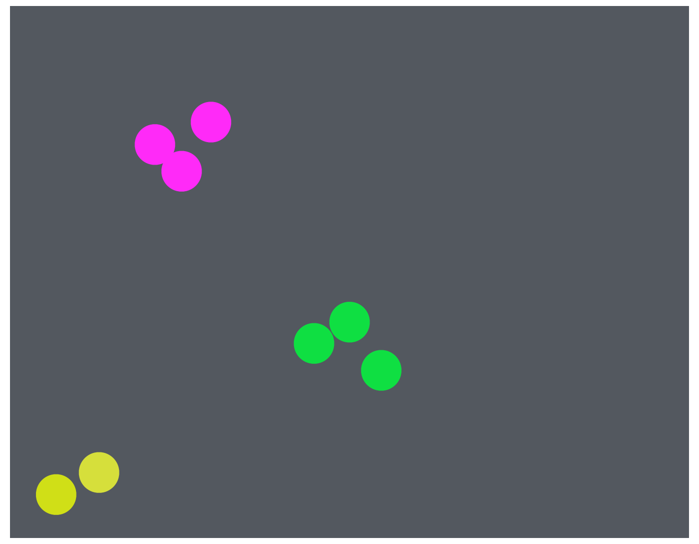
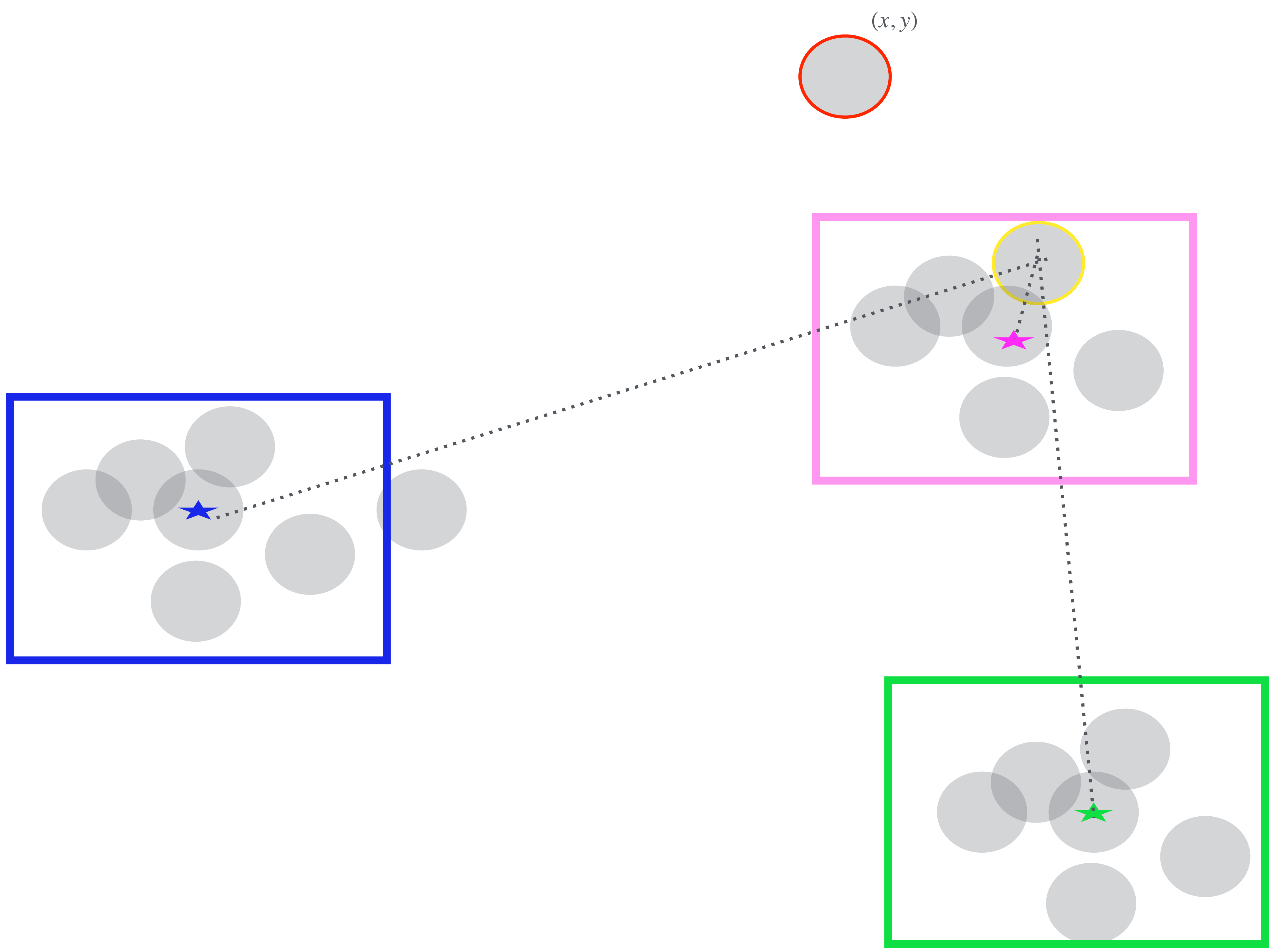
Clustering/Unsupervised Learning

Customer Segmentation

Name	Rank 1	Rank 2	Rank 3
Paul	Snacks	Book	Furniture
Liz	Perfume	Furniture	Book
⋮	⋮	⋮	⋮



Dimensionality Reduction Technique



x_1	x_2
⋮	⋮

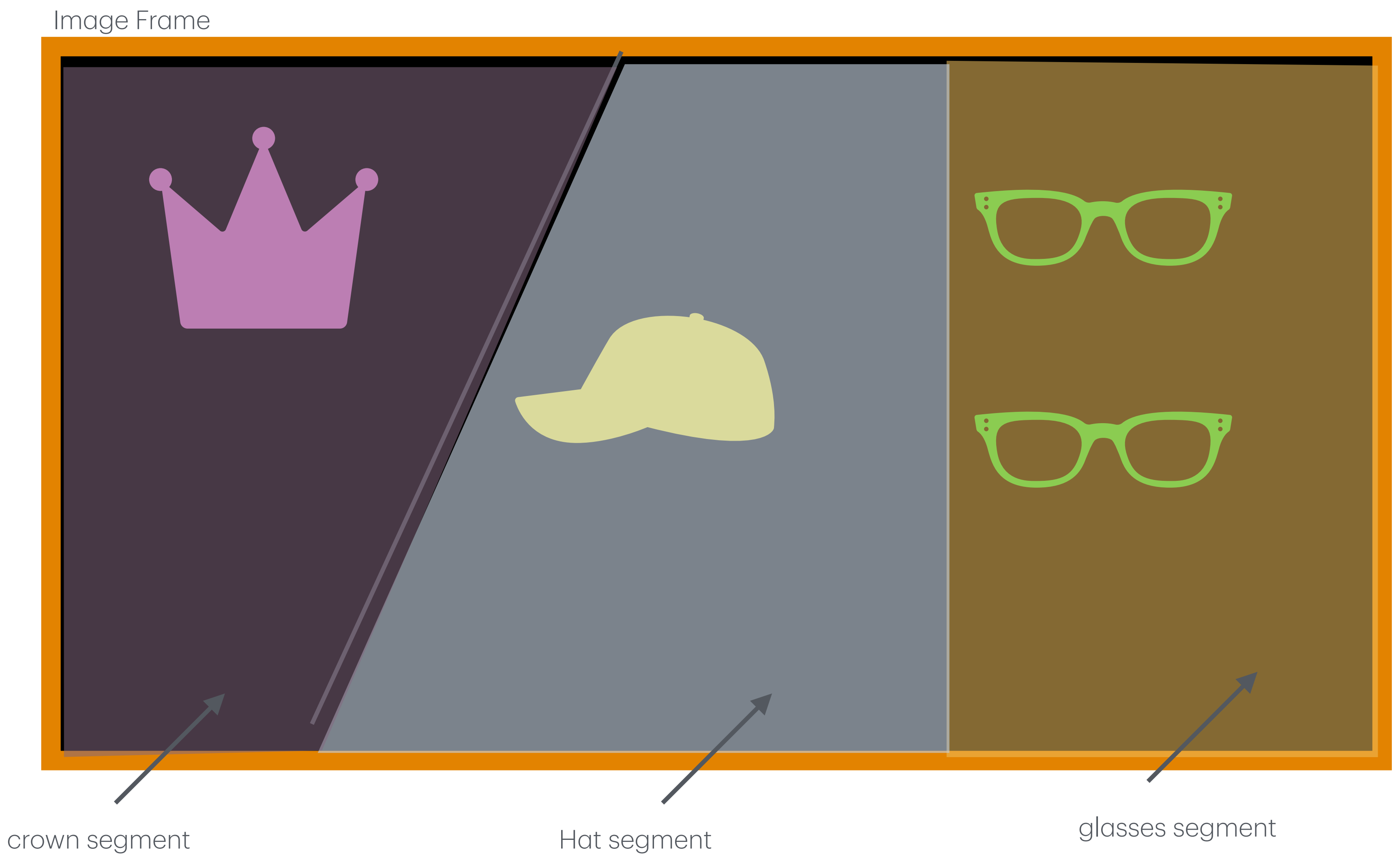
Assign each vector to a favorable cluster (high affinity)

	k_1	k_2	k_3
	0.93	0.4	0.2
	0.2	0.2	0.2
	⋮	⋮	⋮

Distance to blue cluster center

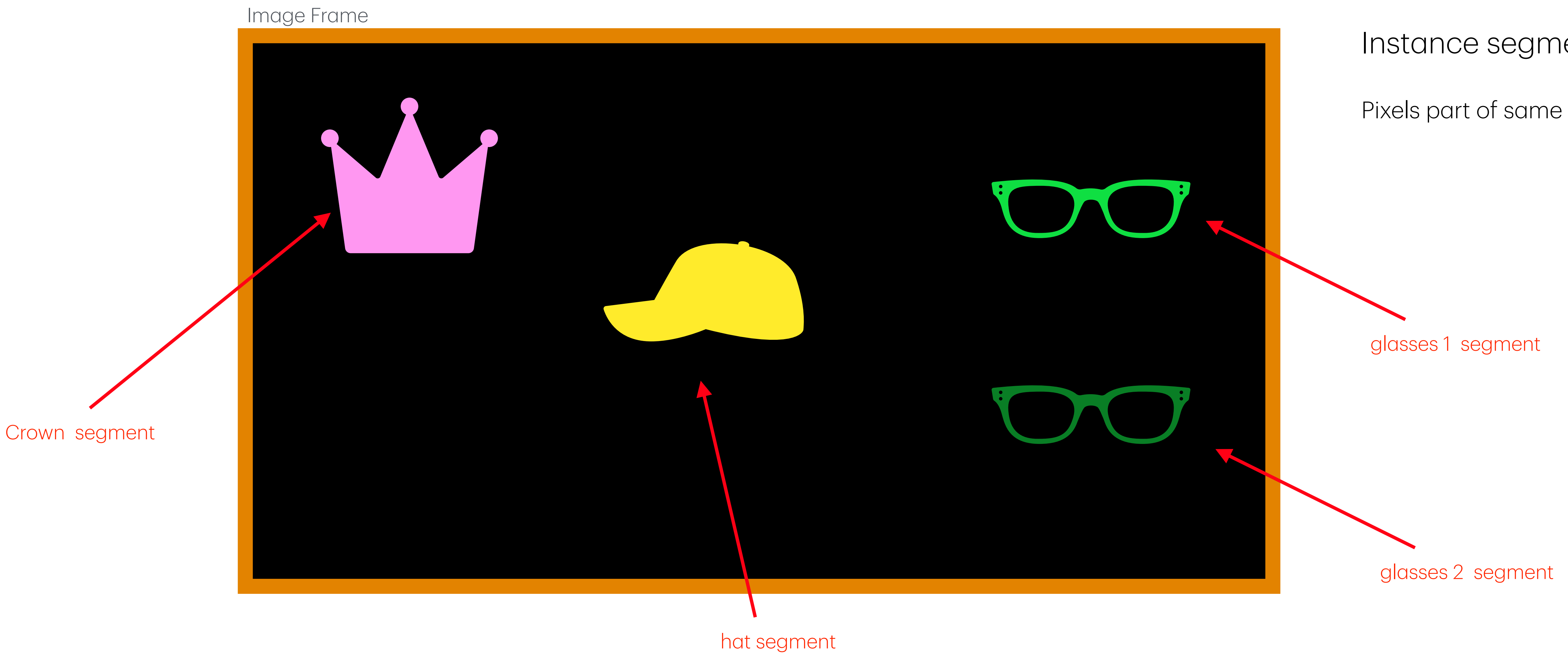
low affinity to all clusters (outlier)

Segmentation



Semantic segmentation:

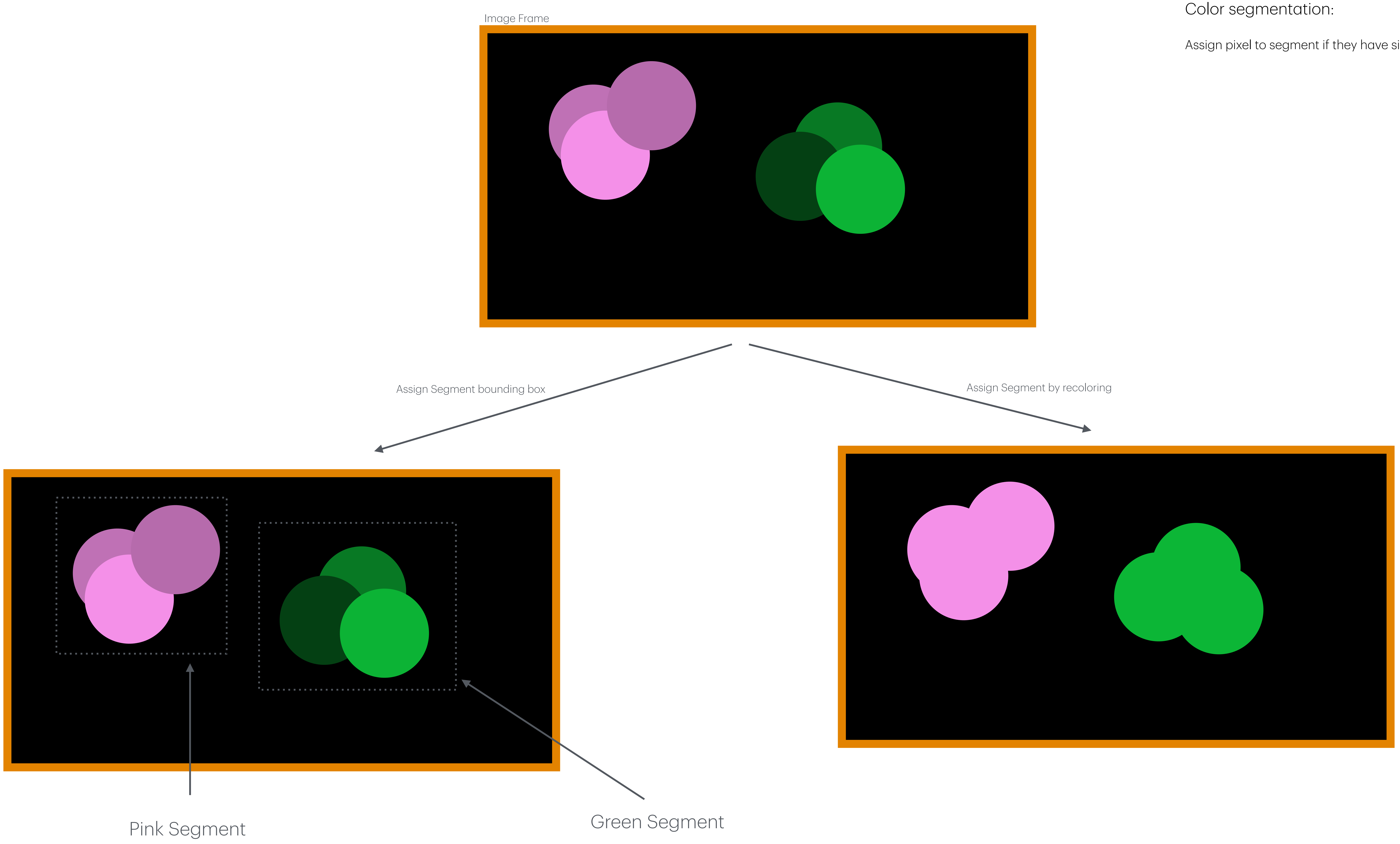
Pixels part of same object get assigned to same segment



Instance segmentation:

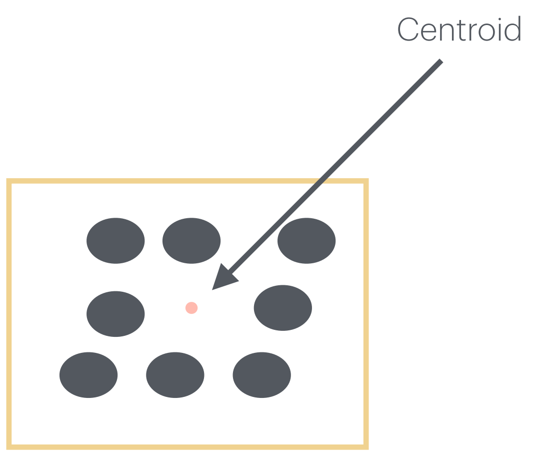
Pixels part of same individual object are assigned same segment (i.e. color)

Segmentation



Color segmentation:

Assign pixel to segment if they have similar color



Algorithm capable of clustering dataset

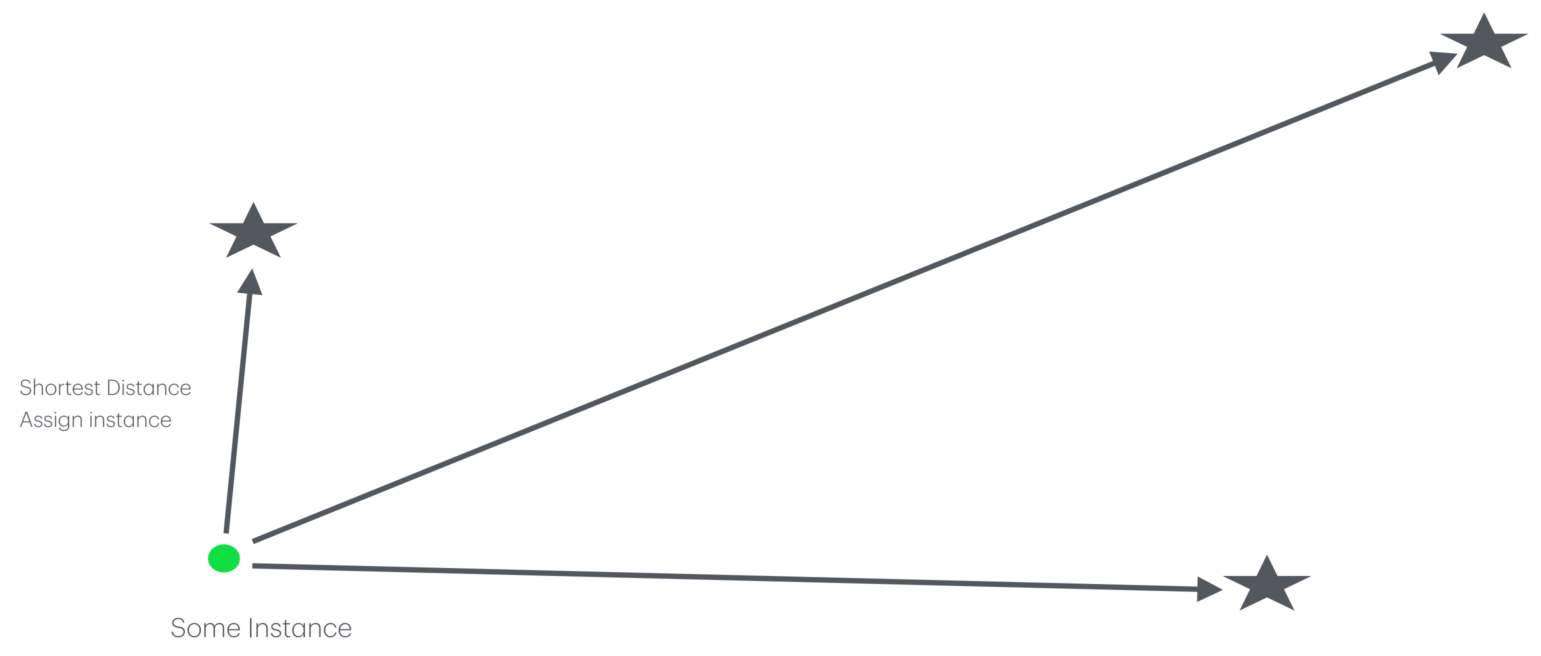
Hard Clustering - instances are assigned to a single cluster

Soft Clustering - instances are assigned a score

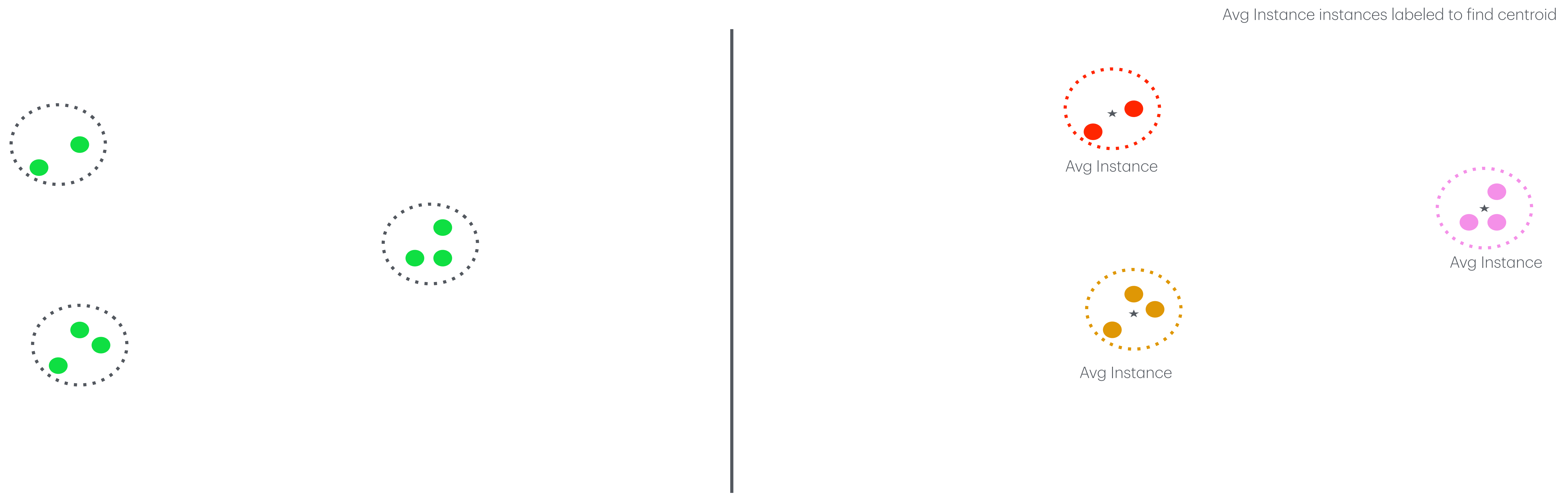
Score is the distance between instance and centroid

Kmeans

If centroids are provided

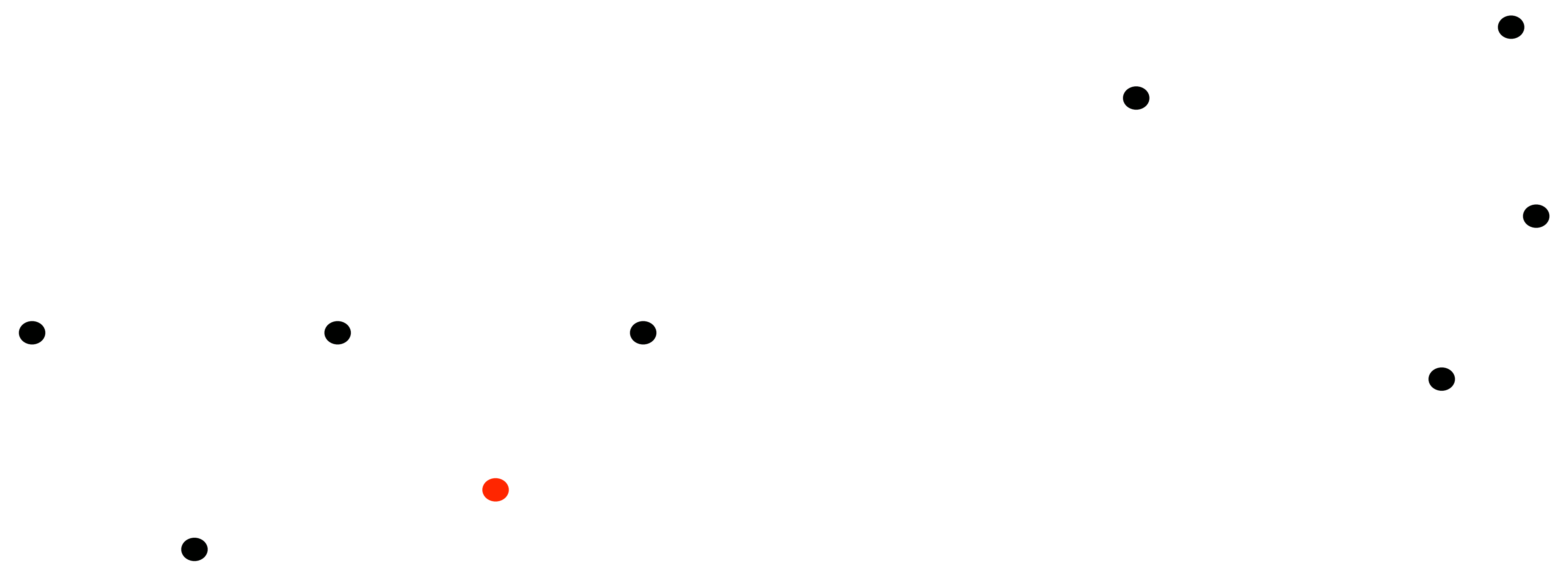


If instances are labeled

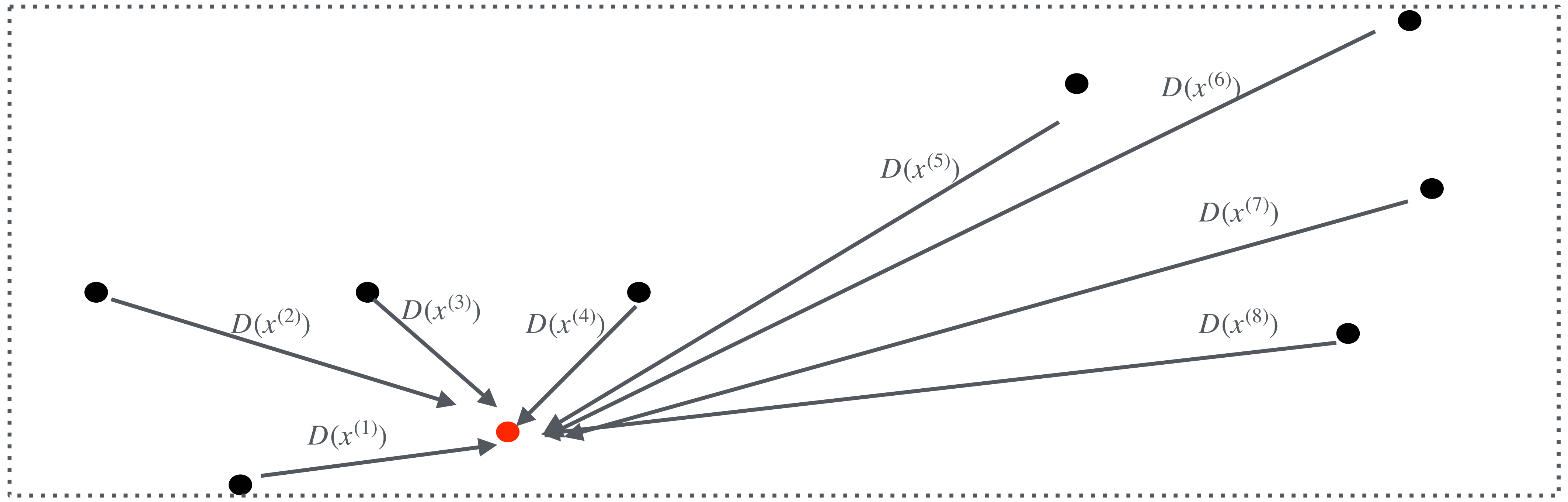


Kmeans Init $Method^1$ k =2

Choose random instance as centroid



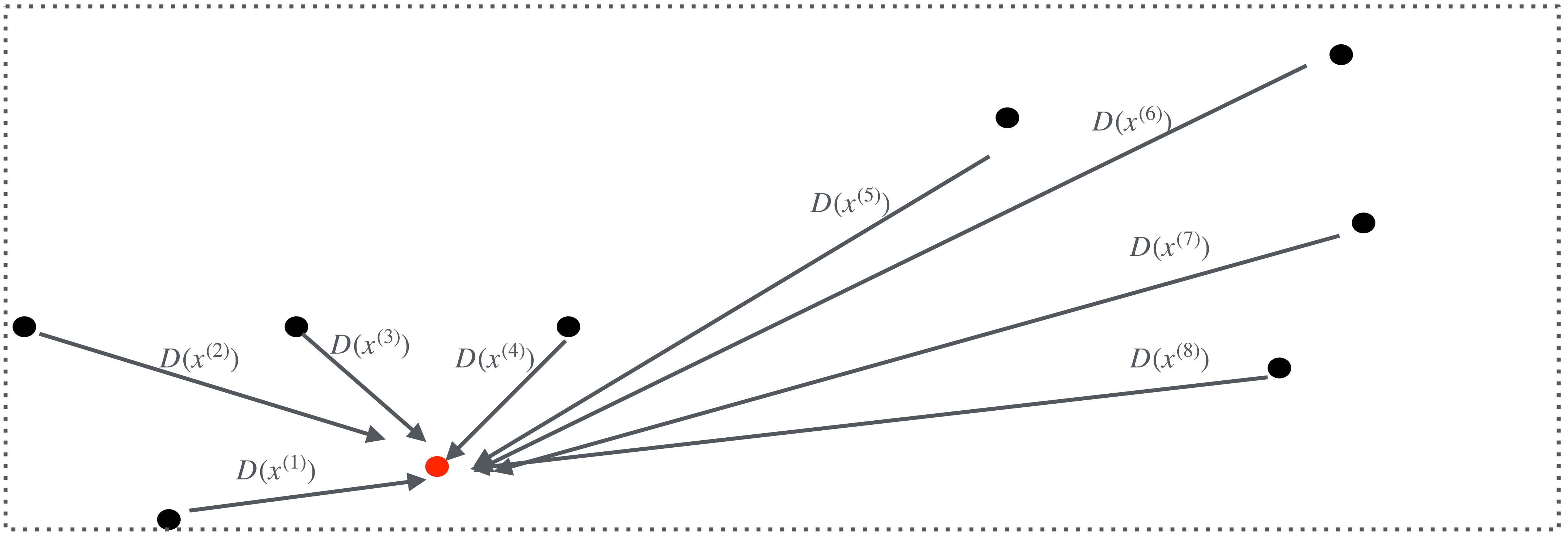
Kmeans Init $Method^1$ k =2

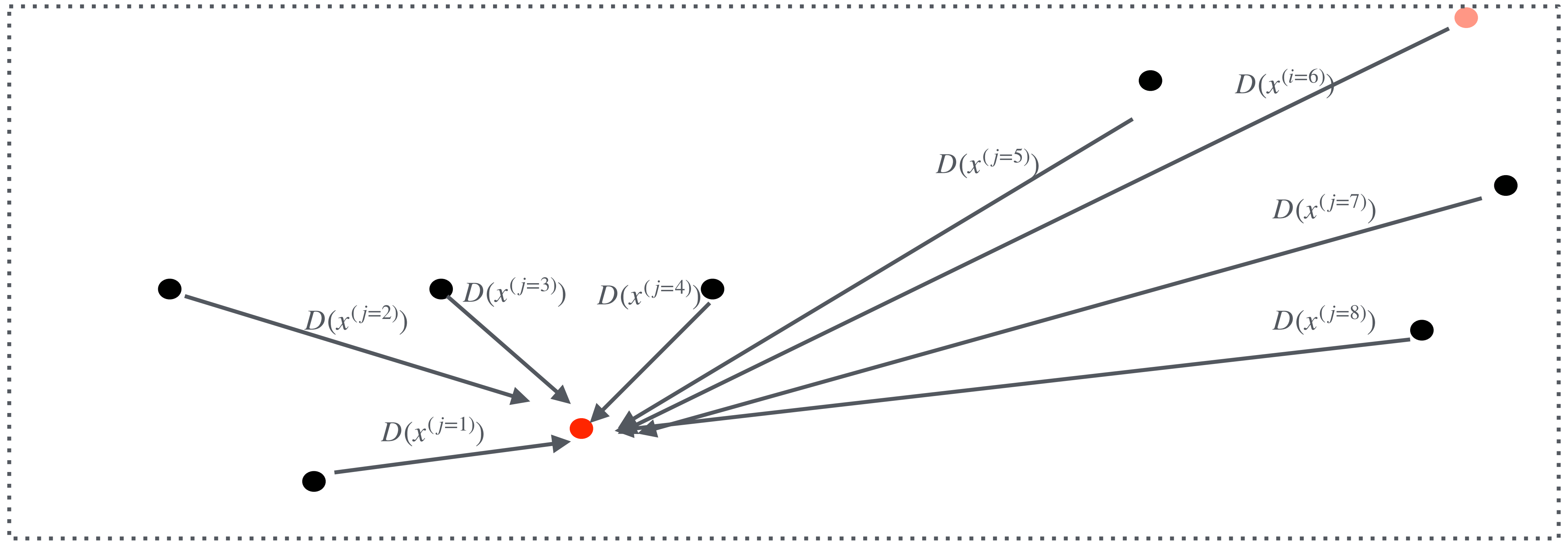


New centroid is instance highest probability:

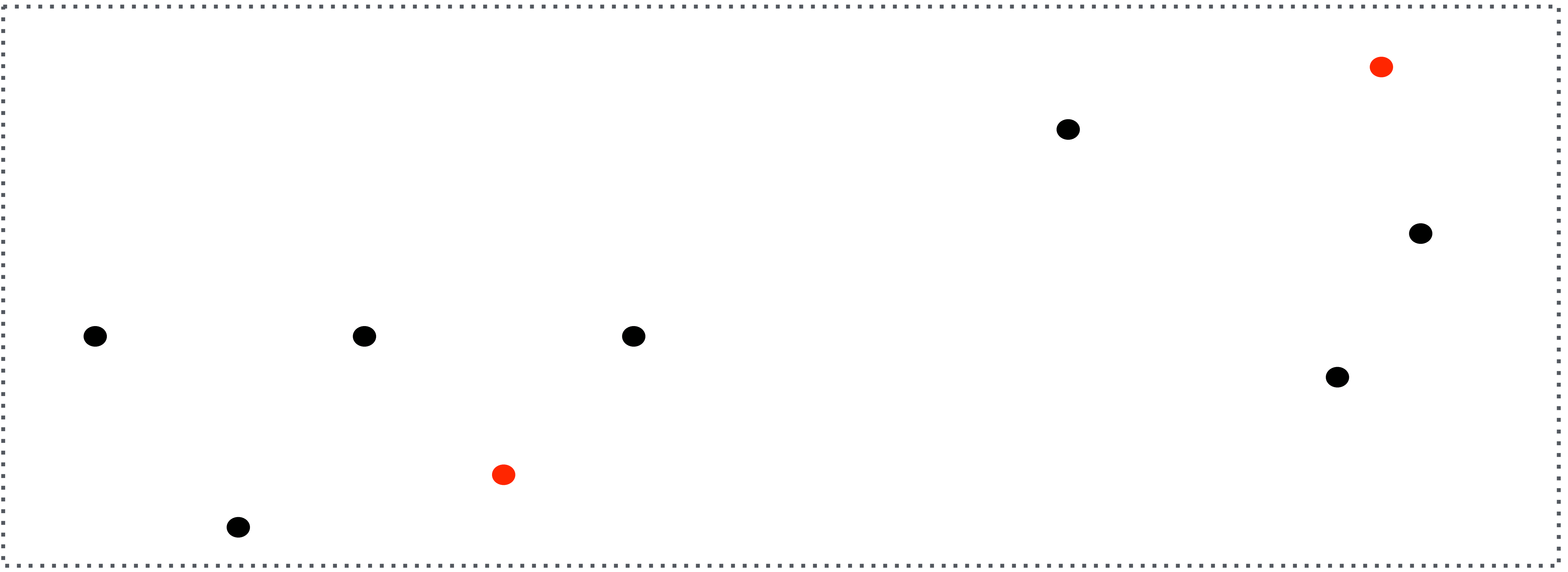
$$\text{New centroid with } \boldsymbol{probability} = \frac{D(x^{(i)})^2}{\sum_{j=1}^{m} D(x^{(j)})^2}$$

Select $D(x^{(6)})$

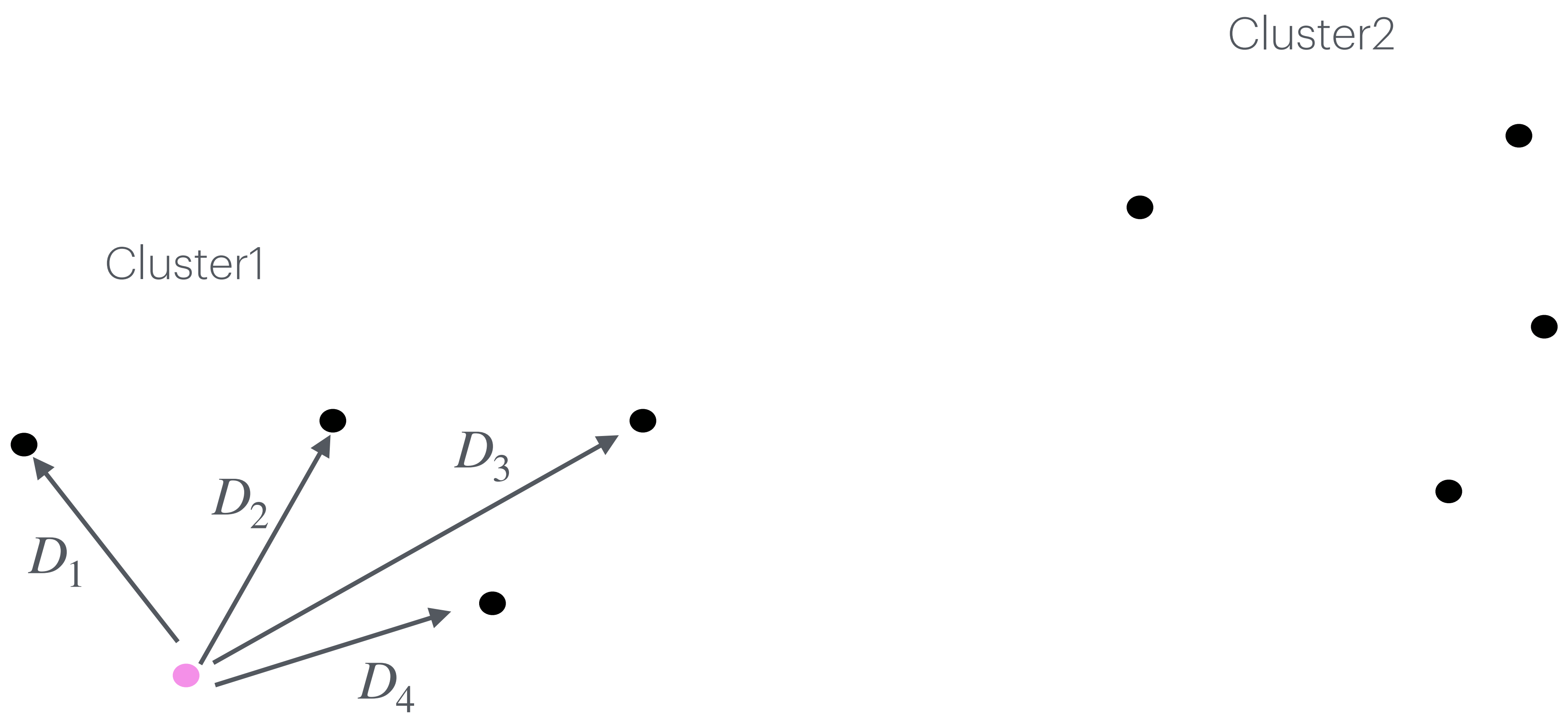




$$\frac{D(x^{(i=6)})^2}{\sum_{j=1\ j\neq 6}^{m} D(x^{(j)})^2}$$

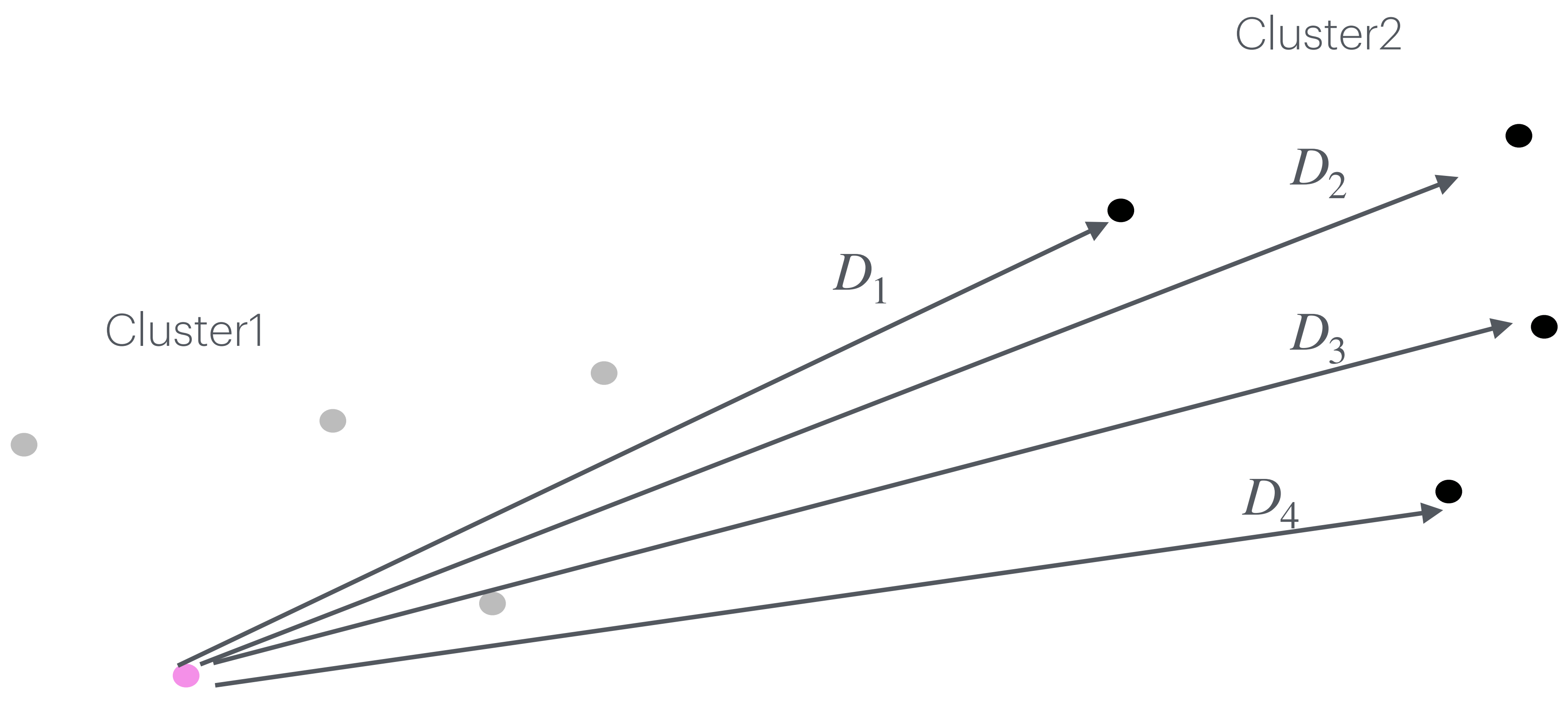


Silhouette Coefficient per instance



$$x^i \ avg \ distance \ to \ other \ x^j \ incluster \ = a = \sum_{j=1}^{N_{thiscluster}-1} D_j$$

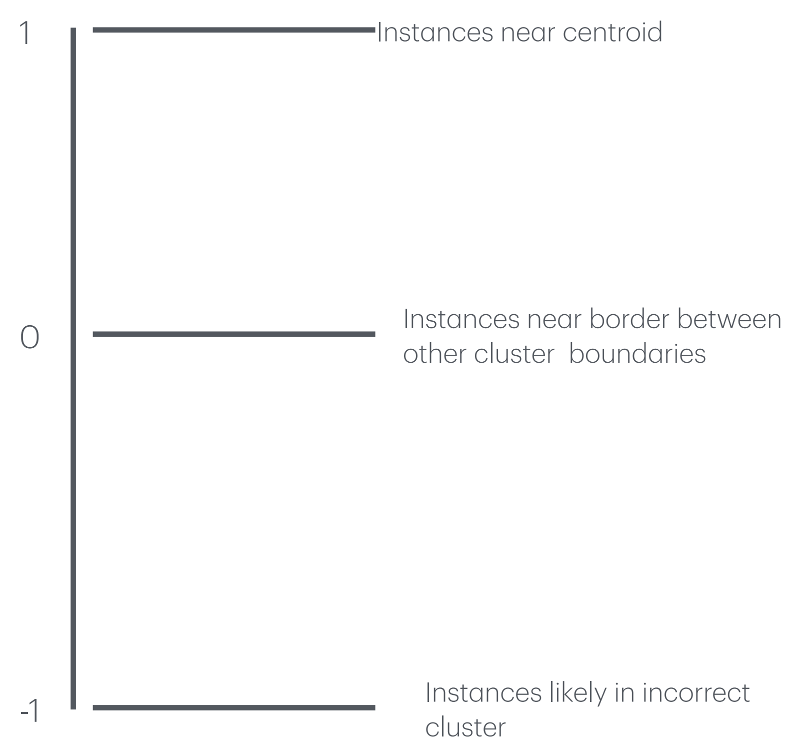
Silhouette Coefficient per instance



$$x^i \text{ avg distance to other } x^j \text{ in nearest cluster } = b = \sum_{j=1}^{N_{thiscluster}} D_j$$

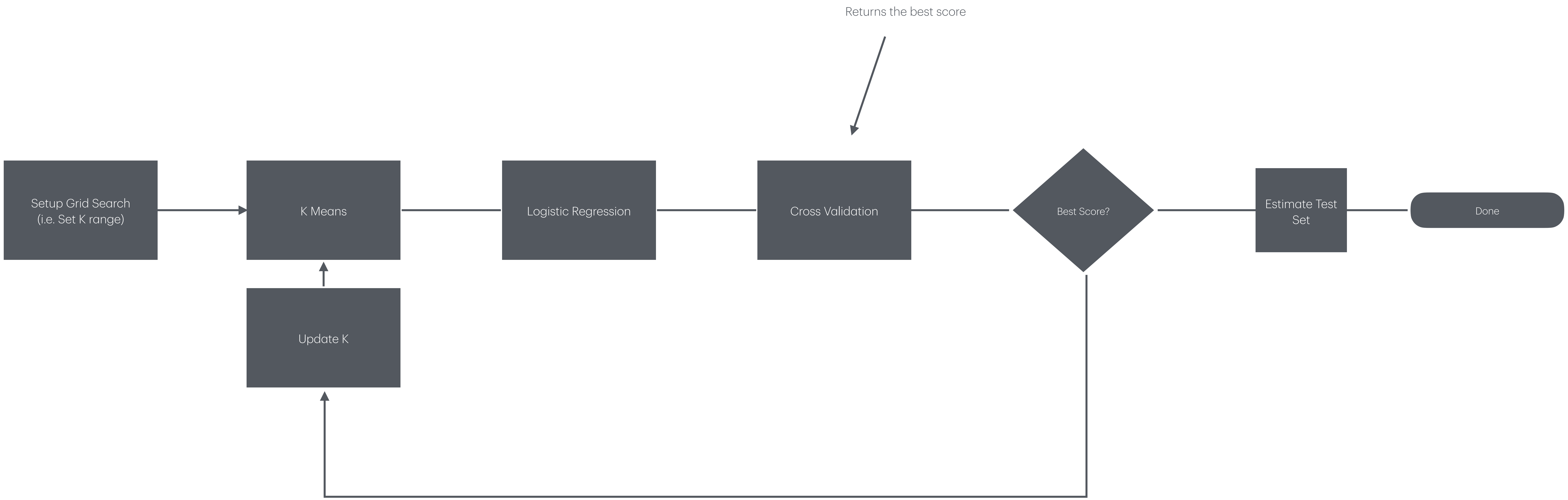
Silhouette Coefficient per instance

$$\text{Silhouette Coefficient} = \frac{(b - a)}{max(a, b)}$$



$$Silouette\ \ Score = \frac{1}{m}\sum_{i}^{m} SilhouetteCoefficient_i$$

Best Method dining K



Semisupervised

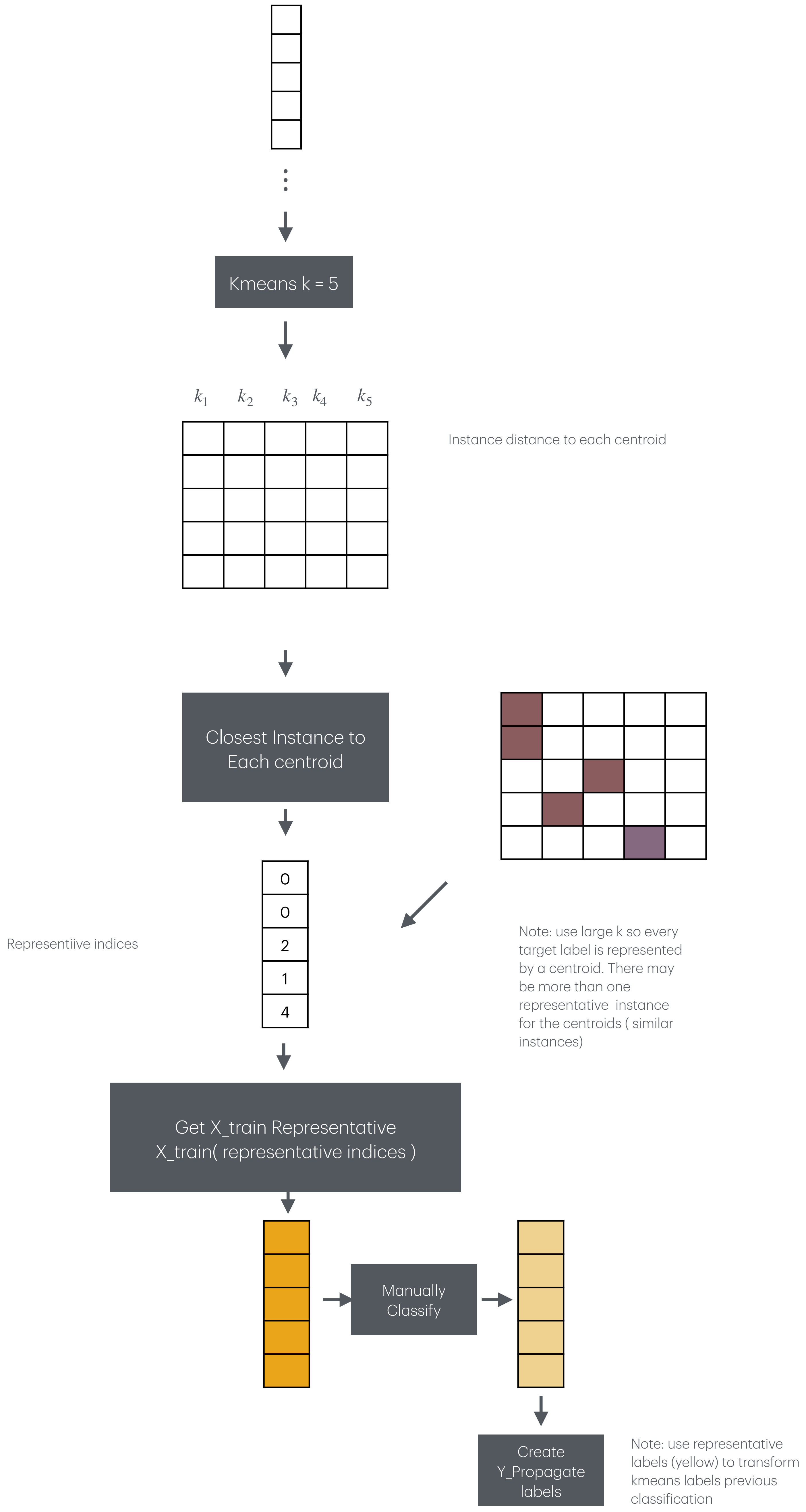
Cluster Training Data

Find Instances closest to centroid (representative Data)

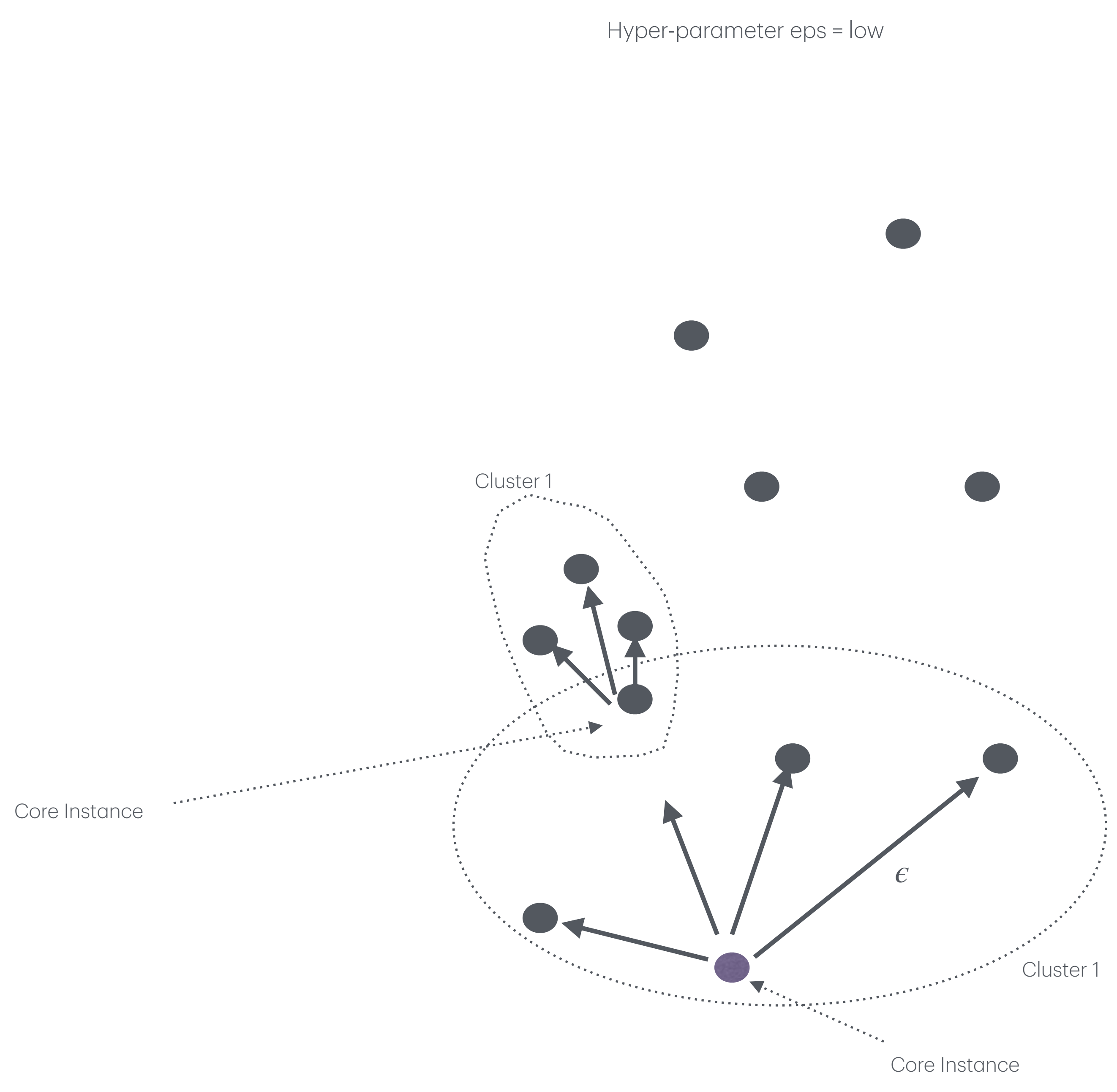
Find Instances closest to centroid

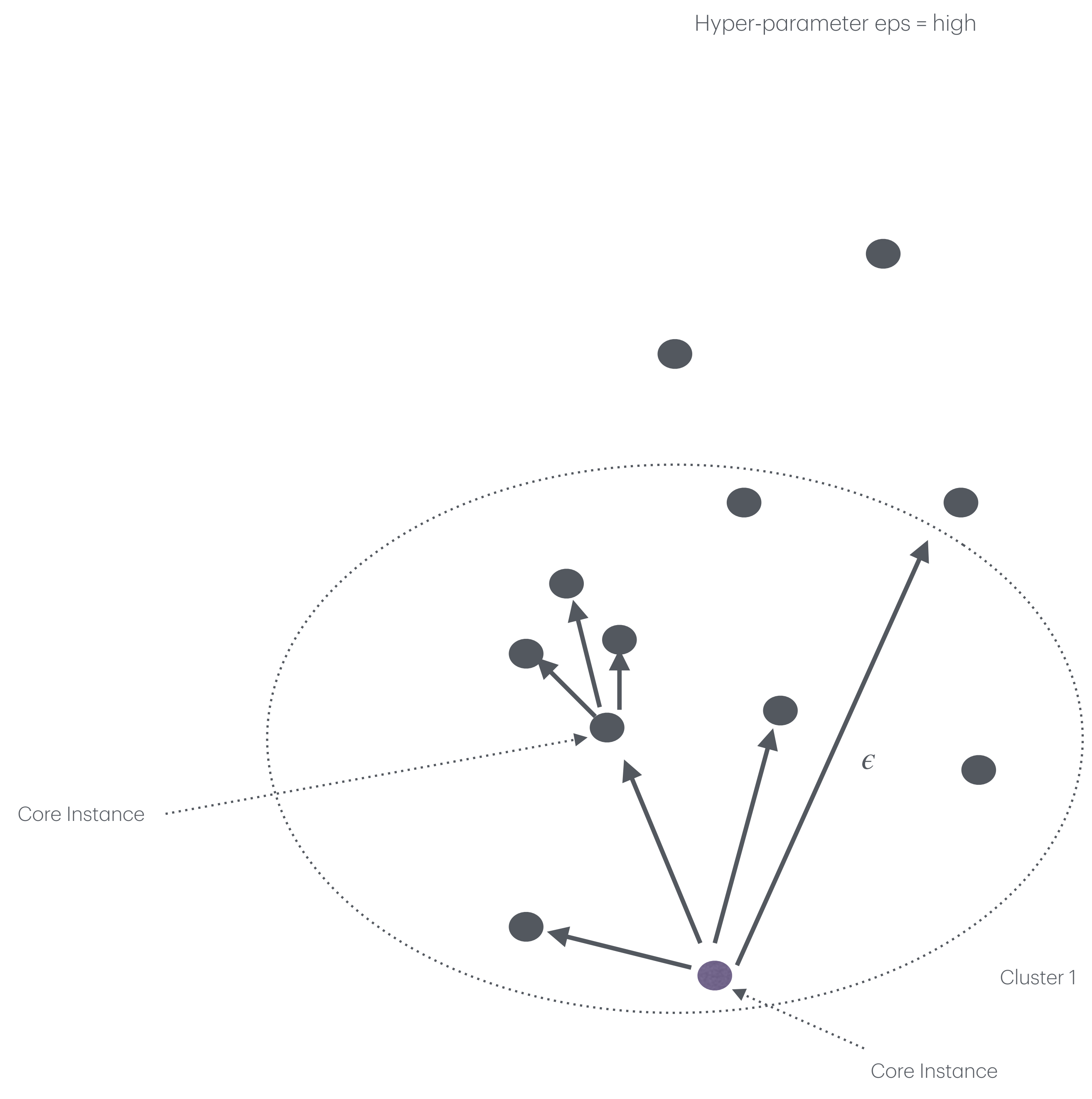
Train (representative_instance, representative_label)

Label Propagation

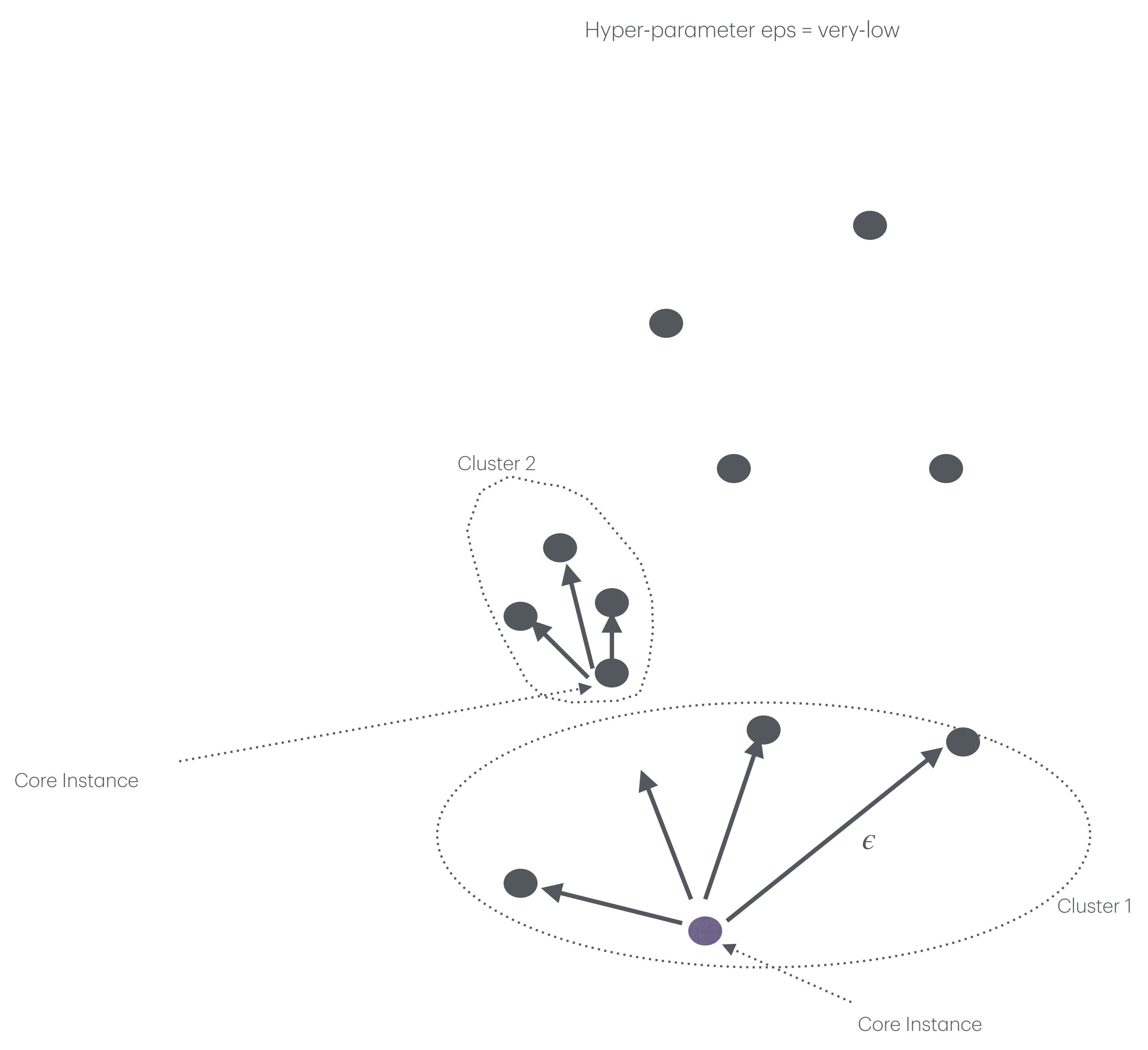


DBSCAN

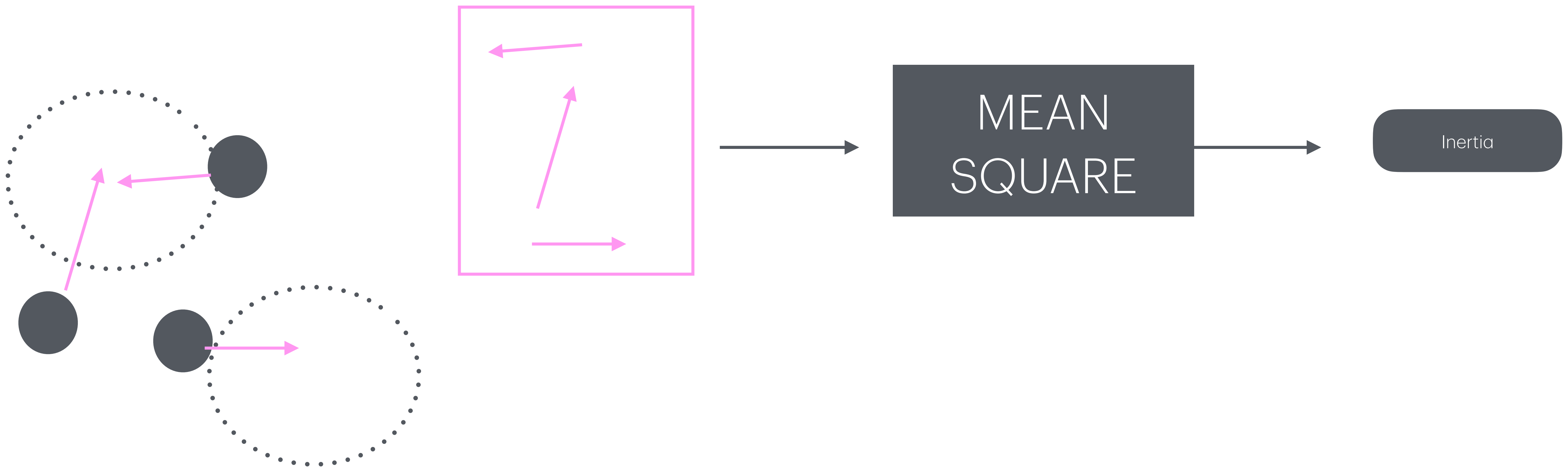

Hyper-parameter eps = low
Cluster 1
Core Instance
ϵ
Cluster 1
Core Instance


Hyper-parameter eps = high
Core Instance
ϵ
Cluster 1
Core Instance

DBSCAN


Hyper-parameter eps = very-low
Cluster 2
Core Instance
ϵ
Cluster 1
Core Instance

Inertia



Distance to each instance and closest centroid

Gaussian Mixture

Probabilistic model assumes instances are generated from a mixture of several Gaussian distributions

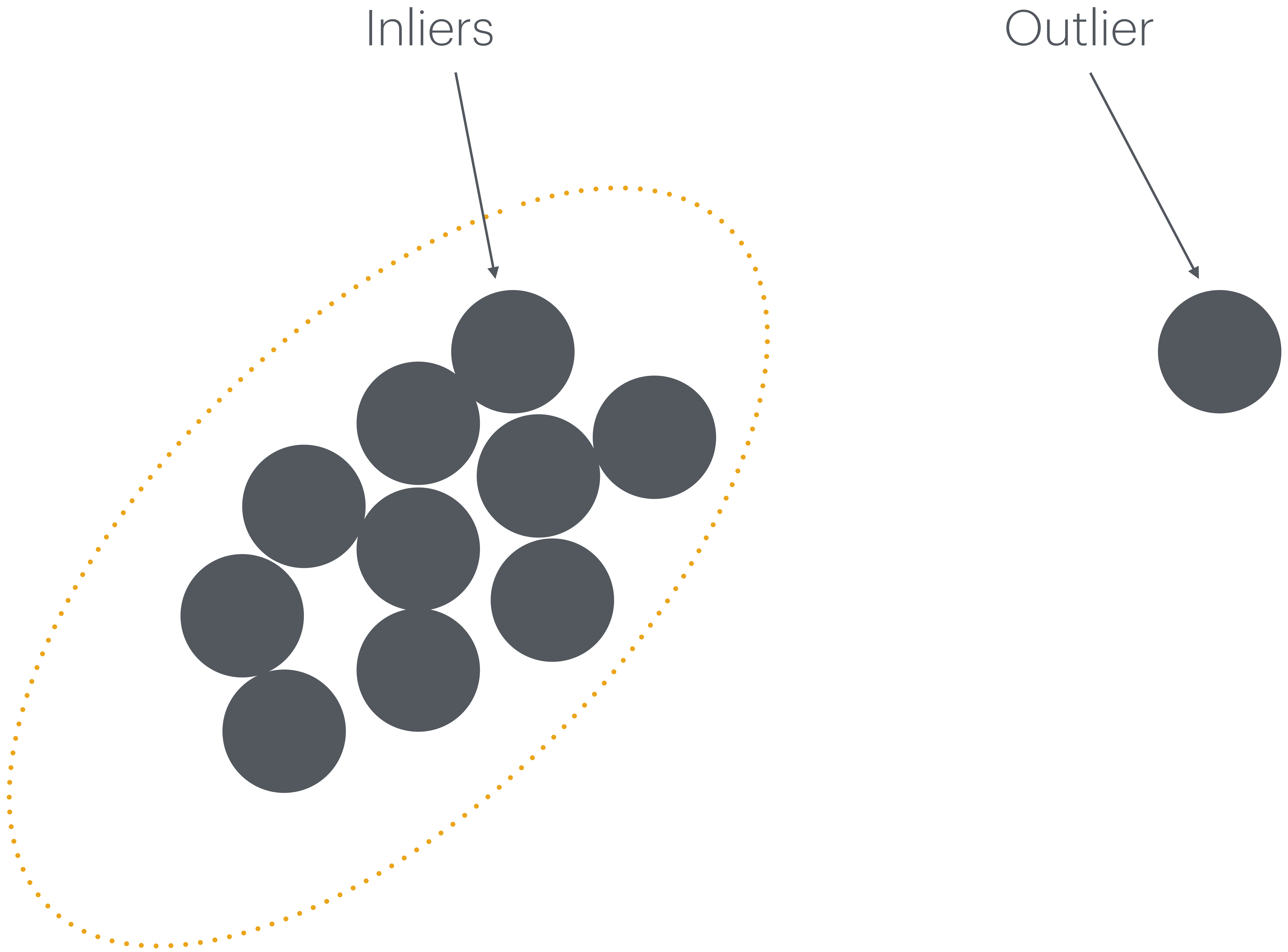
Instances is replaced with sample(s) from Gaussian Distributed Clusters

Model is generative so new instances with can be generated using it

Gaussian Mixture

Primarily used for ***anomaly detection***

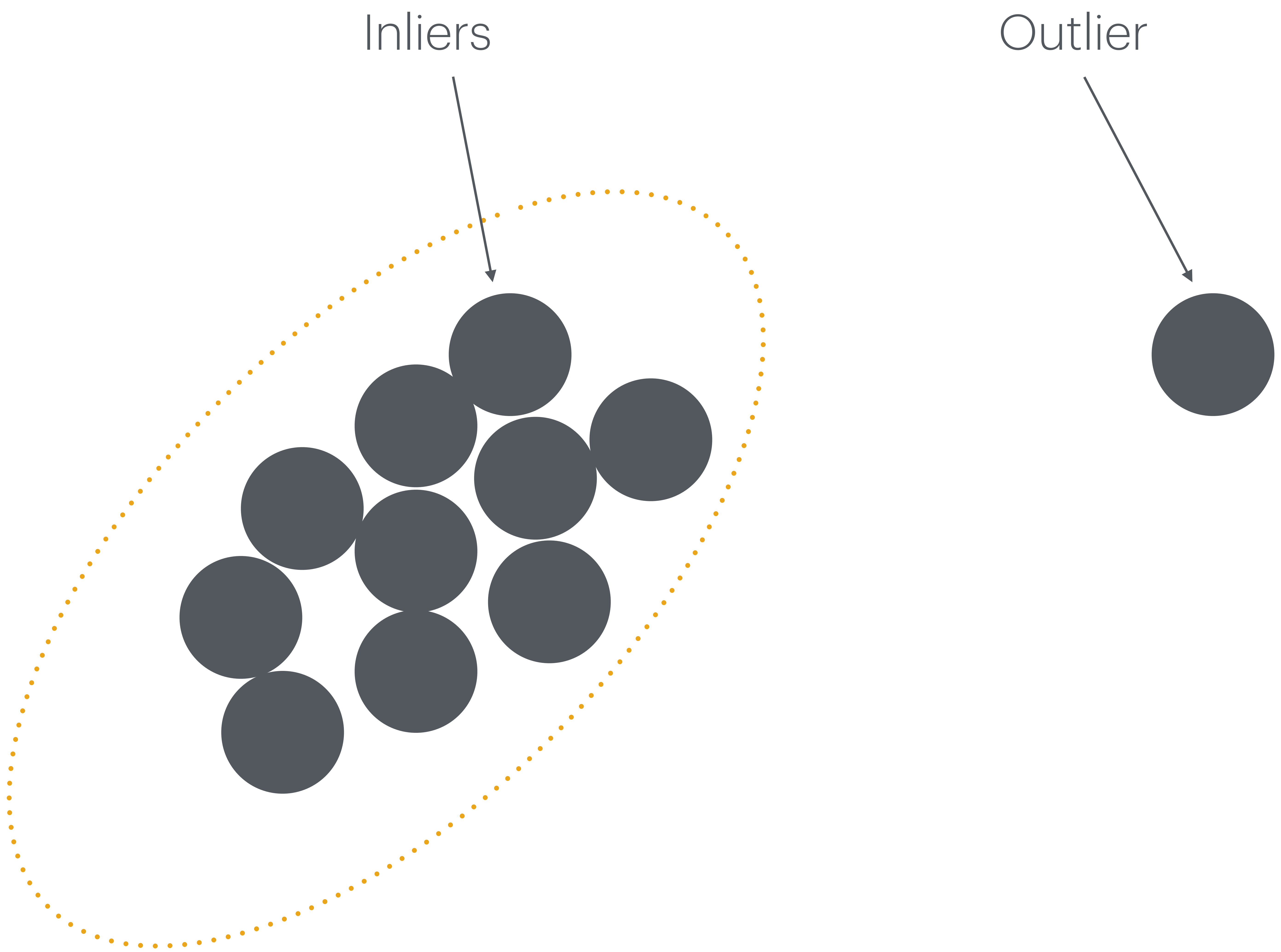
Assumes dataset is clean and uncontaminated.



Gaussian Mixture

Also used for
novelty detection

Assumes dataset is
clean and
uncontaminated.

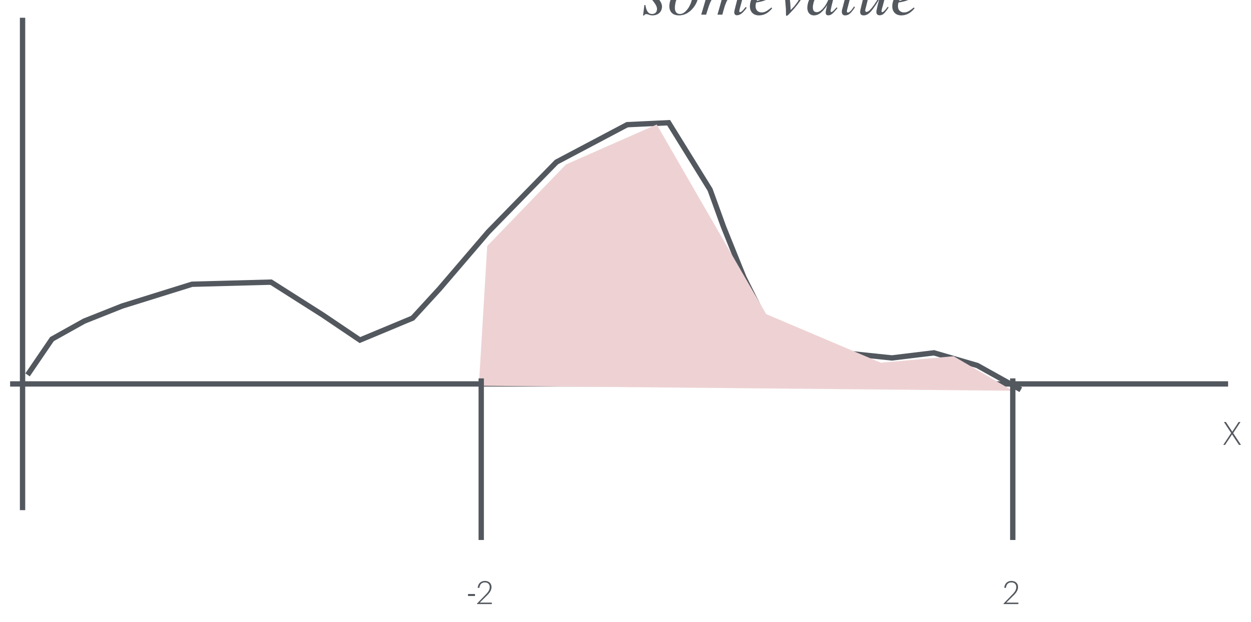


Model = $\left[\theta_1 \quad \theta_2 \quad \theta_3 \quad \dots\right]$

Probability

How **plausible future outcome** x(r.v) is knowing model parameters or weights

PDF$(x, \theta_{somevalue})$

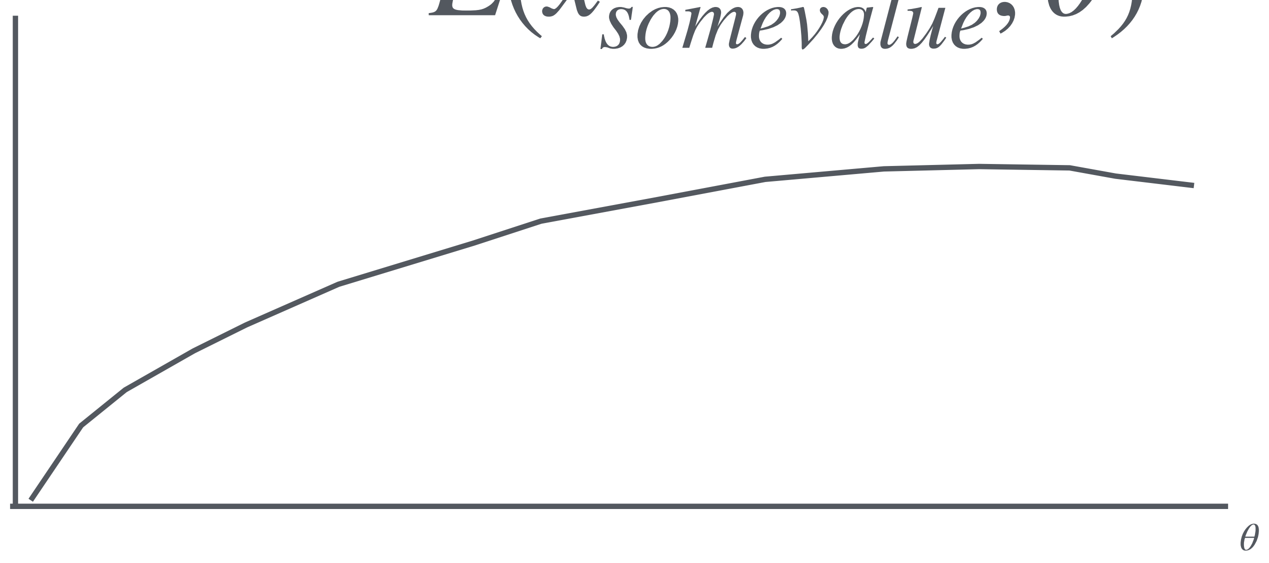


Probability **future outcome(r.v)** is between -2 and -2

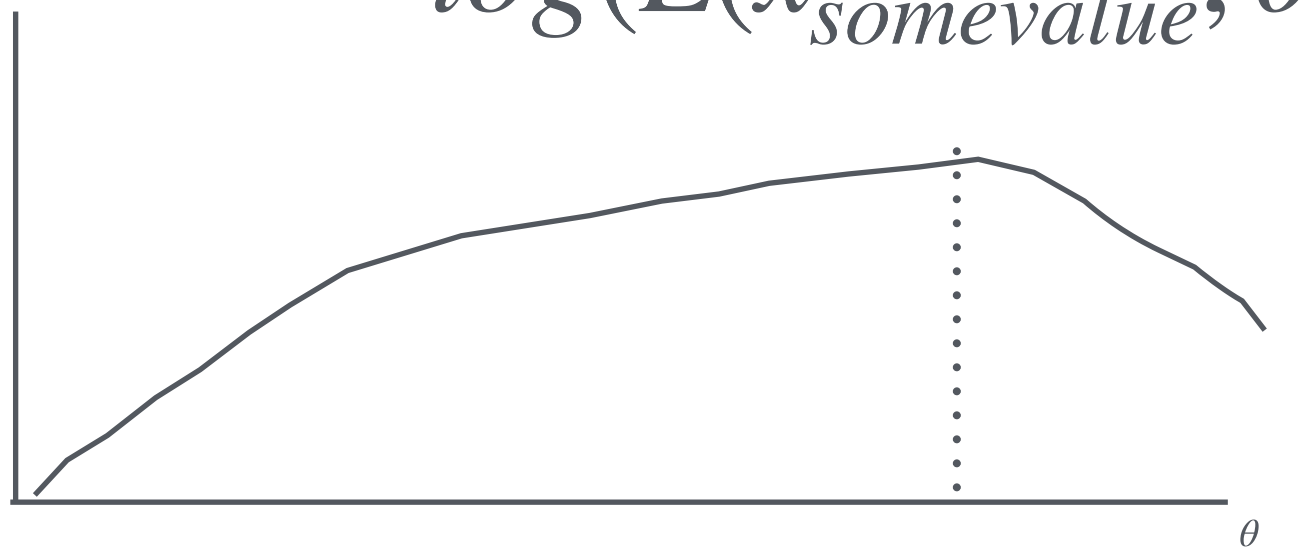
Likelihood

How **plausible parameters** are, after the outcome x(r.v) is known.

$L(x_{somevalue}, \theta)$



$log(L(x_{somevalue}, \theta))$

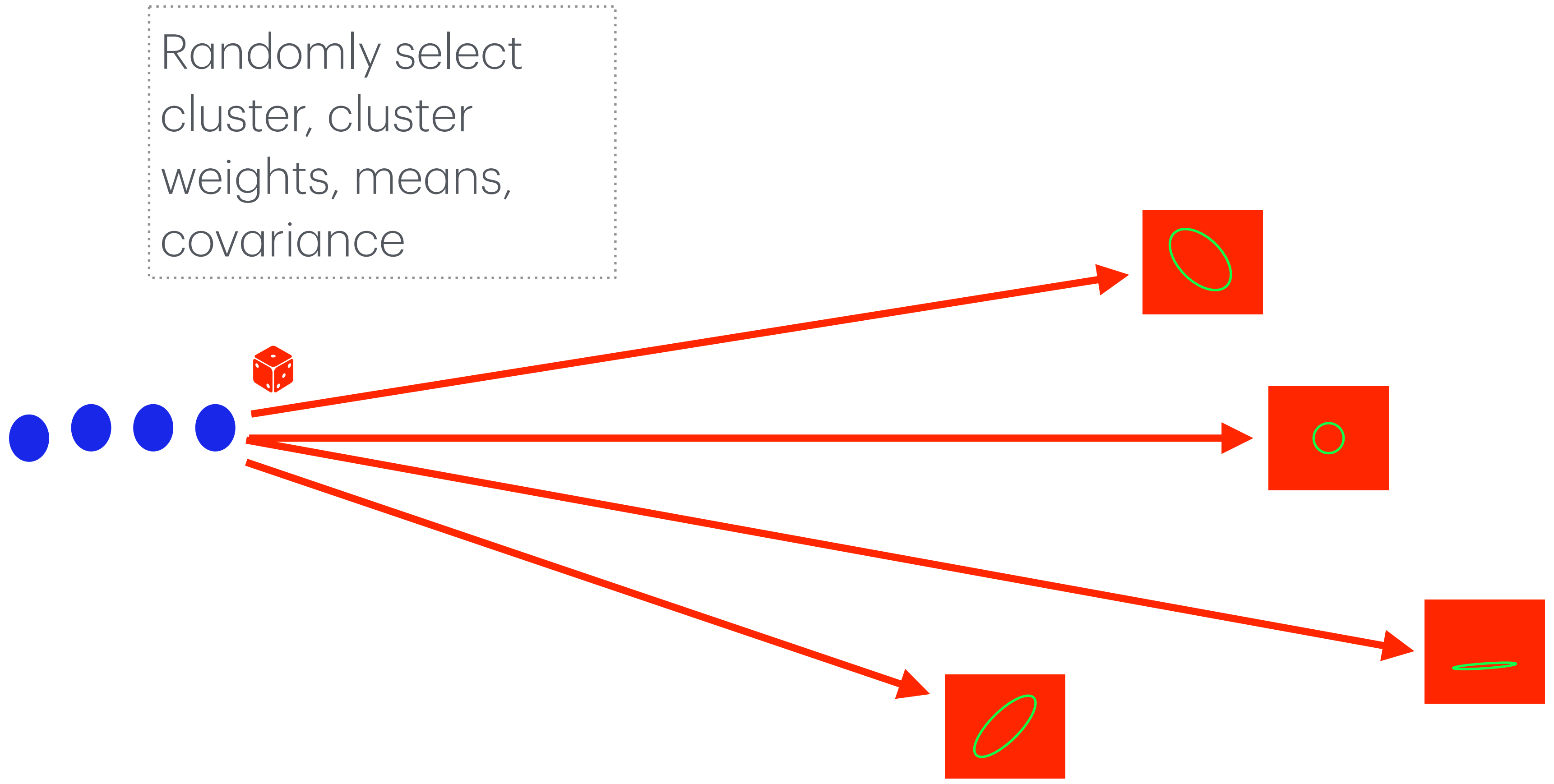


Log of likelihood function reveals **parameters** which make an outcome likely

Bayesian Gaussian Mixture

Find model that minimizes a theoretical information criterion

Bayesian Gaussian Mixture



Beta distribution model

used to generate clusters (i.e. weights, means, covariances)

Φ = [0.3 , 0.6 ...]

- 30 percent of instances go to cluster 1
- 40 percent of remaining instances to to cluster 2

Beta variable α

α large ($\Phi \approx 0$) (many clusters)

α small ($\Phi \approx 1$) (few clusters)

Wishart Distribution model

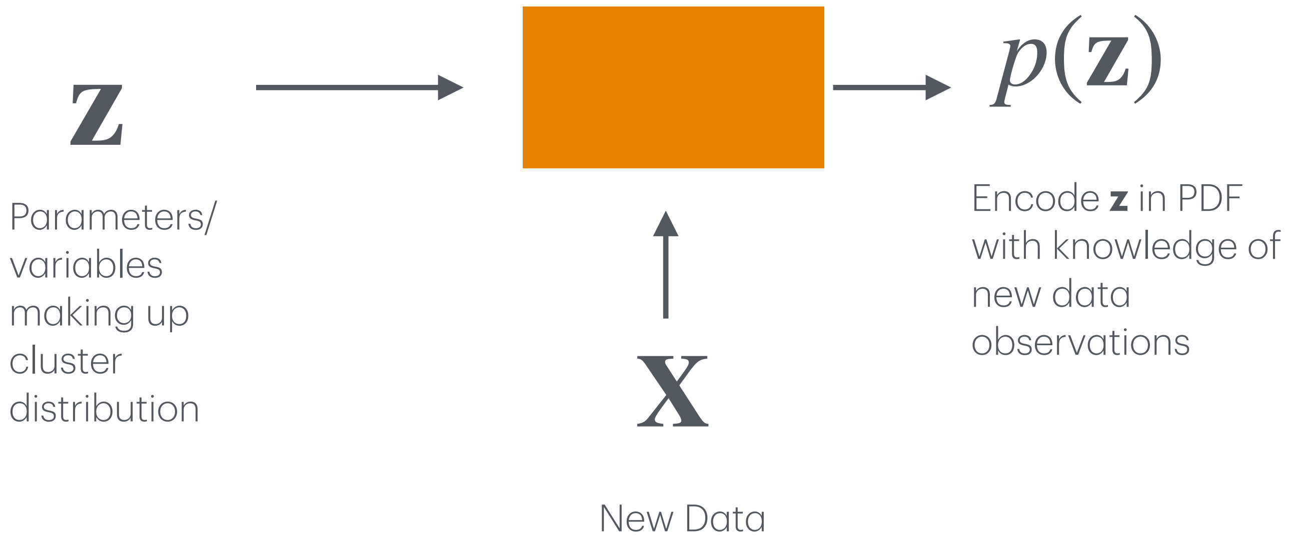
used to generate cluster shapes (samples covariance matrices)

W(d, V)

Bayesian Gaussian Mixture

Assumes clusters and their parameters have been sampled ($\mathbf{z}^i$)

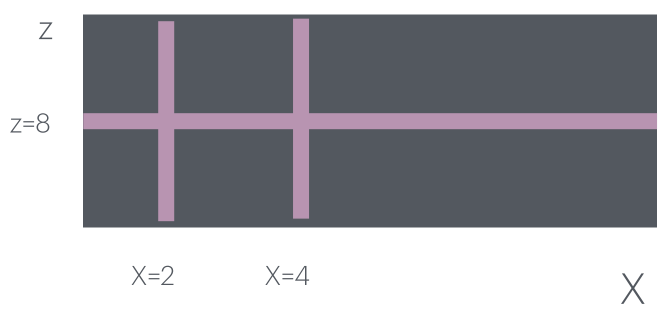
$p(\mathbf{z}^i)$ - encode cluster distribution with prior knowledge



$p(\mathbf{z}\,|\,\mathbf{X})$ = posterior = $\dfrac{p(\mathbf{X}\,|\,\mathbf{z})p(\mathbf{z})}{p(\mathbf{X})}$

How to update **z** probability distribution after observing new data

$p(\mathbf{X}\,|\,\mathbf{z})$
- probability of **X**(observation/event) knowing **z** variables/parameters



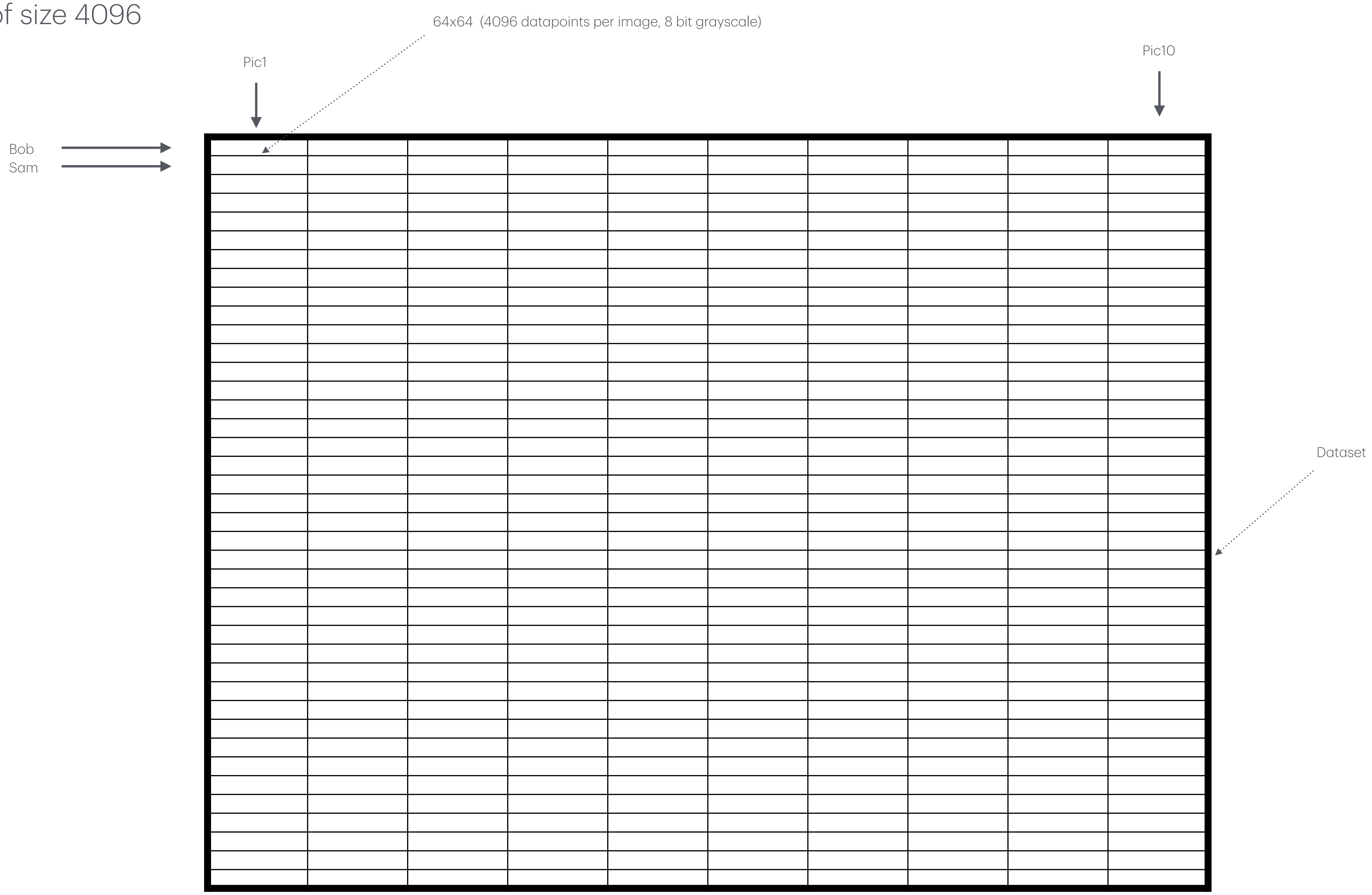
$p(\mathbf{z})$
- *prior* knowledge encoded distribution

Encode examples:
- clusters likely to be less dense (after observing new samples)
- clusters likely to be highly dense
- clusters likely to be spherical shaped

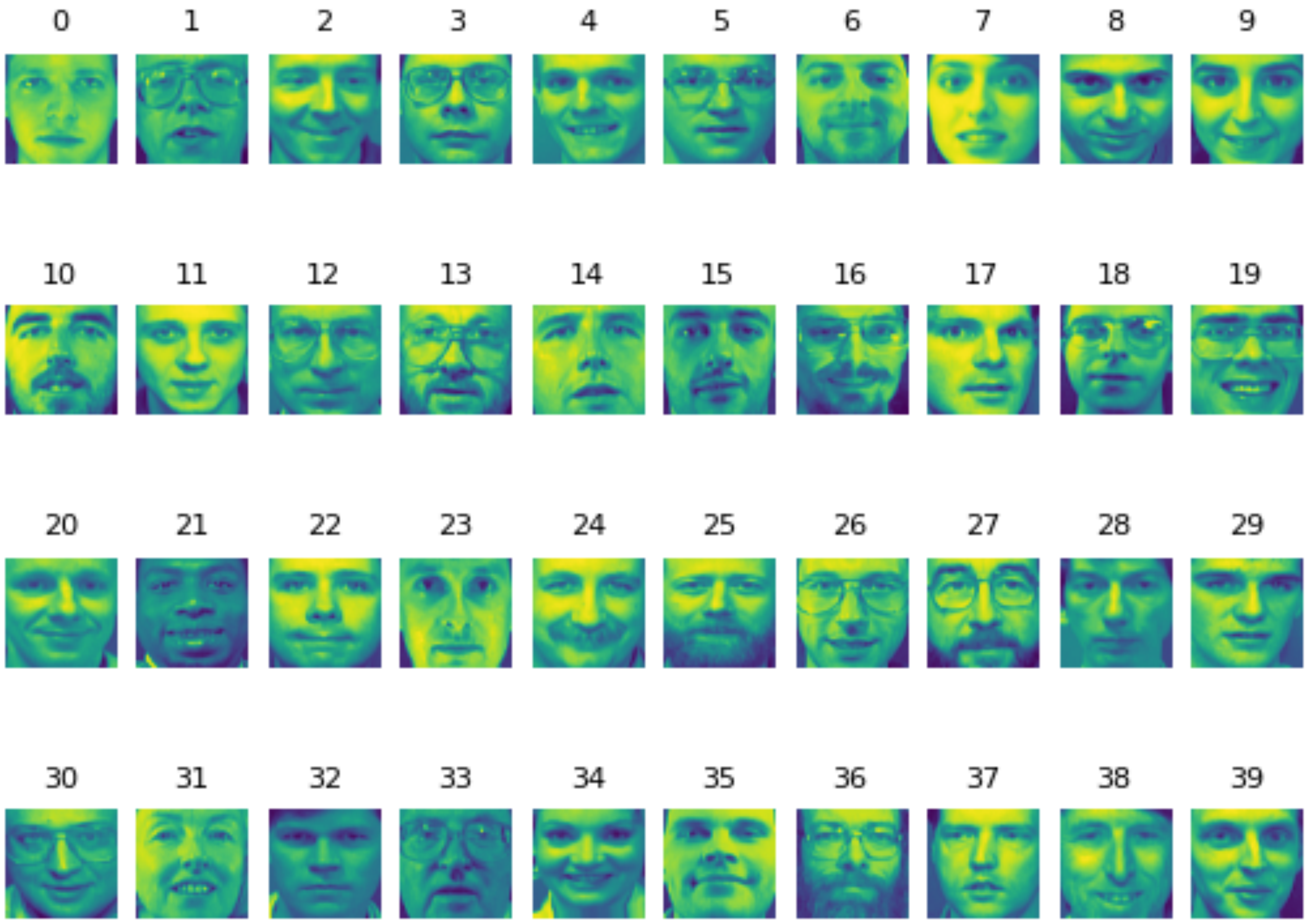
$p(\mathbf{X})$
- probability of observations
- Requires integrating over all possible values of z (impossible as that would consider every possible combination of cluster parameters and cluster assignment)

Olivetti Faces Dataset

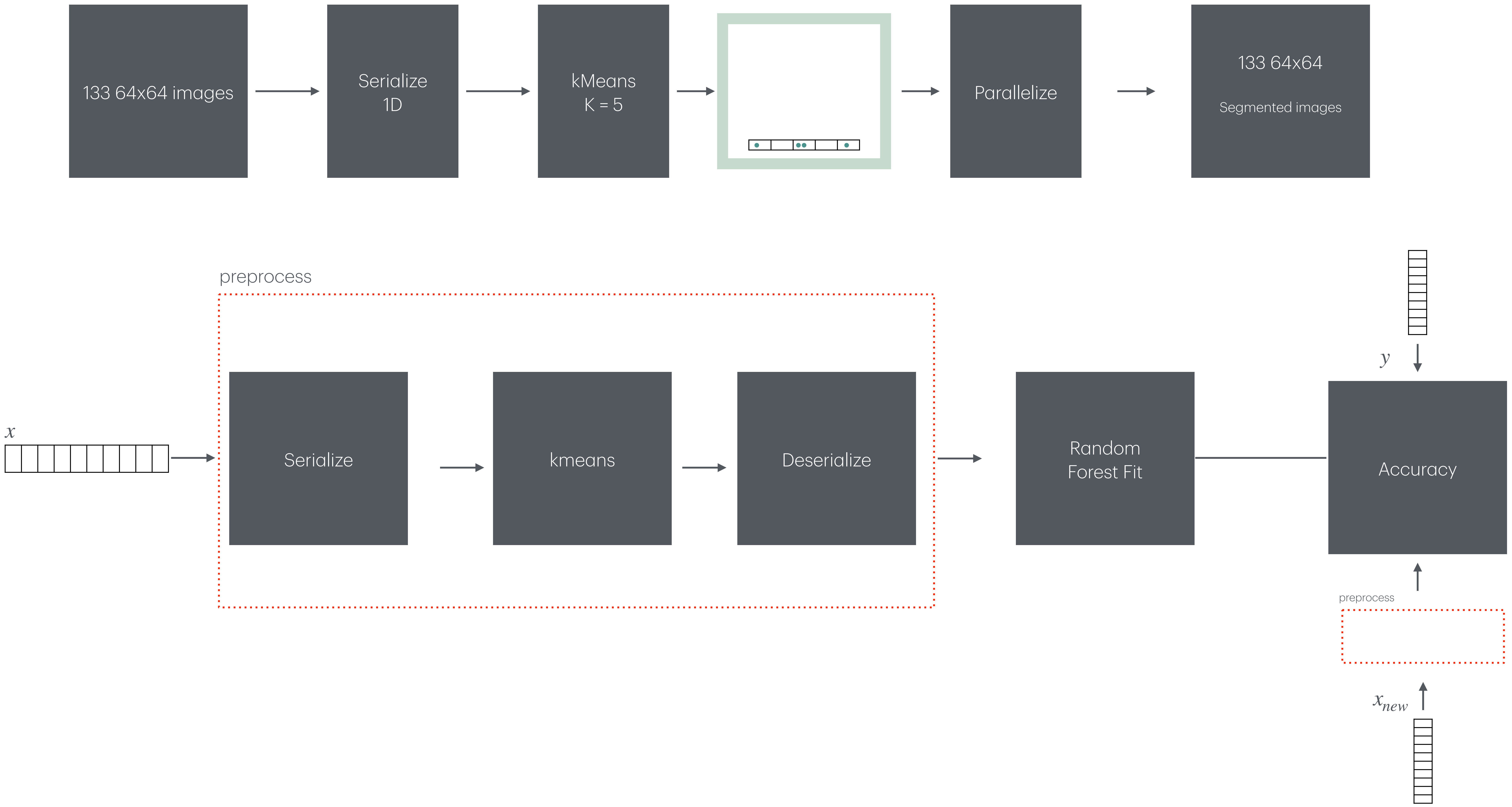
N= 400
Dimension 64x64
Instance - 1D vector of size 4096



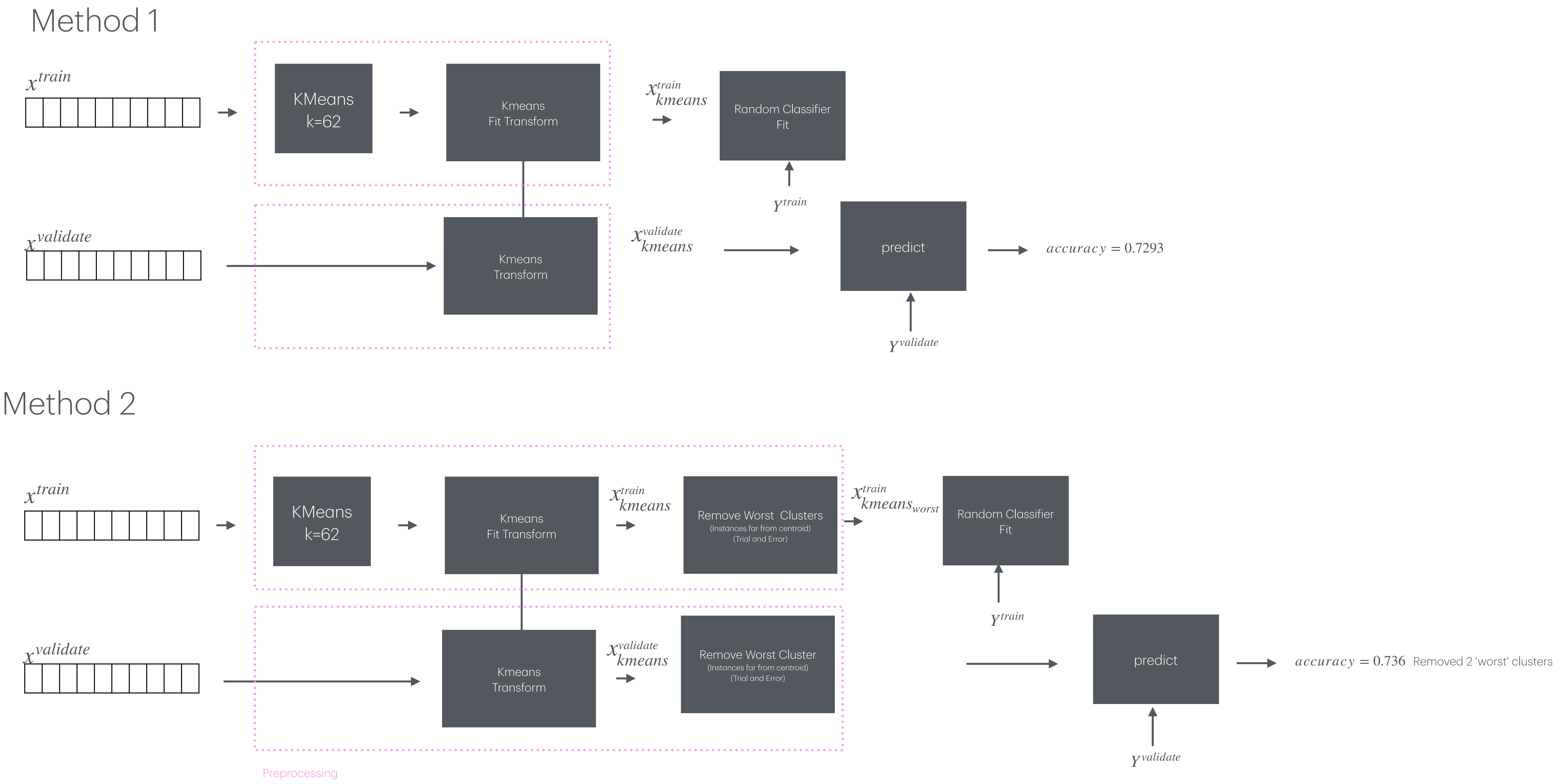
40 Unique People



Pipeline Exercise 11: Image Segmentation

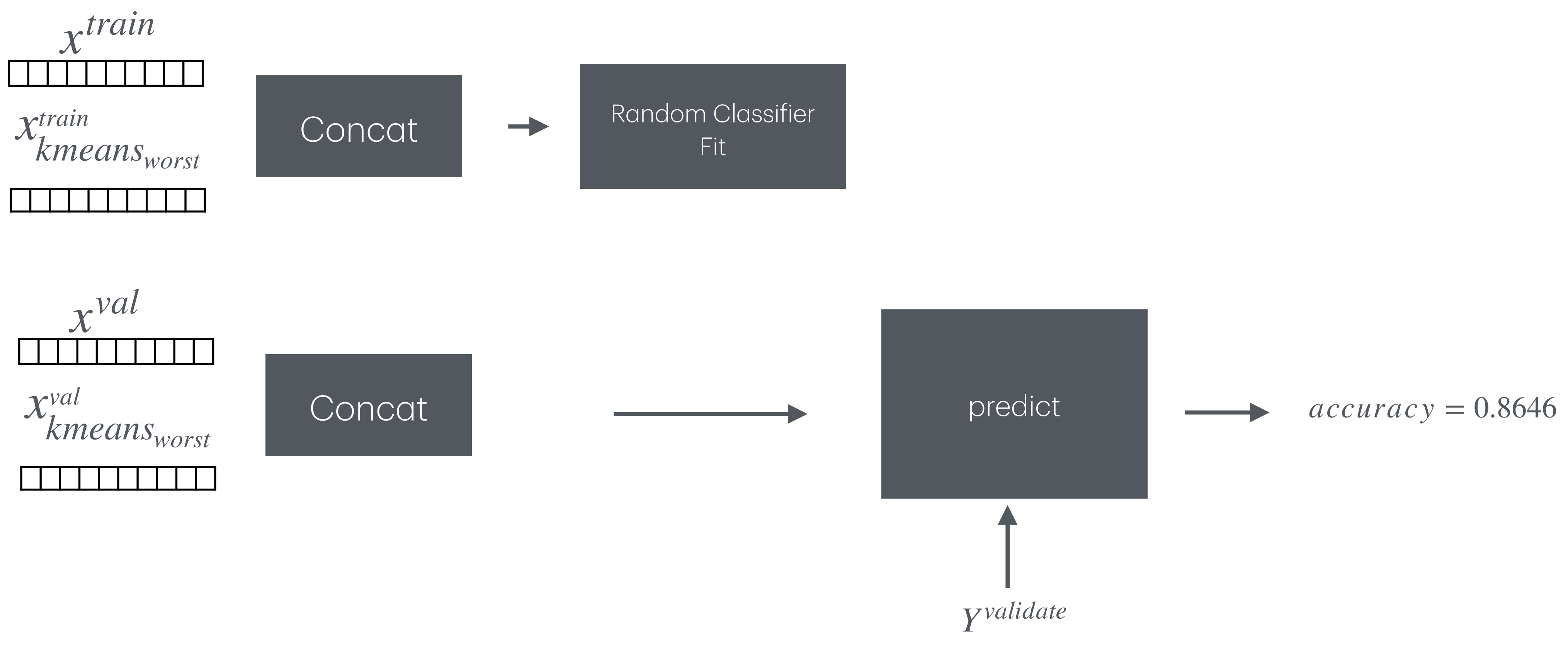


Pipeline Exercise 11: Dimension Reduction

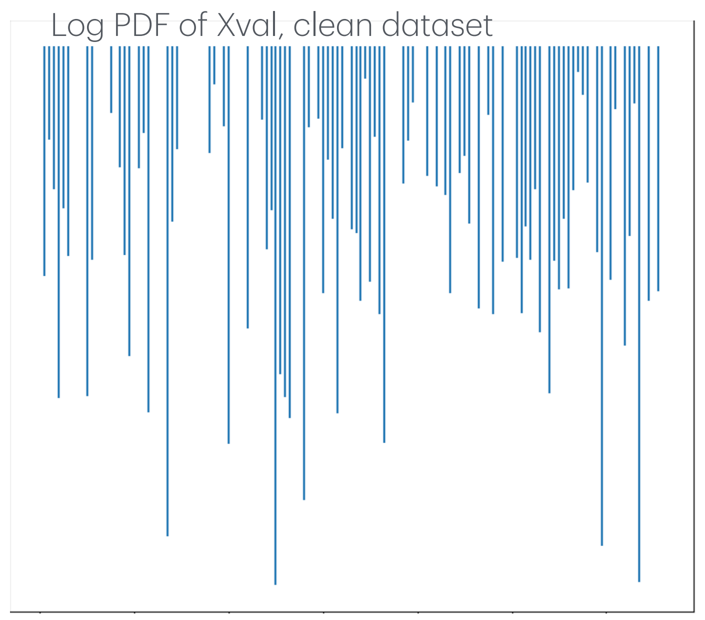


Pipeline Exercise 11: Dimension Reduction

Method 3

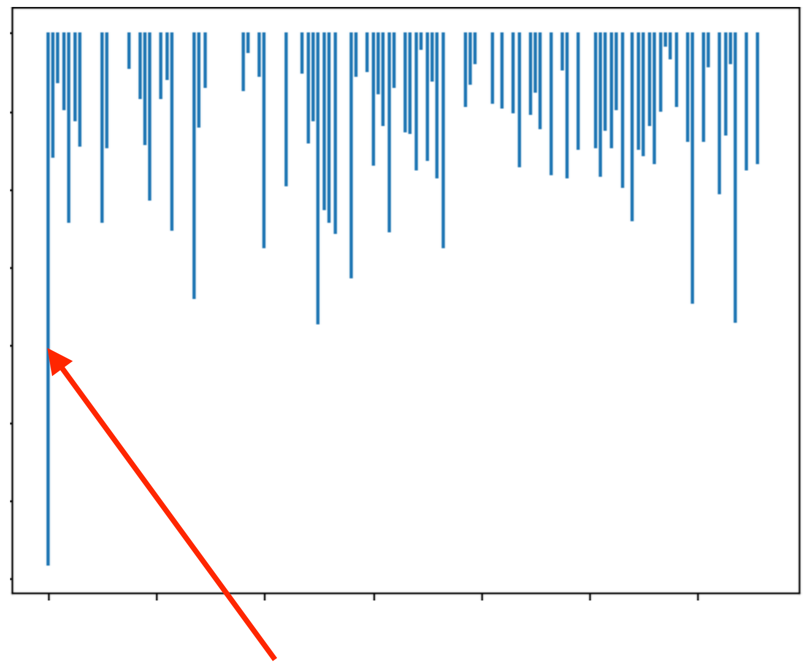


Exercise 12: Anomaly Detection

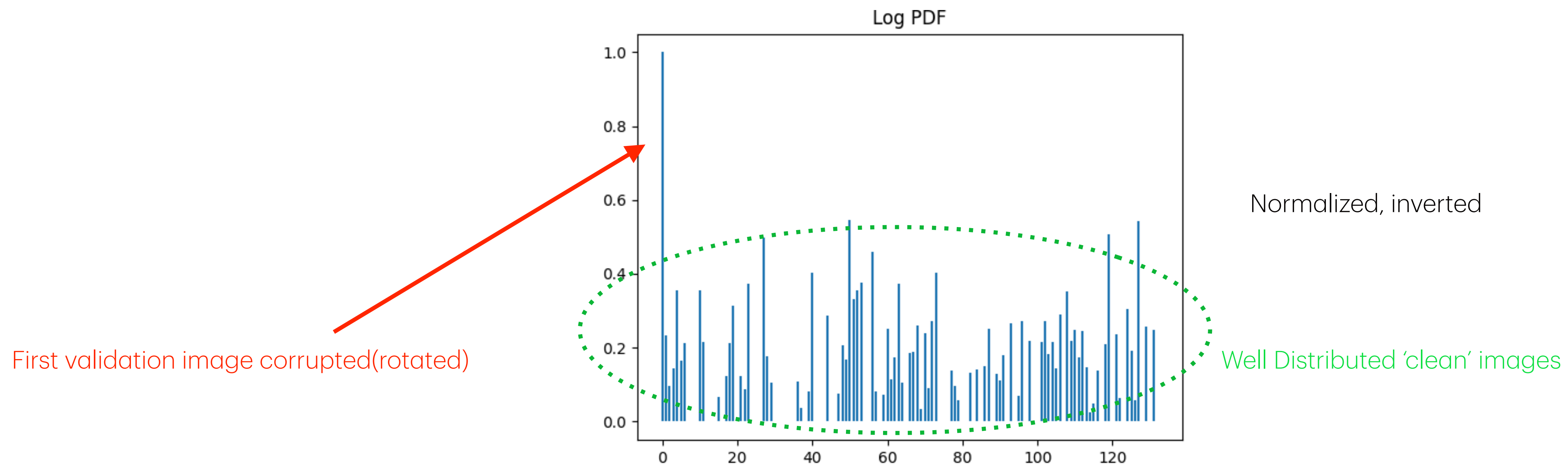


On average data comparable and distributed well

Log PDF of Xval with first image corrupted



First validation image corrupted(rotated)

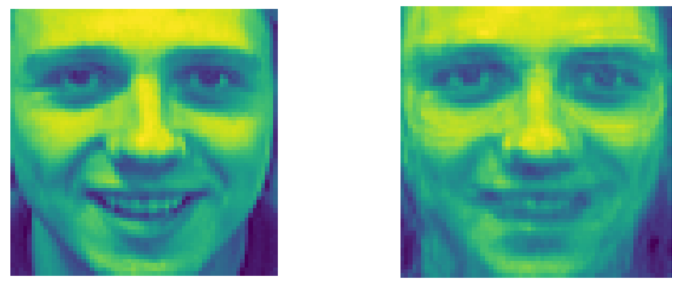


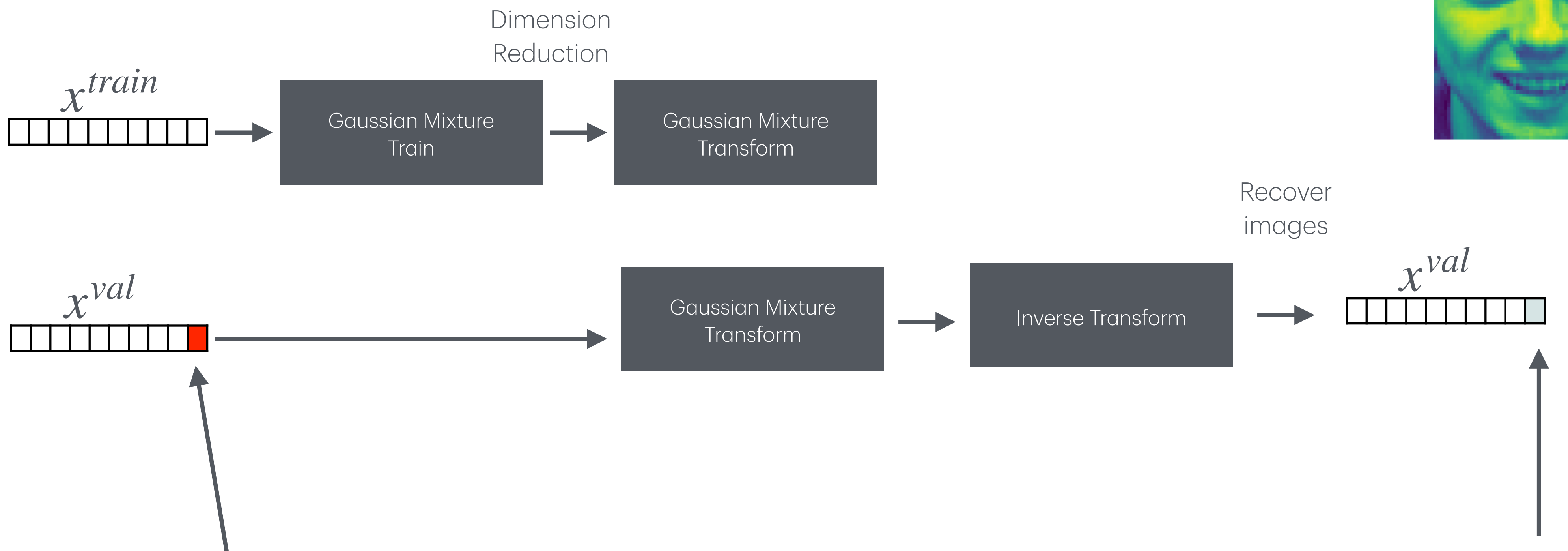
Normalized, inverted

Exercise 13: Reconstruction

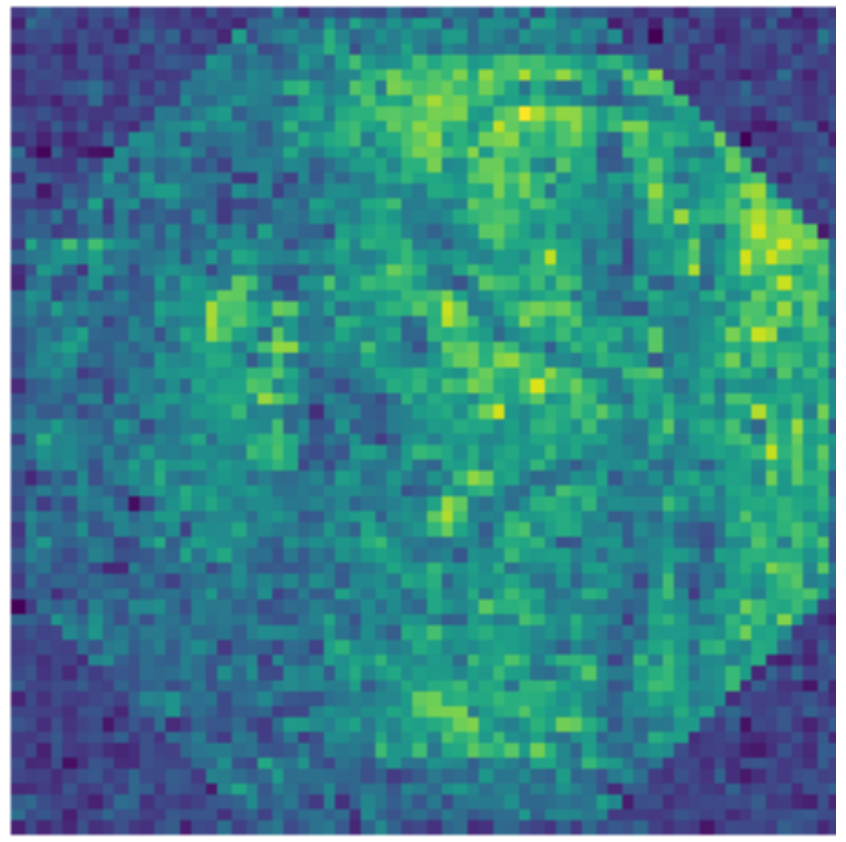
Note : recovered Images are 'good' not' perfect'

Perfect Recovered

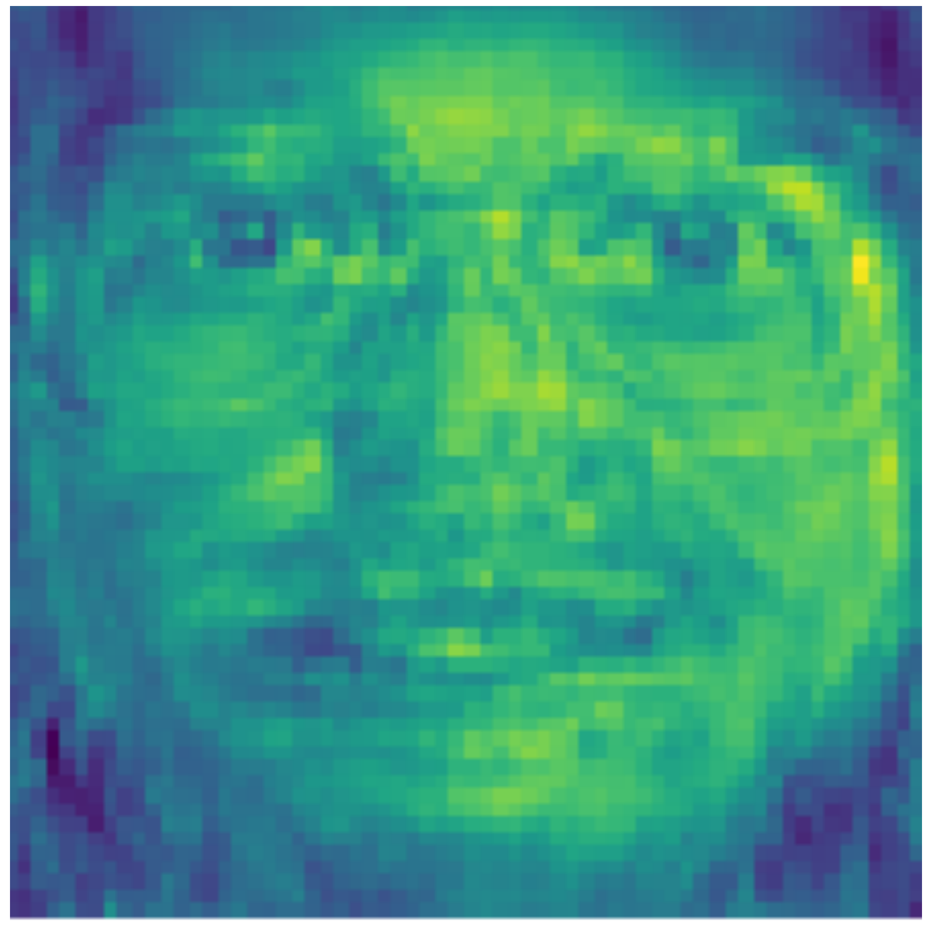




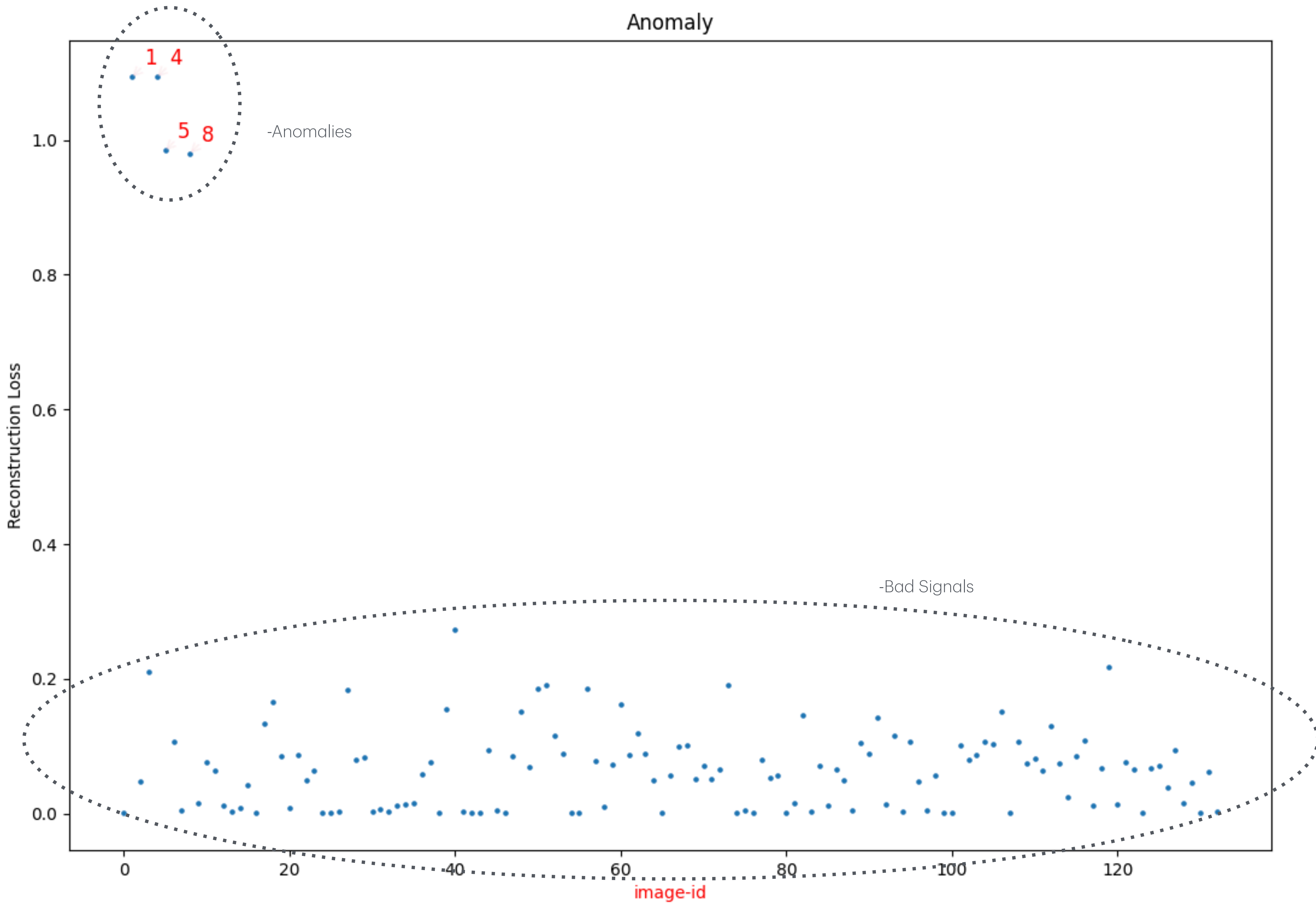
Damaged image



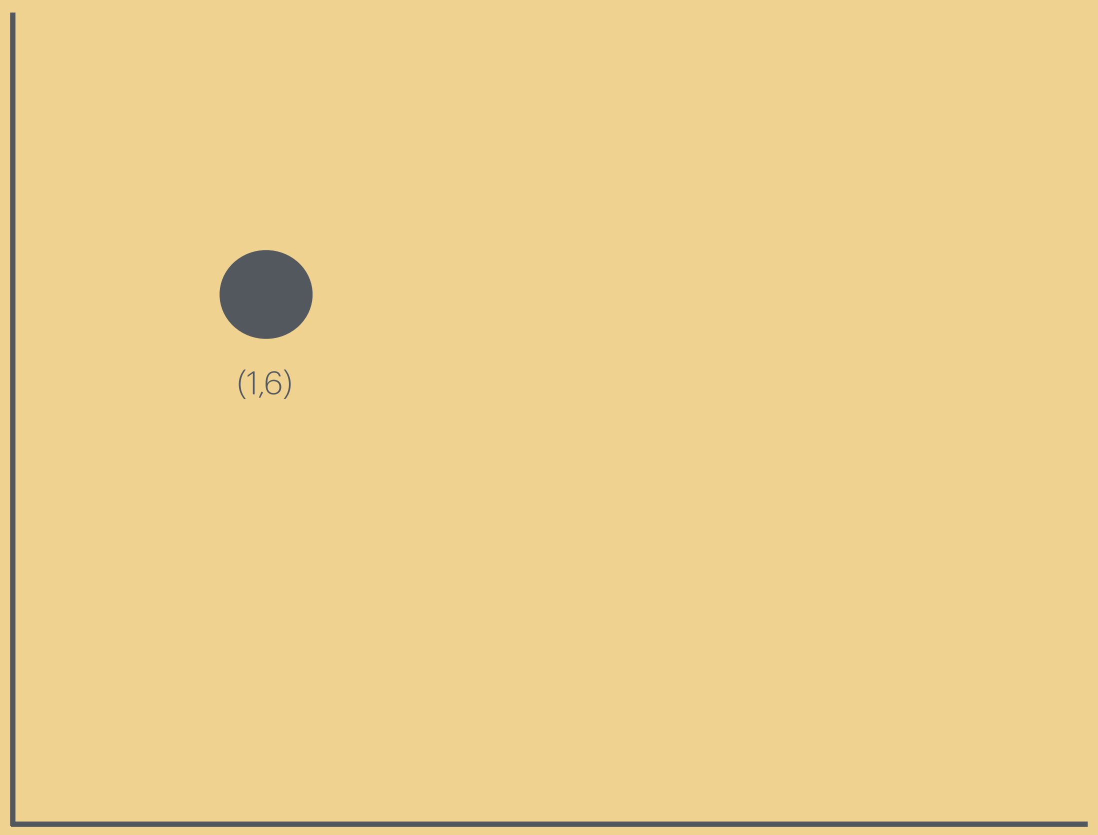
Reconstructs to the dataset in trained dataset:
- Tries to reconstruct a normal face

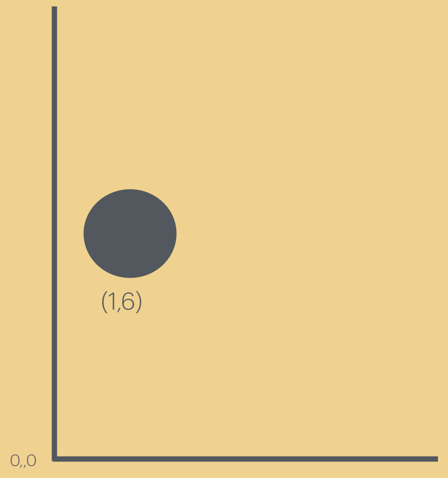


Four images in the validation were distorted. After transforming and taking the inverse the reconstruction set was measured for all datasets

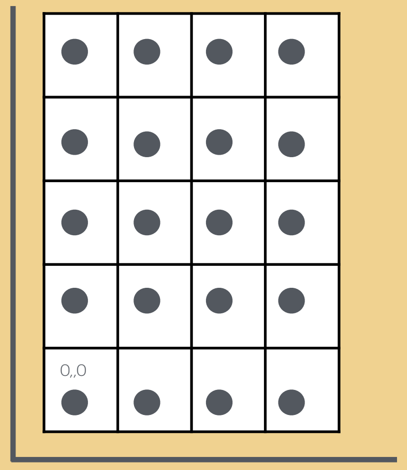


Note: Python: Annotation

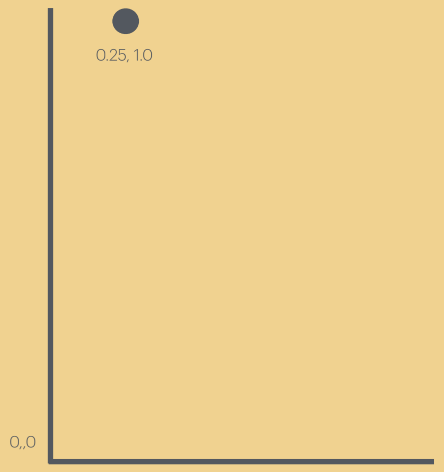




xycoord
- points from lower left



xycoord
- pixels from lower left



xycoord
- fraction