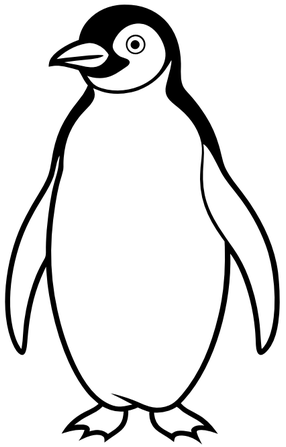
Unsupervised learning with scikit-learn

Supervised vs. Unsupervised

Culmen Length	Culmen Depth	Flipper Length	Body Mass	Species
39.1mm	18.7mm	181.0mm	3.75kg	Adelie
43.5mm	18.1mm	202.0mm	3.40kg	Chinstrap
39.5mm	17.4mm	186.0mm	3.80kg	Adelie
46.1mm	13.2mm	211.0mm	4.50kg	Gentoo

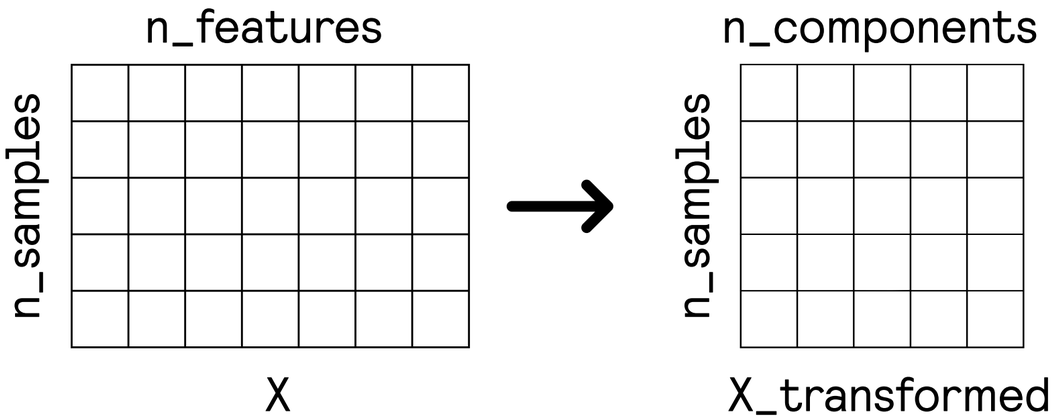


Supervised vs. Unsupervised

Age	Workclass	Education	Marital-status	Occupation	Relationship	Capital-gain	Hours-per-week	Native-country	Class
25	Private	11th	Never-married	Machine-op-inspct	Own-child	0	40	United-States	<=50K
38	Private	HS-grad	Married-civ-spouse	Farming-fishing	Husband	0	50	United-States	<=50K
28	Local-gov	Assoc-acdm	Married-civ-spouse	Protective-serv	Husband	0	40	United-States	>50K
44	Private	Some-college	Married-civ-spouse	Machine-op-inspct	Husband	7688	40	United-States	>50K

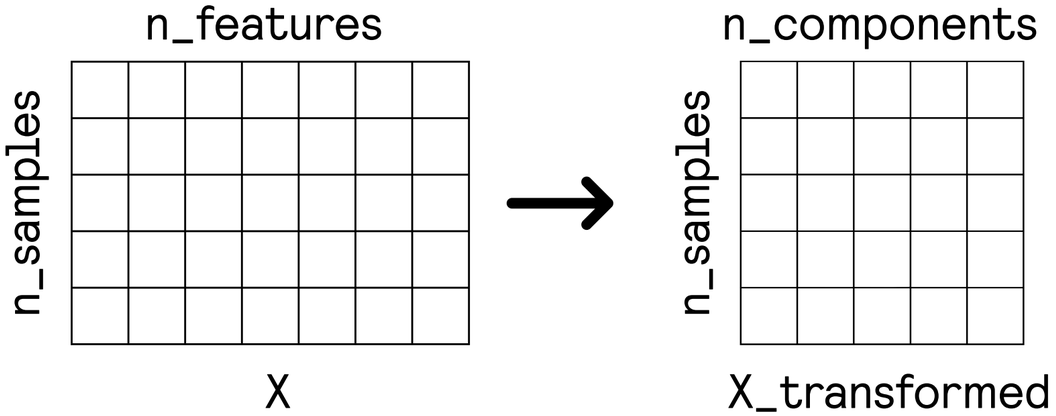
Examples of unsupervised learning

- Dimensionality Reduction
  - PCA
  - TSNE
  - NMF

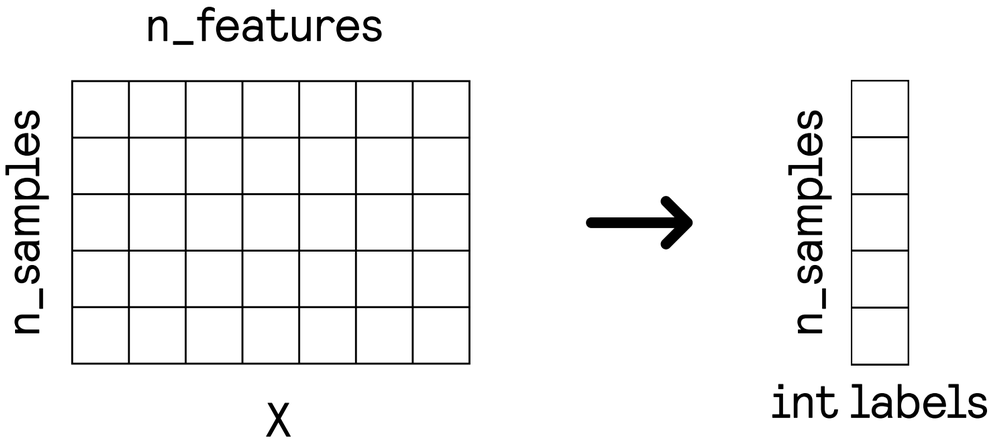


Examples of unsupervised learning

- Dimensionality Reduction
  - PCA
  - TSNE
  - NMF



- Clustering
  - K-means
  - HDBSCAN
  - Agglomerative



Clustering

- Similar features = proximity in the feature space
- Notion of a distance between two points d(x1, x2)
- Choice of feature scale has a large effect!

d ( , ) =

d (

Culmen Length	Culmen Depth	Flipper Length
39.1mm	18.7mm	181.0mm

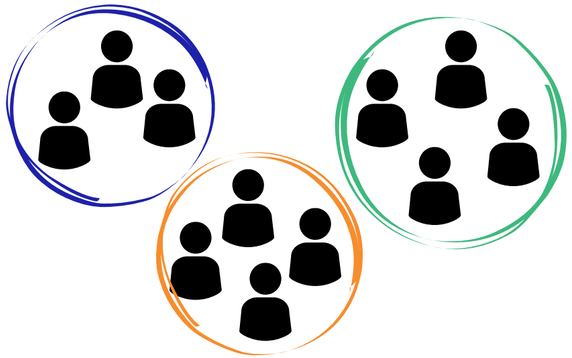
,

Culmen Length	Culmen Depth	Flipper Length
39.5mm	17.4mm	186.0mm

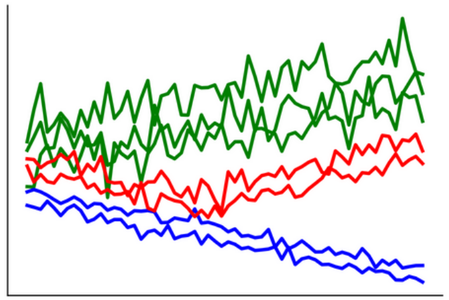
)

Applications of clustering

Document grouping



Market segmentation



Portfolio diversification

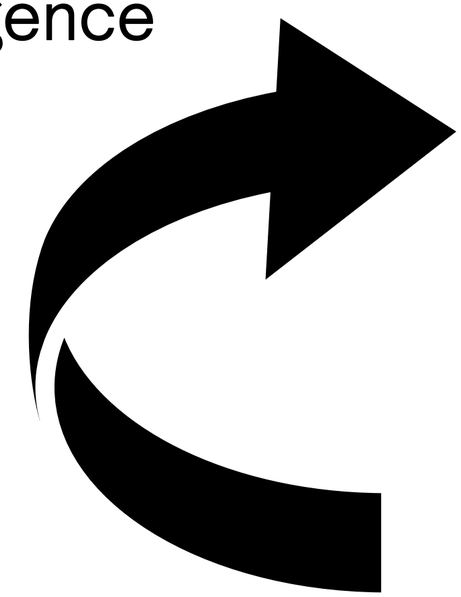
Points of interest
from GPS data

The K-Means algorithm

- Parametric centroid-based clustering
- Clusters are independent of each other
- Aims to minimize the within-cluster sum of squared distances (WCSS) or inertia

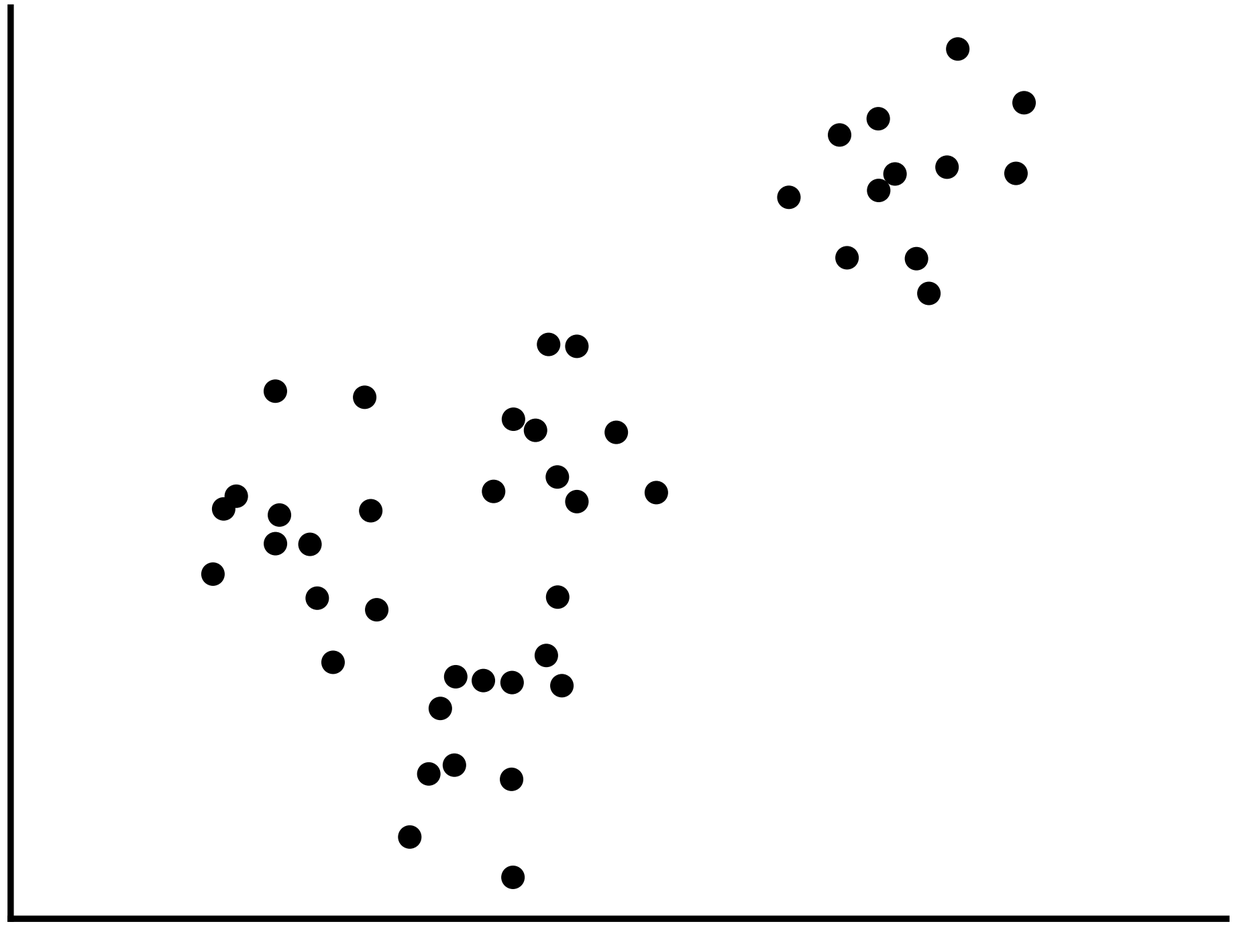
The K-Means algorithm

Repeat until convergence

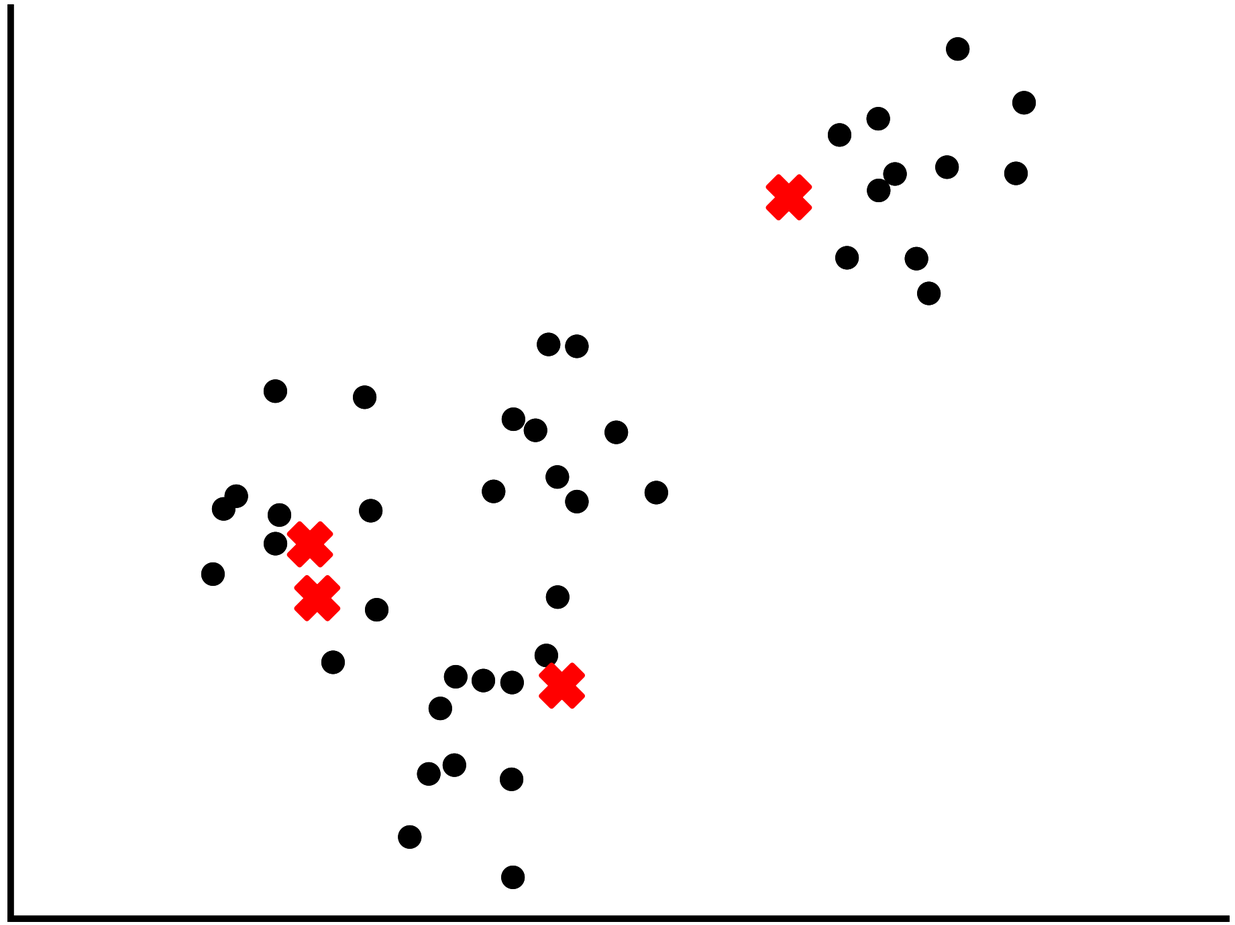


- Centroids initialization
- Assigning each point to a cluster
- Centroids update
- Check convergence

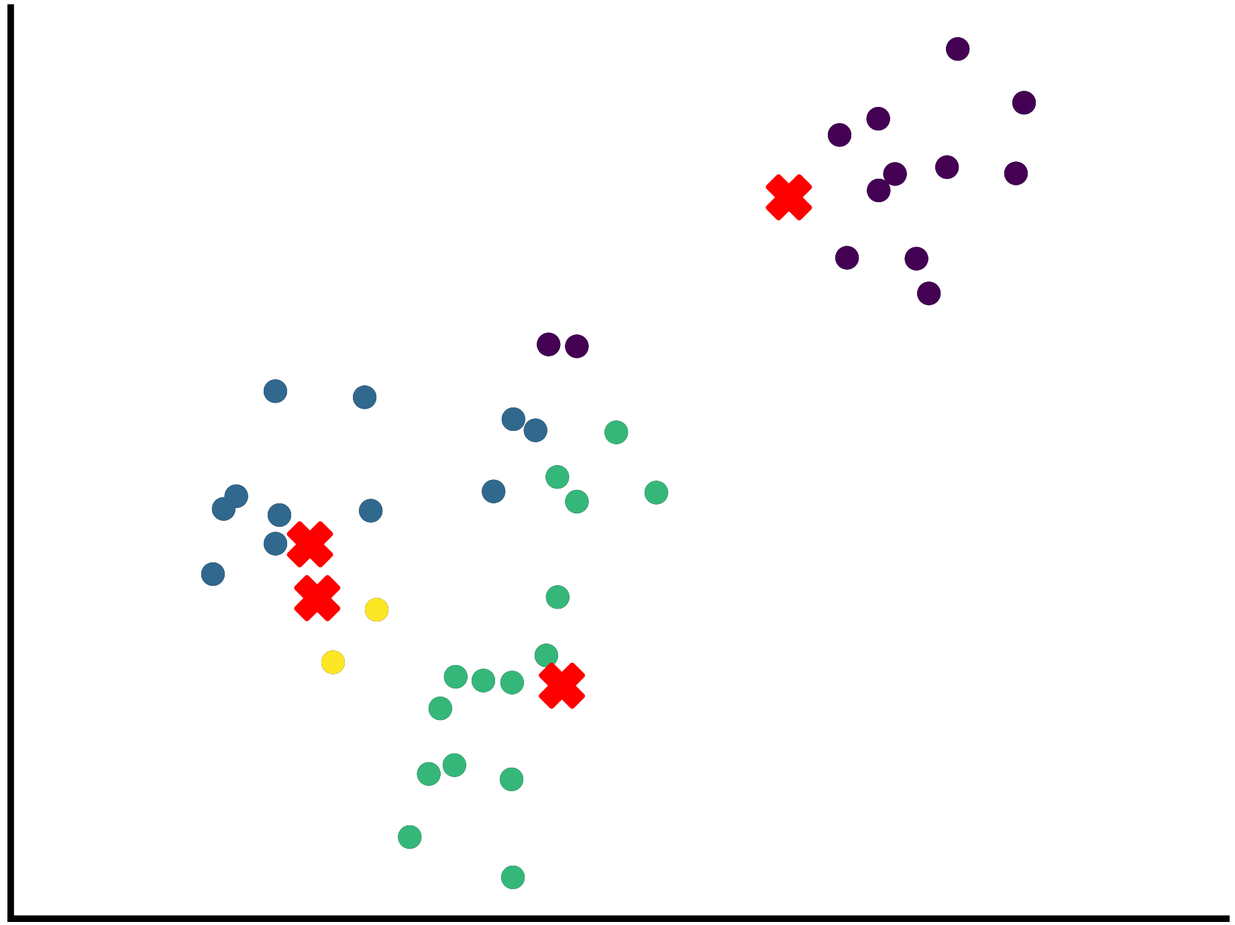
The K-Means algorithm



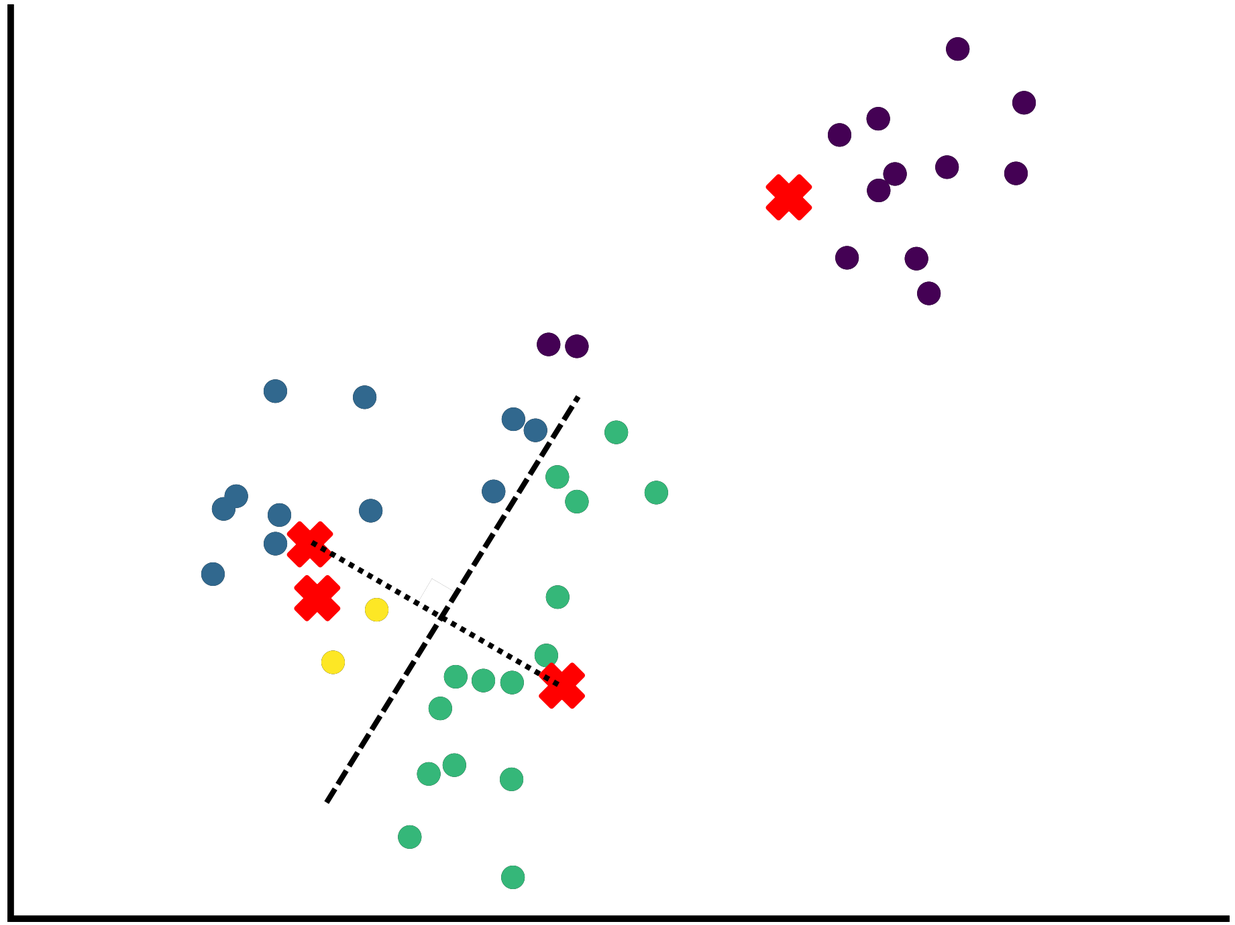
Initialization



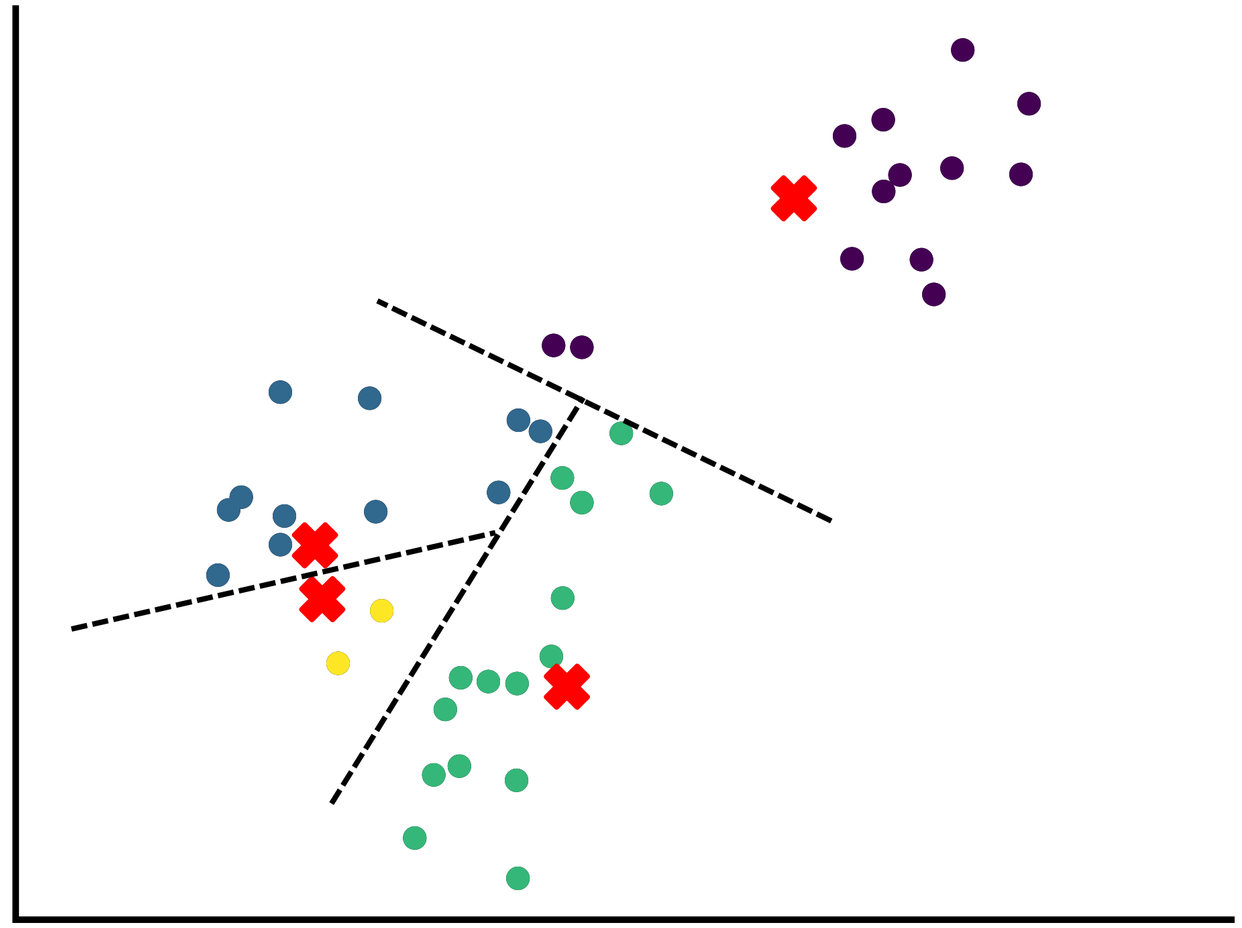
Points Assignment



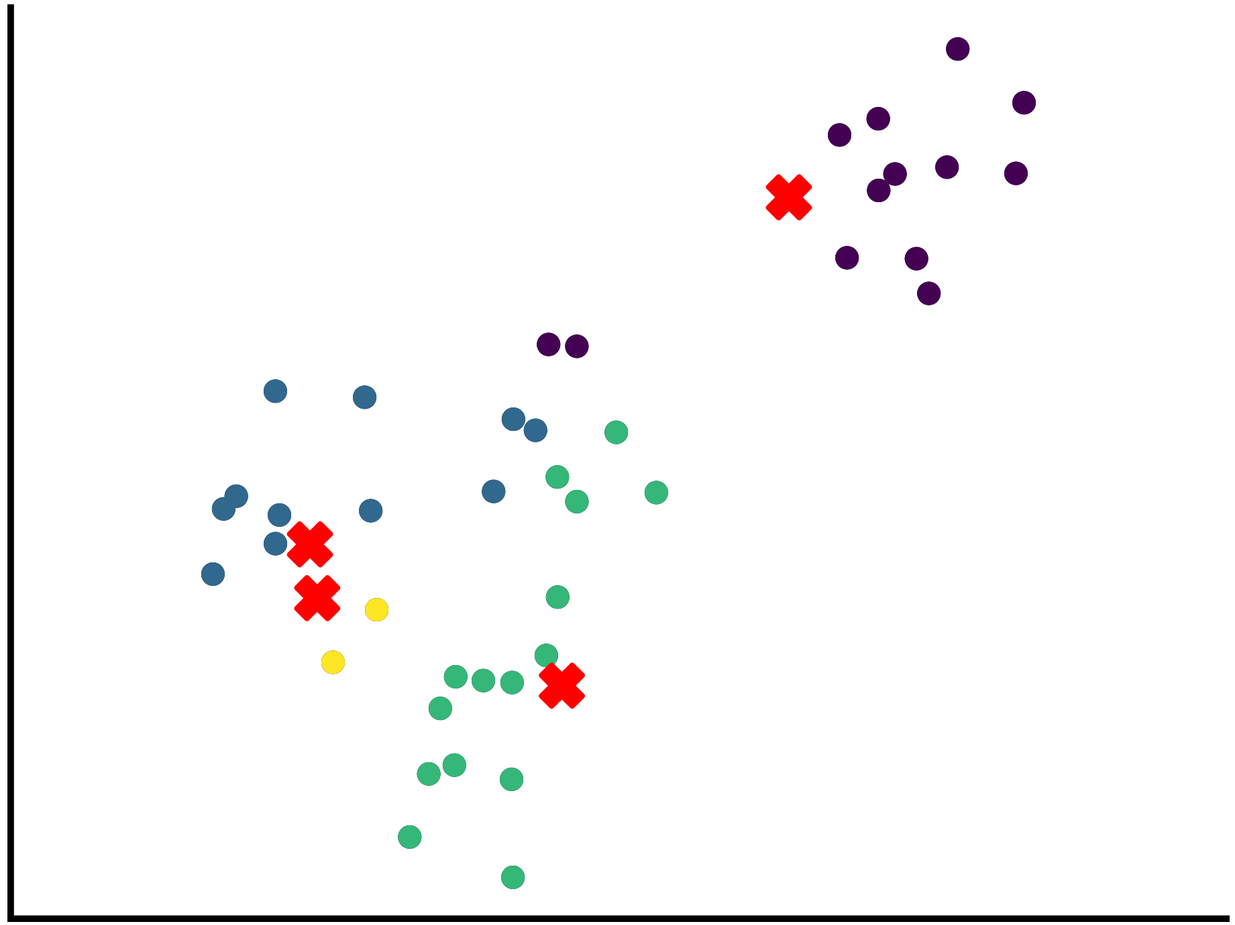
Points Assignment



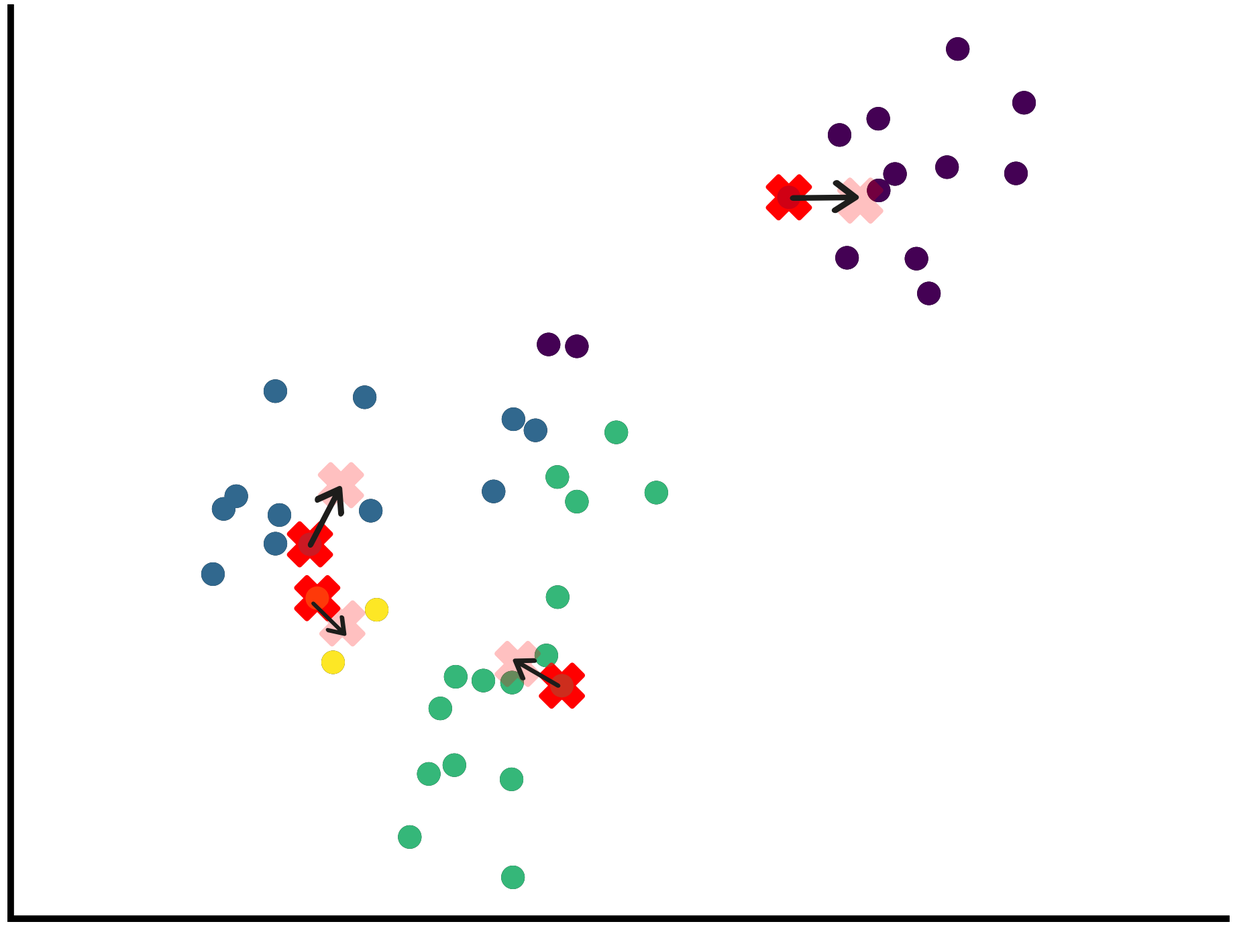
Points Assignment



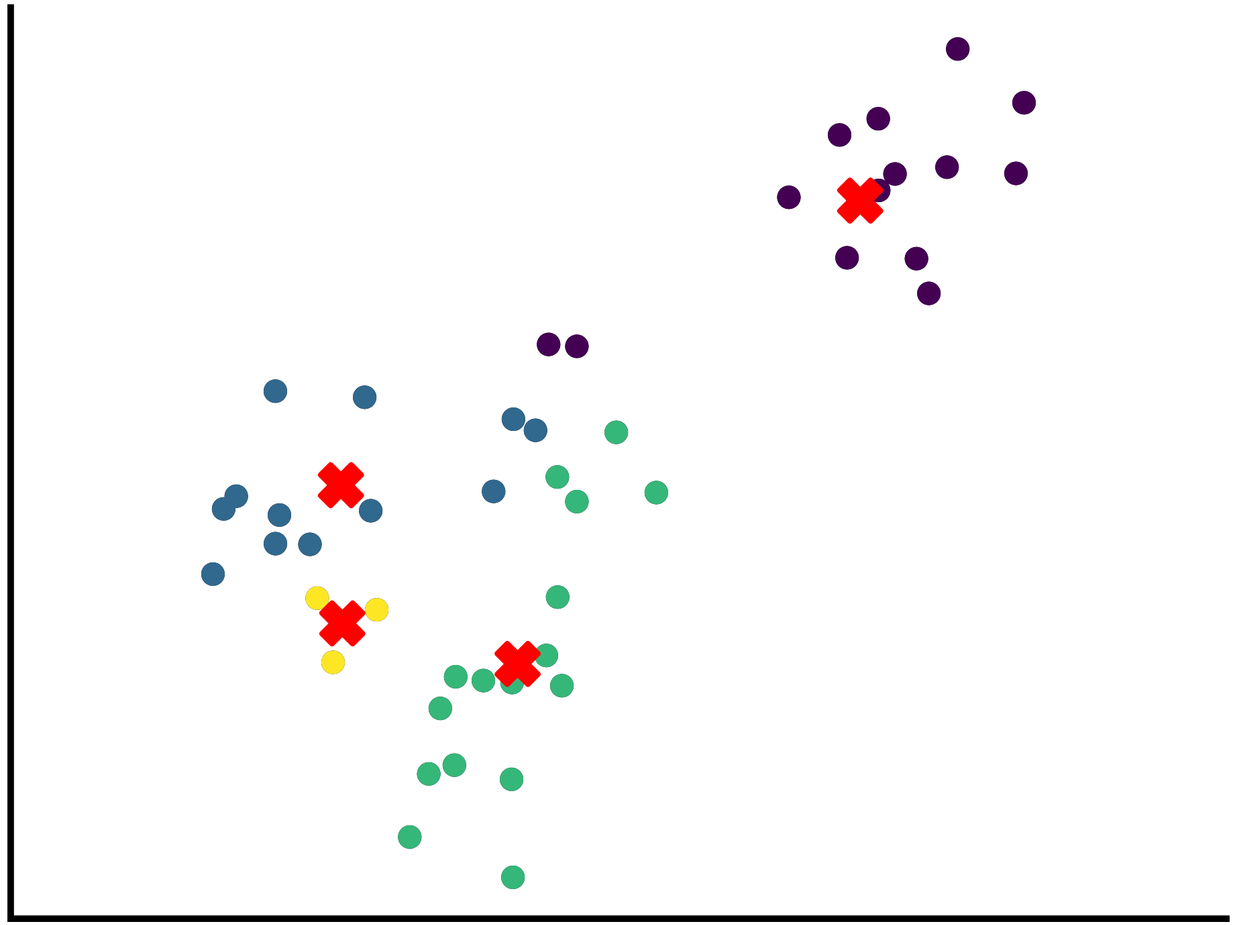
Points Assignment



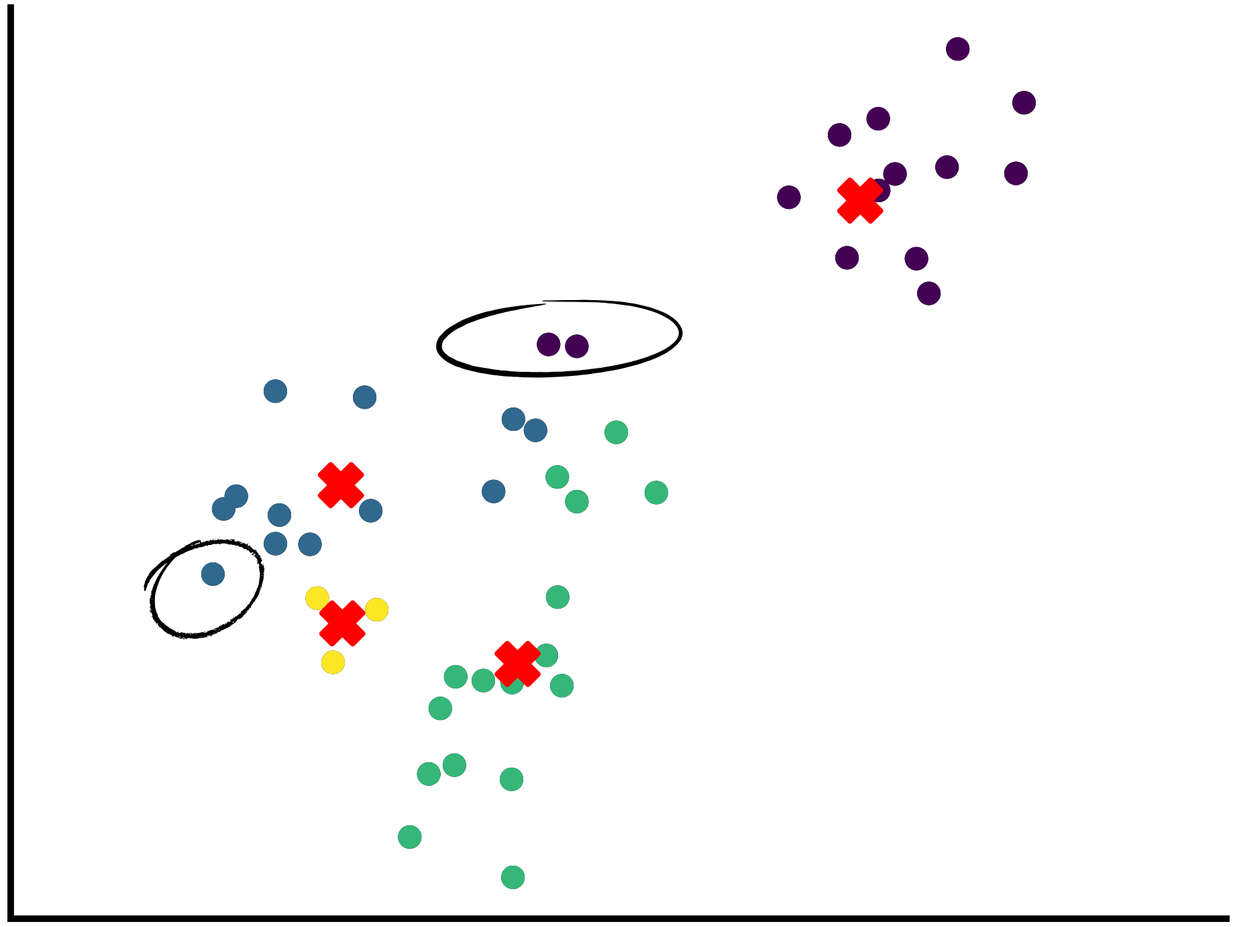
Centroids update



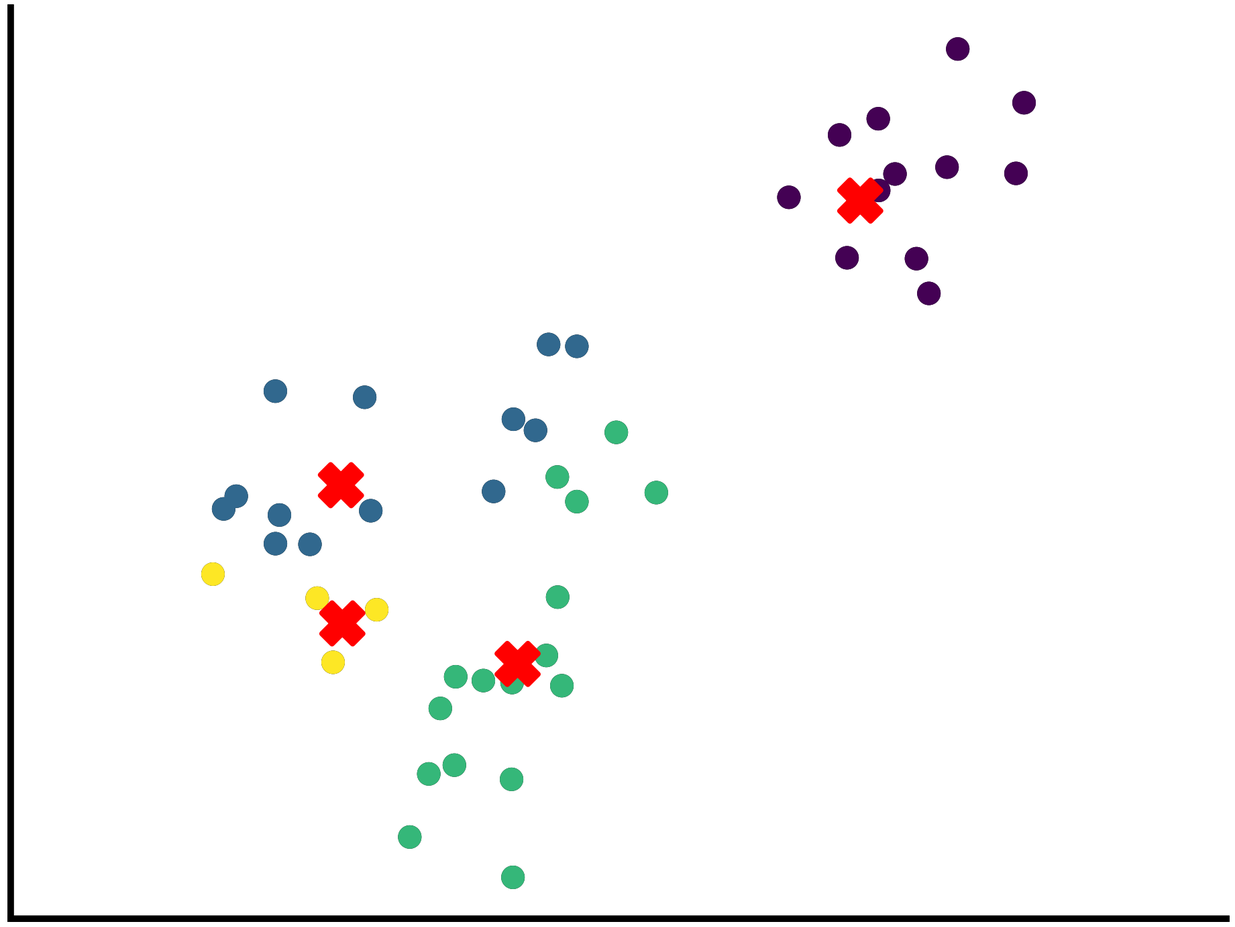
Centroids update



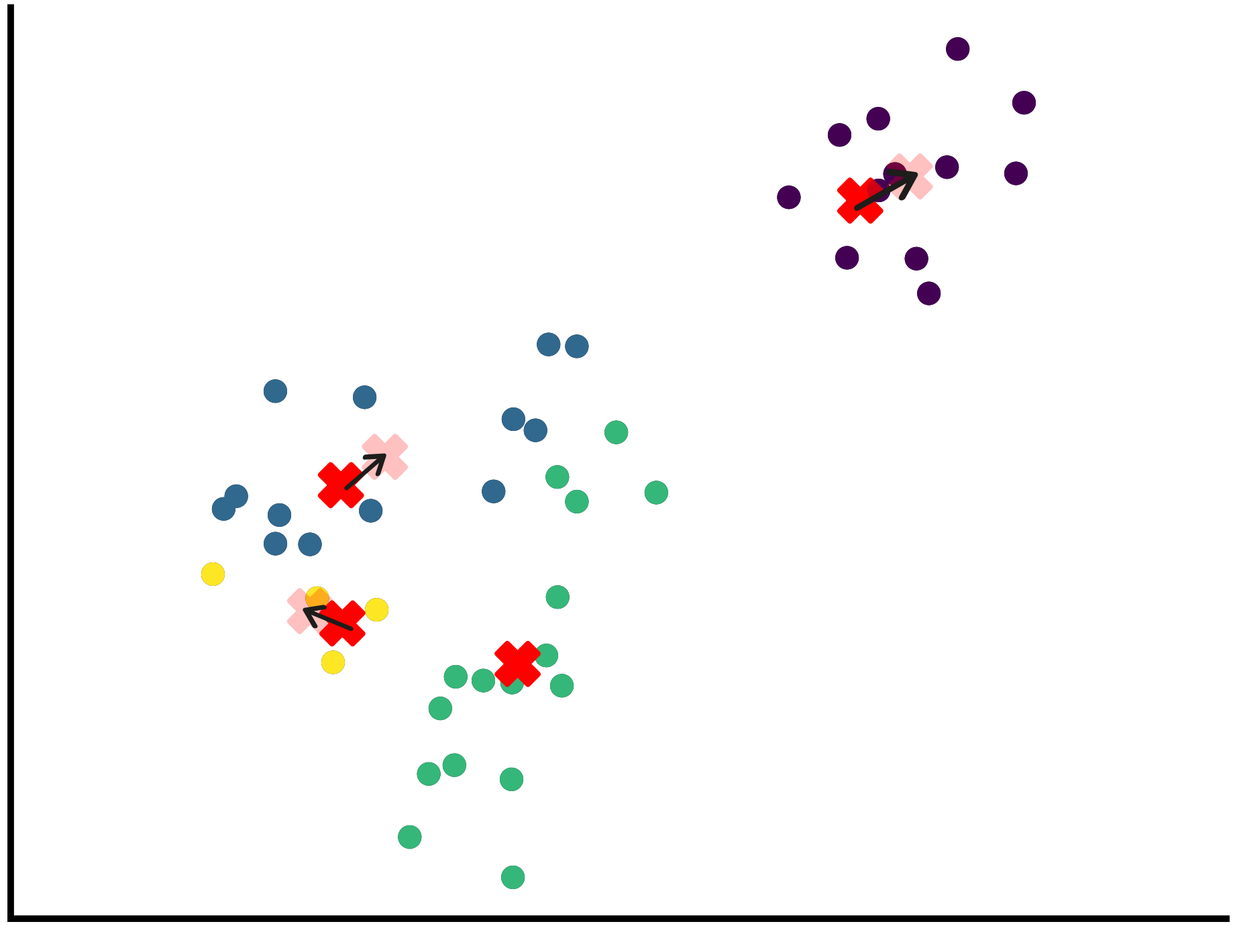
Points Assignment



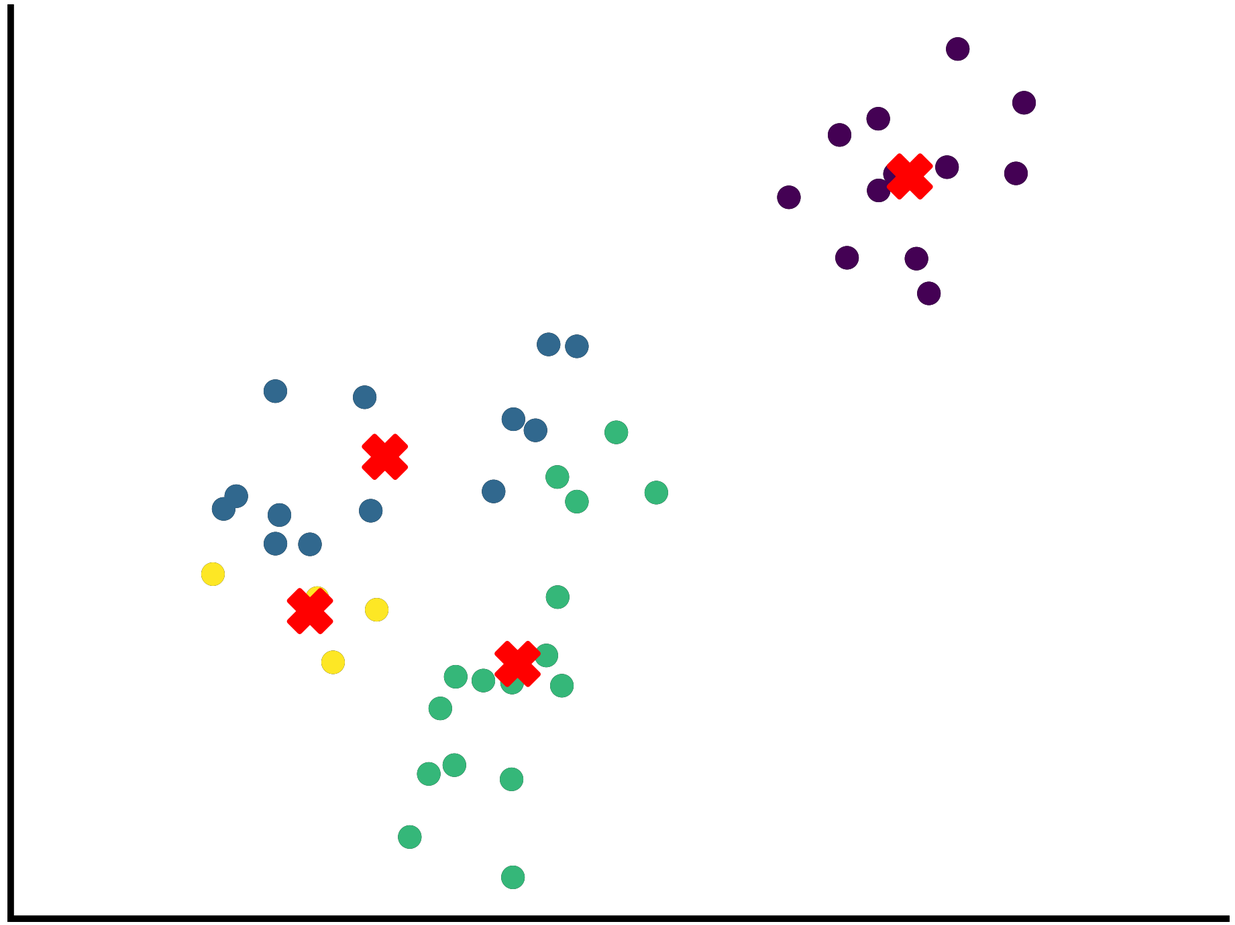
Points Assignment



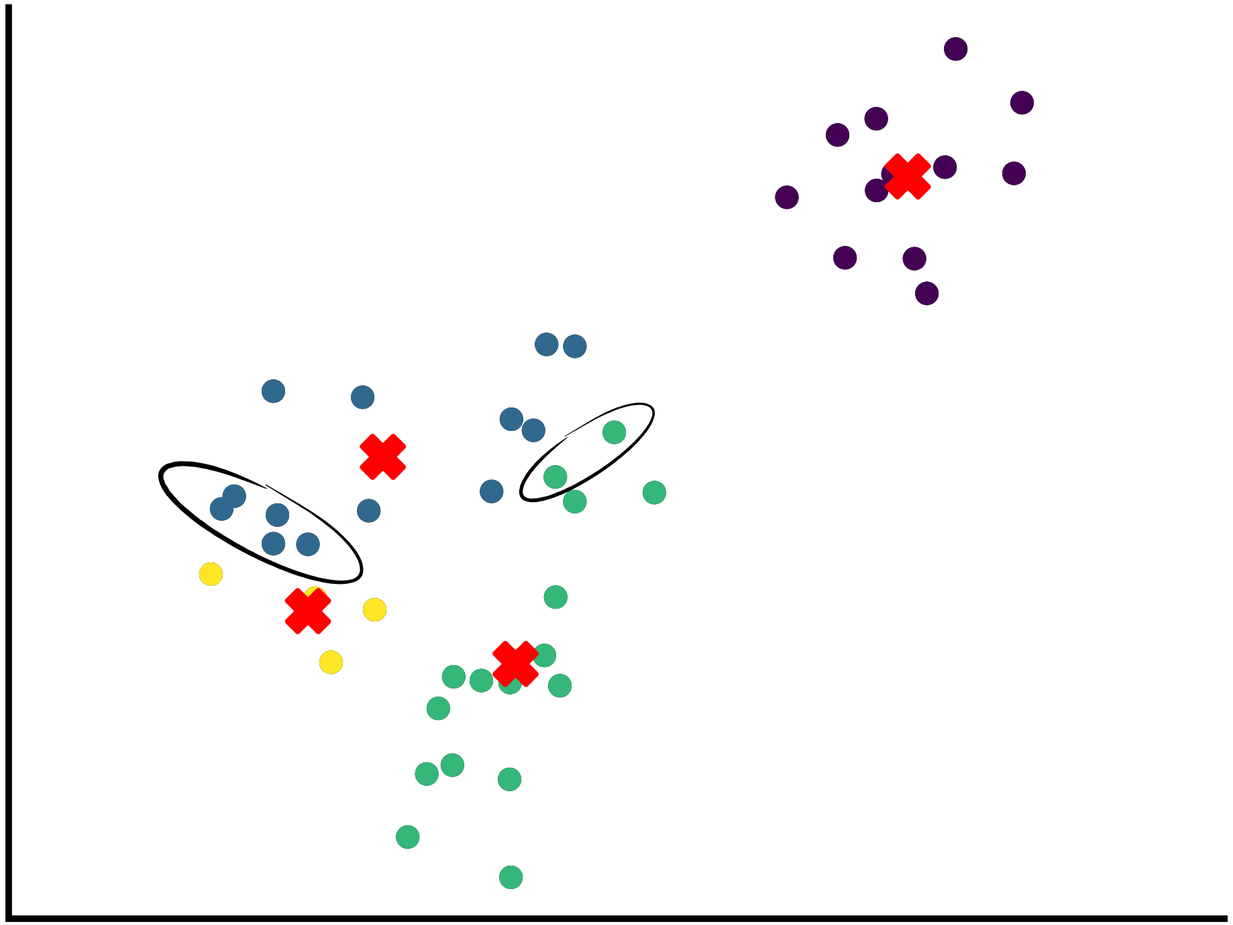
Centroids update



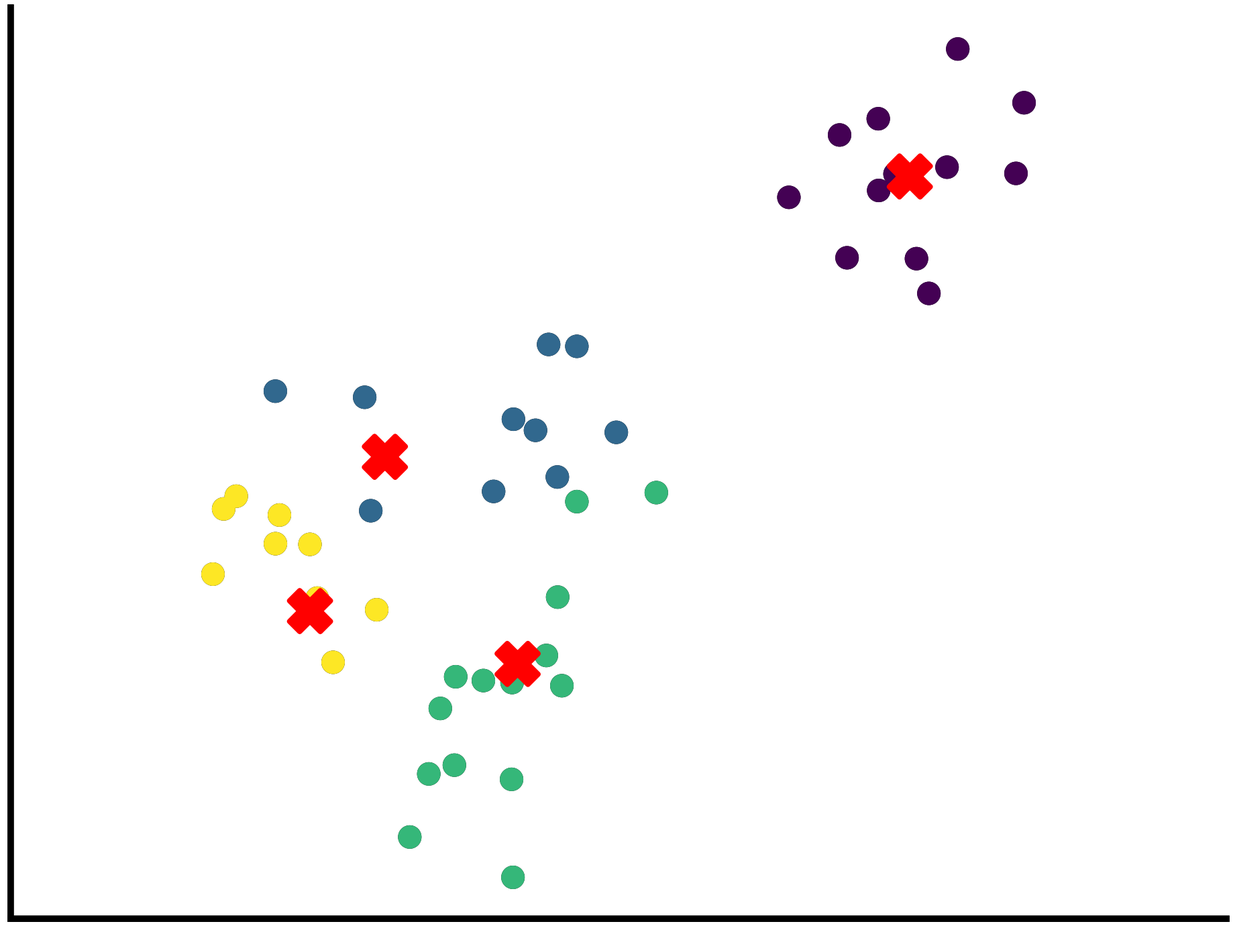
Centroids update



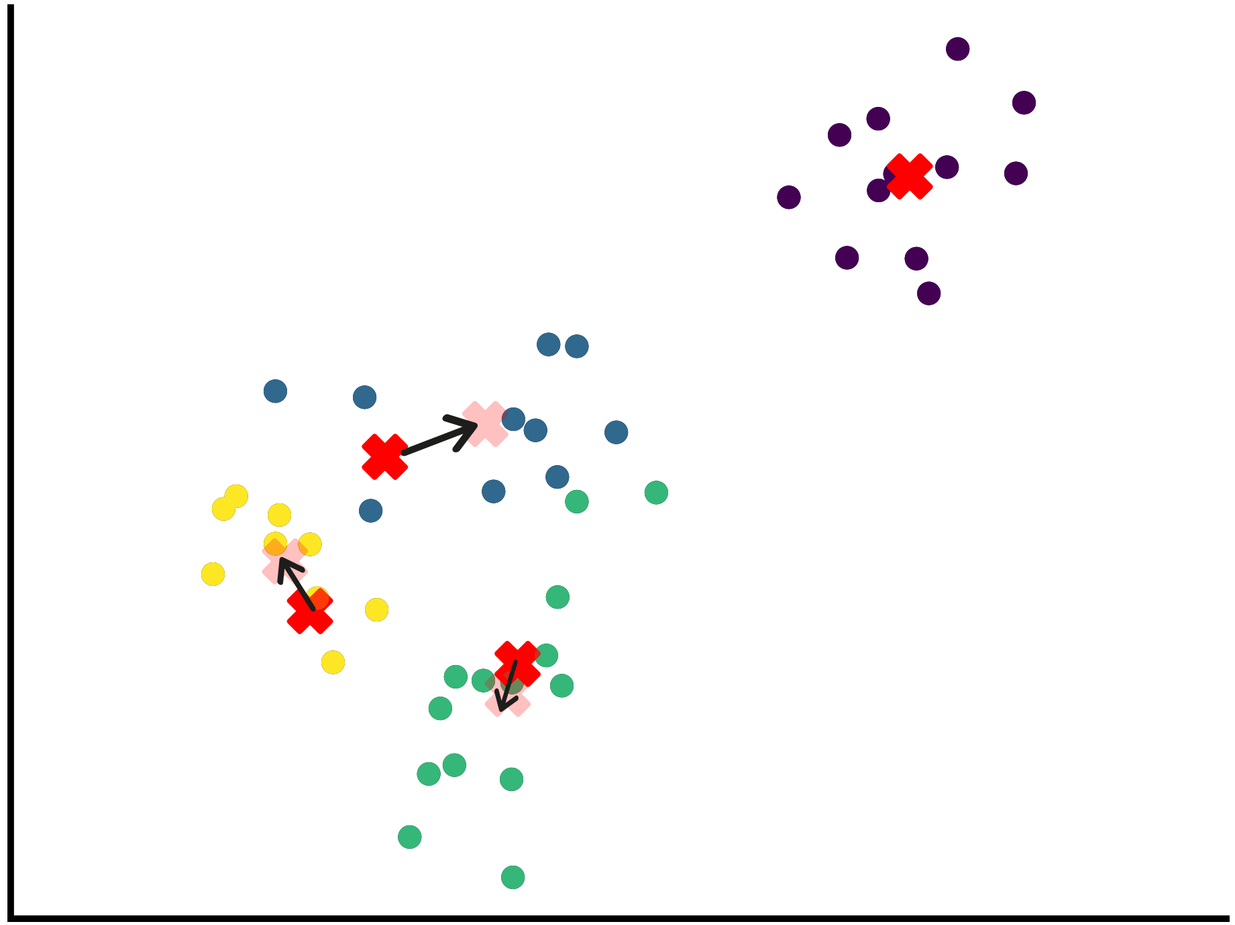
Points Assignment



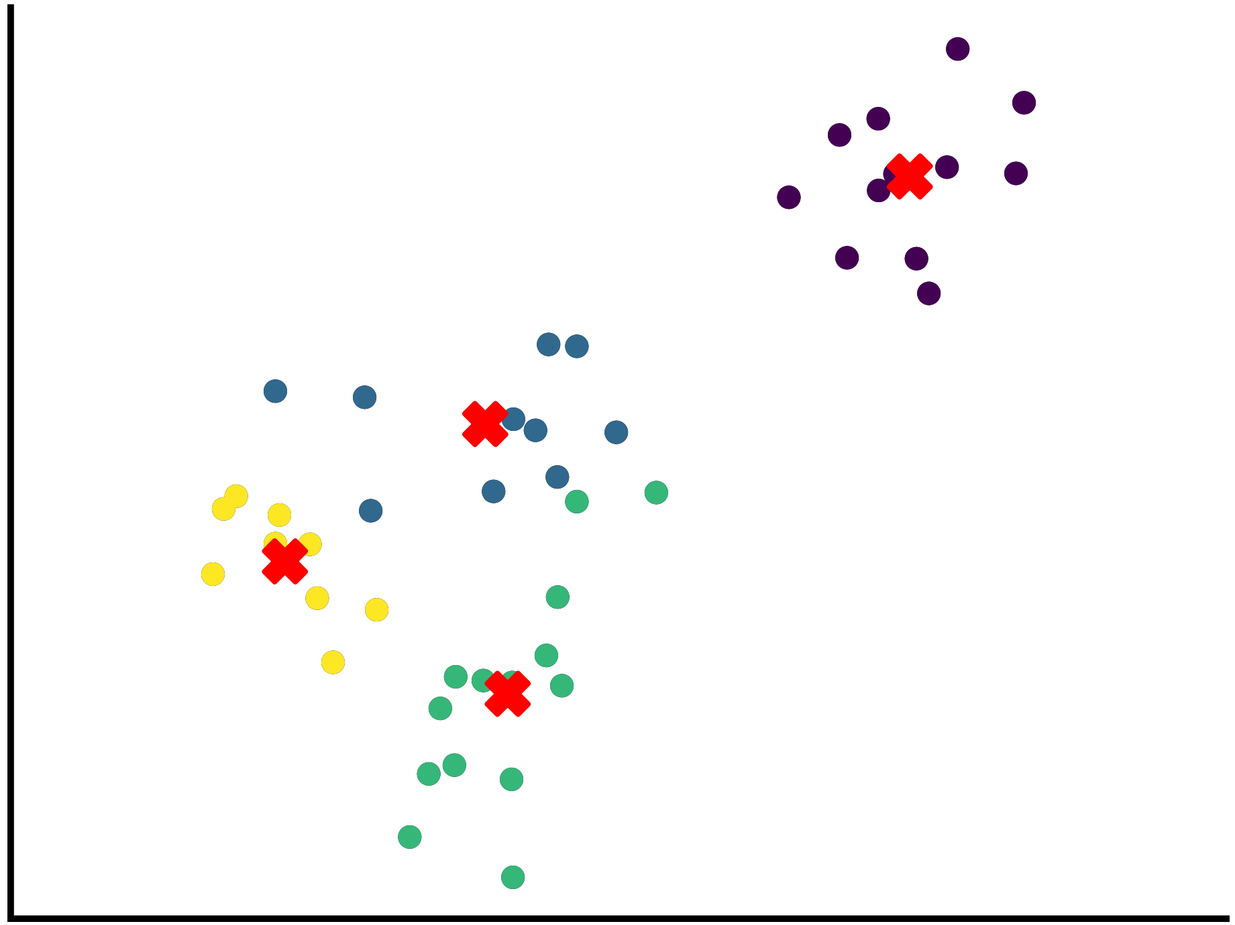
Points Assignment



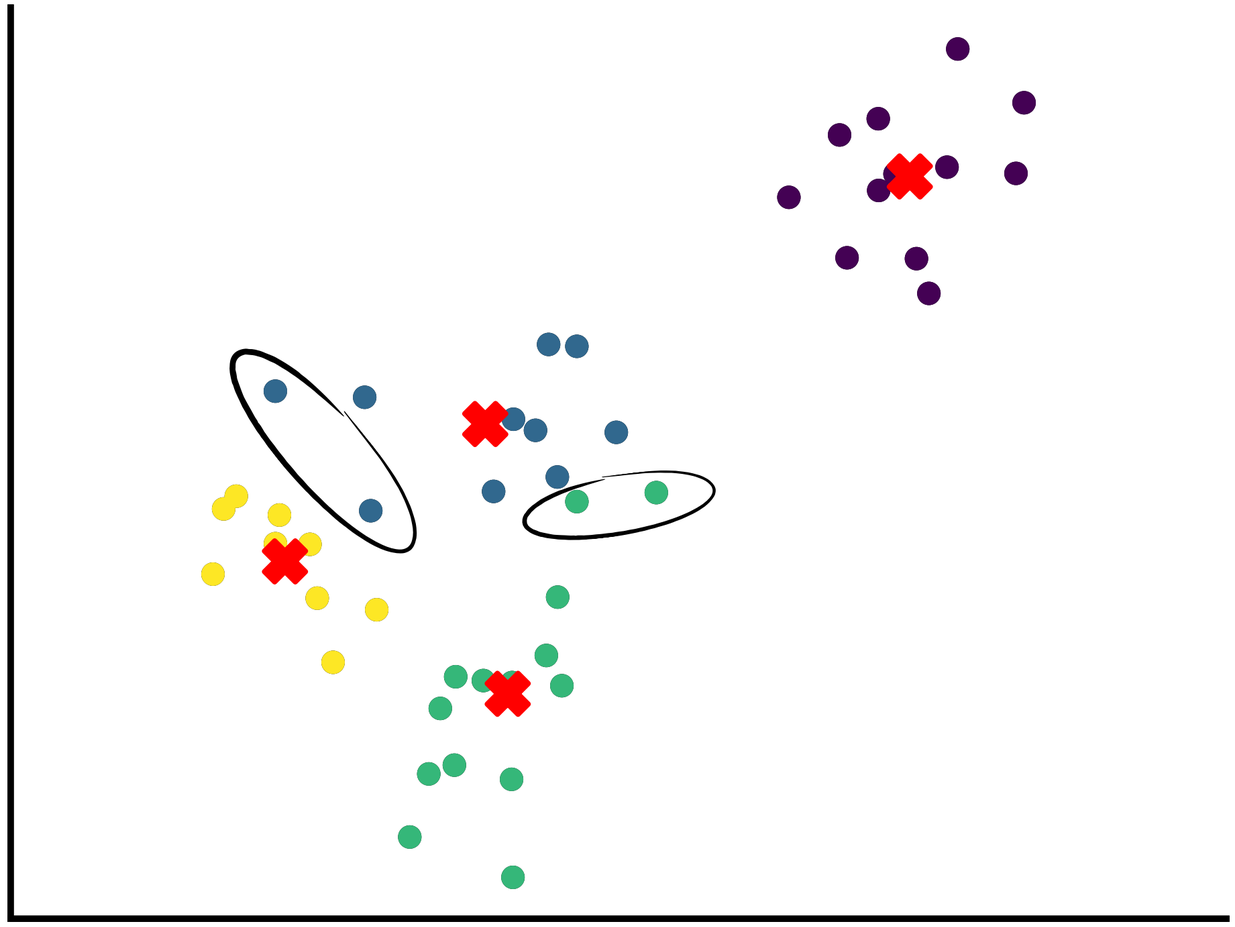
Centroids update



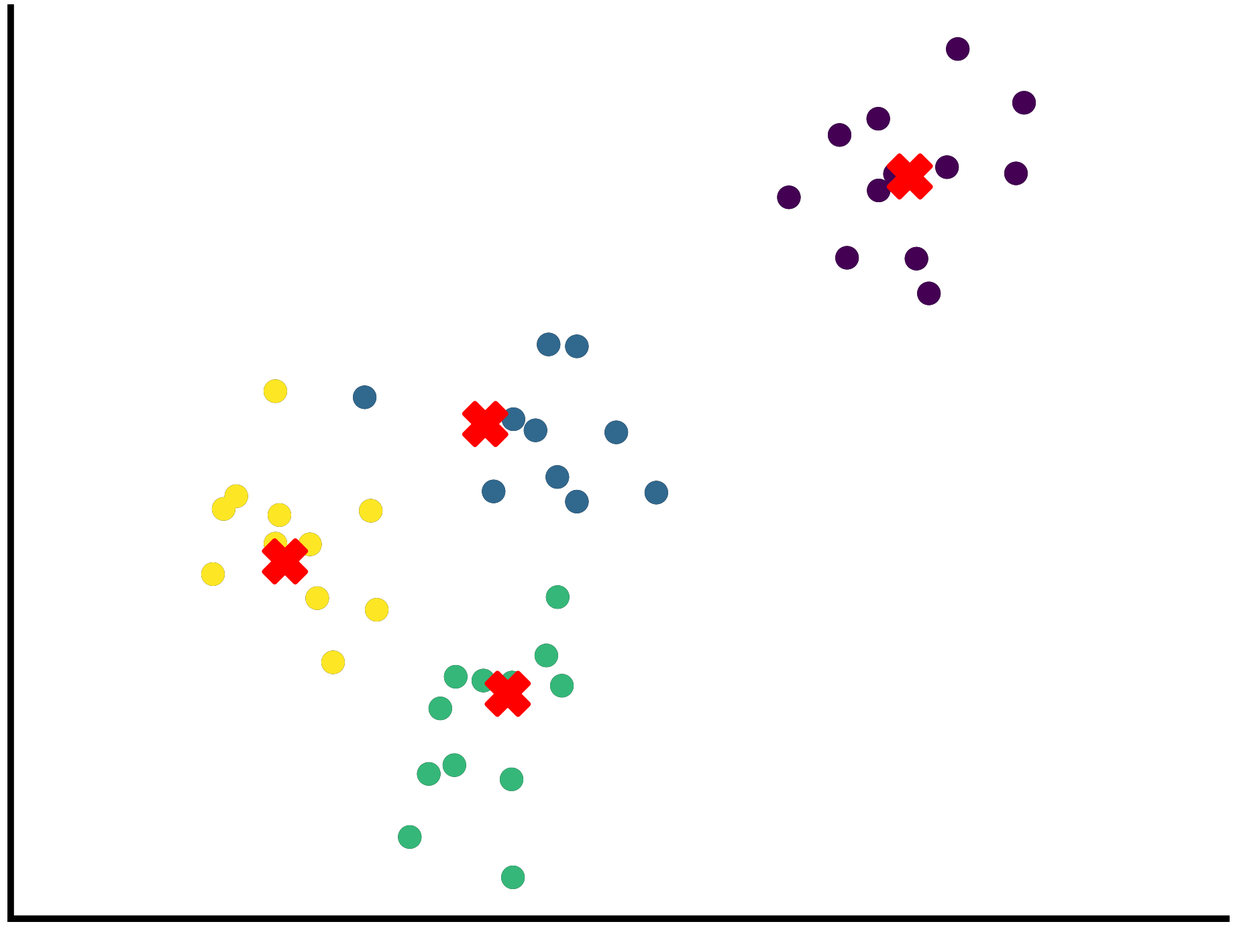
Centroids update



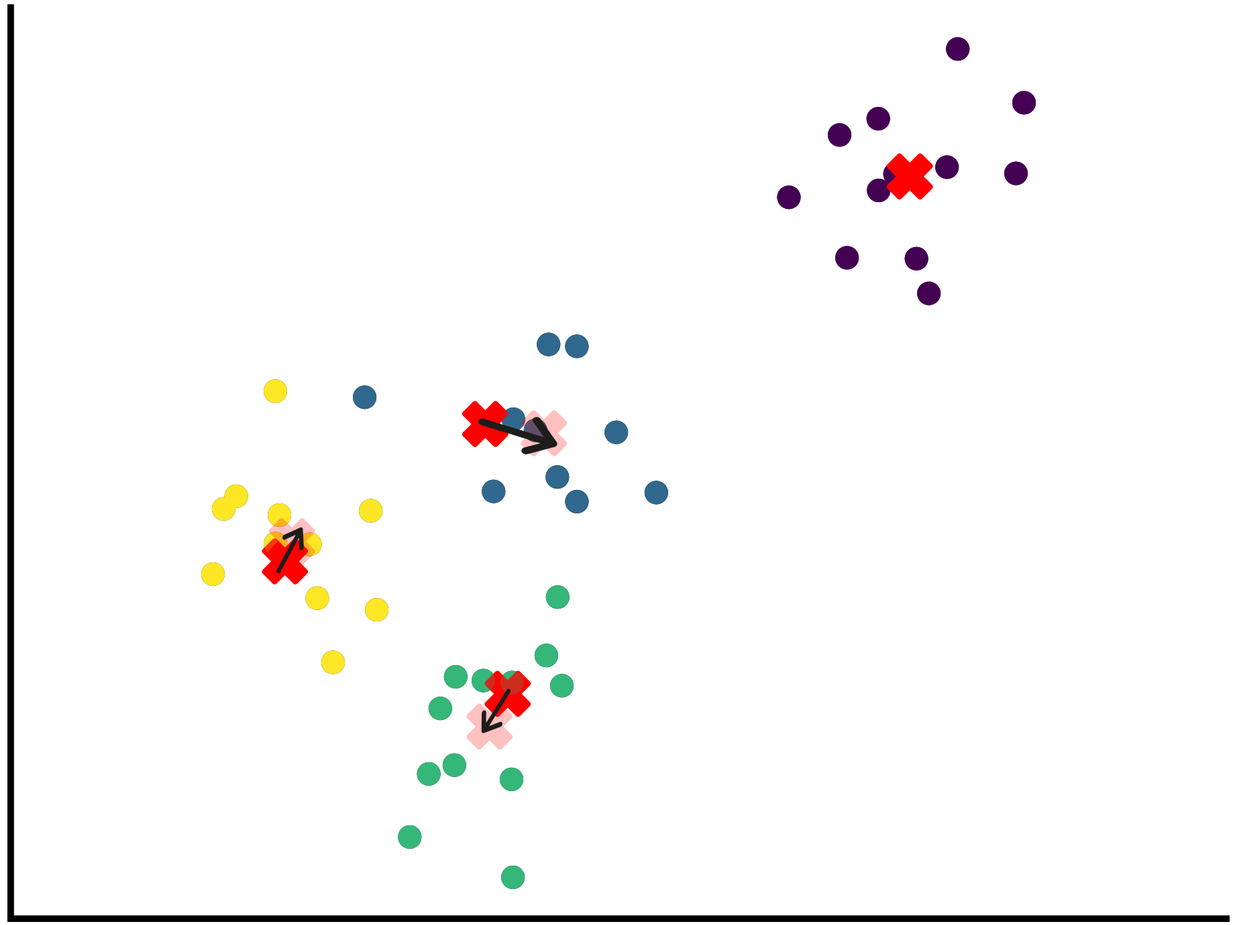
Points Assignment



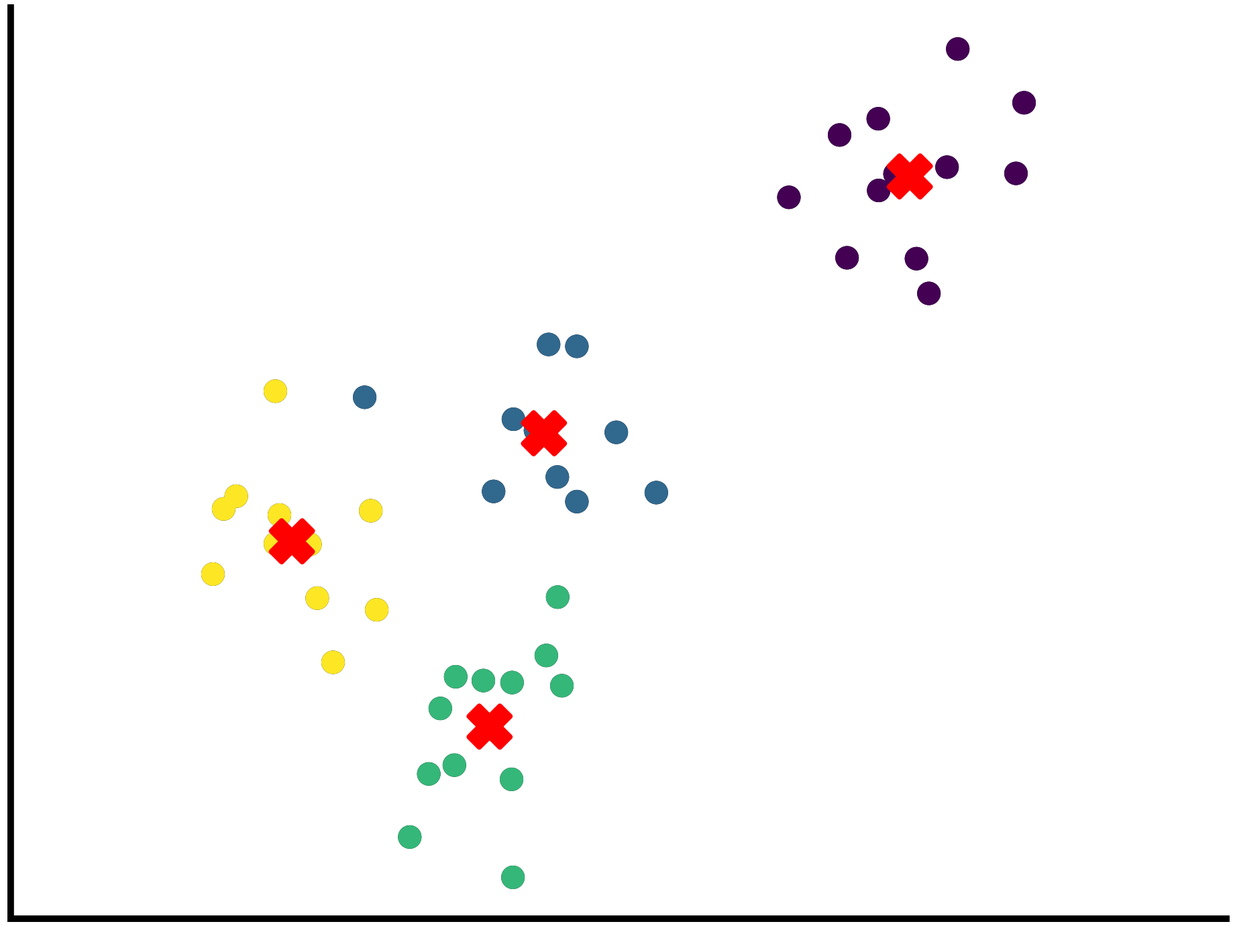
Points Assignment



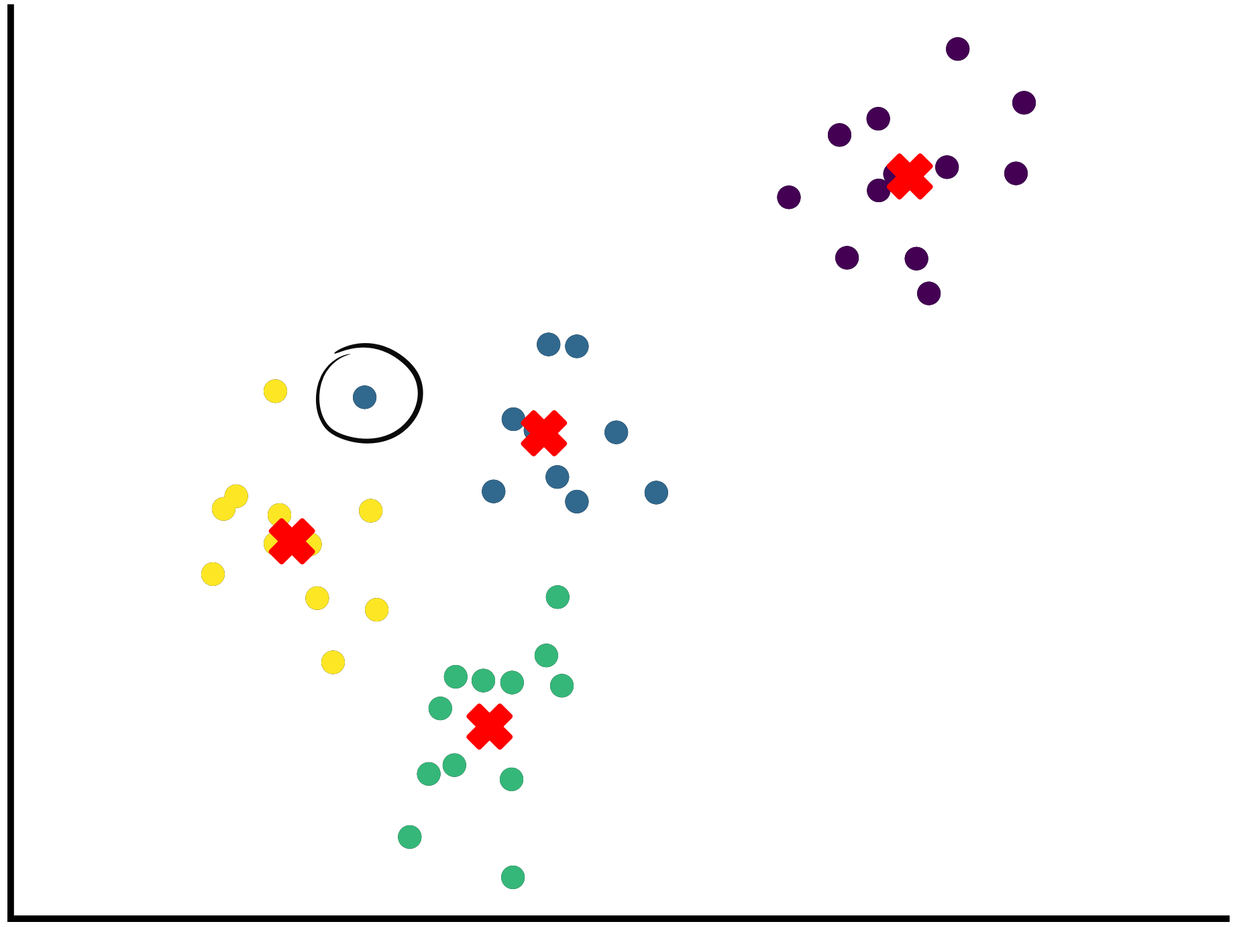
Centroids update



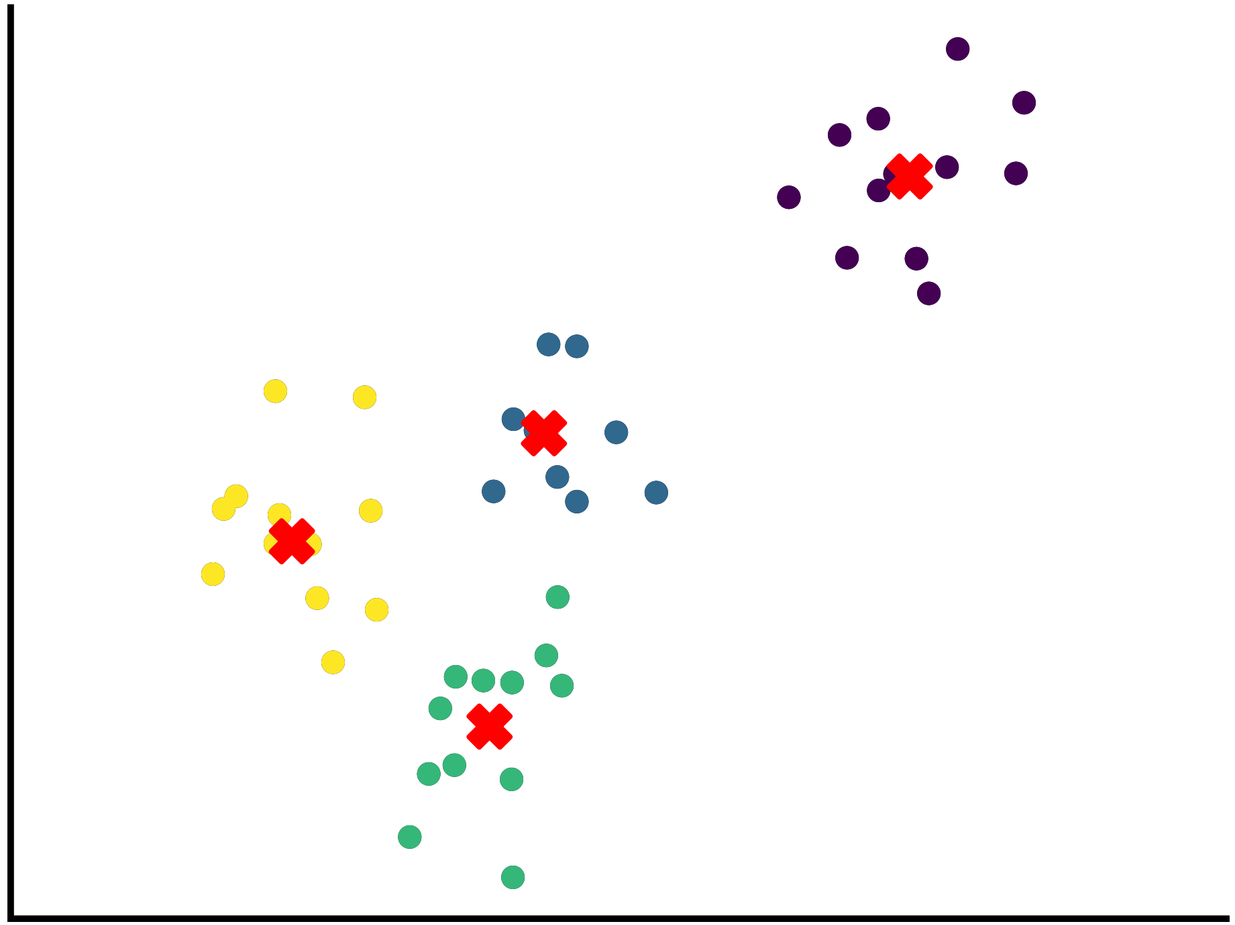
Centroids update



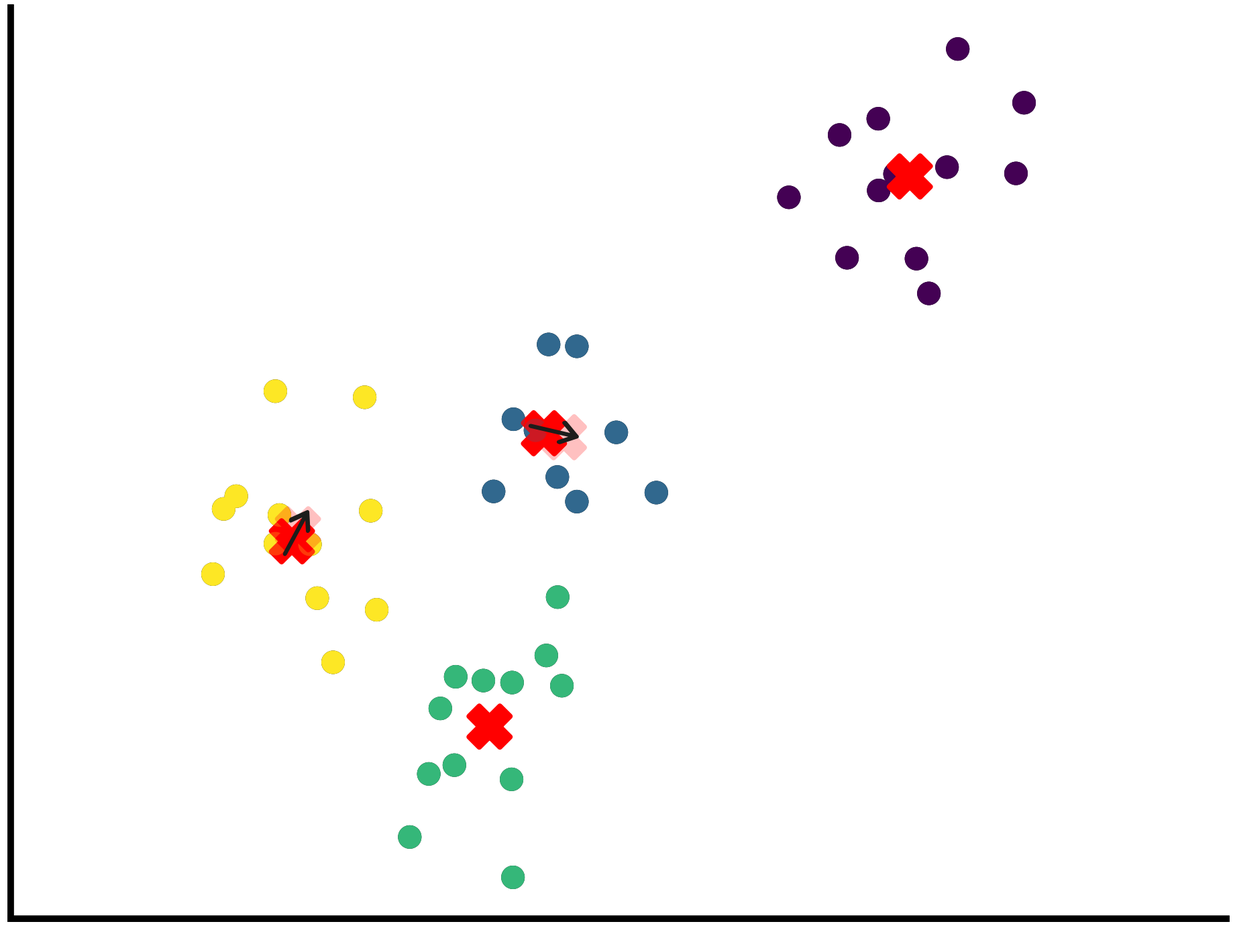
Points Assignment



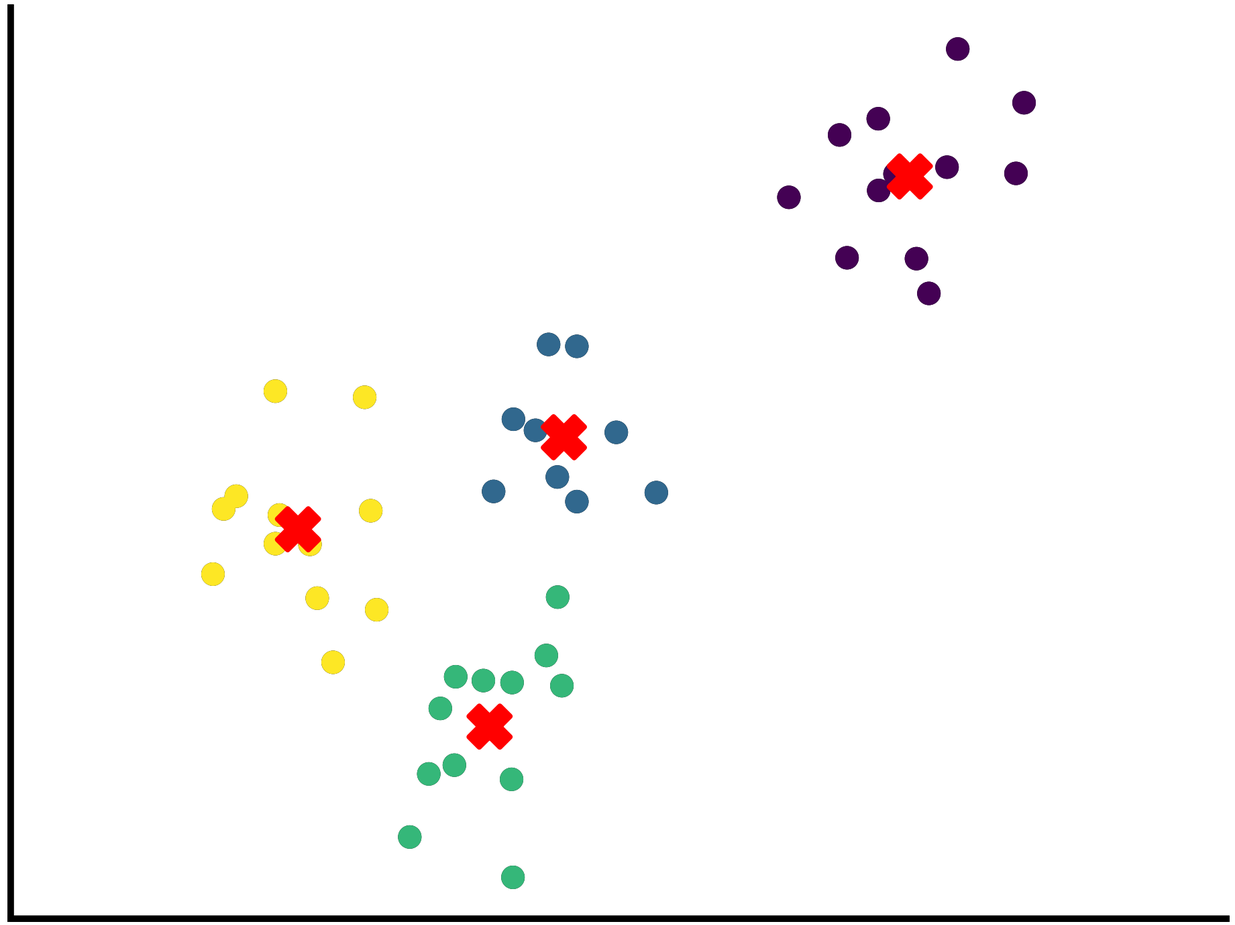
Points Assignment



Points Assignment



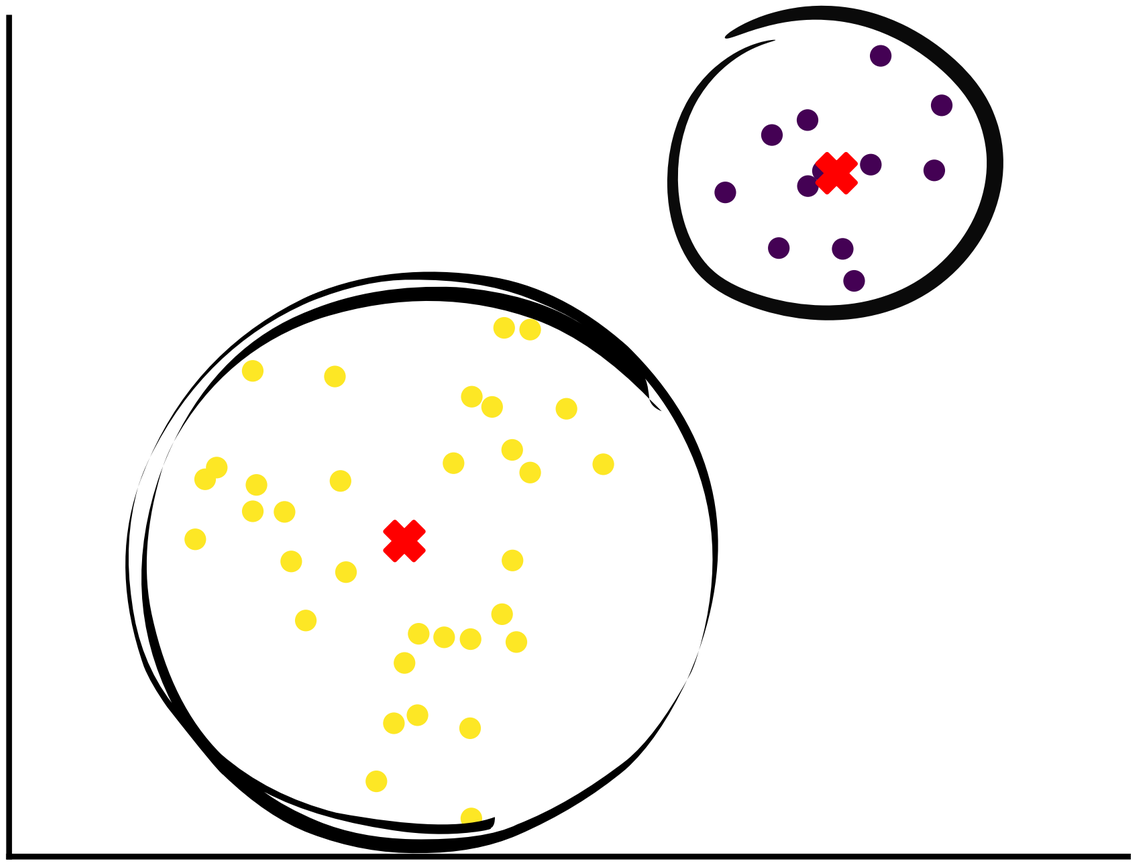
Convergence 🎉



How to choose K (the number of clusters)

- We can define some heuristics, but there is NO one true value of K.
- K-means seeks to minimize the WCSS or inertia: the elbow method compares WCSS values for different values of K.

Elbow method heuristic



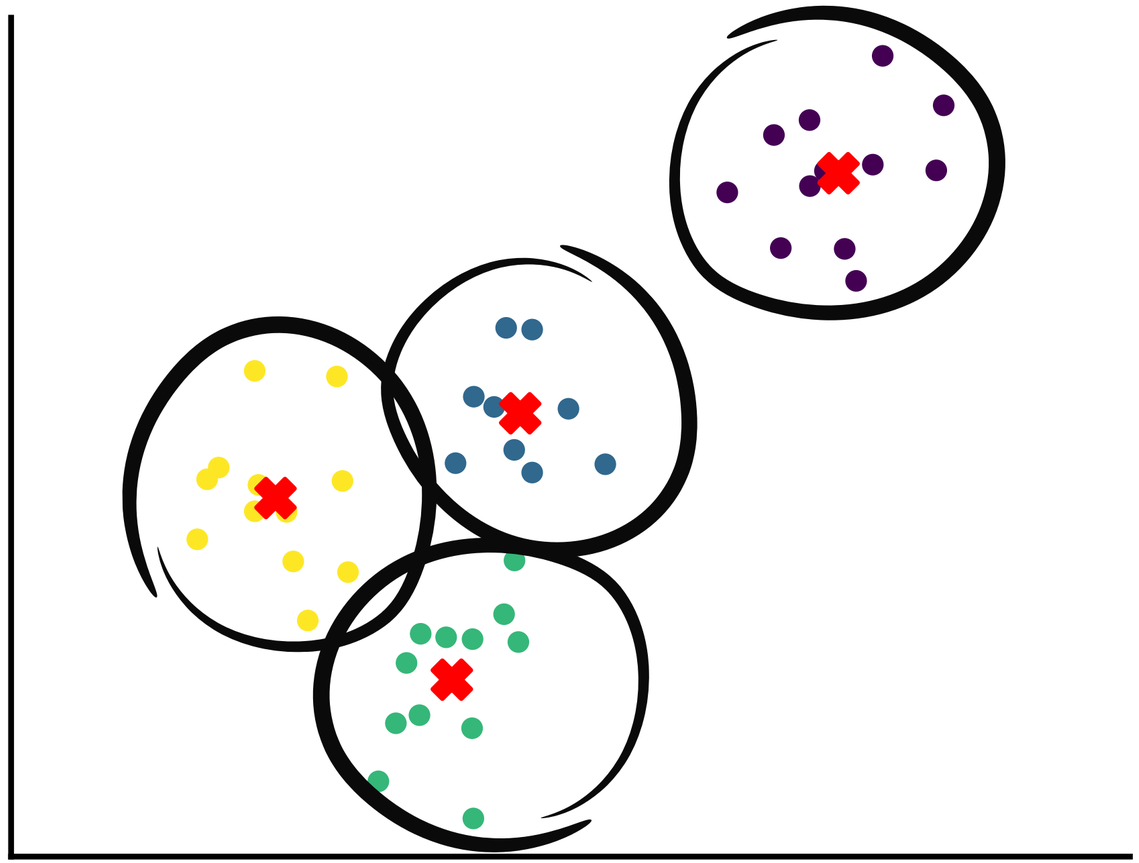
WCSS(K=2)

= D^2(✕ , ●)

+ D^2(✕ , ●)

.skolar

Elbow method heuristic



WCSS(K=4)

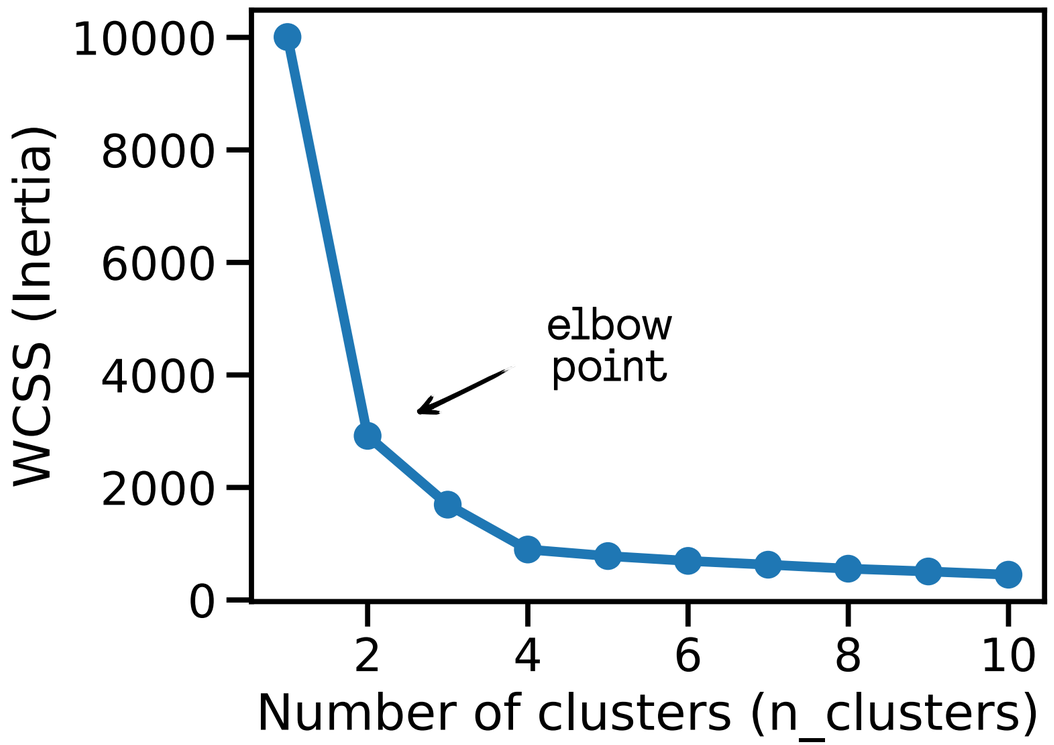
= D^2(✕ , ●)

+ D^2(✕ , ●)

+ D^2(✕ , ●)

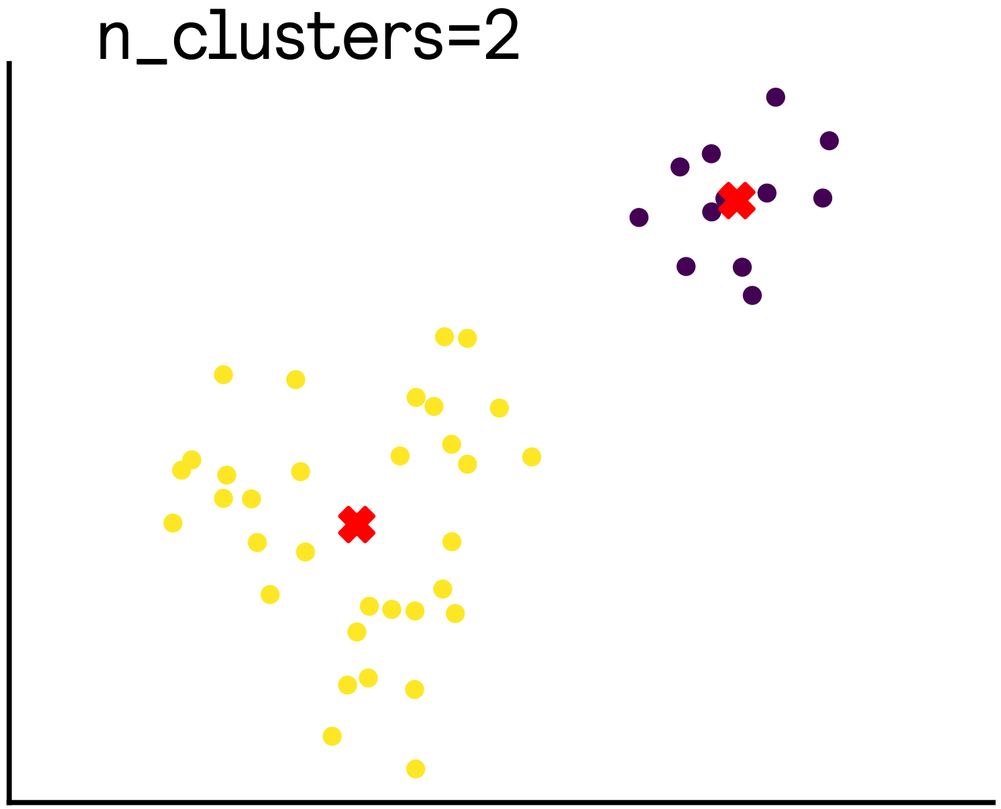
+ D^2(✕ , ●)

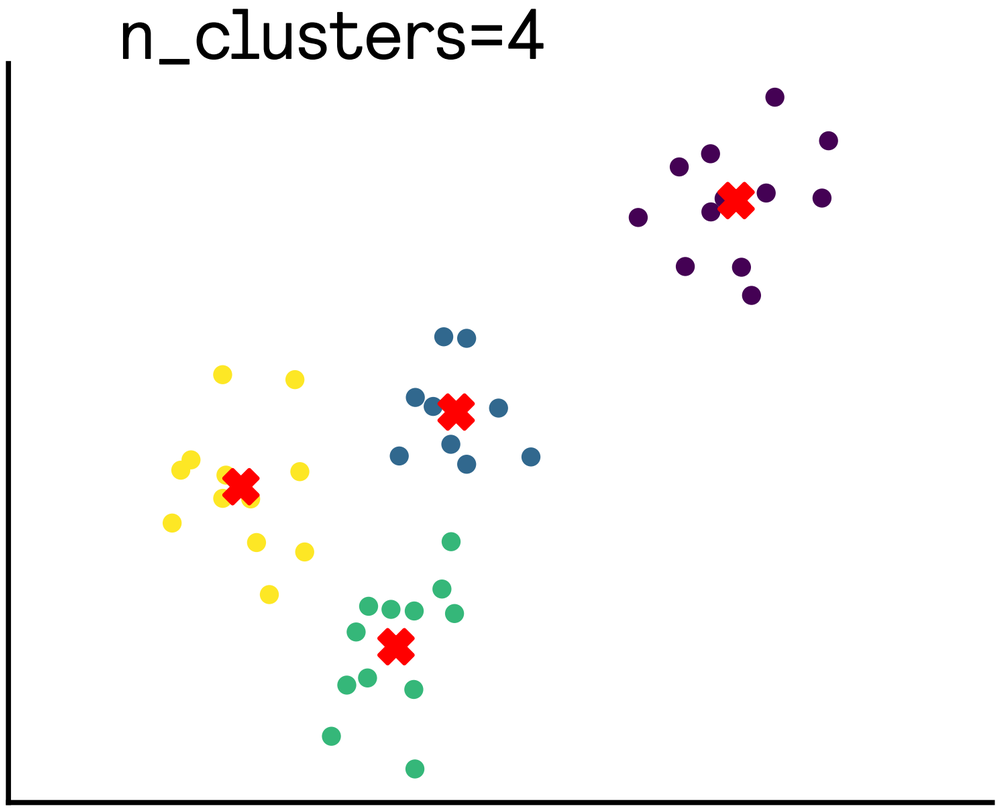
Elbow method heuristic



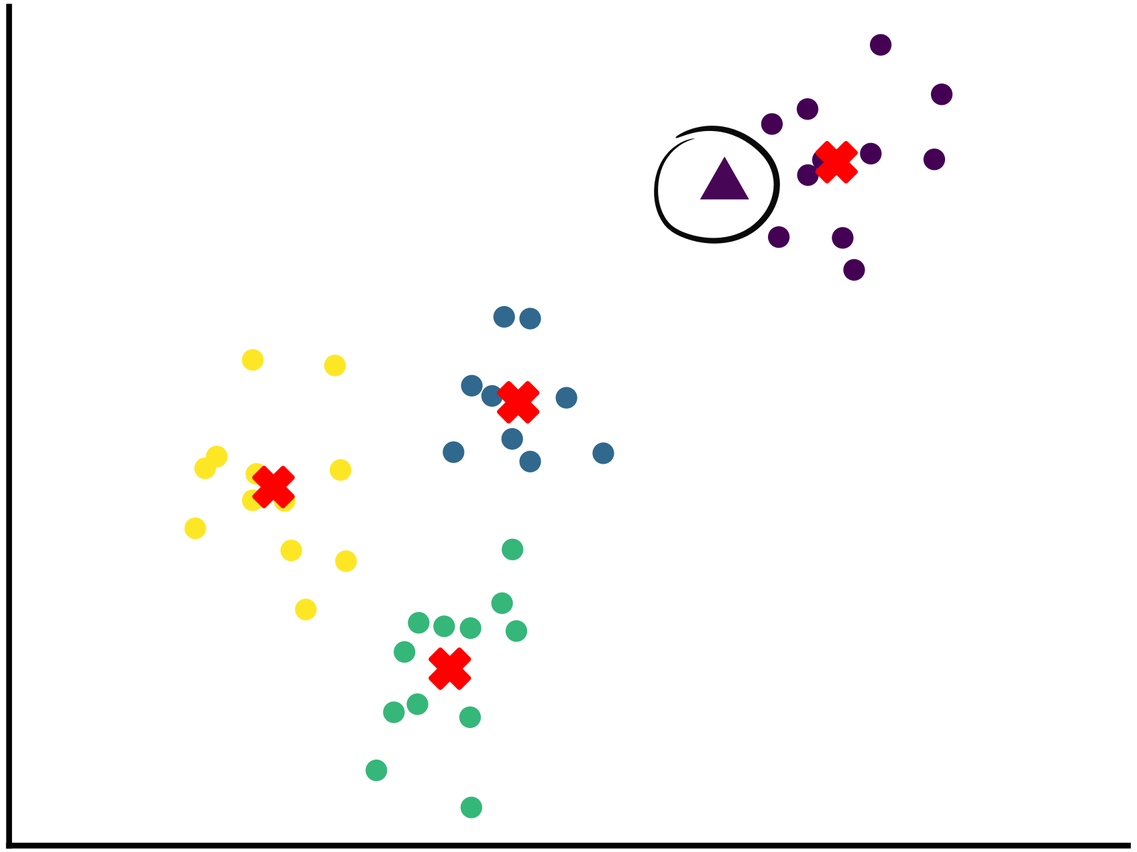
- The higher the number of clusters K, the lower the inertia
- The selected number of clusters is the smallest K for which inertia no longer decreases significantly, which corresponds to the elbow point.

The Silhouette score





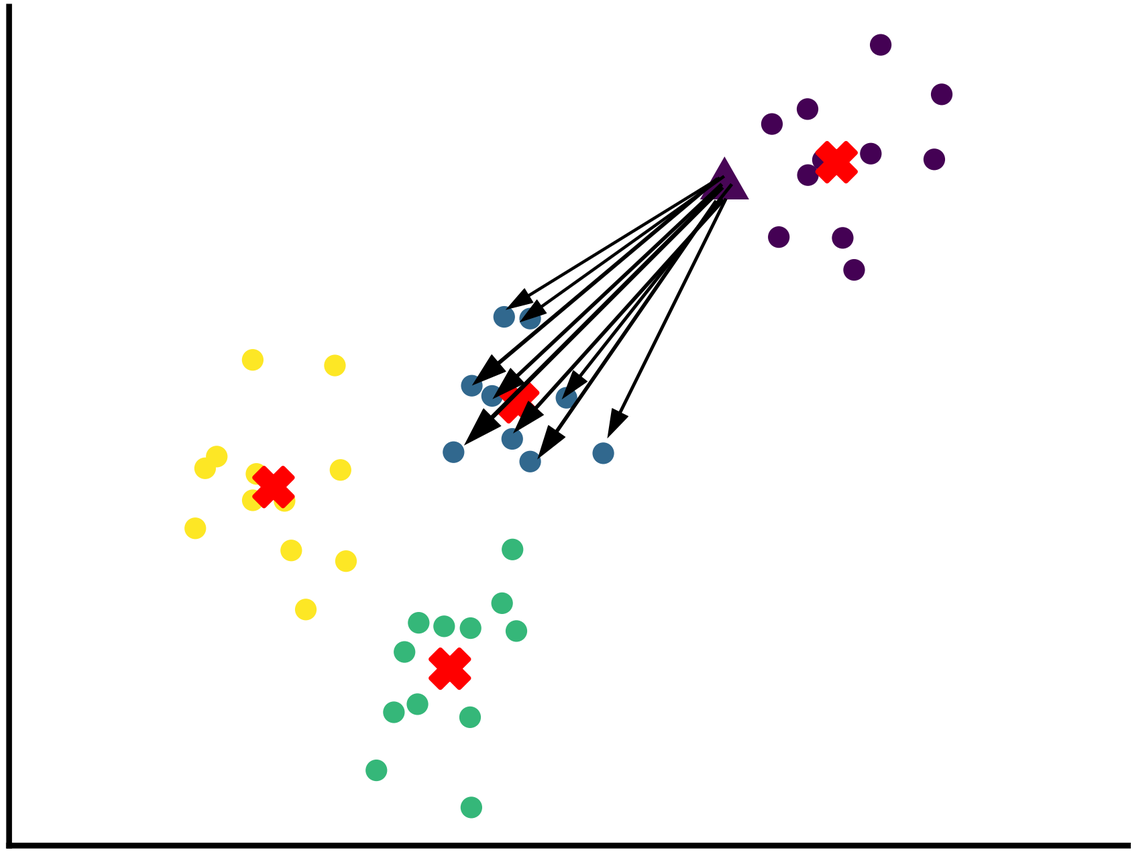
The Silhouette score



$$\text{silhouette}(\blacktriangle) = \frac{\overline{D(\blacktriangle, \bullet)} - \overline{D(\blacktriangle, \bullet)}}{\max\left(\overline{D(\blacktriangle, \bullet)}, \overline{D(\blacktriangle, \bullet)}\right)}$$

Each data point receives its own silhouette score

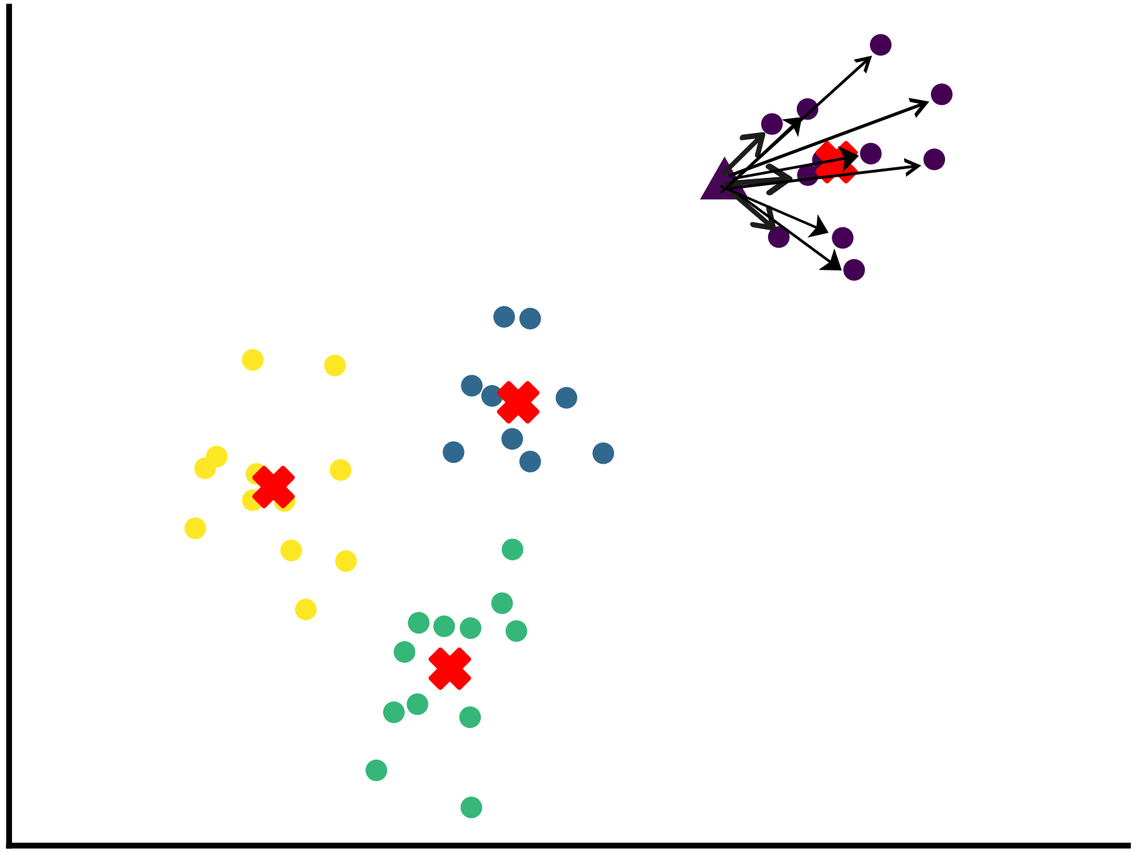
The Silhouette score



$$\text{silhouette}(\blacktriangle) = \frac{\overline{D(\blacktriangle, \bullet)} - \overline{D(\blacktriangle, \bullet)}}{\max(\overline{D(\blacktriangle, \bullet)}, \overline{D(\blacktriangle, \bullet)})}$$

- Mean distance to the closest cluster
- Cluster separation

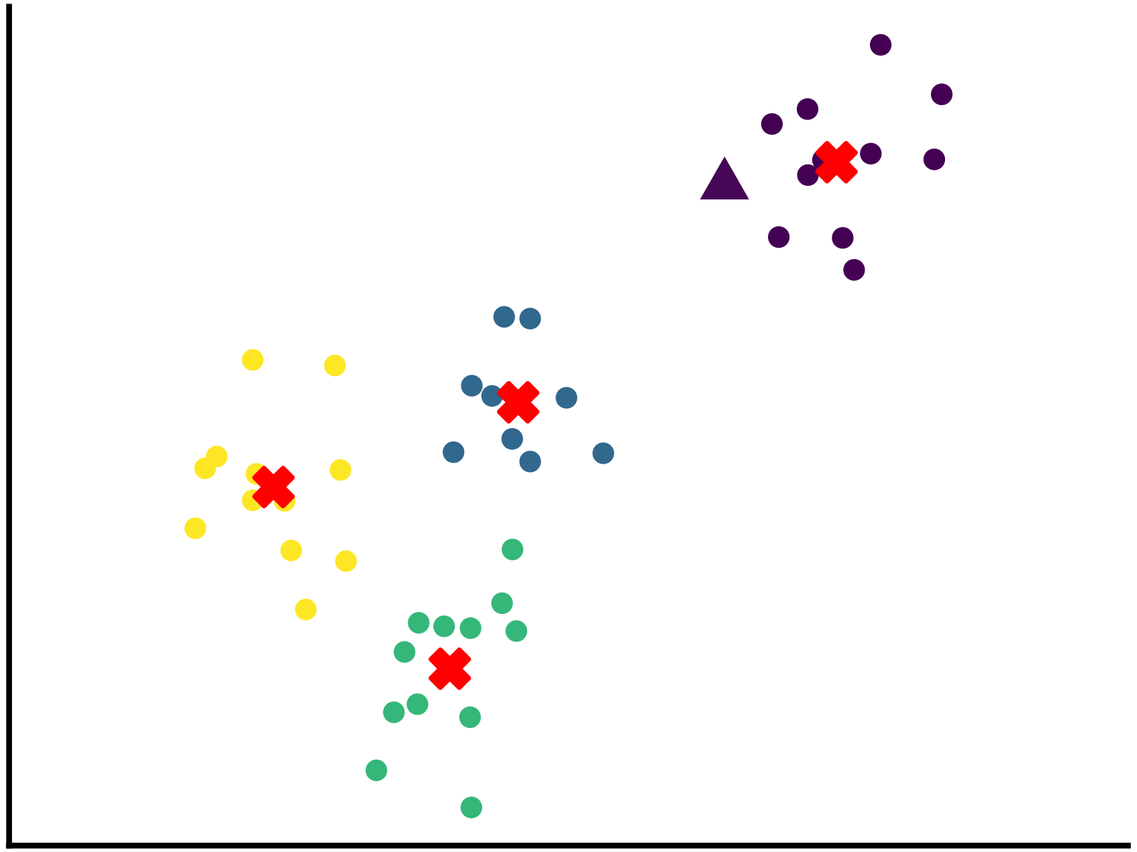
The Silhouette score



$$\text{silhouette}(\blacktriangle) = \frac{\overline{D(\blacktriangle, \bullet)} - \overline{D(\blacktriangle, \bullet)}}{\max(\overline{D(\blacktriangle, \bullet)}, \overline{D(\blacktriangle, \bullet)})}$$

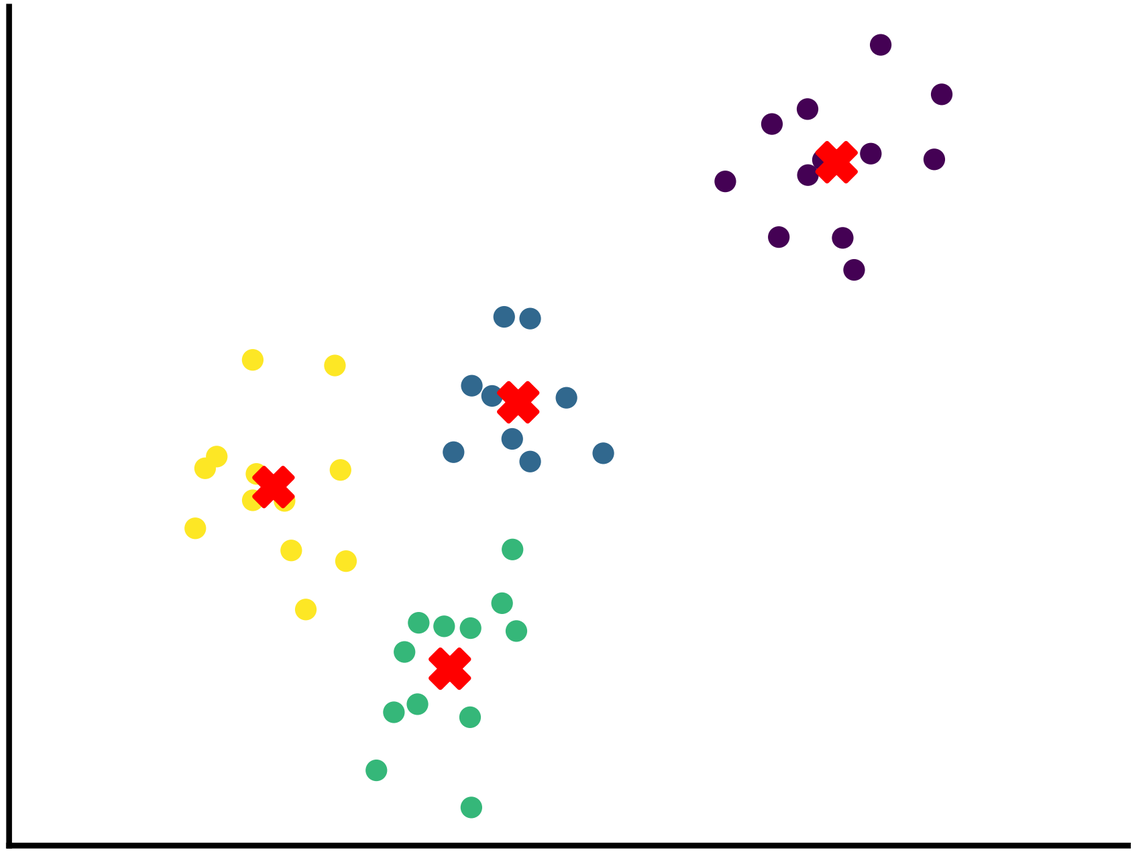
- Mean distance within it's cluster
- Cluster cohesion

The Silhouette score



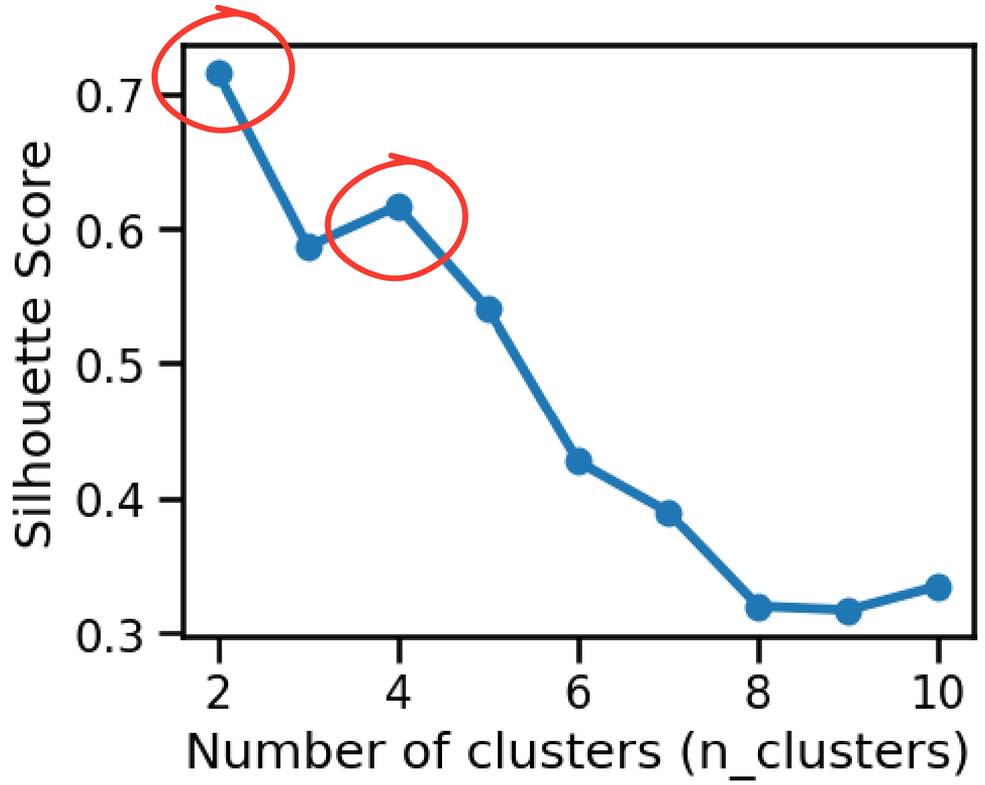
$$\text{silhouette}(\blacktriangle) = \frac{\overline{D(\blacktriangle, \bullet)} - \overline{D(\blacktriangle, \bullet)}}{\max\left(\overline{D(\blacktriangle, \bullet)}, \overline{D(\blacktriangle, \bullet)}\right)}$$

The Silhouette score



The global silhouette score (for K=4) is then the average silhouette over all the data points.

Silhouette as function of n_clusters



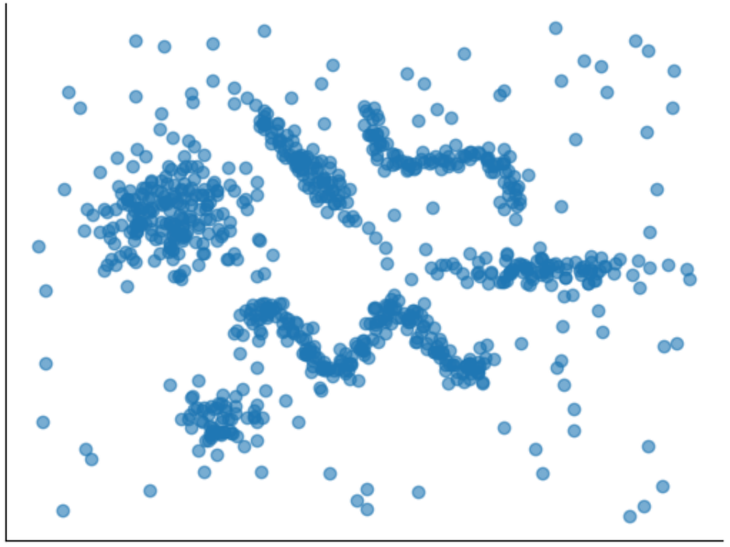
- Higher scores are better
- S ≈ 1 → Clusters are well-separated and distinct
- S ≈ 0 → Clusters are overlapping, points may lie between clusters
- S < 0 → Points may be in the wrong clusters

Disclaimer: Going beyond K-means

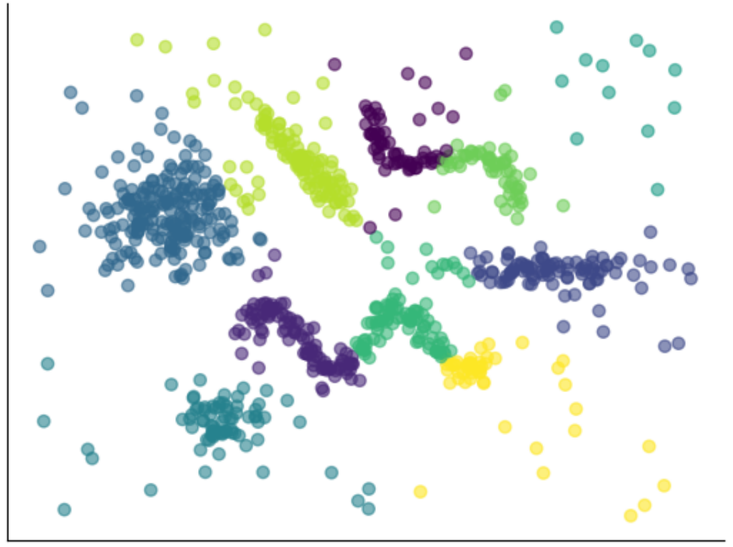
- K-Means favors roughly spherical cluster shapes
  - Feature preprocessing (using scalers or non-linear transformations) might help k-means.
  - But often clusters have no spherical shape, even after preprocessing
- In those cases, alternative to k-means can be tried:
  - Gaussian Mixture Model (favors elongated blobs)
  - HDBSCAN (does not favor any particular shape)
- Sometimes the data has no strong cluster structure at all!
  - K-means can still be useful as a preprocessing step.

Clusters with non-spherical shapes

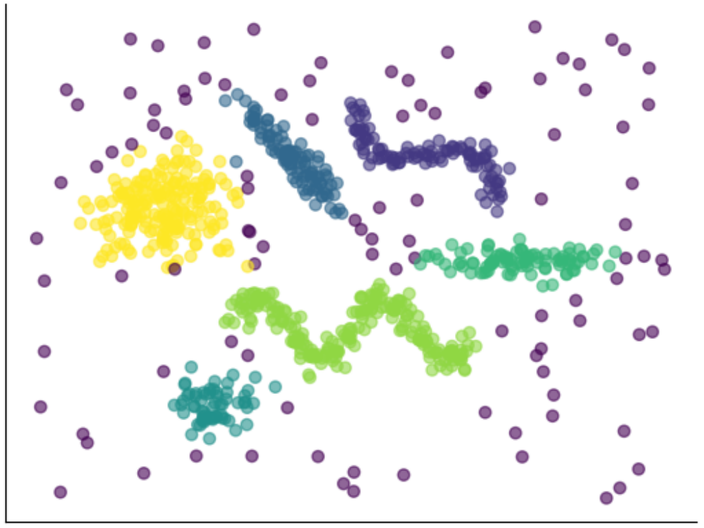
Unlabeled



K-means



HDBSCAN



Main takeaways

- Non-supervised:
  - A data matrix X with n observations but no target y.
  - The goal is to extract from X a structure that generalizes.
- K-means is a centroid-based algorithm, where feature scale matters.
  - We can use some heuristics such as the elbow and silhouette methods, but several values of K can make sense.
  - Sometimes k-means fails to find clusters with the shape it favors.