Mid term 1

- **Exam:**
  - 75-min exam on Oct 16 (lecture time and location)
  - In-person closed-book
  - Can bring a cheatsheet: 1 handwritten piece of paper (letter size, two sides)
  - No need for calculator
- **Practice exam:**
  - Exam and solutions posted on course website (under the files tab)
  - Will go over during the review for mid term 1 (Oct 14)
- **Mid term 1** covers:
  - All the modules we have learned so far including K-Means and PCA (this week)

Project

- **Start after mid term 1** (30%)
  - Projects announcement on 10/21
- **Team of 2-3**
- **Choose from one of the projects options (3 in total)**
  - Different modalities: images, text, and audio clips.
  - Computing resources.
  - Email both instructors if you want to use your own research for this project (i.e., you are actively doing ML research with a faculty member, who can help assess your work)
- **Grading**
  - Performance & report

Lecture 11: Unsupervised Learning (Part 1)

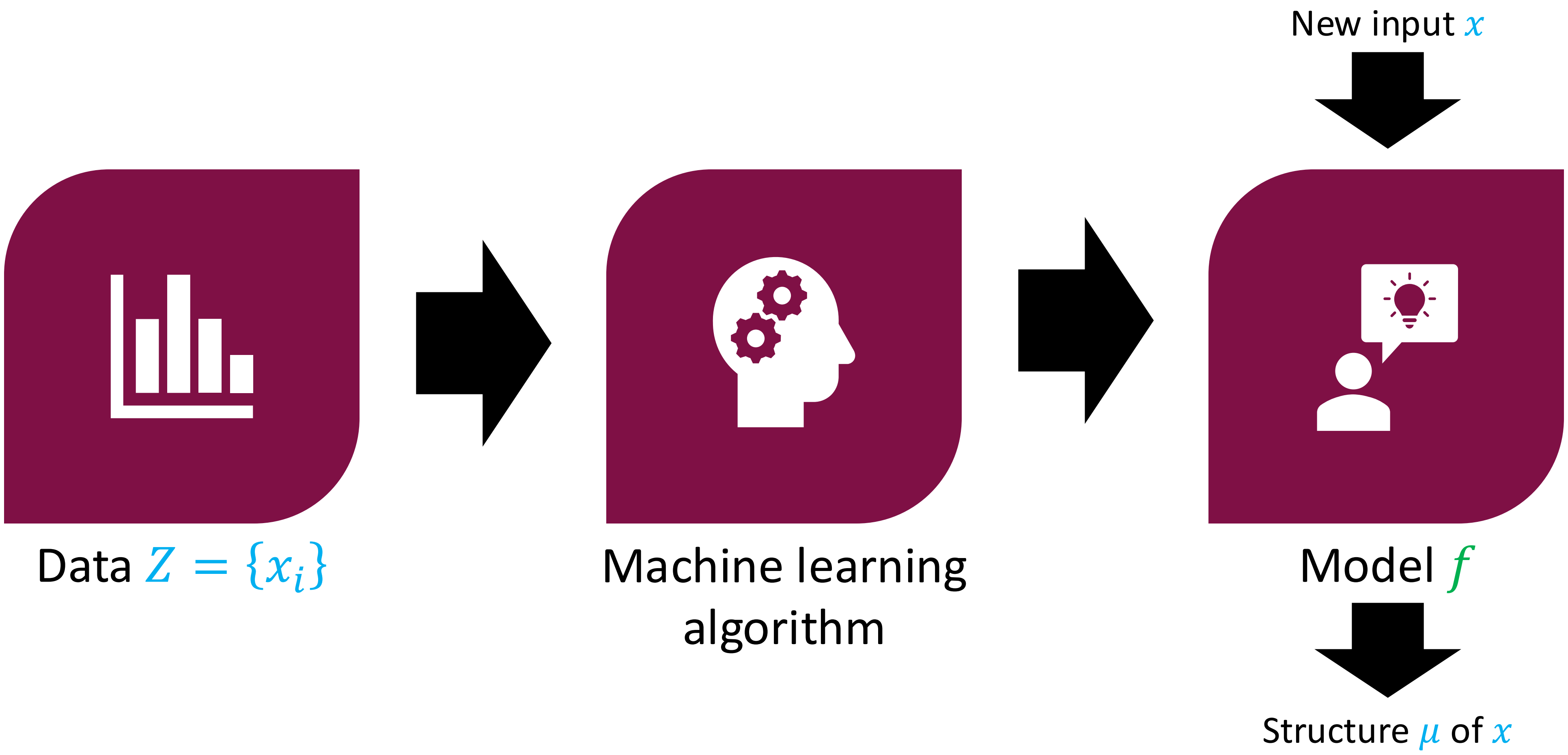
CIS 4190/5190

Fall 2024

Types of Learning

- **Supervised learning**
  - **Input:** Examples of inputs and desired outputs
  - **Output:** Model that predicts output given a new input
- **Unsupervised learning**
  - **Input:** Examples of some data (no “outputs”)
  - **Output:** Representation of structure in the data
- **Reinforcement learning**
  - **Input:** Sequence of interactions with an environment
  - **Output:** Policy that performs a desired task

Unsupervised Learning



Applications of Unsupervised Learning

- **Visualization**
  - Exploring a dataset, or a machine learning model’s outputs
- **Feature Learning**
  - Automatically construct lower-dimensional features
  - Especially useful with a lot of unlabeled data and just a few labeled examples
- **Image Compression**
  - E.g., JPEG is adopting unsupervised machine learning approaches
  - https://jpeg.org/items/20190327_press.html

Applications of Unsupervised Learning

- Visualize the data, find clusters
  e.g. "based on our polling data, there are three main voting blocs, based on age, race, education level, income, political beliefs, and home-ownership. Features like marital status and # children are irrelevant."
- Identify interesting supervised learning problems within your dataset
  e.g. "do our company's profits y_i actually correlate with the weather $\boldsymbol{x}_i$?"
- Generate new data
  e.g. "given all of Bach's work, I could generate new music that would sound like Bach."
- Identify important features in the dataset
  e.g. "Most of the variation between our customers is explained by their age, location, and education level."

Applications of Unsupervised Learning

- Framing an ML problem
- Data curation (sourcing, scraping, collection, labeling)
- Data analysis / visualization
- ML Design (hypothesis class, loss function, optimizer, hyperparameters, features)
- Train model
- Validate / Evaluate
- Deploy (and generate new data)
- Monitor performance on new data

Loss Minimization Framework

- **To design an unsupervised learning algorithm:**
  - **Model family:** Choose a model family $F = \{f_\beta\}_\beta$, where $\mu = f_\beta(x)$ encodes the structure of x
  - **Loss function:** Choose a loss function $L(\beta; Z)$

- **Resulting algorithm:**

$$\hat{\beta}(Z) = \arg\min_\beta L(\beta; Z)$$

Types of Unsupervised Learning

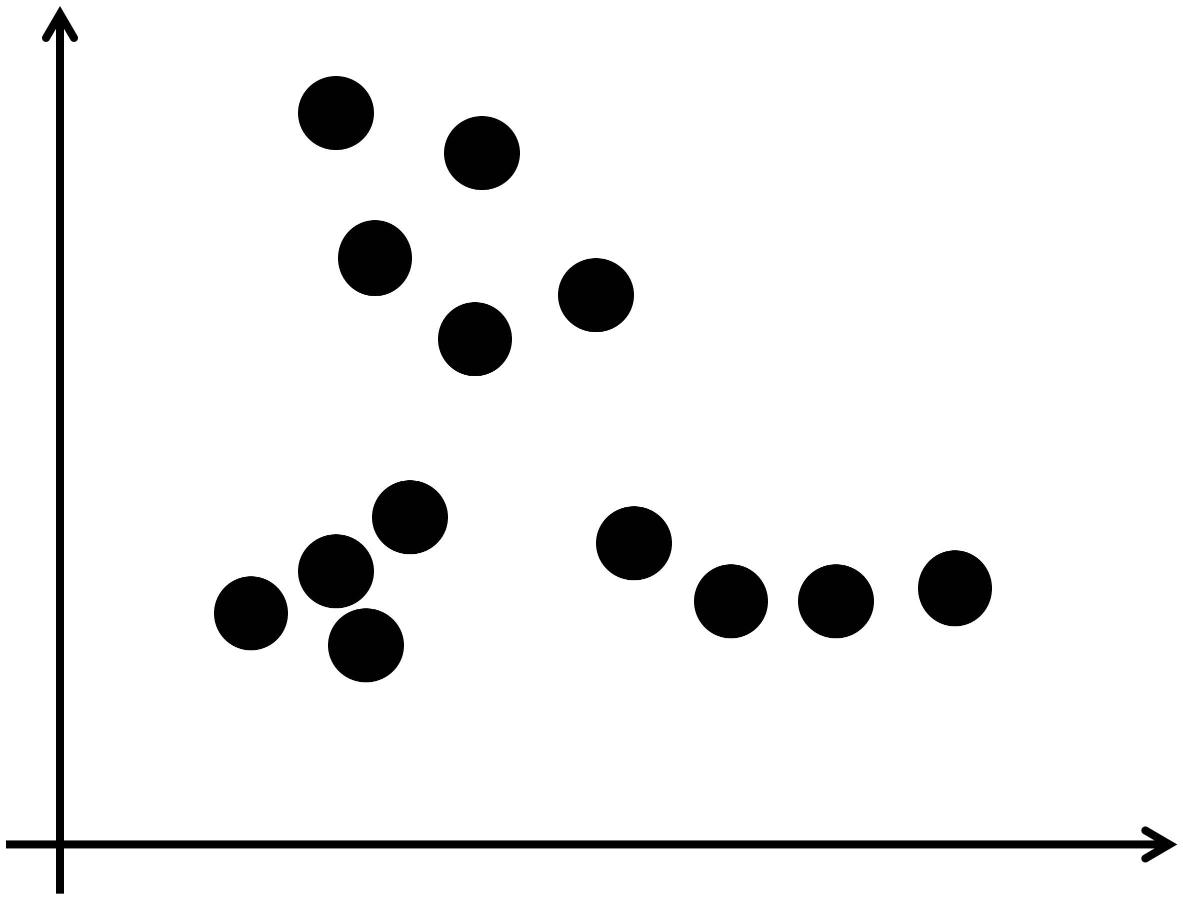
- **Clustering**
  - Map samples $x \in \mathbb{R}^d$ to $f(x) \in \mathbb{N}$
  - Each $k \in \mathbb{N}$ is associated with a representative example $x_k \in \mathbb{R}^d$
  - **Examples:** K-means clustering, greedy hierarchical clustering

- **Dimensionality reduction**
  - Map samples $x \in \mathbb{R}^d$ to $f(x) \in \mathbb{R}^{d'}$, where $d' \ll d$
  - **Example:** Principal components analysis (PCA)
  - Modern deep learning is based on this idea

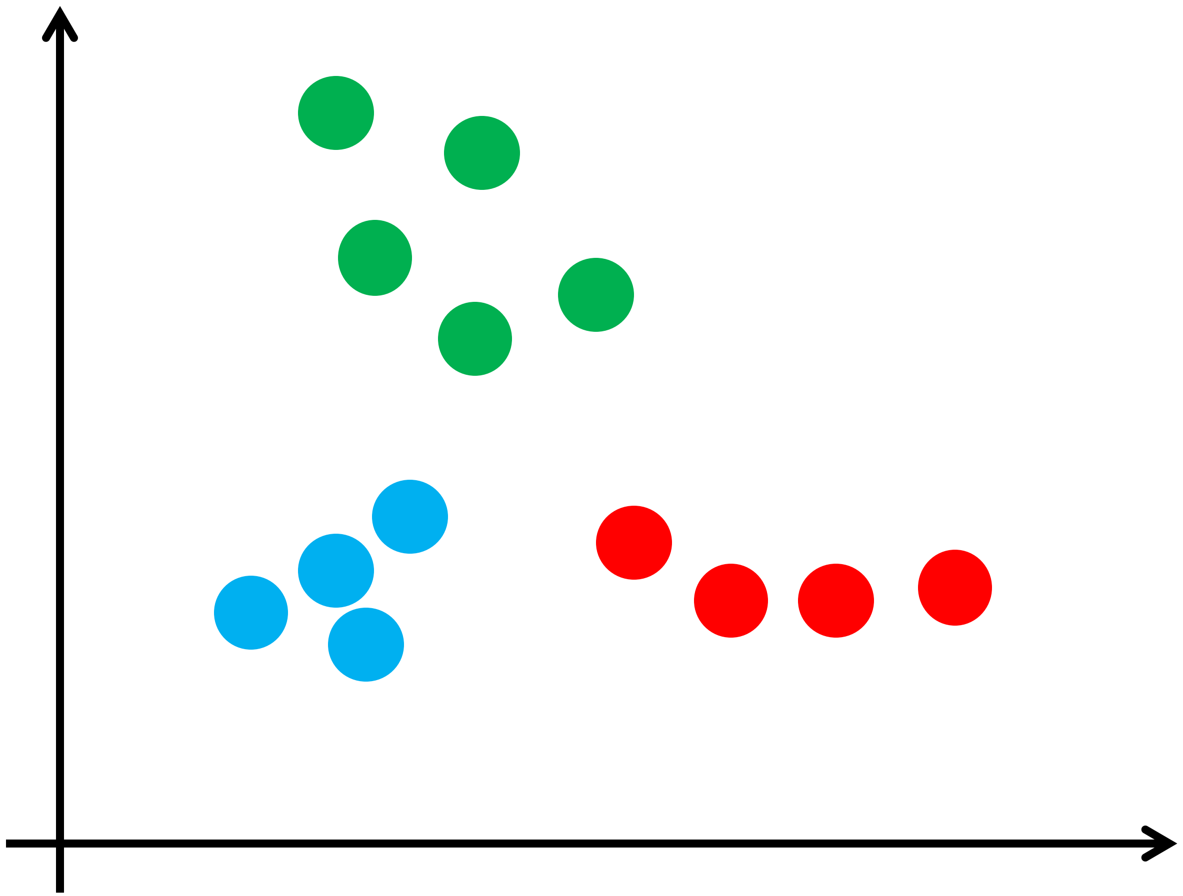
The Clustering Problem

- **Input:** Dataset $Z = \{x_i\}_{i=1}^n$
- **Output:** Model $f(x) \in \{1, \dots, K\}$
  - **Intuition:** Predictions should encode "natural" clusters in the data
  - Here, $K \in \mathbb{N}$ is a hyperparameter

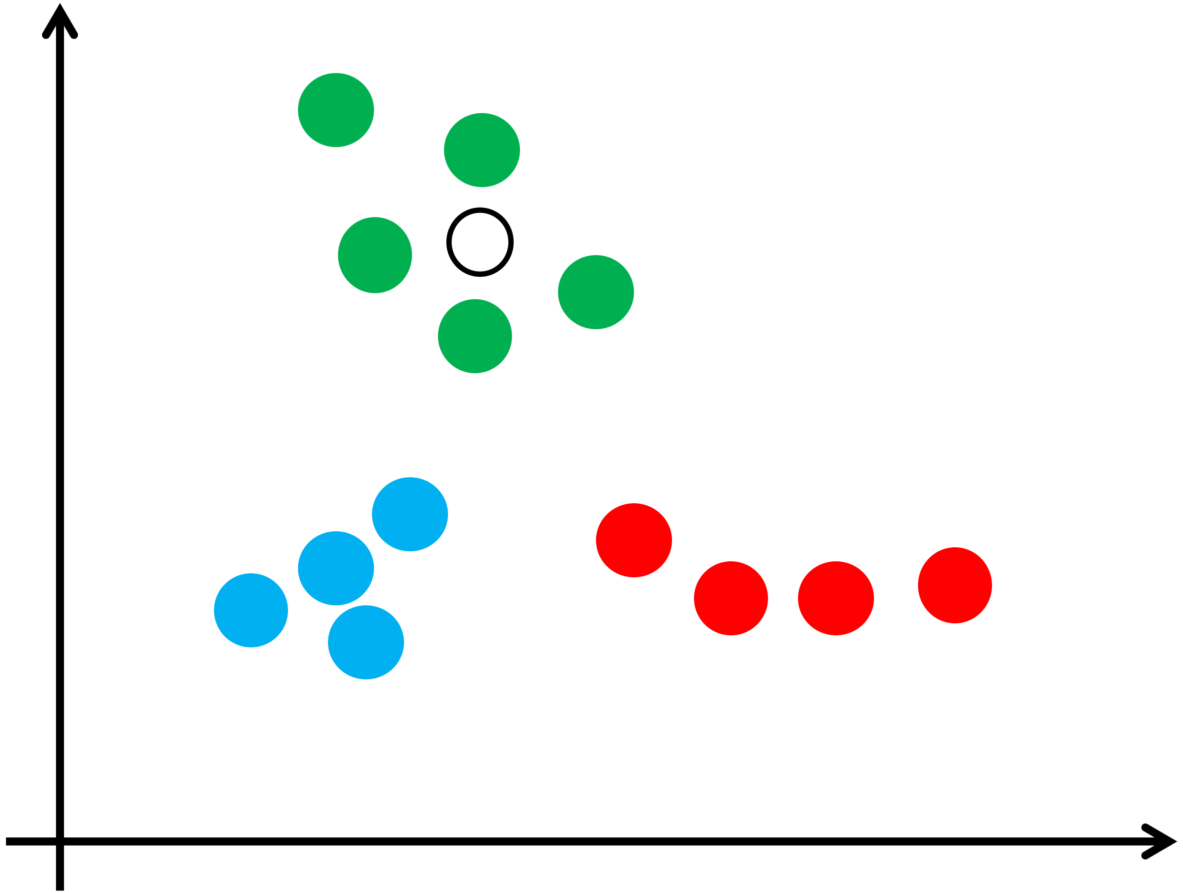
The Clustering Problem



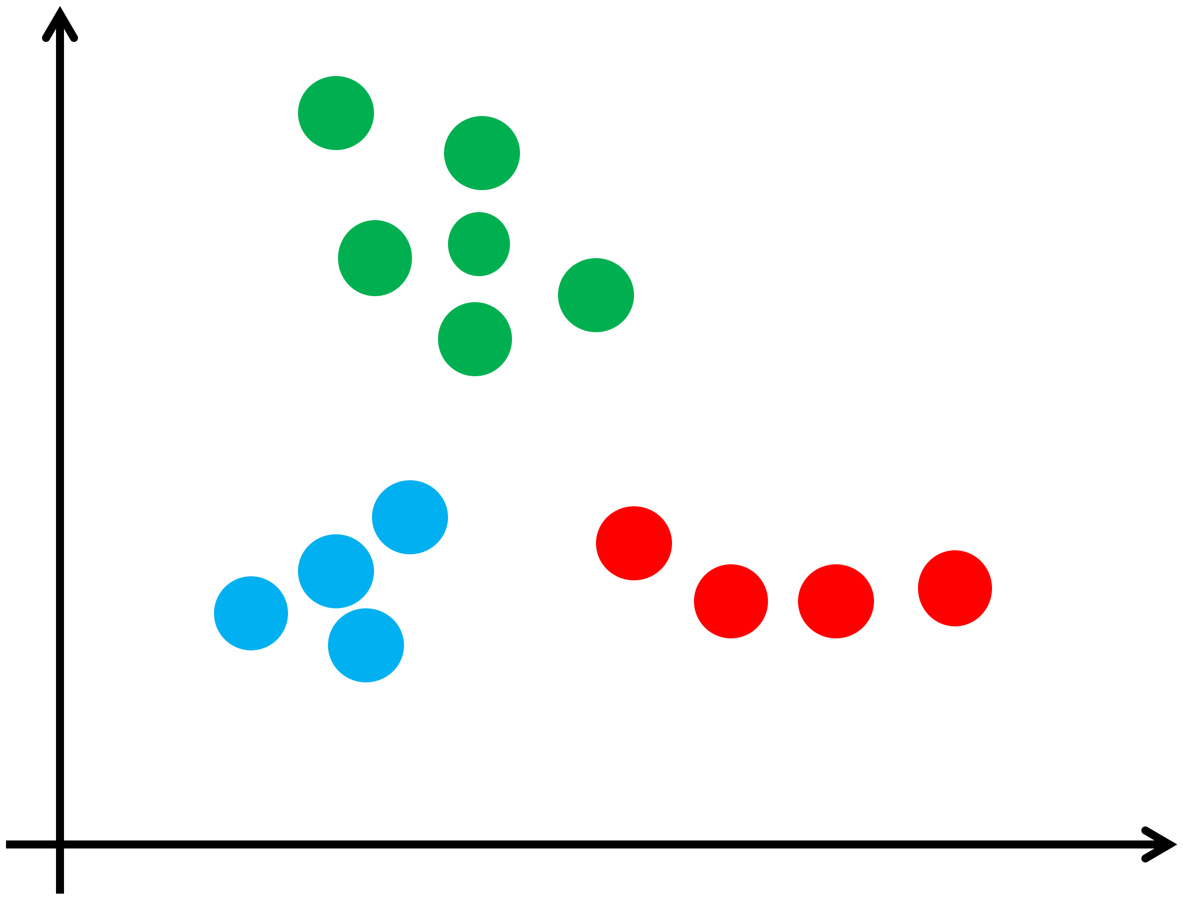
The Clustering Problem



The Clustering Problem



The Clustering Problem

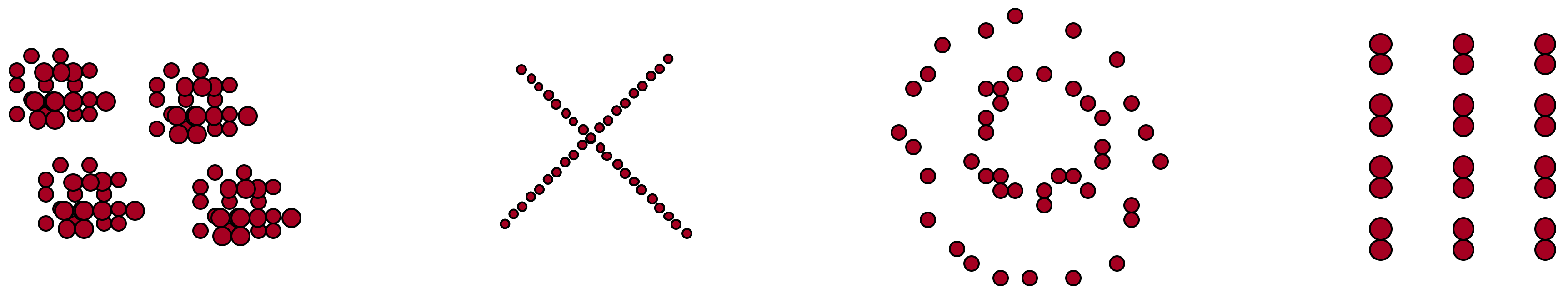


The Clustering Problem

- **Input:** Dataset $Z = \{x_i\}_{i=1}^n$
- **Output:** Model $f(x) \in \{1, \ldots, K\}$
  - **Intuition:** Predictions should encode "natural" clusters in the data
  - Here, $K \in \mathbb{N}$ is a hyperparameter
- How to formalize "naturalness"?
  - Using a loss function!

Clustering Loss

- Loss depends on the structure of the data we are trying to capture



- K-Means clustering aims to minimize specific loss over a specific model family

K-Means Clustering Model Family

- **Parameters:**
- **Centroids** μ_j, for $j \in \{1, \ldots, K\}$
  - One for each cluster (K is a hyperparameter)
  - **Intuition:** μ_j is the "center" of cluster j

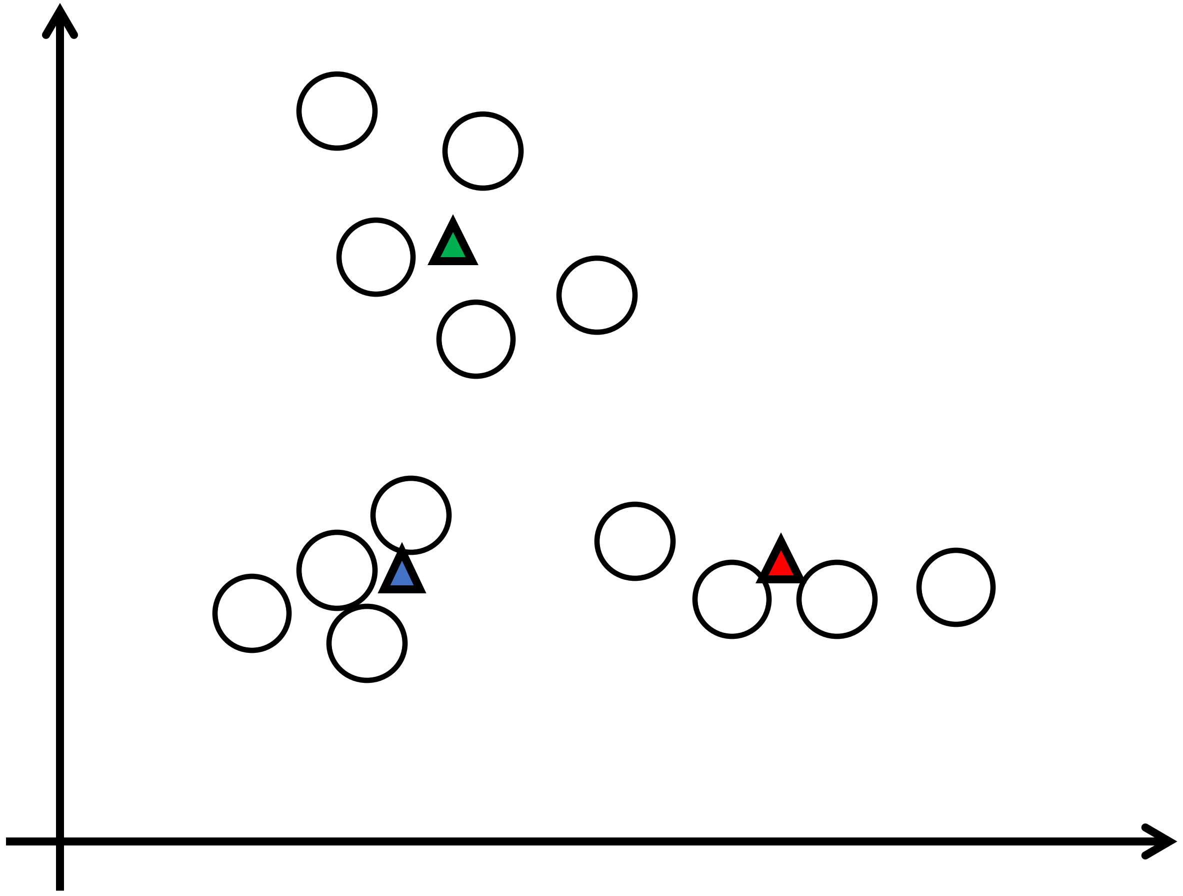
- **Assignment:** assign each data point x it to the nearest cluster:

$$f_\mu(x) = \arg\min_j \|x - \mu_j\|_2^2$$

  - Can use other distance functions

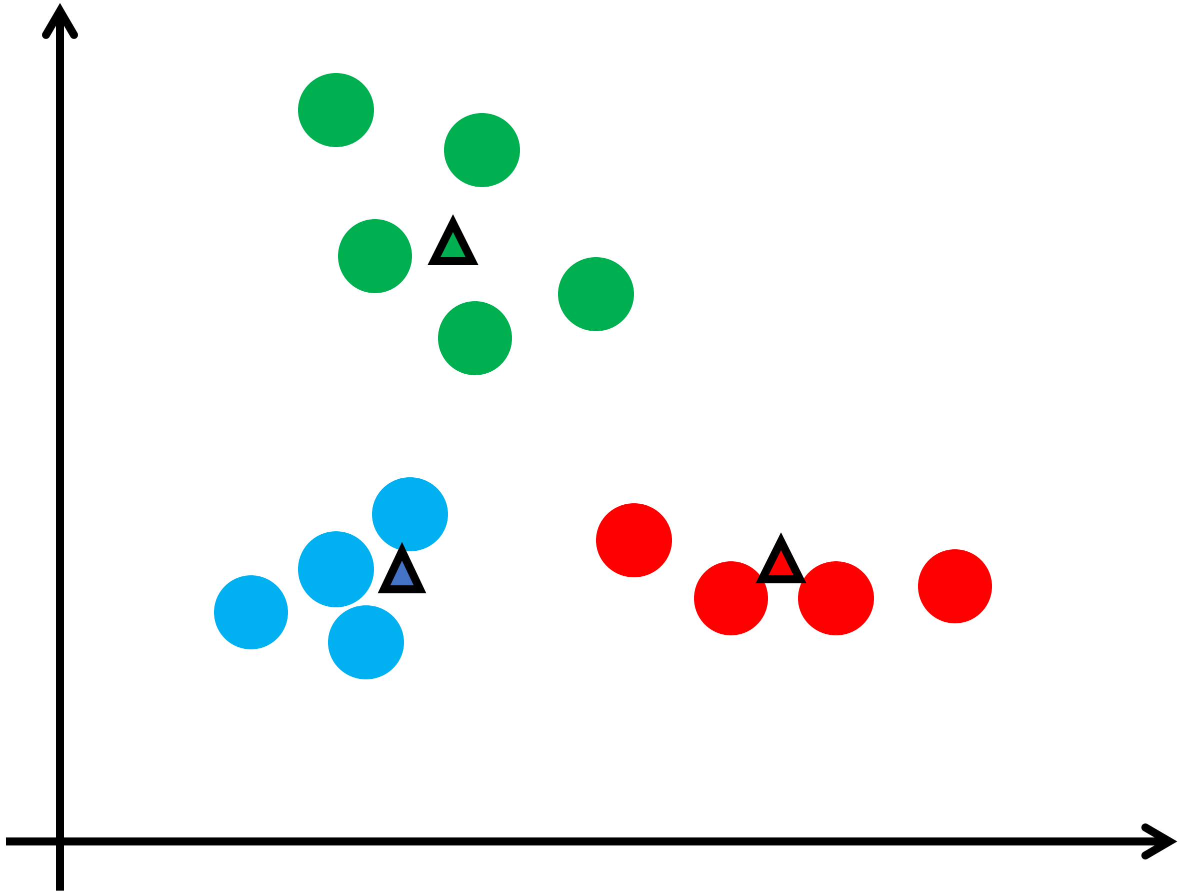
K-Means Clustering Loss

- Compute MSE of each point in the training data to its centroid



K-Means Clustering Loss

- Compute MSE of each point in the training data to its centroid



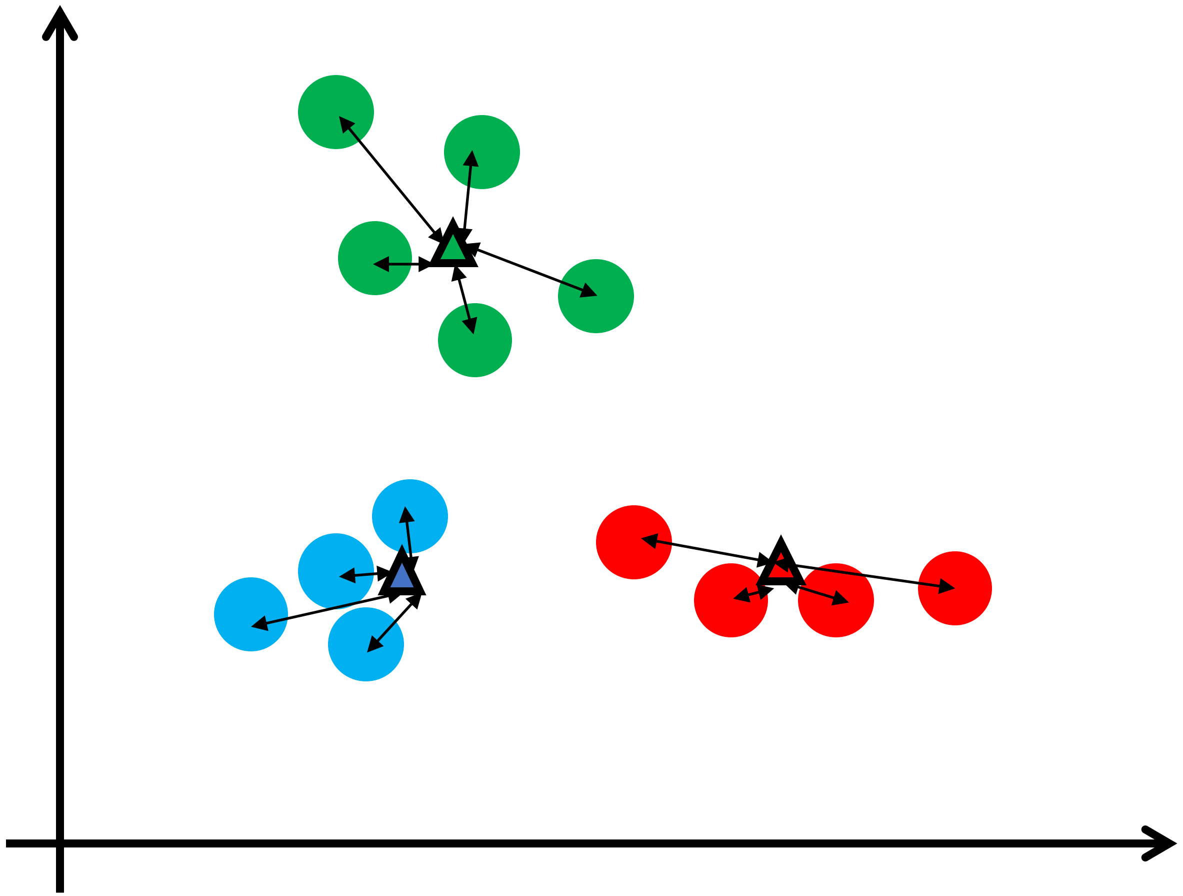
K-Means Clustering Loss

- K-means clustering chooses centroids that minimize loss of training examples Z

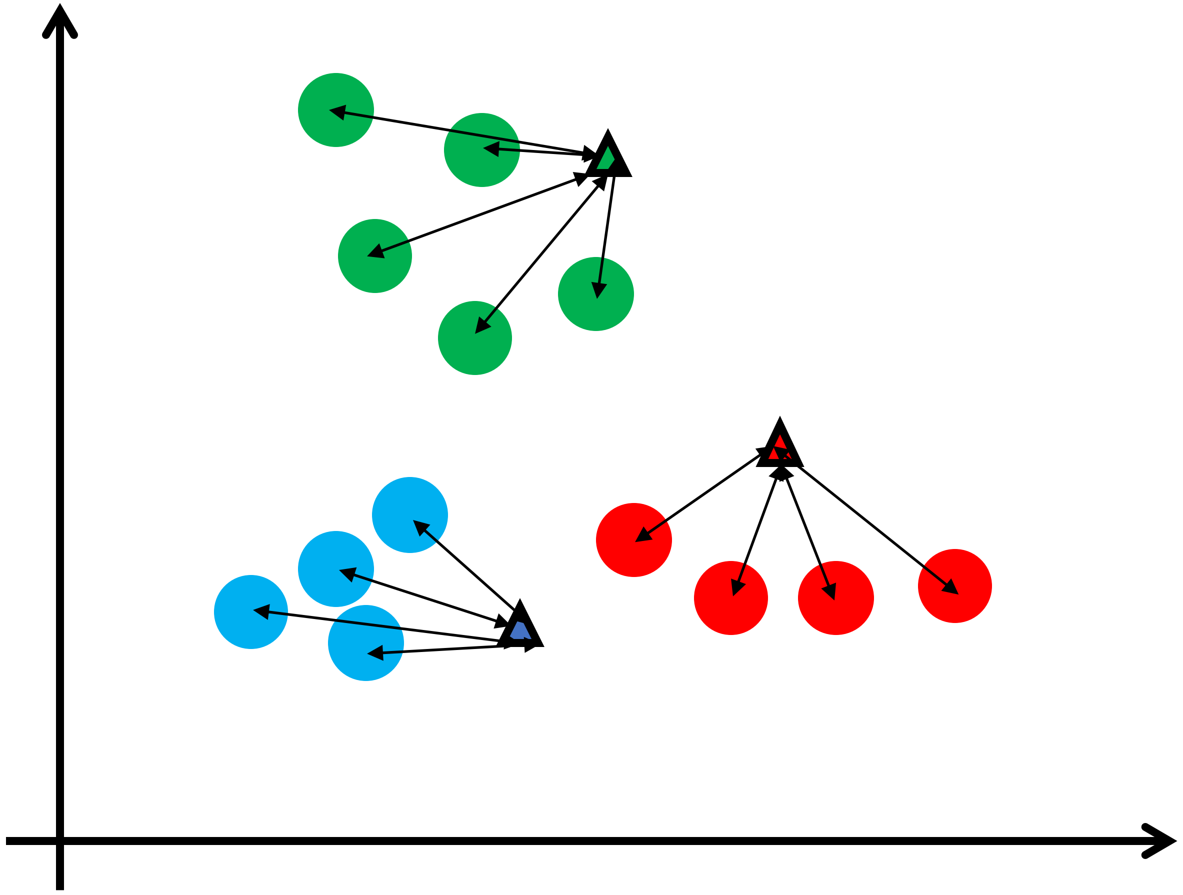
- Compute MSE of each point in the training data to its nearest centroid:

$$L(\mu; Z) = \sum_{i=1}^{n} \left\| x_i - \mu_{f_\mu(x_i)} \right\|_2^2$$

K-Means Clustering Loss



K-Means Clustering Loss



K-Means Clustering Summary

- **Model family:** $f_\mu(x) = \arg\min_j \left\| x - \mu_j \right\|_2^2$

- **Loss:** $L(\mu; Z) = \sum_{i=1}^{n} \left\| x_i - \mu_{f_\mu(x_i)} \right\|_2^2$

- **Optimizer:**
  - If we know the assignment of points to clusters
  - **Mean of point per cluster** is the vector that minimizes the squared loss!

  - Without knowledge of true assignments, this optimization is non-convex and has many local optimums

K-Means Clustering Summary

- **Model family:** $f_\mu(x) = \arg\min_j \left\| x - \mu_j \right\|_2^2$
- **Loss:** $L(\mu; Z) = \sum_{i=1}^{n} \left\| x_i - \mu_{f_\mu(x_i)} \right\|_2^2$
- **Optimizer:** Alternating minimization
  - Given an initial (potentially random) estimate of means,
  - Find every cluster assignment
  - Recompute means (changing them)
  - Iterate until convergence.

K-Means Clustering Algorithm

Kmeans(Z):

for $j \in \{1, \ldots, k\}$:

$\mu_{1,j} \leftarrow \text{Random}(Z)$

for $t \in \{1,2, \ldots\}$:

for $i \in \{1, \ldots, n\}$:

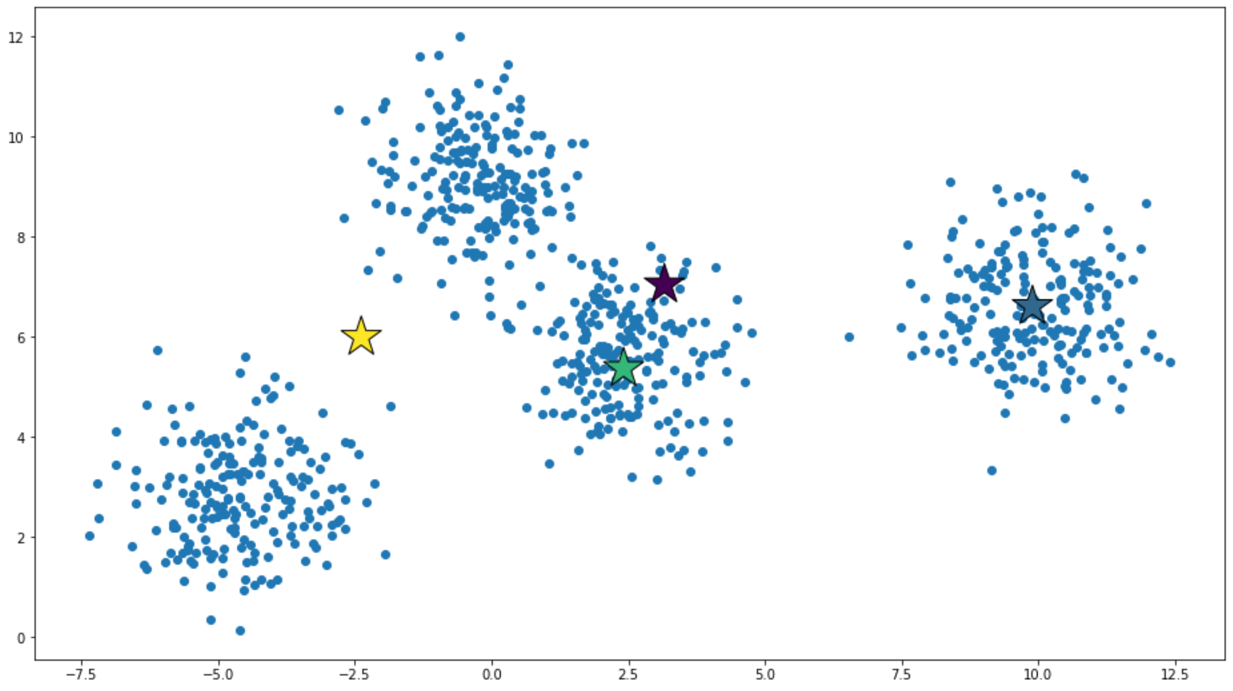
$j_{t,i} \leftarrow f_{\mu_t}(x_i)$

for $j \in \{1, \ldots, k\}$:

$\mu_{t,j} \leftarrow \text{mean}(\{\, x_i \mid j_{t,i} = j \,\})$

if $\mu_t = \mu_{t-1}$:

return μ_t



K-Means Clustering Algorithm

Kmeans(Z):

> **for** $j \in \{1, \dots, k\}$:
>
> $\quad \mu_{1,j} \leftarrow \text{Random}(Z)$

for $t \in \{1,2, \dots\}$:

$\quad$ **for** $i \in \{1, \dots, n\}$:

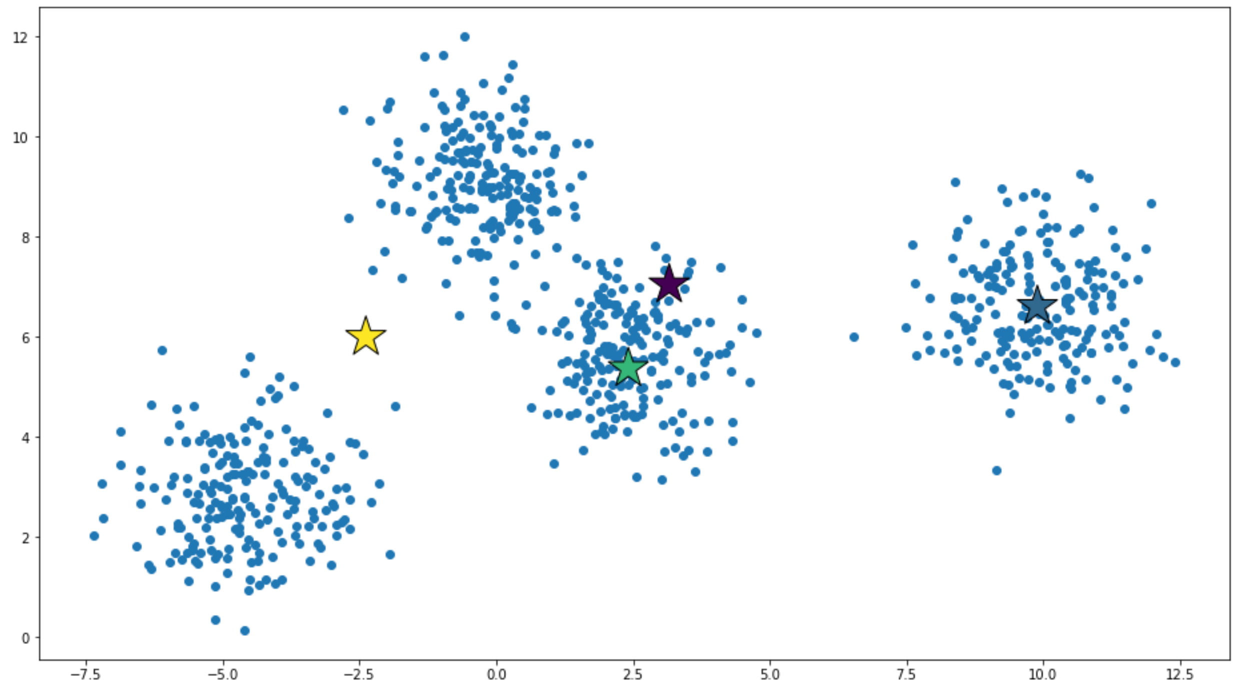
$\quad\quad j_{t,i} \leftarrow f_{\mu_t}(x_i)$

$\quad$ **for** $j \in \{1, \dots, k\}$:

$\quad\quad \mu_{t,j} \leftarrow \text{mean}(\{\, x_i \mid j_{t,i} = j \,\})$

$\quad$ **if** $\mu_t = \mu_{t-1}$:

$\quad\quad$ **return** μ_t



K-Means Clustering Algorithm

Kmeans(Z):

for $j \in \{1, \dots, k\}$:

$\mu_{1,j} \leftarrow \text{Random}(Z)$

for $t \in \{1, 2, \dots\}$:

for $i \in \{1, \dots, n\}$:

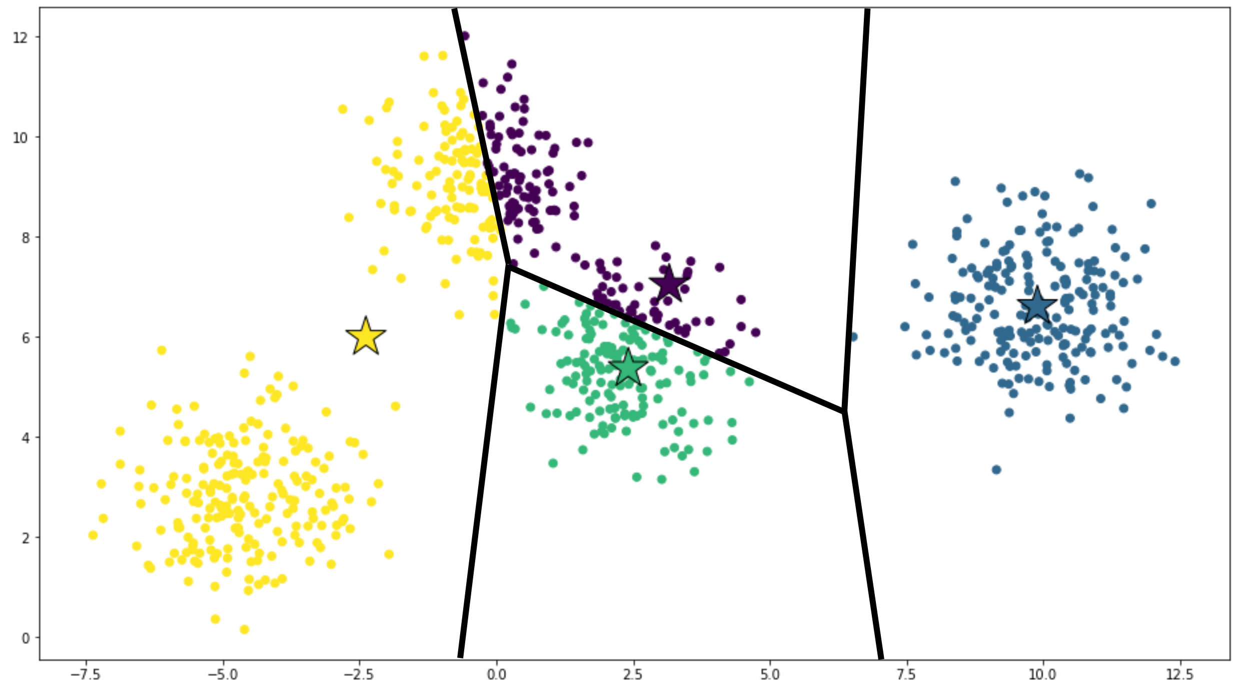
$j_{t,i} \leftarrow f_{\mu_t}(x_i)$

for $j \in \{1, \dots, k\}$:

$\mu_{t,j} \leftarrow \text{mean}(\{\, x_i \mid j_{t,i} = j \,\})$

if $\mu_t = \mu_{t-1}$:

return μ_t



K-Means Clustering Algorithm

Kmeans(Z):

for $j \in \{1, \dots, k\}$:

$\mu_{1,j} \leftarrow \text{Random}(Z)$

for $t \in \{1,2, \dots\}$:

for $i \in \{1, \dots, n\}$:

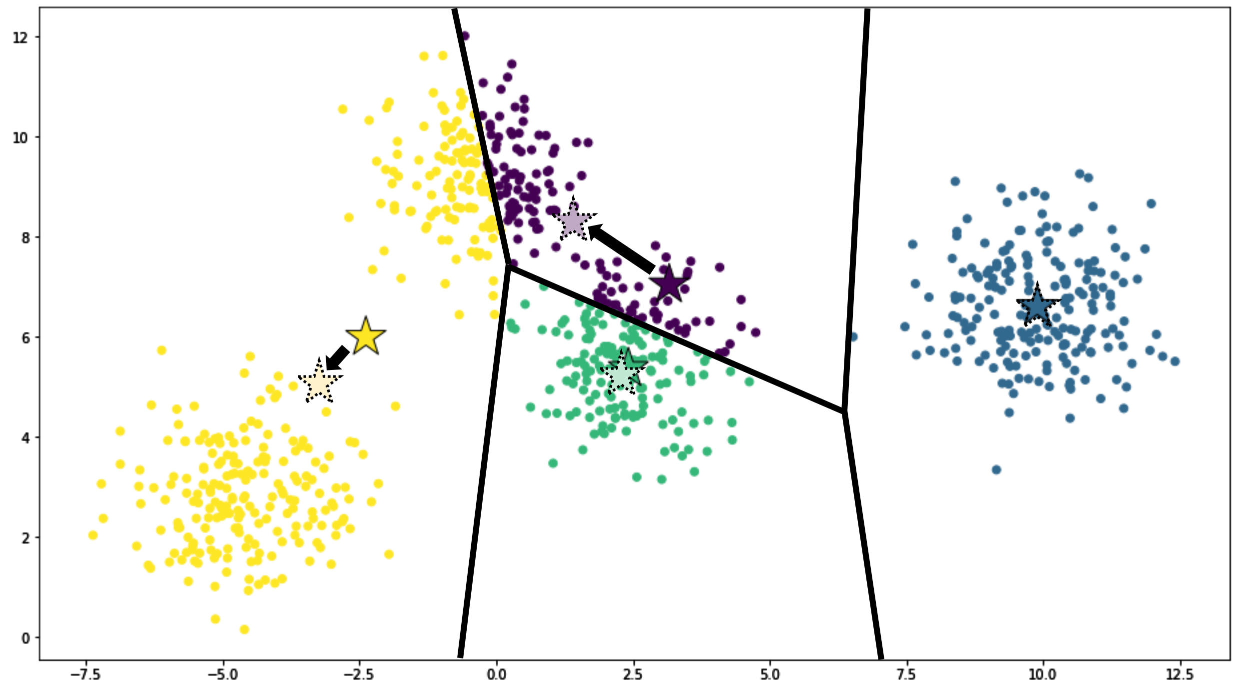
$j_{t,i} \leftarrow f_{\mu_t}(x_i)$

for $j \in \{1, \dots, k\}$:

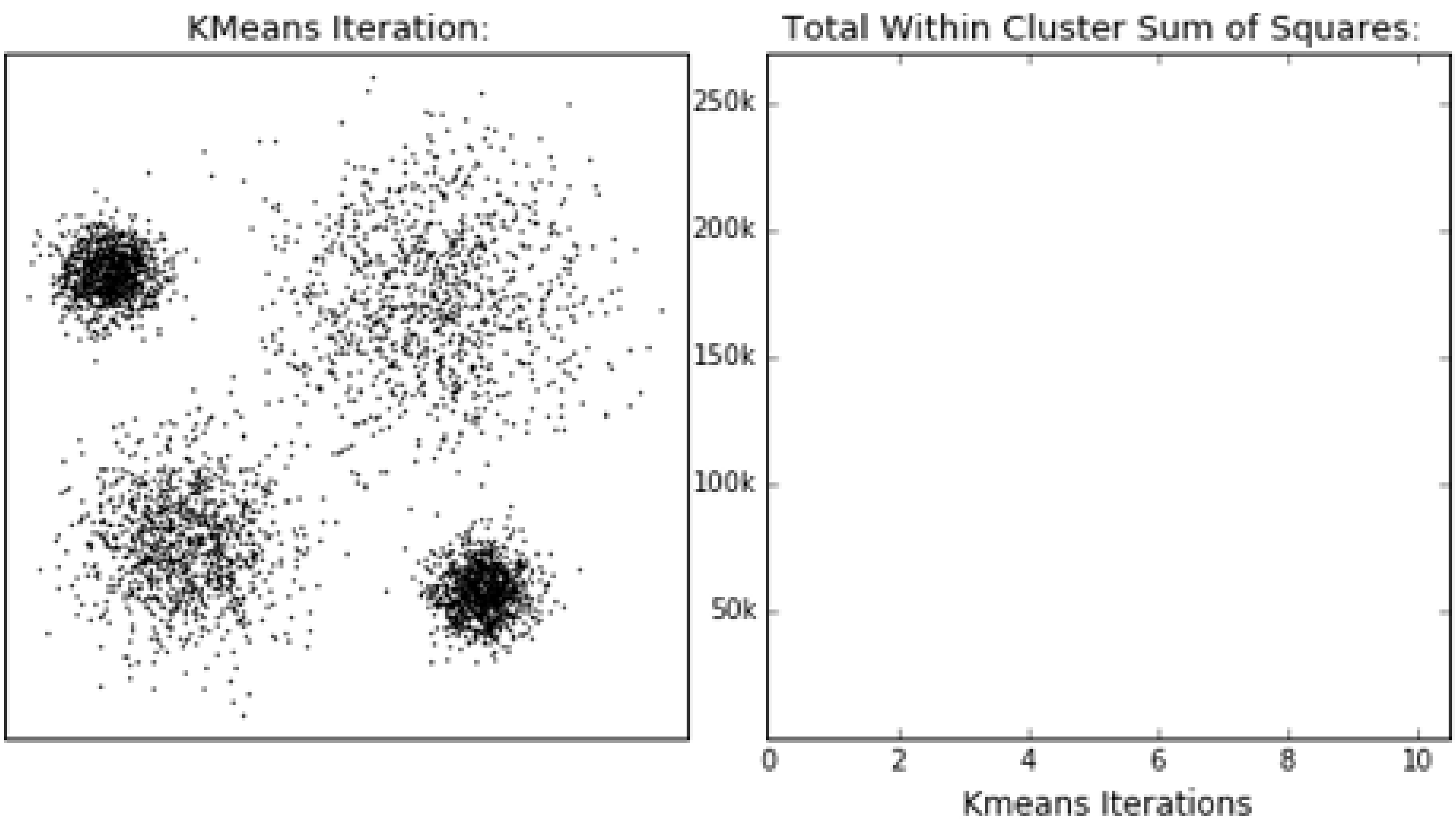
$\mu_{t,j} \leftarrow \text{mean}(\{\, x_i \mid j_{t,i} = j \,\})$

if $\mu_t = \mu_{t-1}$:

return μ_t



K-Means Clustering Algorithm

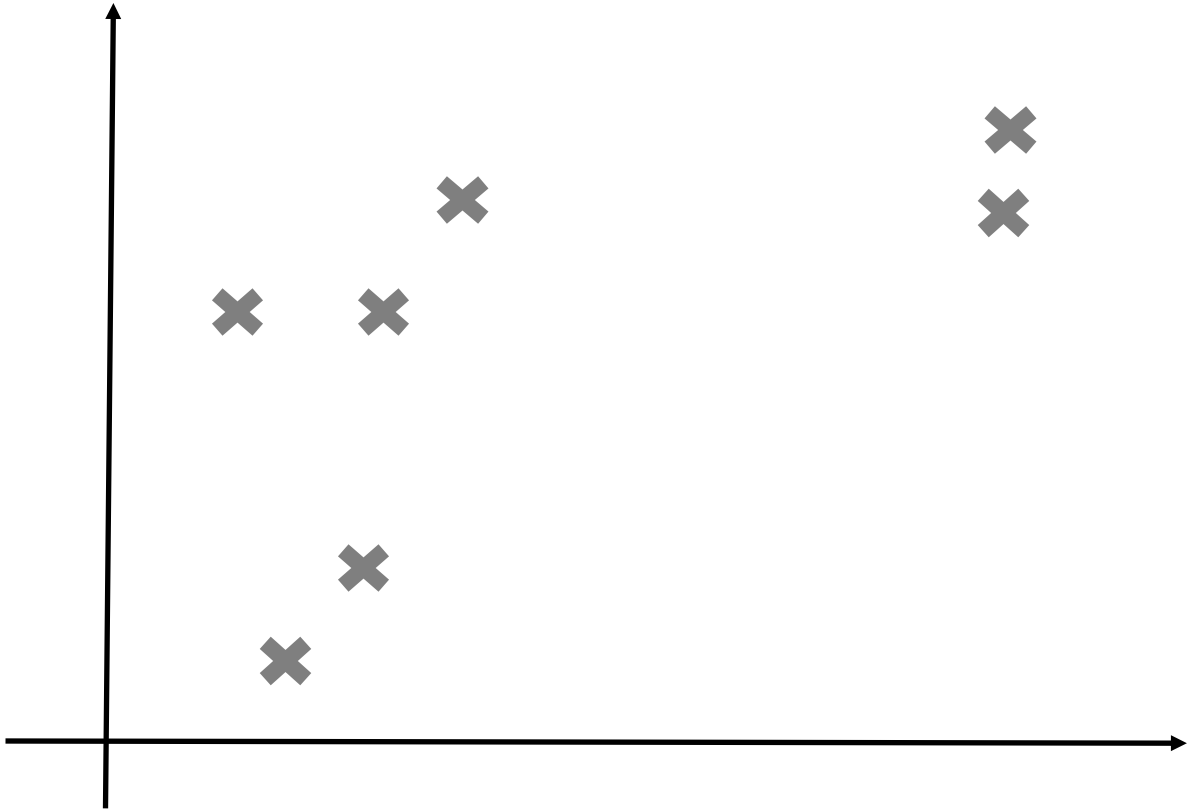


https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/

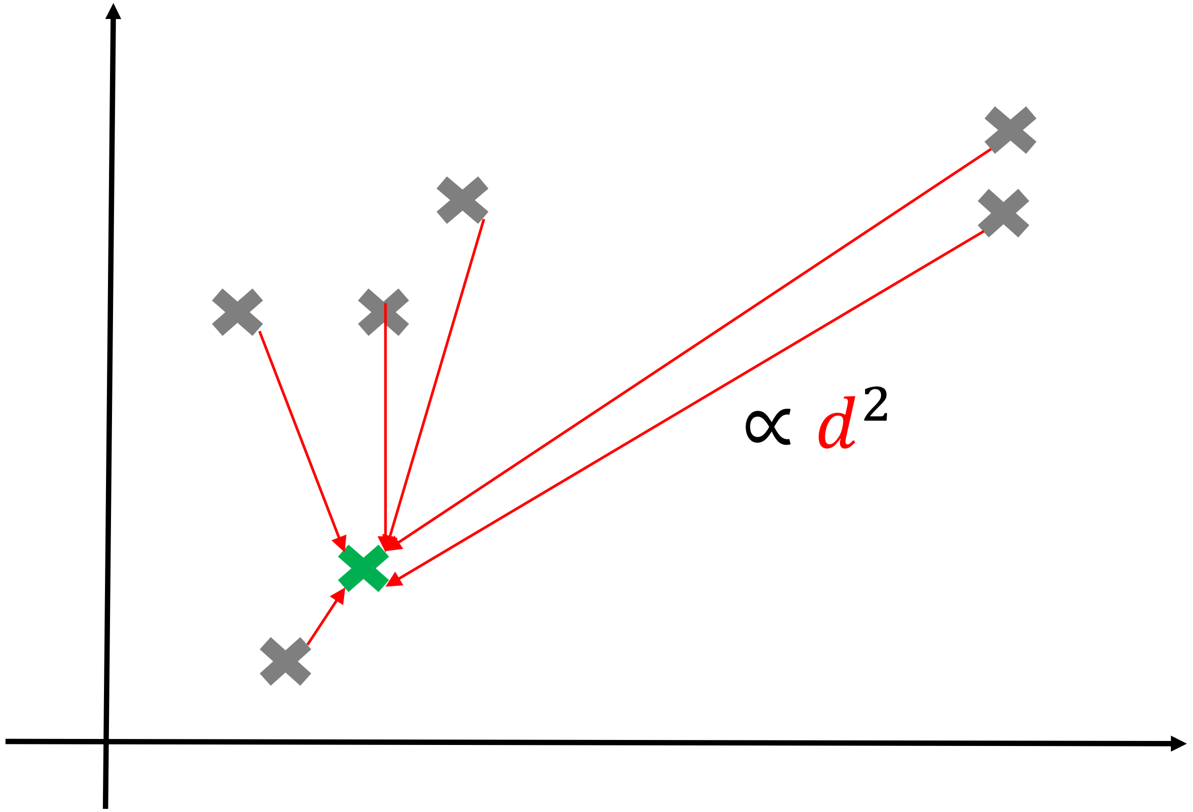
Random Initialization

- Sensitive to initialization
- One strategy is to run multiple times with different random centroids and choose the model with lowest MSE
- **Alternative:** K-means++
  - Randomly initialize first centroid to some $x \in Z$
  - Subsequently, choose centroid randomly according to $p(x) \propto d_x^2$, where d_x is the distance to the nearest centroid so far
  - Upweights points that are farther from existing centroids

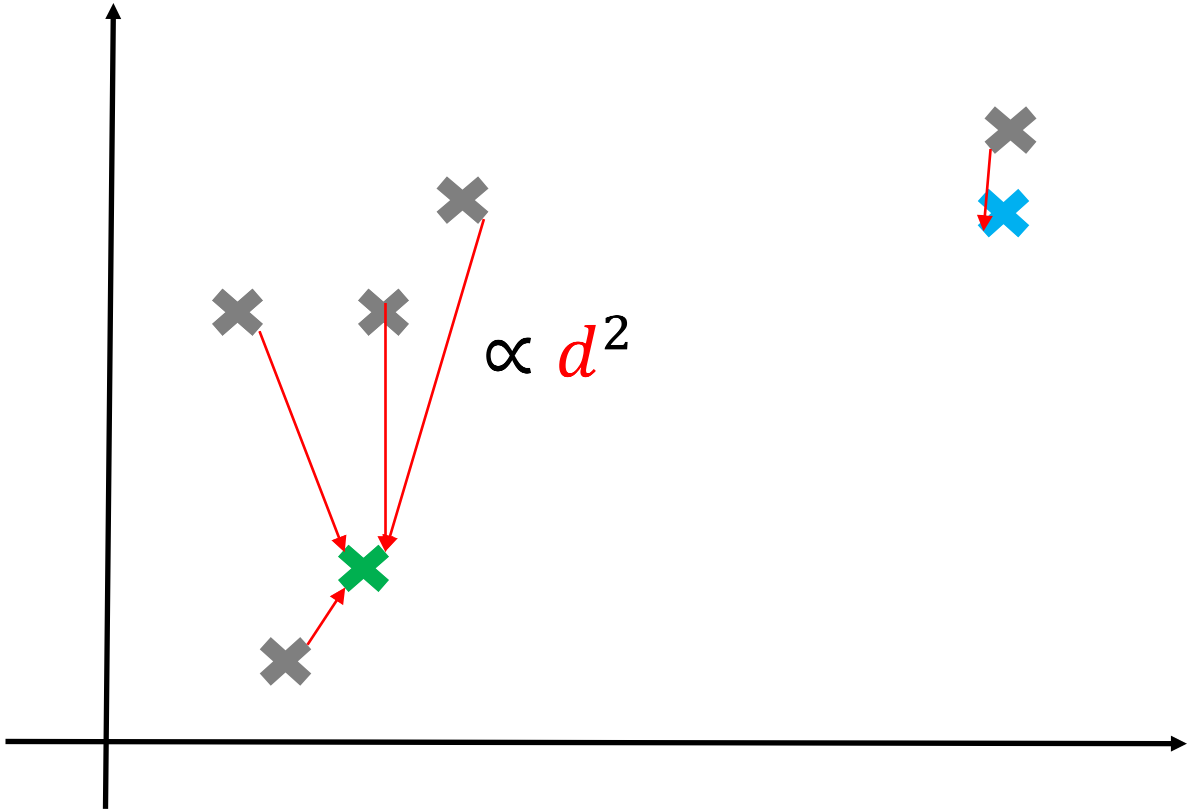
K-Means++ : Address initialization challenge



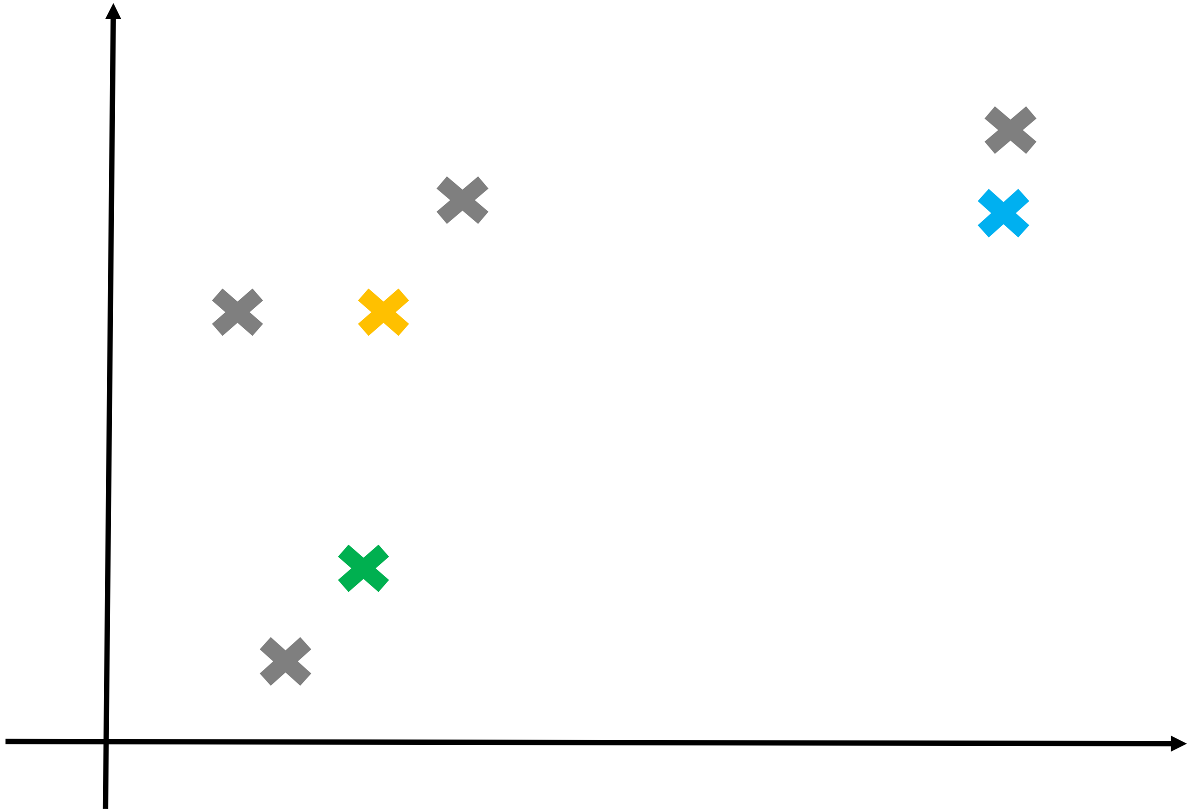
K-Means++



K-Means++



K-Means++



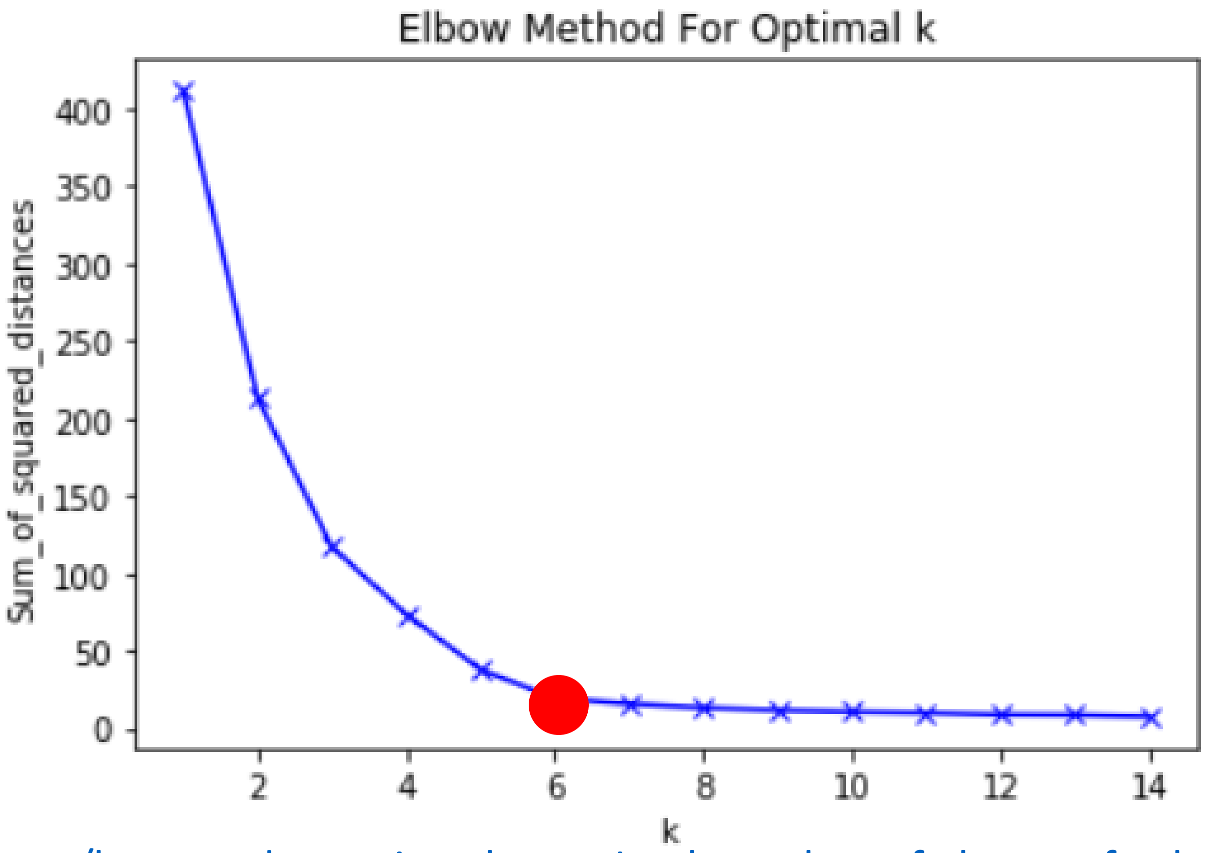
Then, run alternating minimization

Number of Clusters

- As K becomes large
  - MSE becomes small
  - Many clusters → might be less useful

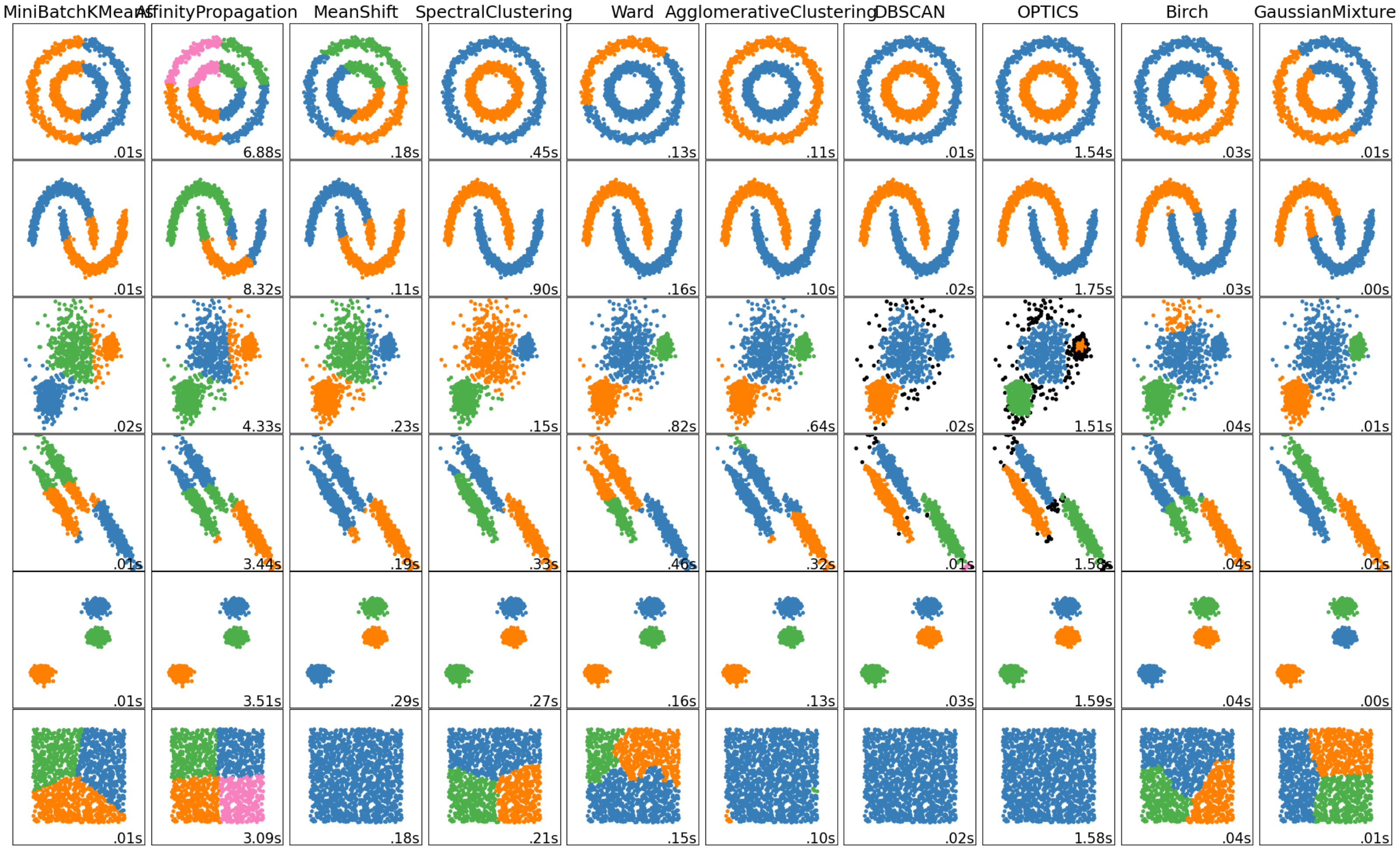
- Choice of K is subjective

Number of Clusters



https://blog.cambridgespark.com/how-to-determine-the-optimal-number-of-clusters-for-k-means-clustering-14f27070048f

Many Clustering Algorithms



https://scikit-learn.org/stable/modules/clustering.html#clustering