Machine Learning for All

Session 7. Unsupervised Machine Learning

César E. Montiel-Olea and Ana Yarygina (SPD)

March 8th, 2019

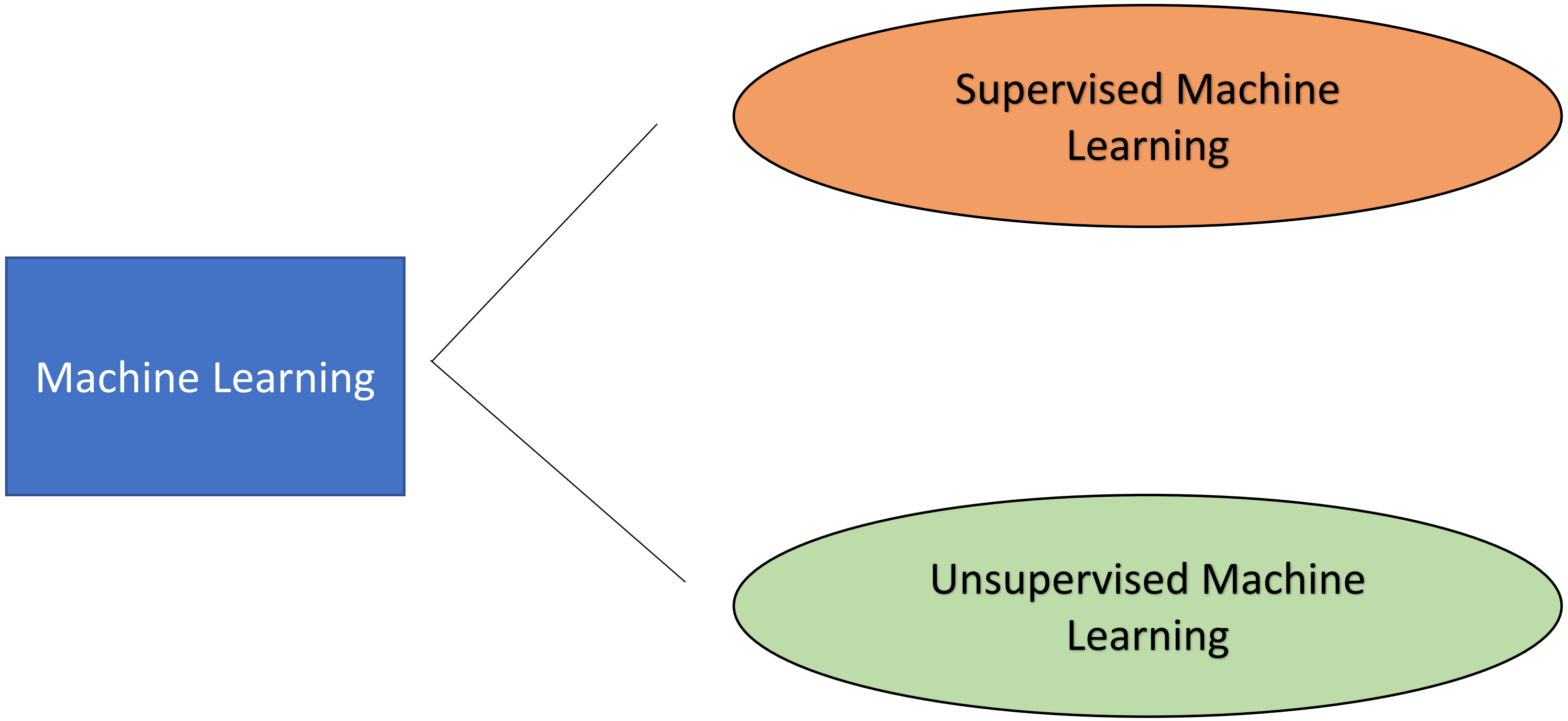
Roadmap

- Lessons learned from Supervised Machine Learning
- What is Unsupervised Machine Learning?
- Popular applications for Unsupervised Machine Learning
- The K-means clustering algorithm

Roadmap

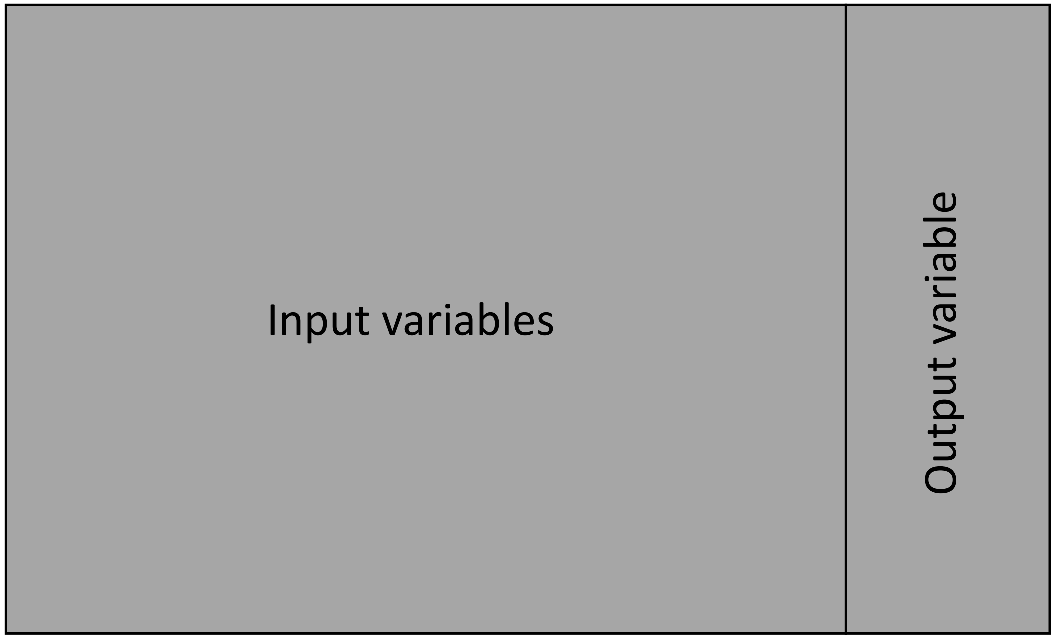
- **Lessons learned from Supervised Machine Learning**
- What is Unsupervised Machine Learning?
- Popular applications for Unsupervised Machine Learning
- The K-means clustering algorithm

Popular types of Machine Learning algorithms

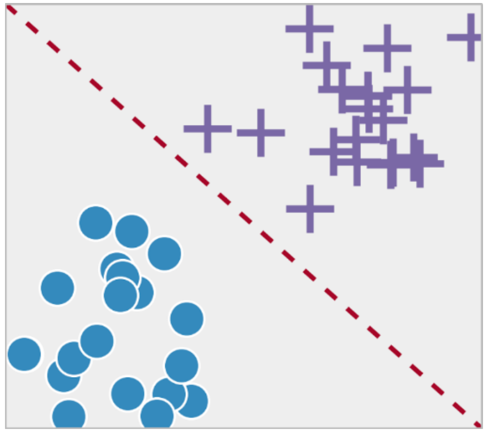


Lessons from Supervised Machine Learning

Data examples

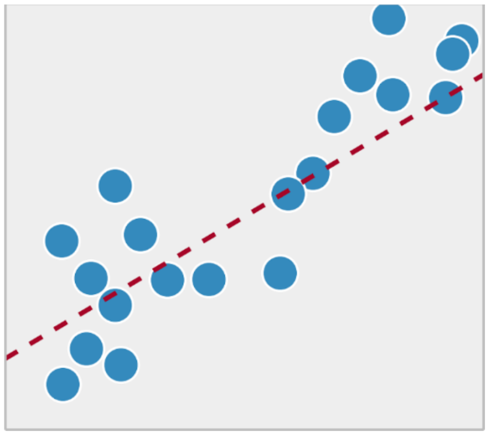


If categorical → classification problem



→ Spam problem

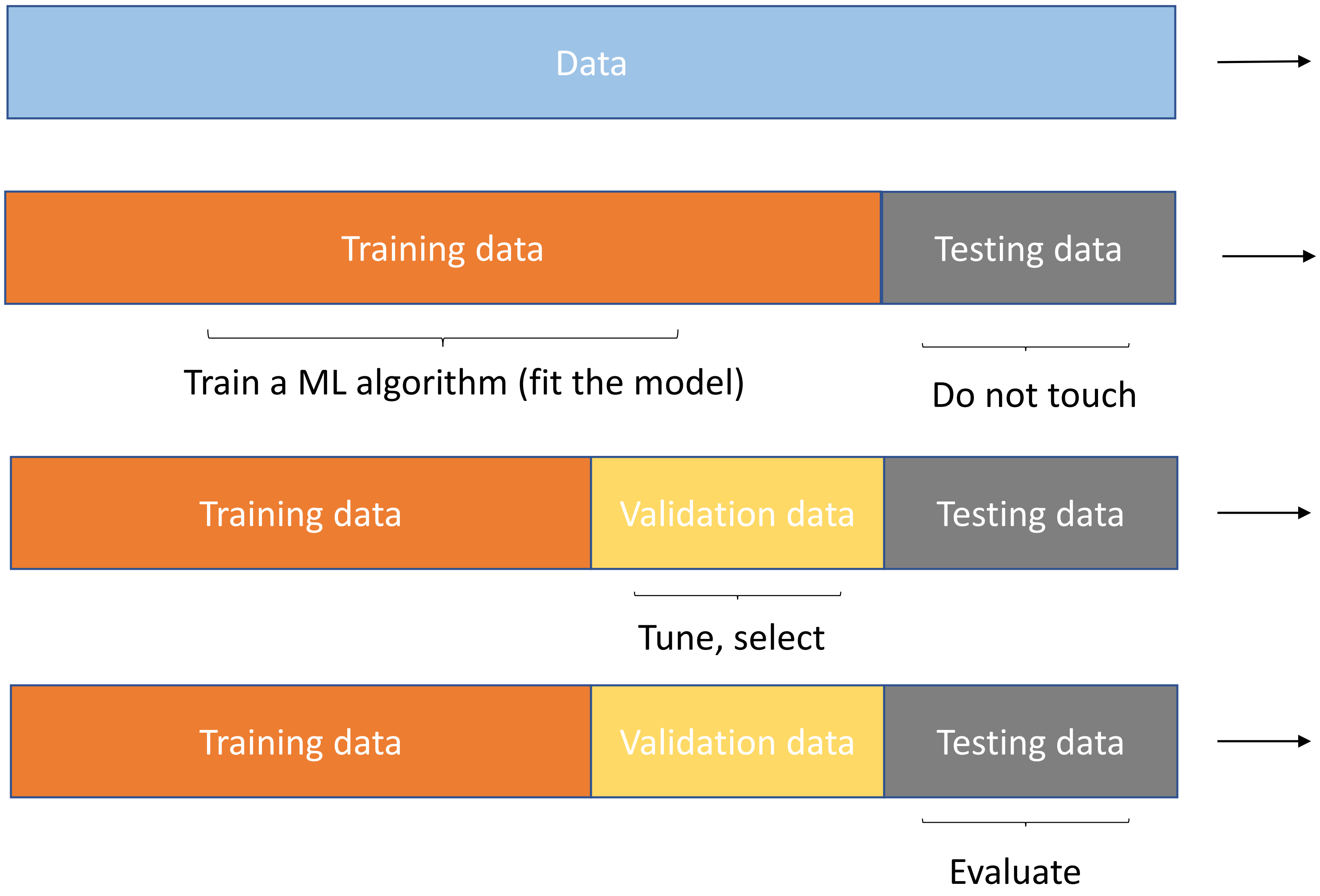
If continuous→ regression problem

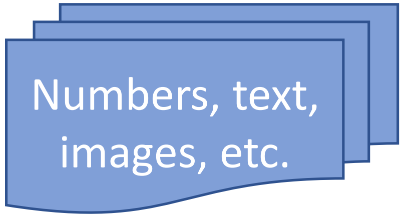


→ Predict housing values

Source of photo: https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

Lessons from Supervised Machine Learning





Randomly split the data

If there are tuning parameters, perform cross-validation

Evaluate performance using model obtained from training data

Lessons from Supervised Machine Learning

- In general we are interested in finding a **learning method which predicts well out of sample** (i.e. in the testing data)

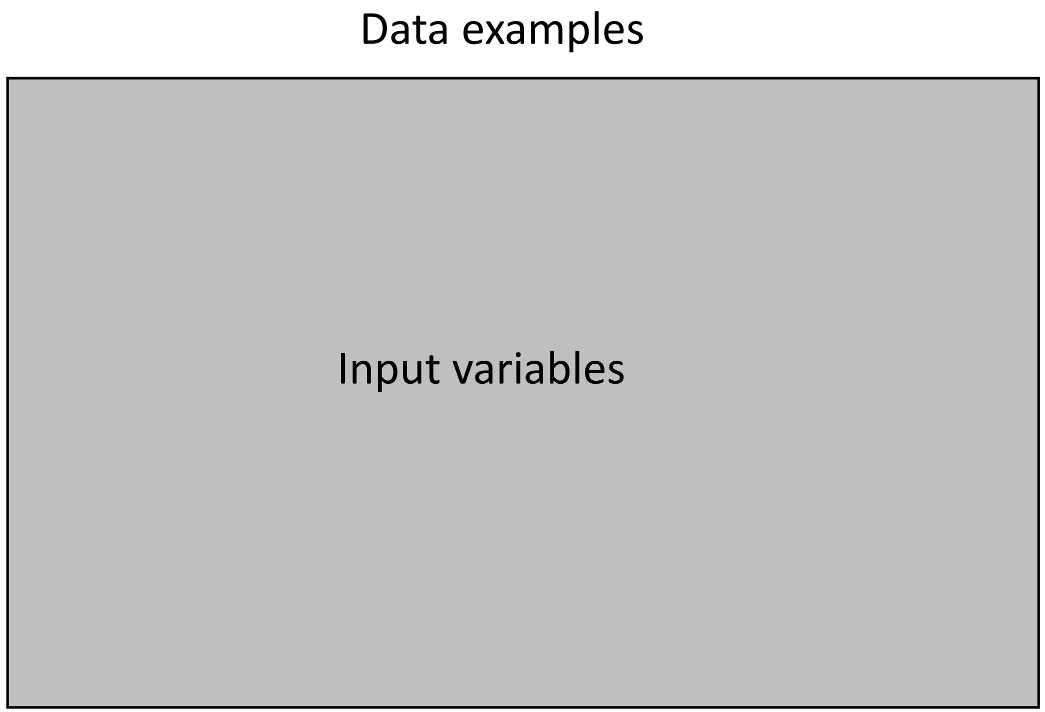
- A **very flexible or complex model** can in principle perform well in the training data but bad in the testing data. Moreover, it could lead to **overfitting,** which we want to avoid.

- Which learning method to use depends on the data, the problem that we seek to solve, and the computational capacity.

Roadmap

- Lessons learned from Supervised Machine Learning
- **What is Unsupervised Machine Learning?**
- Popular applications for Unsupervised Machine Learning
- The K-means clustering algorithm

What is Unsupervised Machine Learning?



Key differences with respect to SML

1. We are not interested in predicting an outcome because we just have inputs or covariates.

2. We want to find structures or patterns in the data without knowing the solution.

3. We perform it as part of an exploratory data analysis

Popular algorithms: Principal Component Analysis, K-means clustering, Hierarchical Clustering

What is Unsupervised Machine Learning?

More technical description:

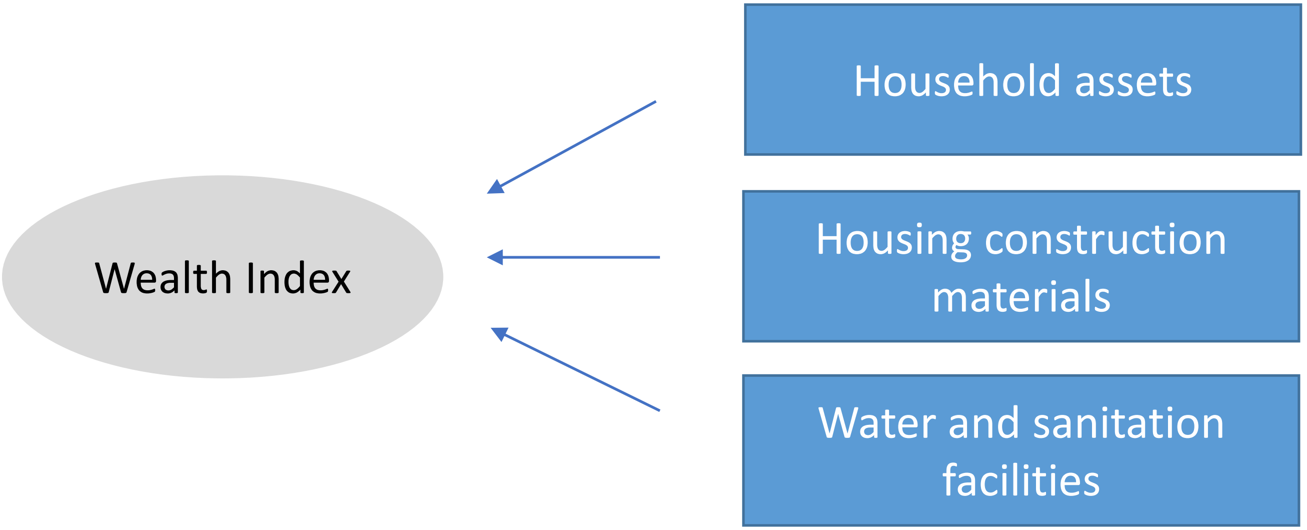
- No label information is available (not outcome to predict or classify)
- There is no training and testing data
- Our purpose is to fit model when only $x_1, \ldots, x_n$ are available

Roadmap

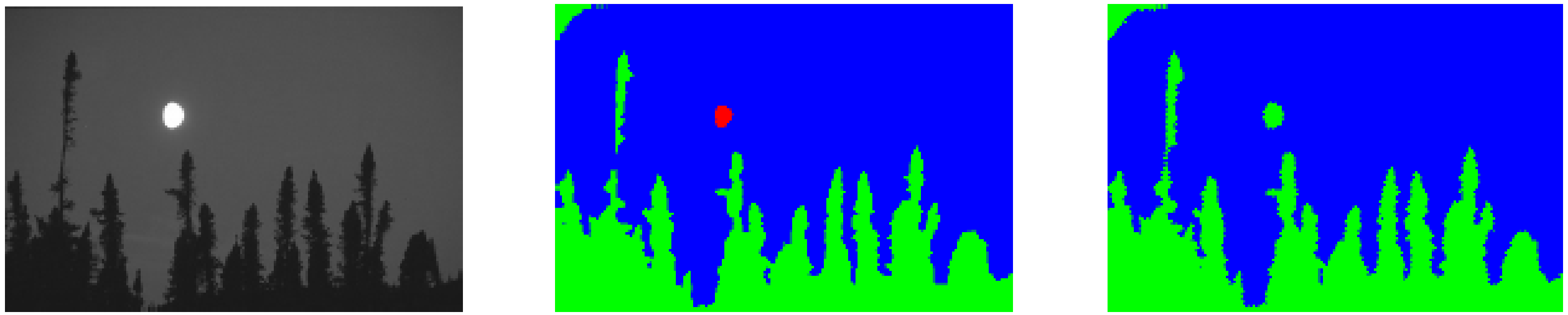
- Lessons learned from Supervised Machine Learning
- What is Unsupervised Machine Learning?
- **Popular applications for Unsupervised Machine Learning**
- The K-means clustering algorithm

Popular applications for Unsupervised Machine Learning

- Dimension Reduction: Find a low-dimensional representation of the data



- Image segmentation: Partitioning images into coherent groups



Roadmap

- Lessons learned from Supervised Machine Learning
- What is Unsupervised Machine Learning?
- Popular applications for Unsupervised Machine Learning
- **The K-means clustering algorithm**

A simple K-means clustering for crime data

We start with a set of data points that we wish to accommodate into K different, non-overlapping clusters

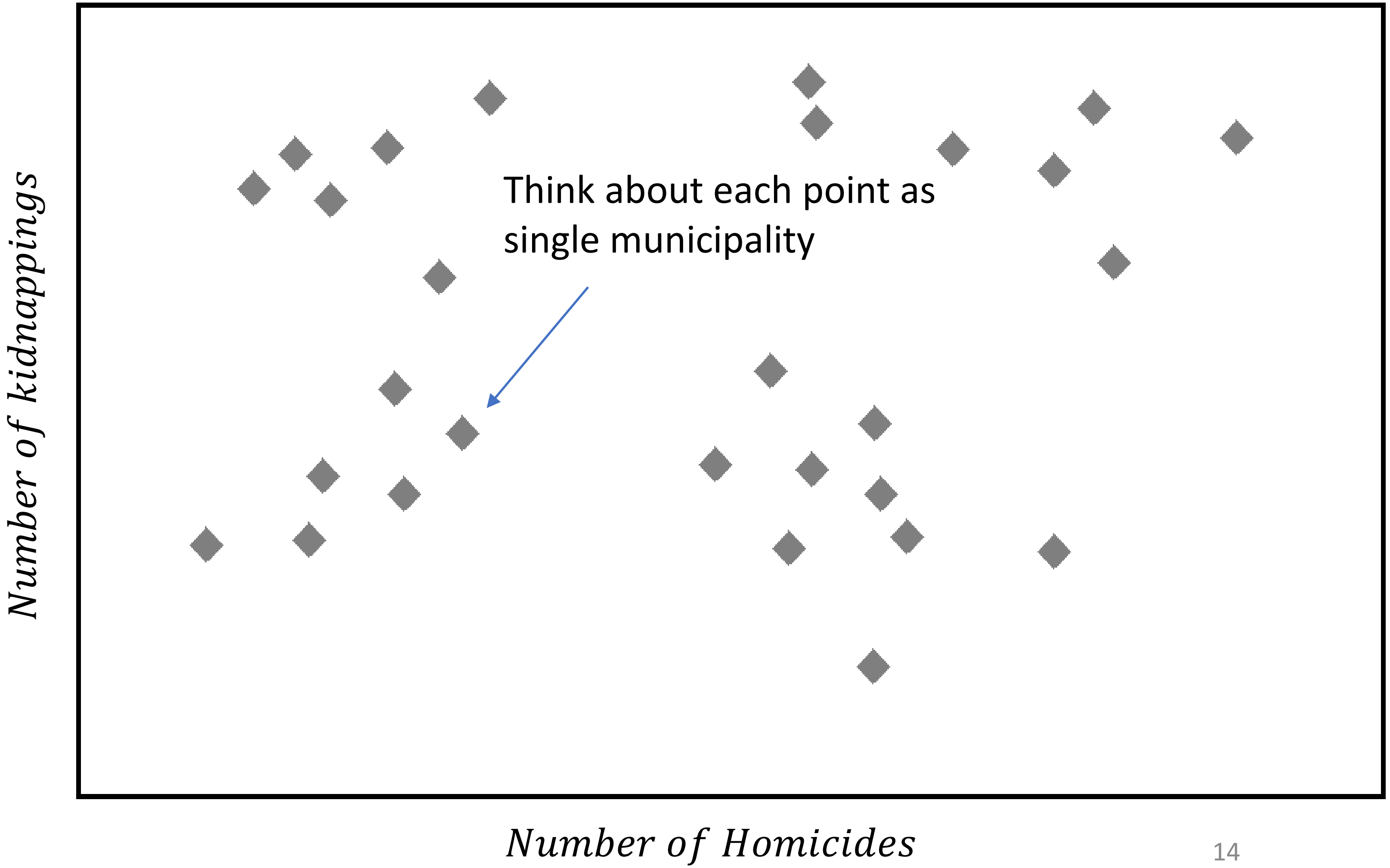
Inputs:

- K (number of clusters) that we want
- Data points $x_i, \dots, x_n \in \mathbb{R}^d$

We want to assign each data point to a cluster.

Clustering tries to find homogenous groups in the data

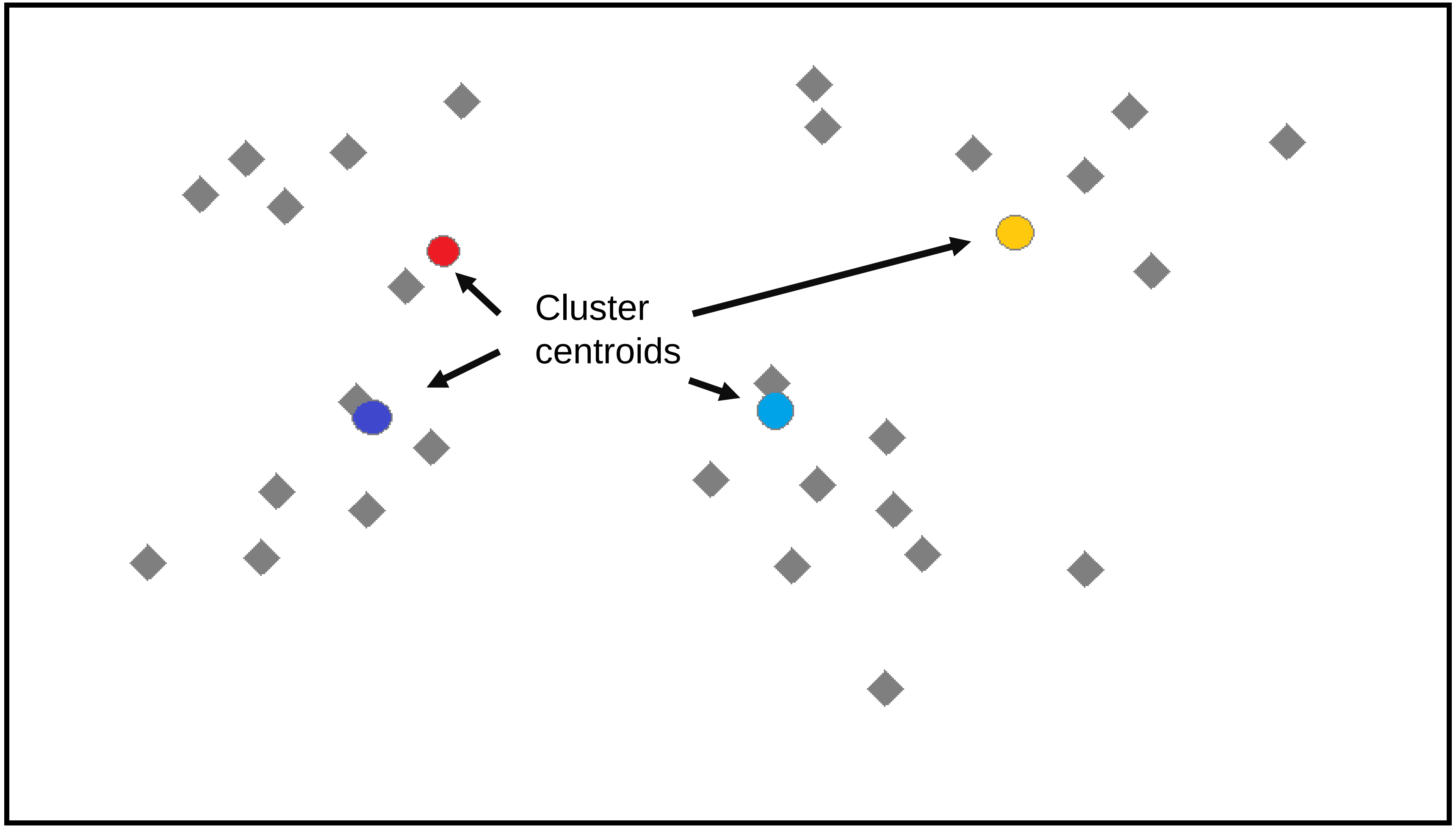
Each diamond represents a data point. In this example, we assume that each data point is a $\mathbb{R}^2$ dimensional vector



Step 1. Initialization

We randomly define K initial cluster centroids (or center means), one for each cluster $\mu_1, \dots, \mu_K \in \mathbb{R}^d$

Think about the cluster centroids as random municipalities in my data that have both homicide and kidnapping counts.



In this case, each color represents a different cluster centroid. In this image, we have four cluster centroids, i.e., K = 4

Step 2. Cluster assignment

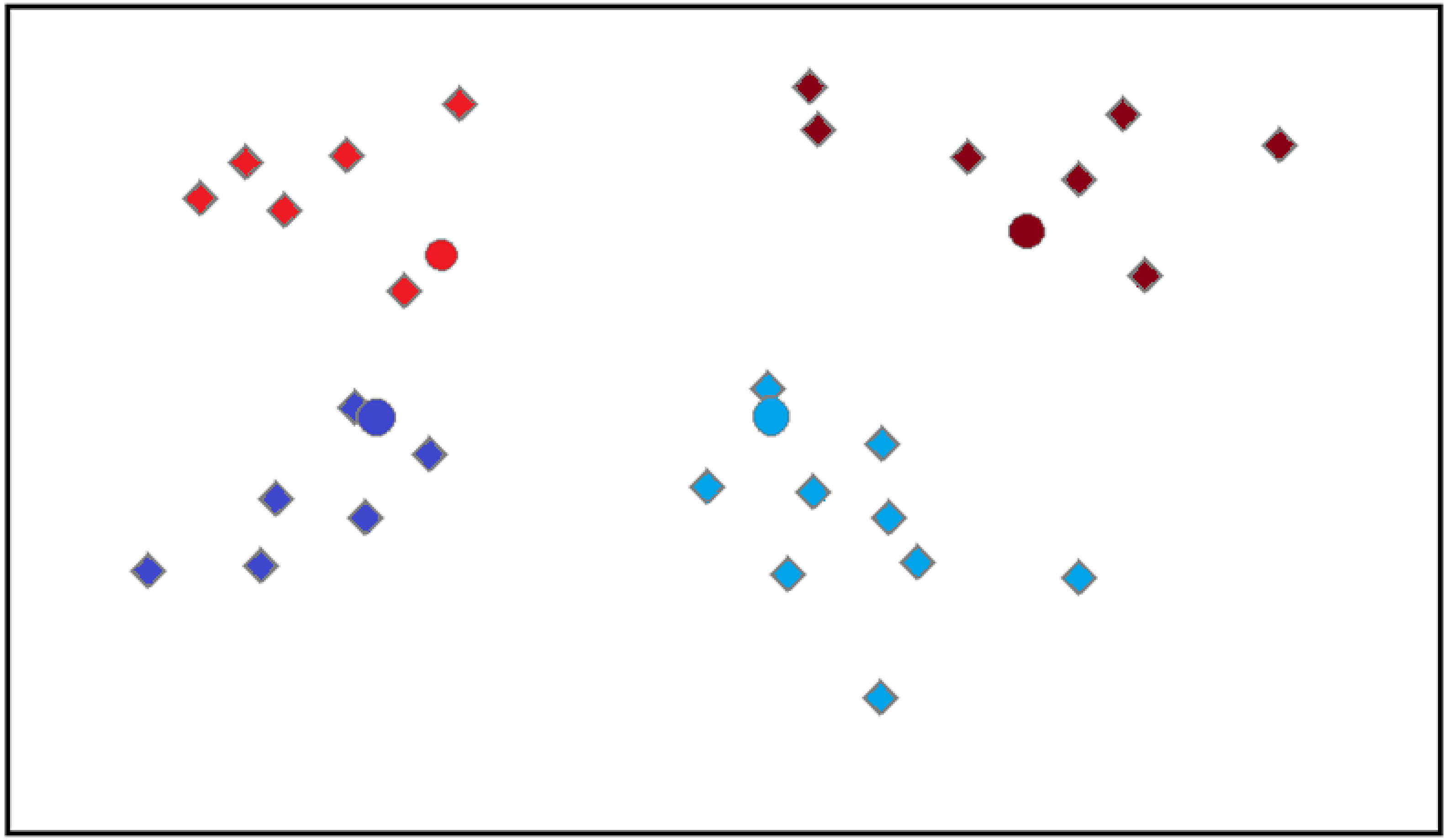
The algorithm assigns each data point (or municipality) to the closest cluster centroid

Why do we minimize the distance?

By using the **Within Cluster Sum of Squares**:

$$\min_{k} \|x_i - \mu_k\|_2^2$$

We want the Within Cluster Sum of Squares to be as small as possible.



In this case, we are going to colored our data points (or municipalities) based on the color of the closest centroid mean.

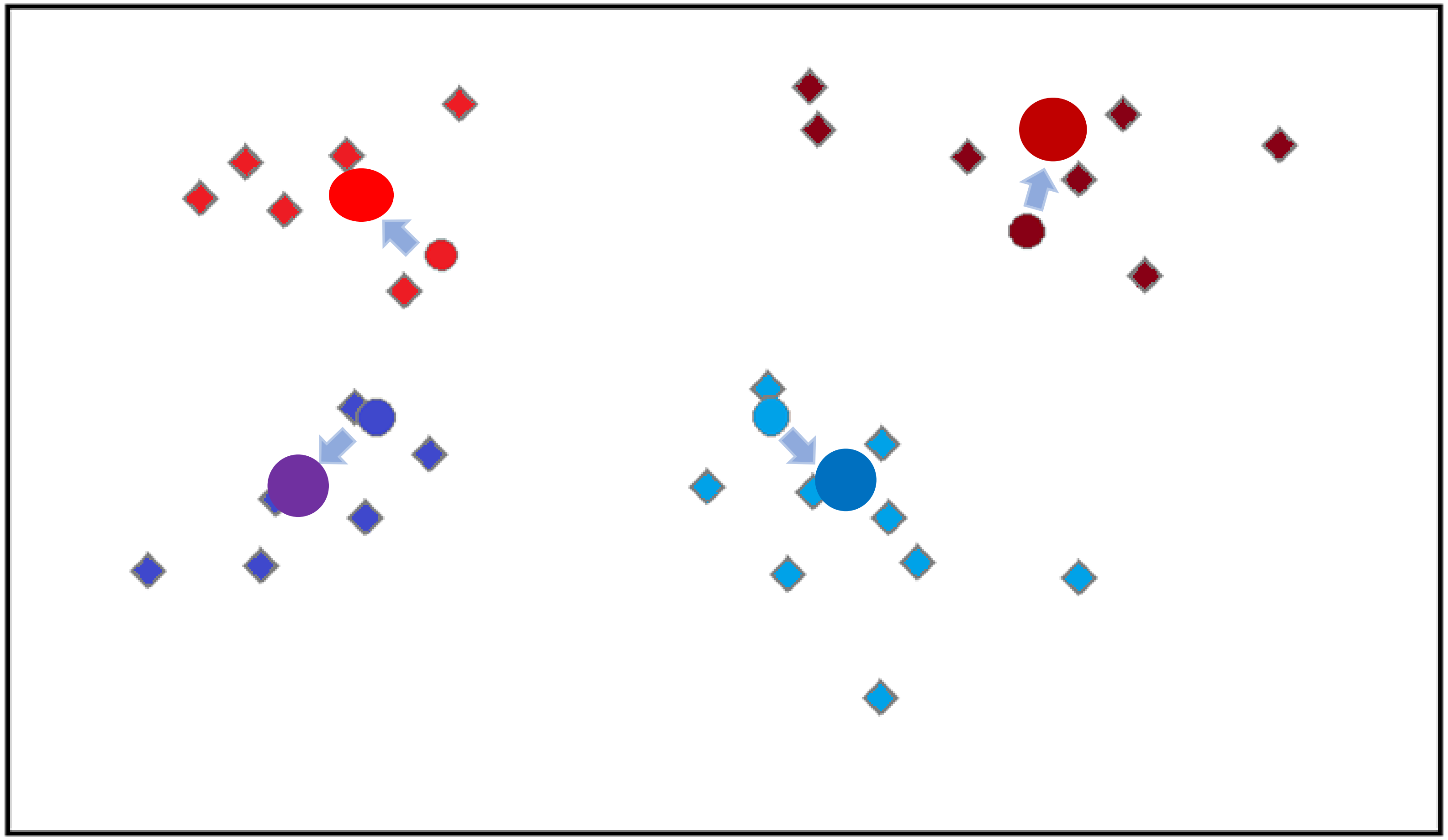
Step 3. Move (update) centroids

We now compute the average of the points in each cluster

Example:

$x_1, x_3, x_5, x_8 \Rightarrow c_1 = 2, c_3 = 2, c_5 = 2, c_8 = 2$

$$\mu_2 = \frac{1}{4}[x_1, x_3, x_5, x_8]$$



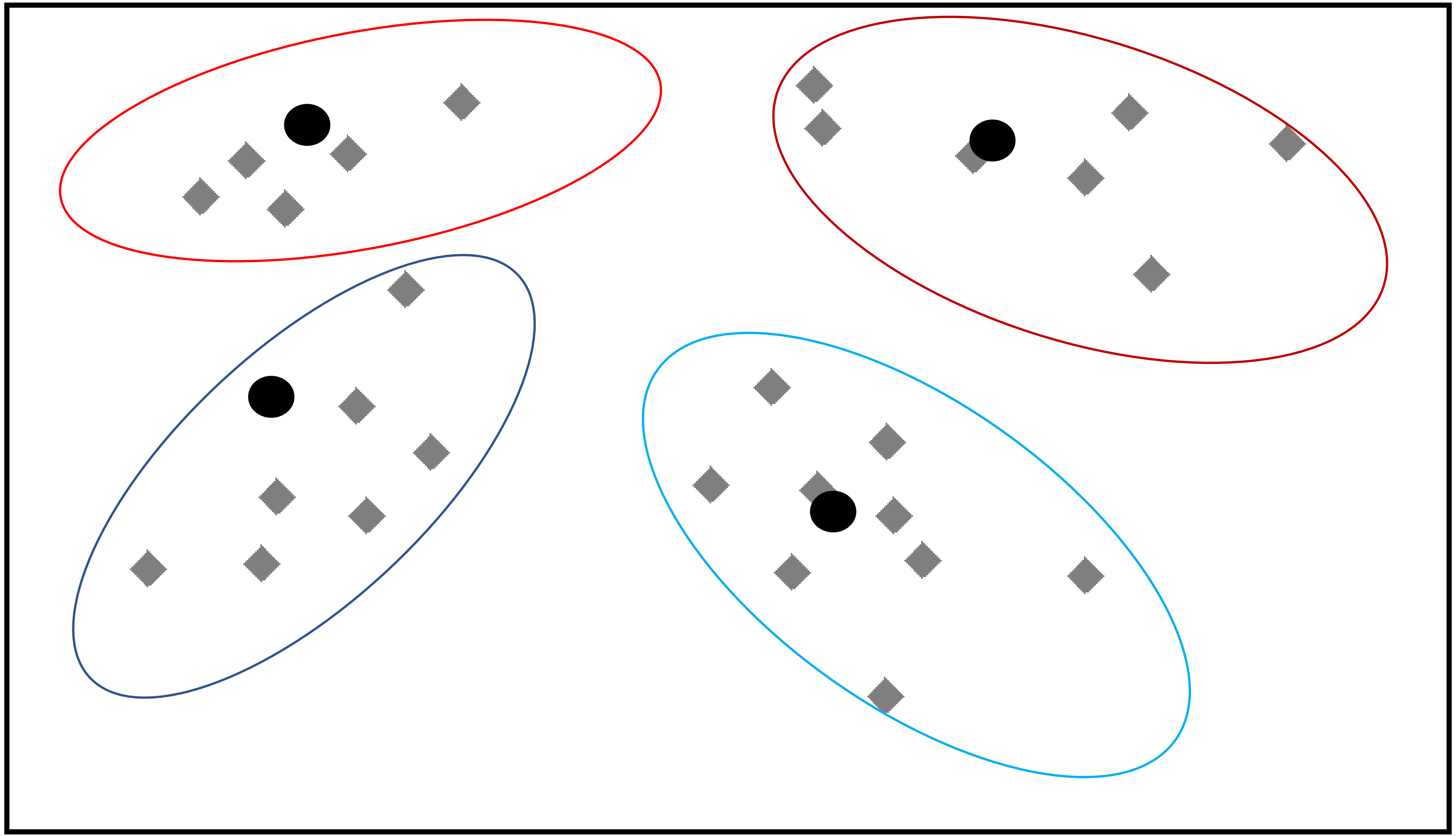
Now the cluster centroids have been updated (to facilitate visualization the new centroids are shown in a bigger size).

Step 4. We iterate steps 2 and 3 until the centroids no longer move

When will they stop moving?

When the Within Cluster Sum of Squares stops improving

We finally have our cluster centers and our 4 clusters of municipalities in terms of homicides and kidnappings.



The black circles are our final cluster centroids. Ideally, we want the algorithm to converge to a global optimum, but in general it converges to a local optima. For this reason, we normally run the algorithm multiple times using different random initializations.

K-means clustering formally

Suppose we have the set of points $x_i, \dots, x_n \in \mathbb{R}^d$ and we seek to accommodate them into k different clusters

1) Initialization: Randomly choose k "cluster centers" or means $\mu_1, \dots, \mu_K \in \mathbb{R}^d$

   For K clusters we encode assignments to clusters as $m_i = k \Leftrightarrow x_i$ assigned to cluster k, and $C_k = \{i | m_i = k\}$ denote the sets containing the indices for the observations that correspond to each cluster k

2) Iterate until convergence ($j =$ iteration number).

   Assign each point x_i to the closest mean (in Euclidean distance)

$$m_i^{(j+1)} \coloneqq \underset{k \epsilon \{1,\dots K)}{\arg\min} \left\| x_i - \mu_k^{(j)} \right\|$$

3) Recompute each $\mu_k^{(j)}$ as the mean of all points assigned to it:

$$\mu_k^{(j)} \coloneqq \sum_{i | m_i^{(j+1)} = k} \frac{x_i}{\left| \{i | m_i^{(j+1)} = k\} \right|}$$

4) Terminate when total change of means satisfies for example $\sum_{k=1}^{K} \left\| \mu_k^{(j+1)} - \mu_k^{(j)} \right\| < \tau$

Now let’s code with Ana!