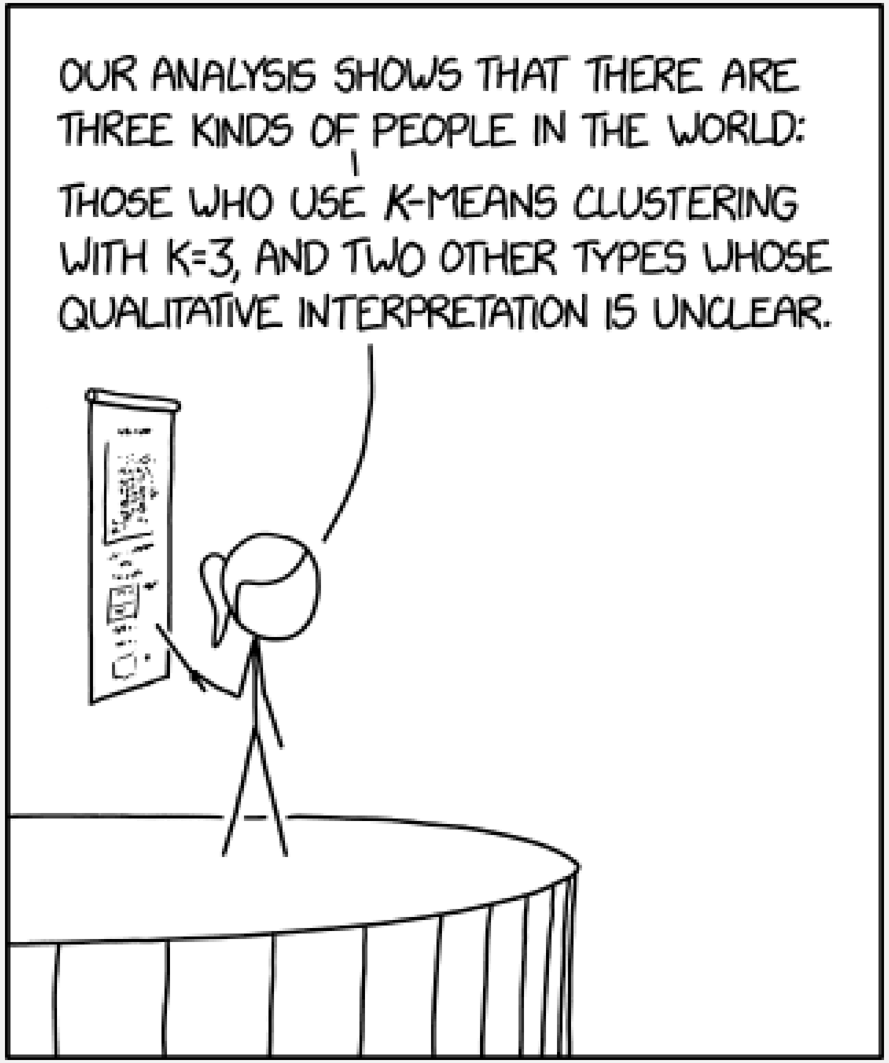
	Pros	***Cons***
***k*-means clustering**		
Agglomerative clustering		
DBSCAN		



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

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.