Unsupervised Learning Algorithms

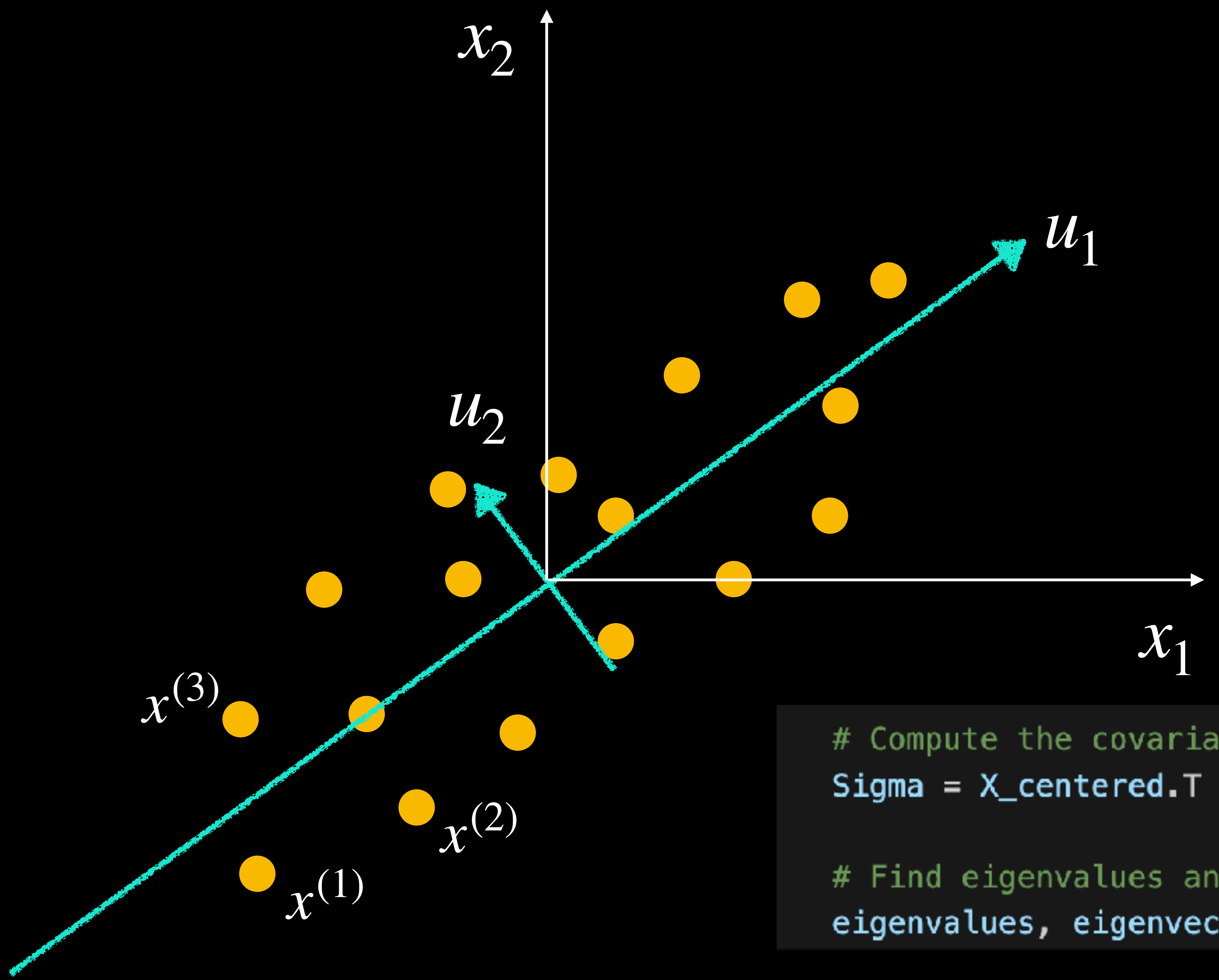
Prepared by: Joseph Bakarji

Dimensionality Reduction

Principal Component Analysis

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
…	…




```python
# Compute the covariance matrix
Sigma = X_centered.T @ X_centered

# Find eigenvalues and eigenvectors of the covariance matrix
eigenvalues, eigenvectors = np.linalg.eig(Sigma)
```

or

```python
U, S, Vt = np.linalg.svd(data_centered)
```

Dimensionality Reduction

Principal Component Analysis

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
…	…

Scikit-Learn

```
class sklearn.decomposition.PCA(n_components=None, *, copy=True, whiten=False,
svd_solver='auto', tol=0.0, iterated_power='auto', n_oversamples=10,
power_iteration_normalizer='auto', random_state=None)
```

svd_solver : *{'auto', 'full', 'covariance_eigh', 'arpack', 'randomized'}, default='auto'*

"auto" :

The solver is selected by a default 'auto' policy is based on `X.shape` and `n_components`: if the input data has fewer than 1000 features and more than 10 times as many samples, then the "covariance_eigh" solver is used. Otherwise, if the input data is larger than 500x500 and the number of components to extract is lower than 80% of the smallest dimension of the data, then the more efficient "randomized" method is selected. Otherwise the exact "full" SVD is computed and optionally truncated afterwards.

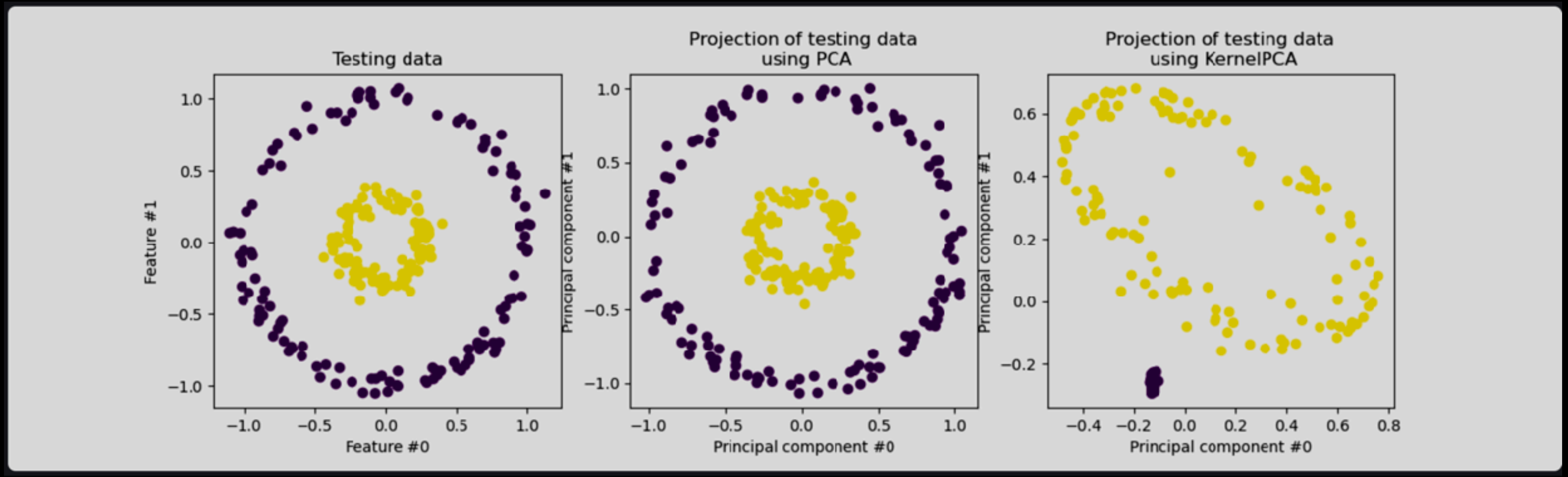
Dimensionality Reduction

Kernel PCA

Extension of PCA which achieves non-linear dimensionality reduction through the use of kernels

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
…	…



Define a nonlinear feature space through a kernel

$$\phi(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ \vdots \end{bmatrix}$$

→ **Perform PCA on the new space**

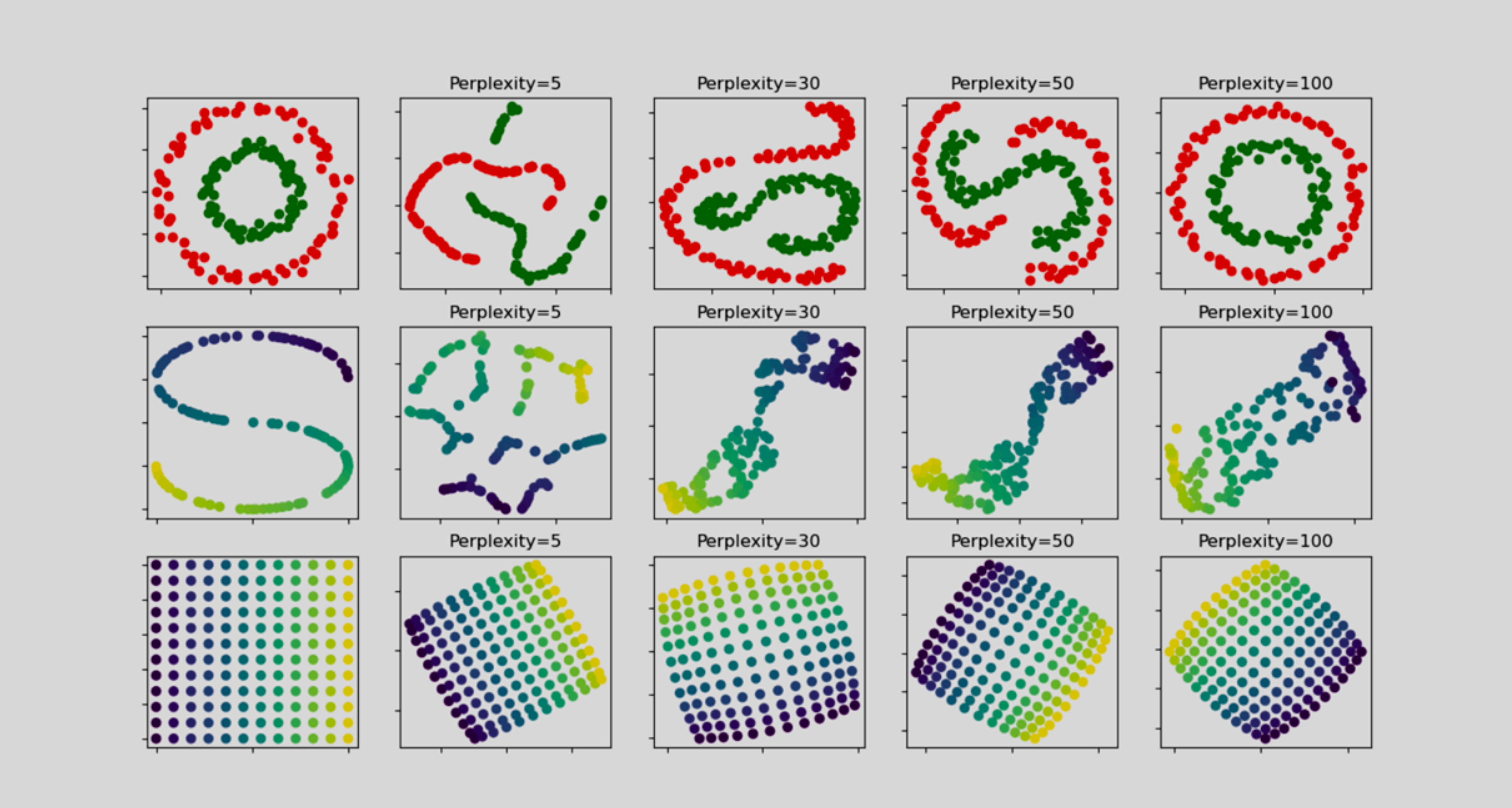
Dimensionality Reduction

t-Distributed Stochastic Neighbor Embedding (t-SNE)

Represents high-dimensional data in a lower-dimensional space while preserving the relationships between data points

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
…	…



https://scikit-learn.org/1.5/auto_examples/manifold/plot_t_sne_perplexity.html#sphx-glr-auto-examples-manifold-plot-t-sne-perplexity-py

Dimensionality Reduction

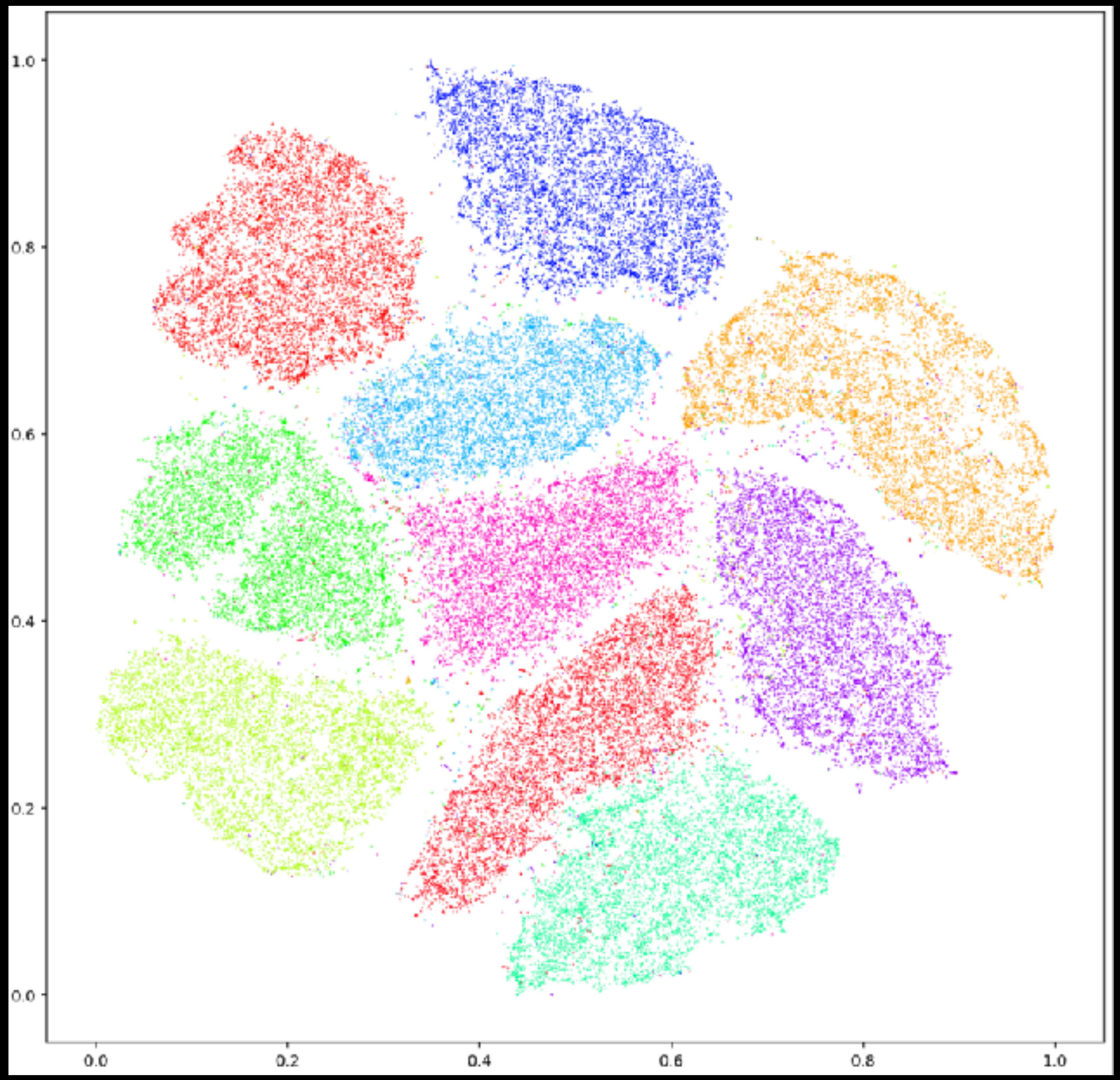
t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE of the MNIST data

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
…	…

→



Dimensionality Reduction

t-Distributed Stochastic Neighbor Embedding (t-SNE)

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
…	…

The similarity between two data points, x_i & x_j is calculated using a conditional probability

$$P(x_j | x_i) = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}\right)}$$

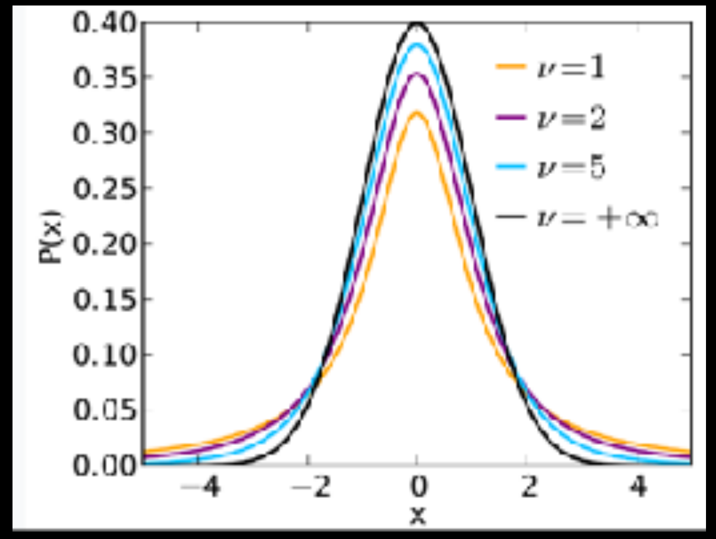
Then a joint distribution $P(x_i, x_j)$ is created by the mean

$$P(x_i, x_j) = [P(x_i | x_j) + P(x_j | x_i)]/2n$$

In lower-dimensions (2-3D), the joint distribution between points in the reduced space is given by

$$Q(x_i, x_j) = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

Which is a Student's t-distribution



https://scikit-learn.org/1.5/auto_examples/manifold/plot_t_sne_perplexity.html#sphx-glr-auto-examples-manifold-plot-t-sne-perplexity-py

Dimensionality Reduction

t-Distributed Stochastic Neighbor Embedding (t-SNE)

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
…	…

$$P(x_j \mid x_i) = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}\right)}$$

$$P(x_i, x_j) = [P(x_i \mid x_j) + P(x_j \mid x_i)]/2n$$

$$Q(x_i, x_j) = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

"Distance" between them is minimized using gradient descent

$$\mathrm{KL}(P \| Q) = \sum_{i \neq j} P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

The Kullback-Leibler (KL) divergence is Often used to minimize the distance between two distributions

https://scikit-learn.org/1.5/auto_examples/manifold/plot_t_sne_perplexity.html#sphx-glr-auto-examples-manifold-plot-t-sne-perplexity-py

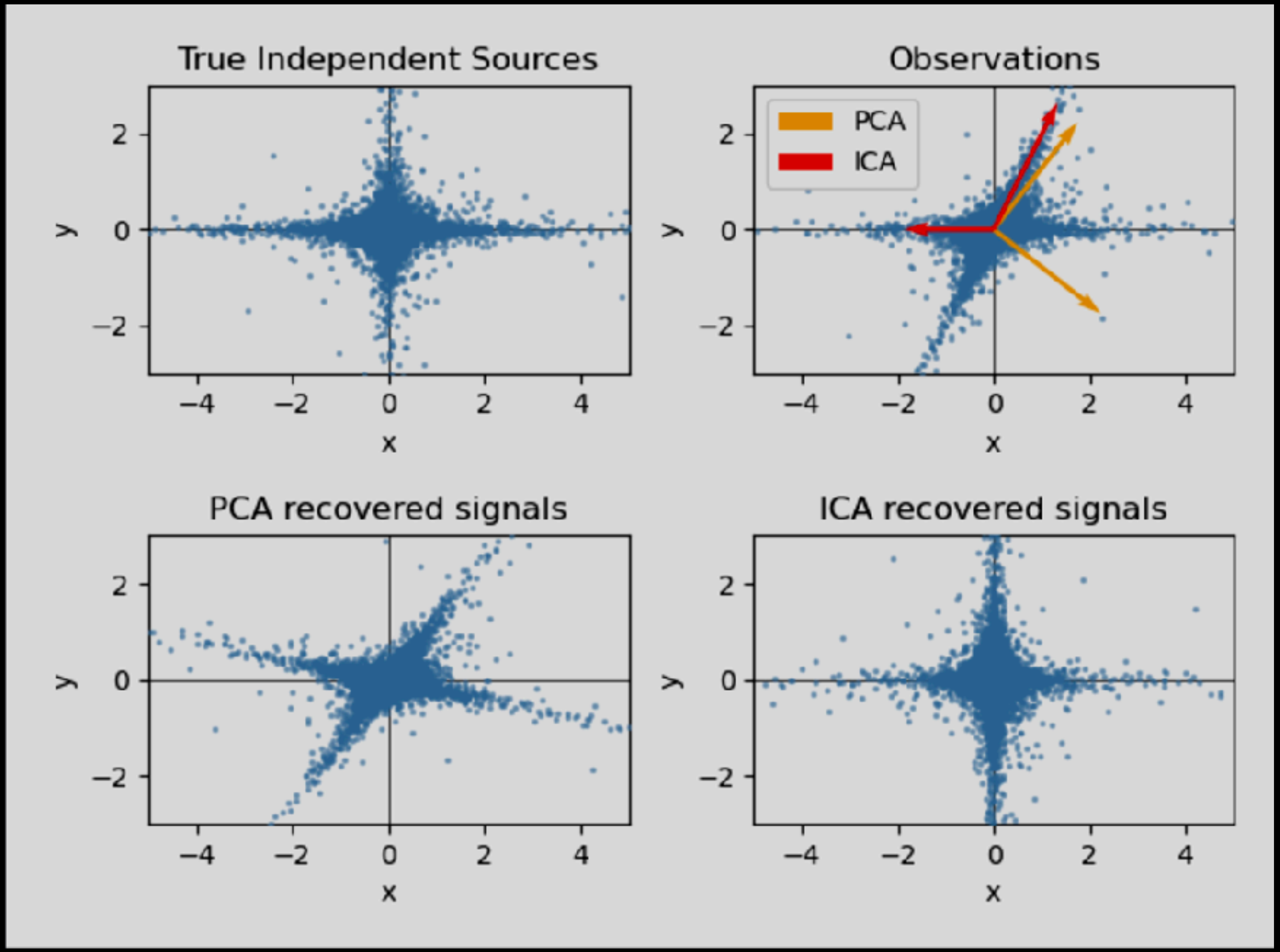
Dimensionality Reduction

Independent Component Analysis (ICA)

The goal of ICA is to express observed data X (A matrix of n observed signals) as a linear combination of statistically independent source signals

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
…	…



https://scikit-learn.org/stable/auto_examples/decomposition/plot_ica_vs_pca.html#sphx-glr-auto-examples-decomposition-plot-ica-vs-pca-py

Dimensionality Reduction

Independent Component Analysis (ICA)

The goal of ICA is to express observed data X (a matrix of n observed signals) as a linear combination of statistically independent source signals

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
…	…

The observed data X_i can be represented as a linear combination of the source signals S_j with the mixing matrix elements A_{ij}:

$$X_i = \sum_j A_{ij} S_j$$

This equation states that each observed variable X_i is a sum of contributions from each source S_j, weighted by the mixing coefficients A_{ij}. In matrix form, this can be written compactly as:

$$X = AS$$

where X is the vector of observed variables, A is the mixing matrix, and S is the vector of source signals. The goal of ICA is to find the inverse (or unmixing matrix W) such that:

$$S = WX$$

https://scikit-learn.org/stable/auto_examples/decomposition/plot_ica_vs_pca.html#sphx-glr-auto-examples-decomposition-plot-ica-vs-pca-py

Dimensionality Reduction

Factor Analysis

In unsupervised learning we only have a dataset $X = \{x_1, x_2, \ldots, x_n\}$. How can this dataset be described mathematically? A very simple `continuous latent variable` model for X is

$$x_i = W h_i + \mu + \epsilon$$

The vector h_i is called "latent" because it is unobserved. ϵ is considered a noise term distributed according to a Gaussian with mean 0 and covariance Ψ (i.e. $\epsilon \sim \mathcal{N}(0, \Psi)$), μ is some arbitrary offset vector. Such a model is called "generative" as it describes how x_i is generated from h_i. If we use all the x_i's as columns to form a matrix $\mathbf{X}$ and all the h_i's as columns of a matrix $\mathbf{H}$ then we can write (with suitably defined $\mathbf{M}$ and $\mathbf{E}$):

$$\mathbf{X} = W\mathbf{H} + \mathbf{M} + \mathbf{E}$$

In other words, we *decomposed* matrix $\mathbf{X}$.

If h_i is given, the above equation automatically implies the following probabilistic interpretation:

$$p(x_i | h_i) = \mathcal{N}(W h_i + \mu, \Psi)$$

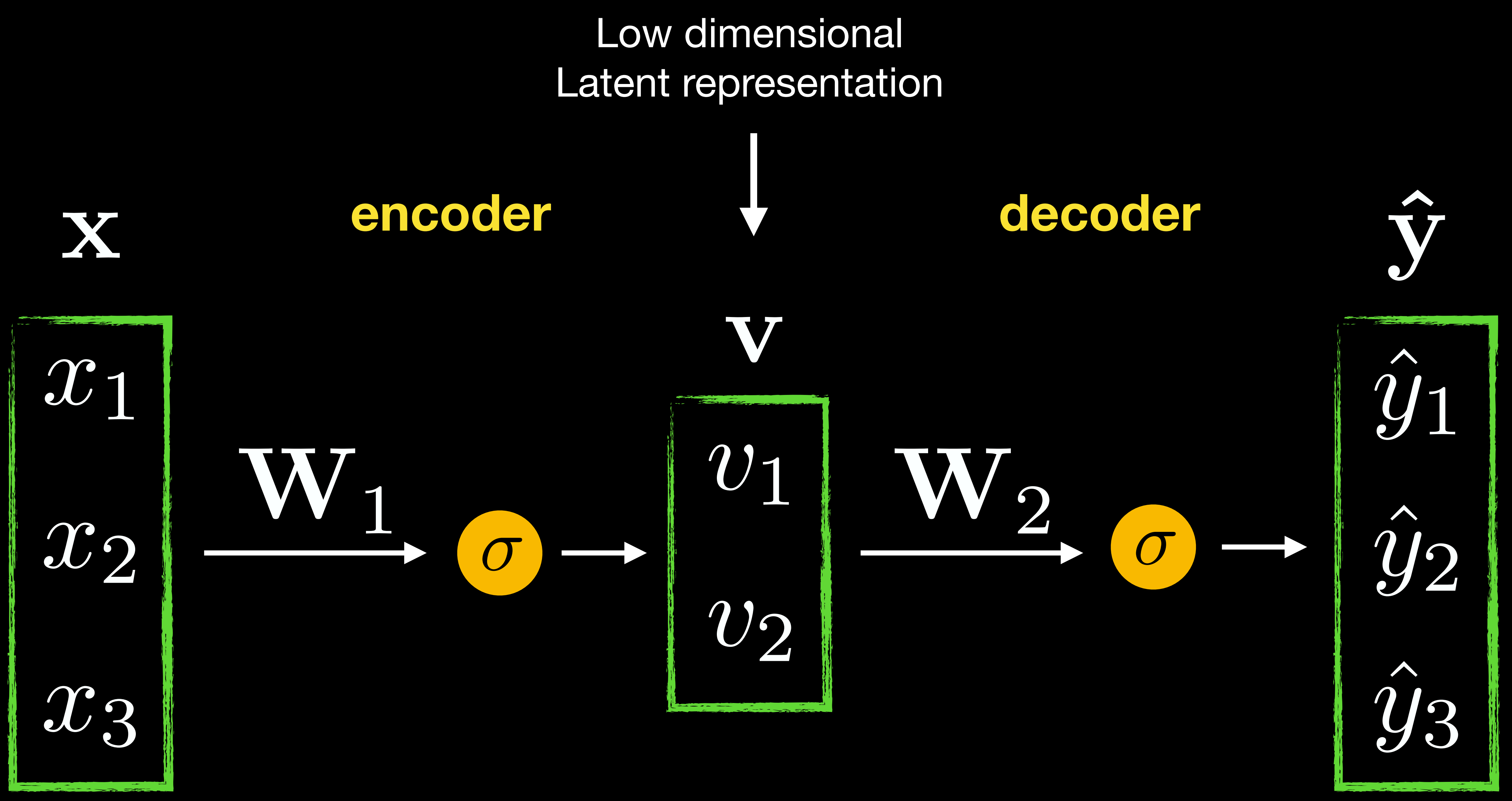
For a complete probabilistic model we also need a prior distribution for the latent variable h. The most straightforward assumption (based on the nice properties of the Gaussian distribution) is $h \sim \mathcal{N}(0, \mathbf{I})$. This yields a Gaussian as the marginal distribution of x:

$$p(x) = \mathcal{N}(\mu, WW^T + \Psi)$$

https://scikit-learn.org/stable/modules/decomposition.html#factor-analysis

Dimensionality Reduction

Neural Networks: Auto-encoders



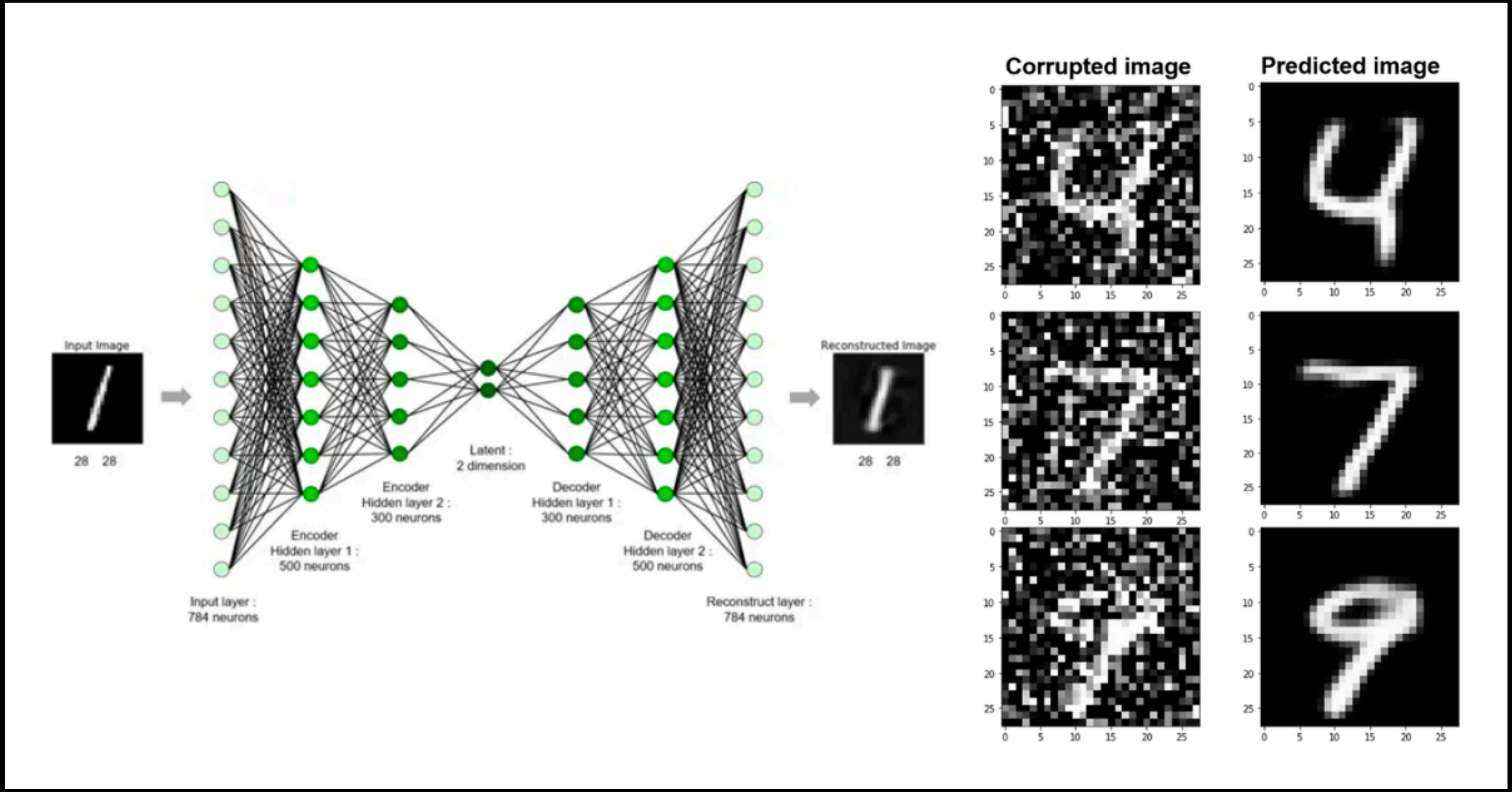
$$\mathcal{L} = \|f_{\mathbf{W}_1 \mathbf{W}_2}(\mathbf{x}) - \mathbf{x}\|^2$$

if $\sigma = I$, network reduced to SVD decomposition

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$$

Denoising

Neural Networks: Auto-encoders

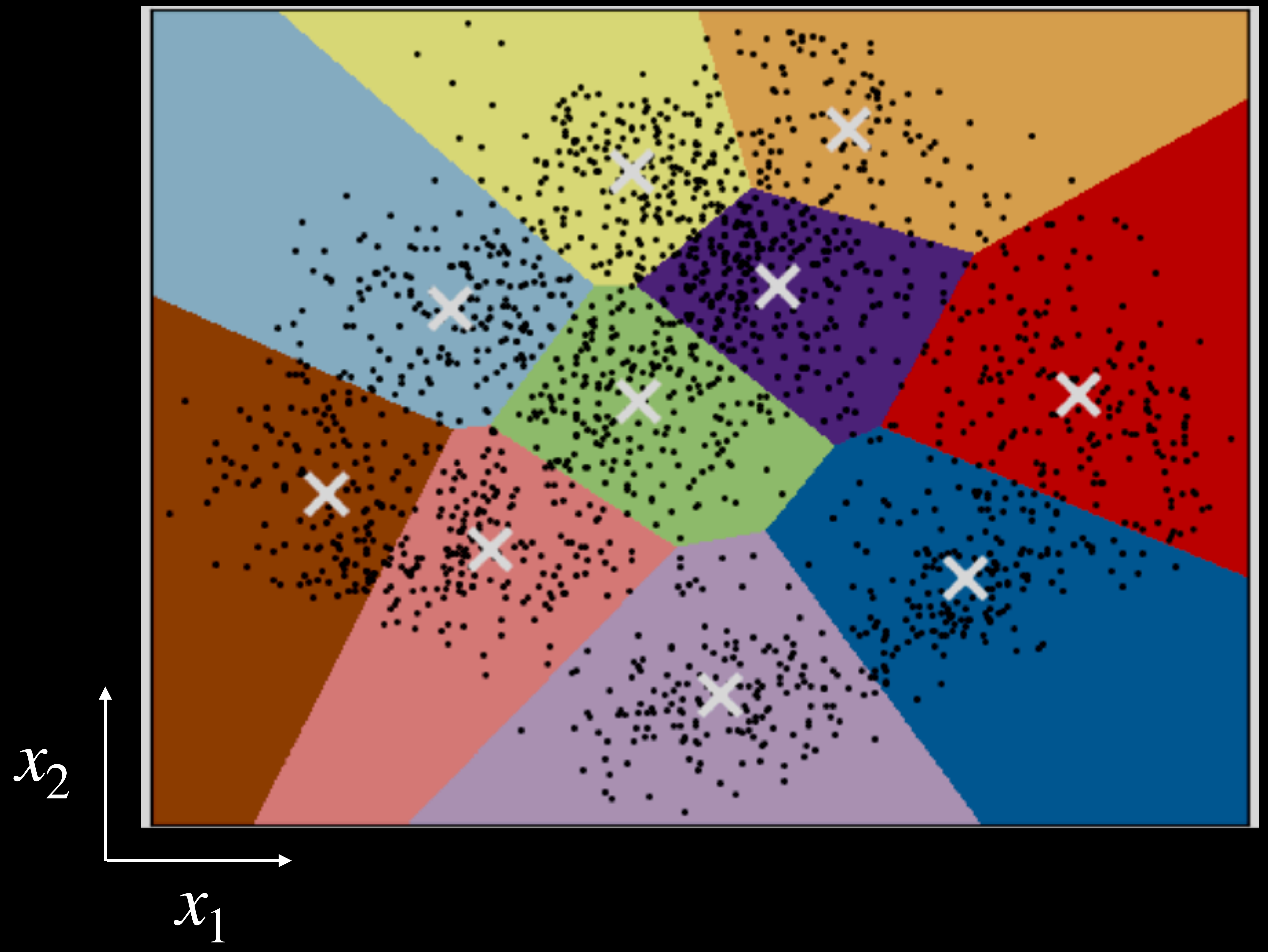


Clustering

K-Means

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
…	…

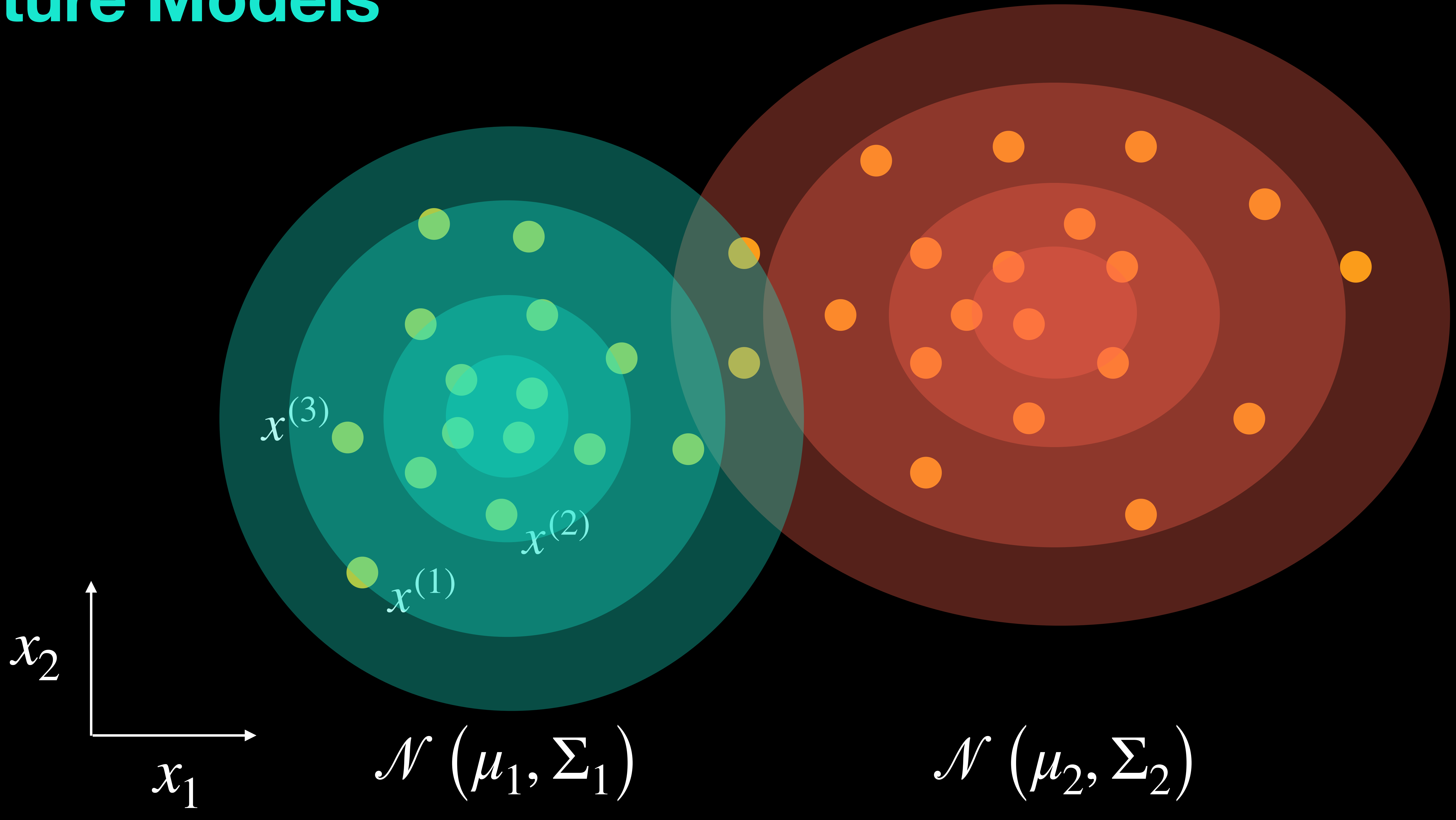


Clustering

Gaussian Mixture Models

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
...	...

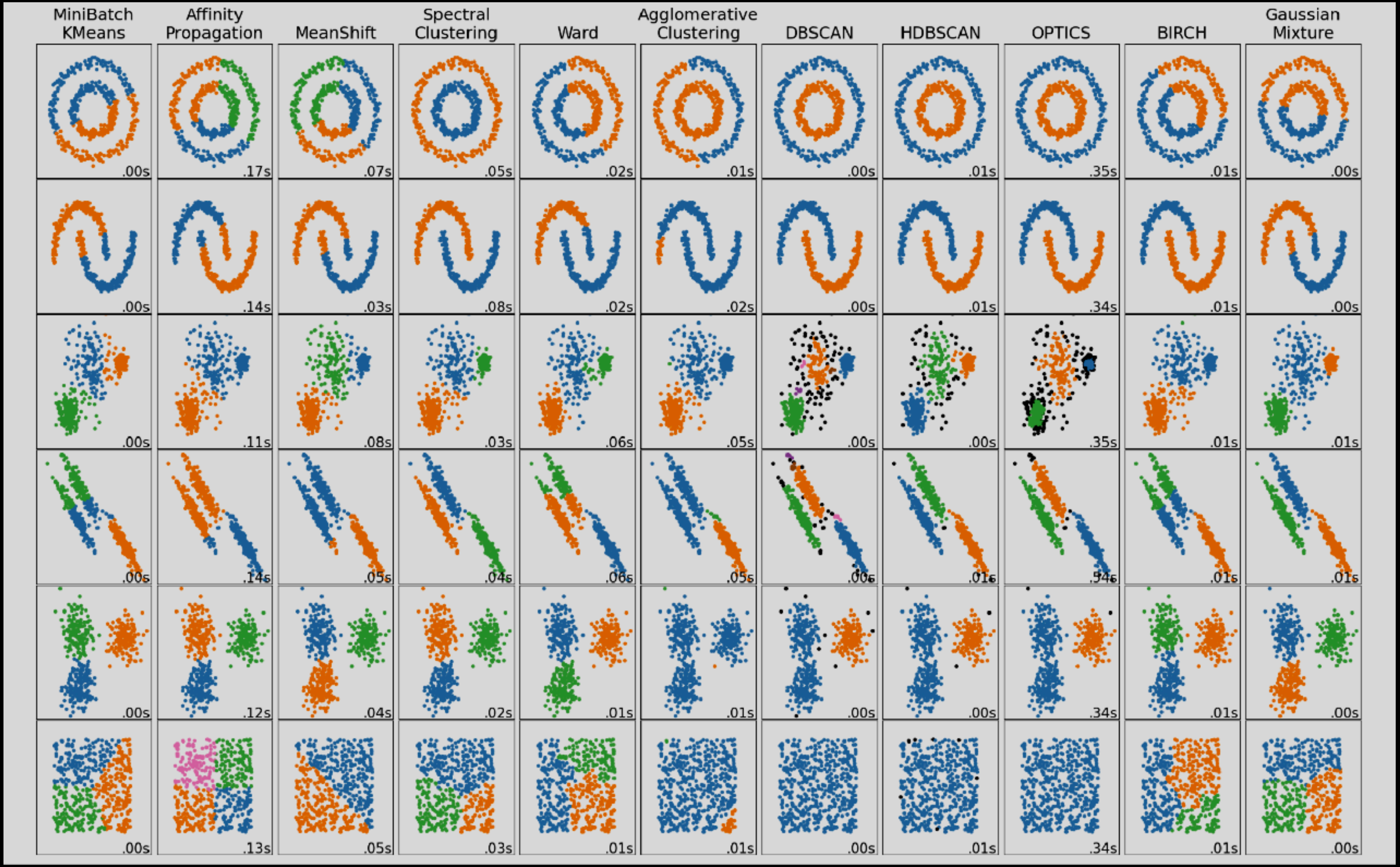


Clustering

A zoo of clustering methods

Dataset

x_1	x_2
1.2	1.2
3.2	5.4
4.3	6.4
3.2	5.4
…	…

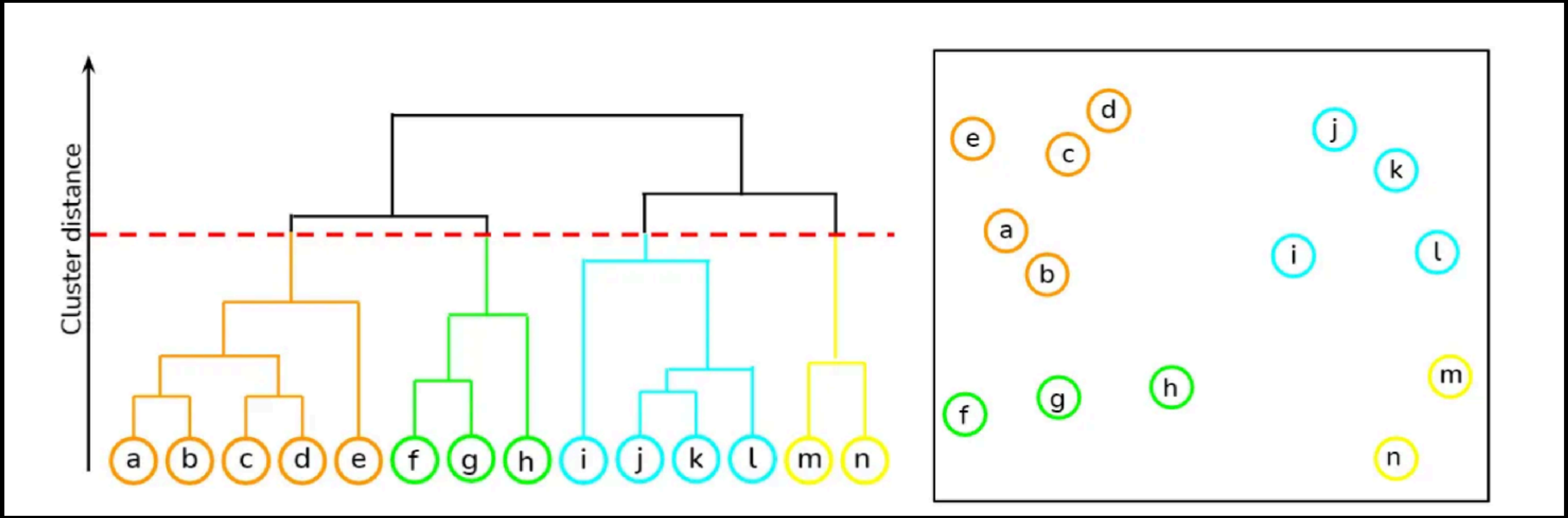


https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering

Clustering

Hierarchical Clustering

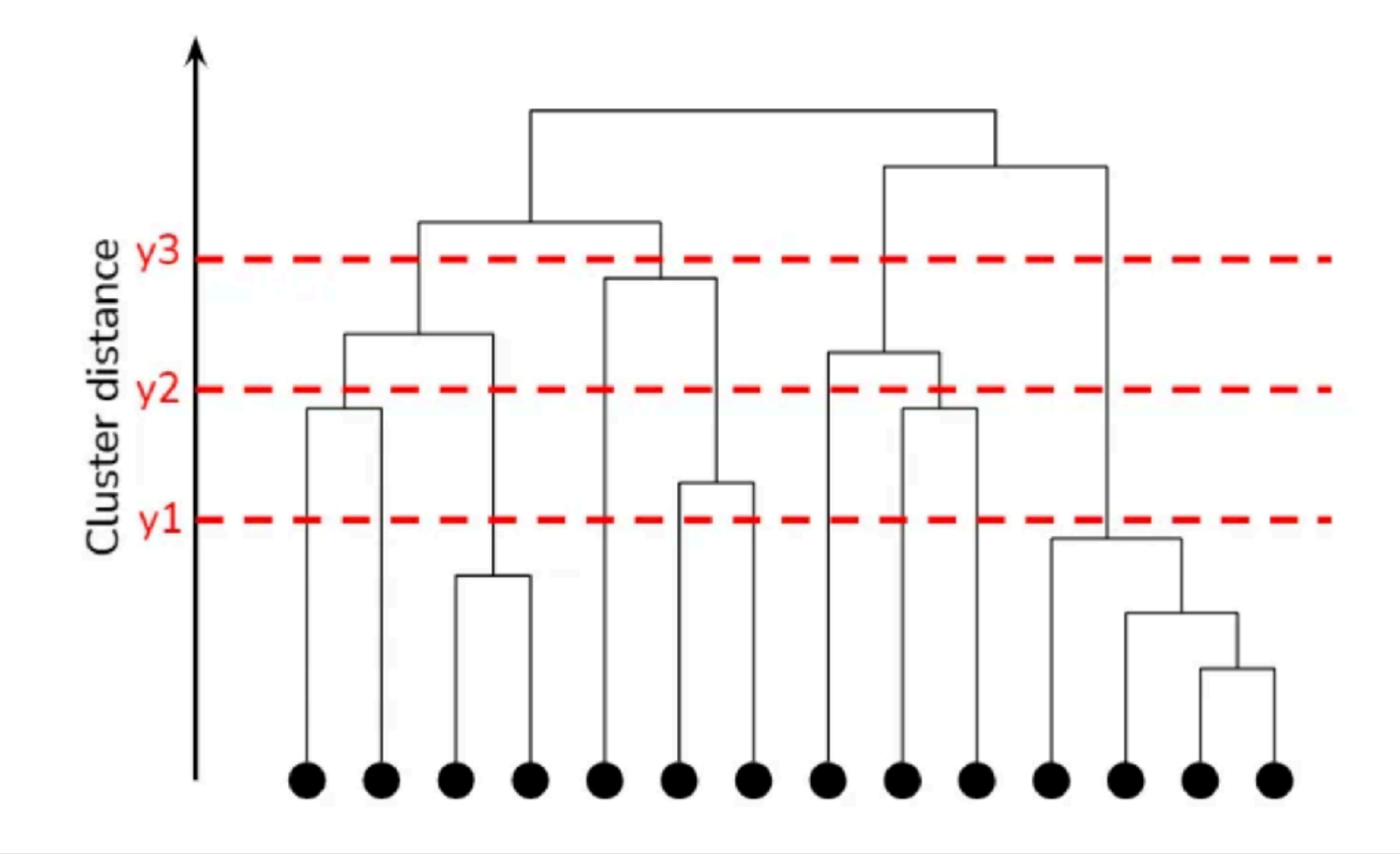
Dendrogram



https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8

Clustering

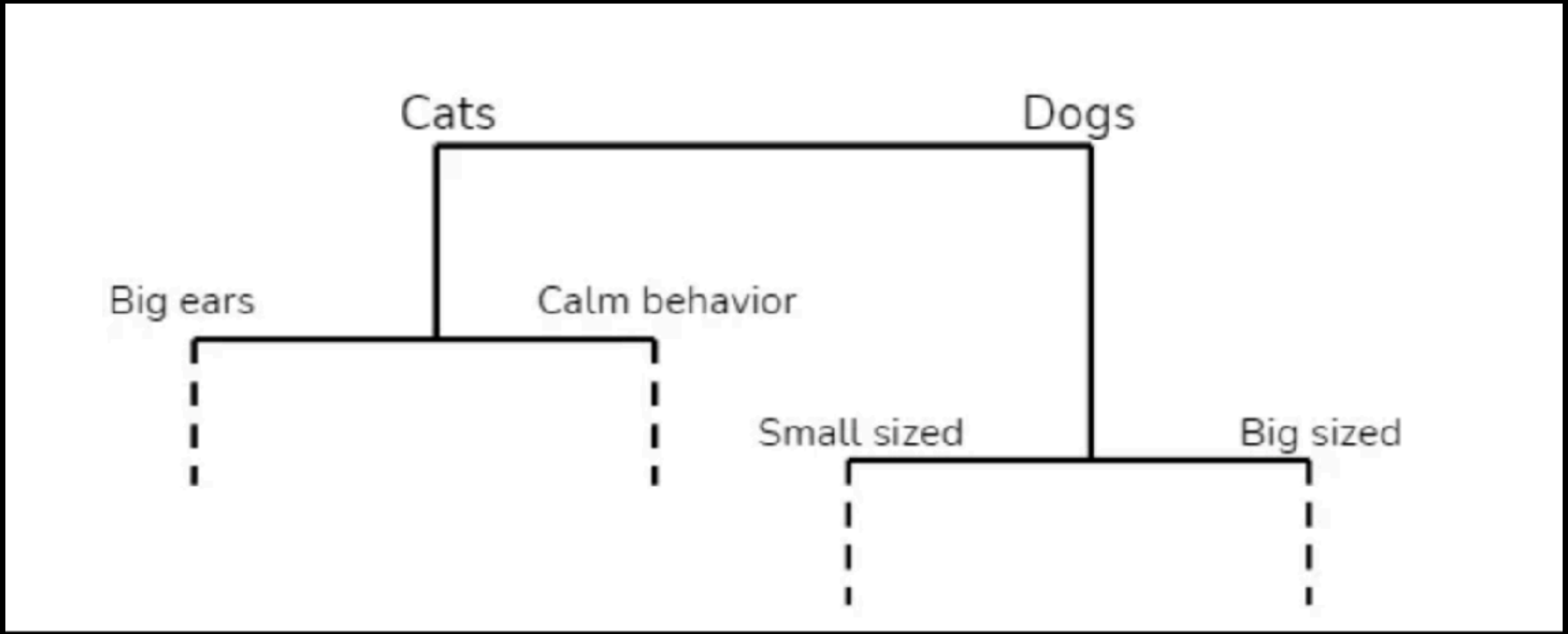
Hierarchical Clustering



https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8

Clustering

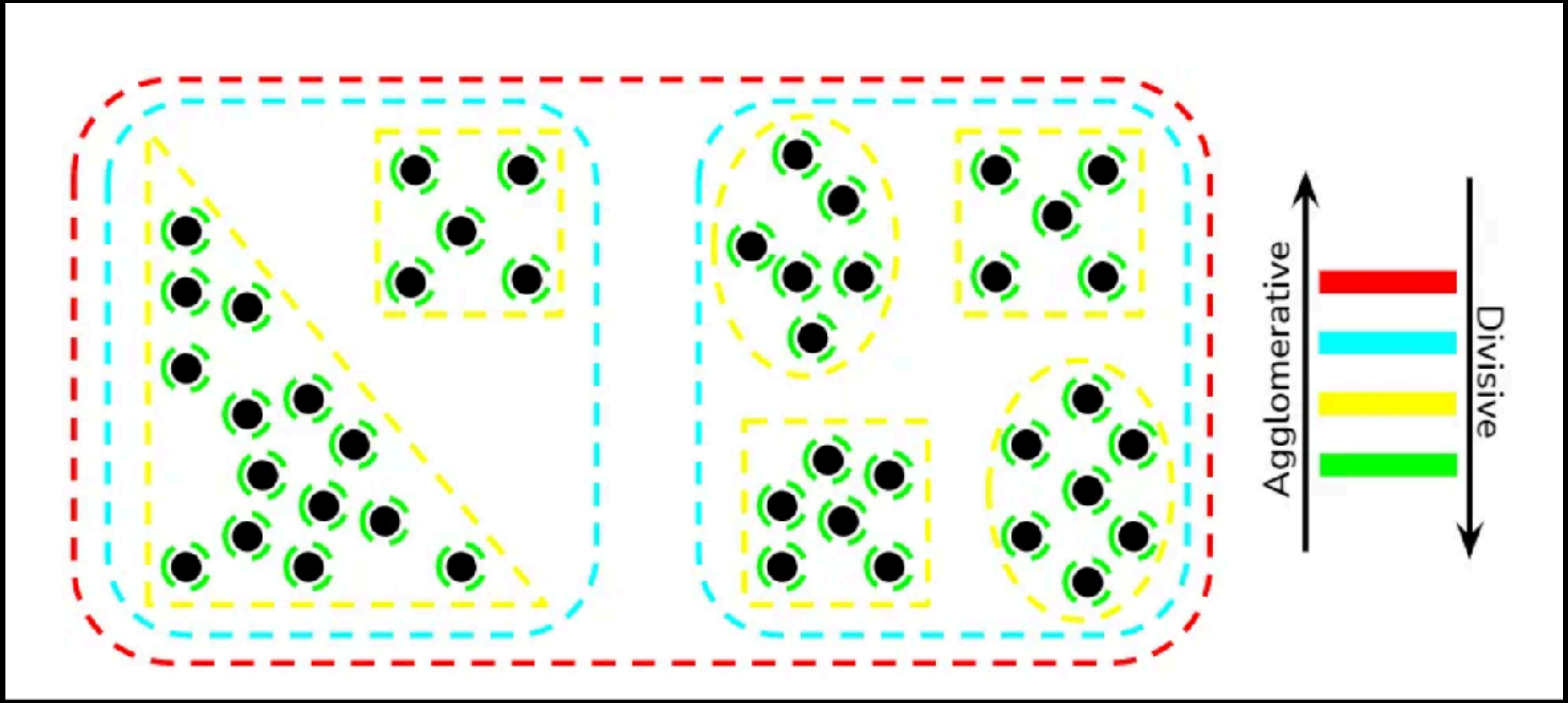
Hierarchical Clustering



https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8

Clustering

Hierarchical Clustering



https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8

Clustering

Hierarchical Clustering

Objective function for Ward's method

$$\sum_{C} \sum_{x \in C} ||x - \mu_C||^2$$

- C represents a cluster in the set of all clusters
- x is a data point within cluster C
- μ_c is the centroid (mean) of cluster C
- $||x - \mu_C||^2$ is the squared Euclidean distance between a point x and the centroid μ_c

https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8