Using Machine Learning Tools PG

Week 7 – Unsupervised Learning

COMP SCI 7317

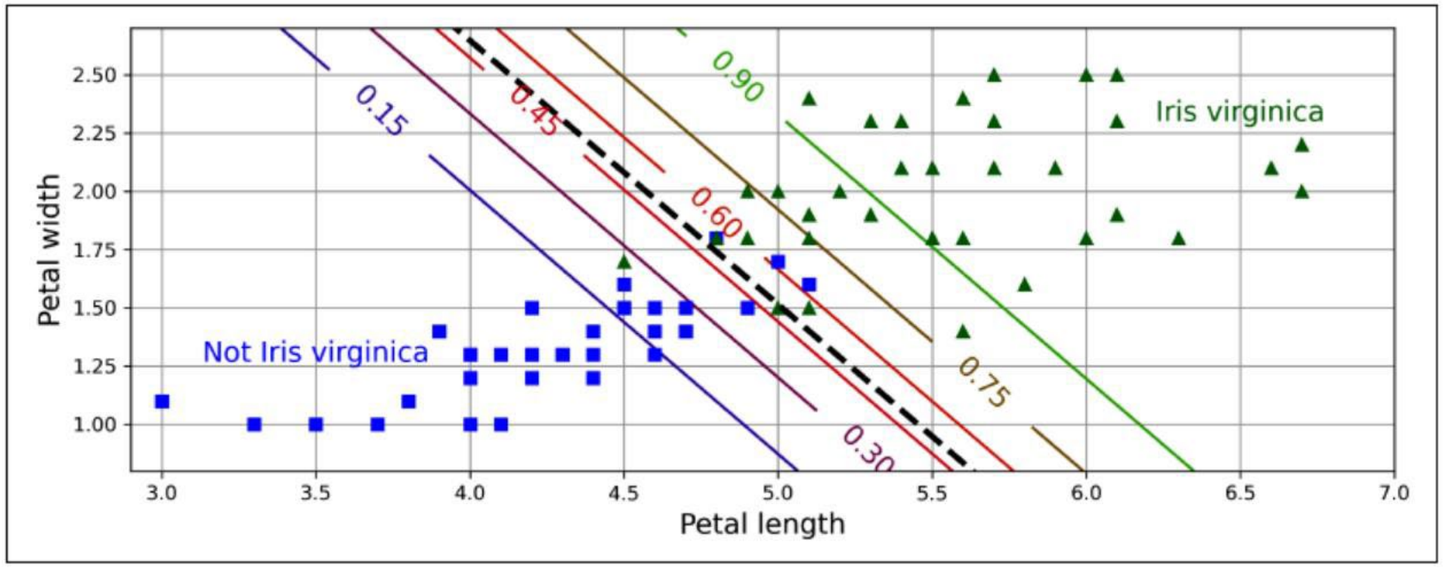
Trimester 2, 2024

From last week…

1. **Regularisation Continued**
   - L1, L2, Elastic Net
2. **Logistic Regression**
   - Logistic & Logit Regression
   - Training a logistic model
   - Softmax
3. **Support Vector Machines (SVM)**
   - Linear SVM - Hard Margin
   - Linear SVM - Soft Margin
   - Kernel trick

Logistic regression: Decision boundary

- **Decision boundary:** Helps decide which class a new data point belongs to, based on its features.
  - In logistic regression, the decision boundary is linear/monotonic, meaning it creates a straight line (in 2D) or a flat plane (in 3D) to separate classes.
  - Normal to the decision boundary is direction in which the probability of being in a specific class increases most (due to s-shaped sigmoid).

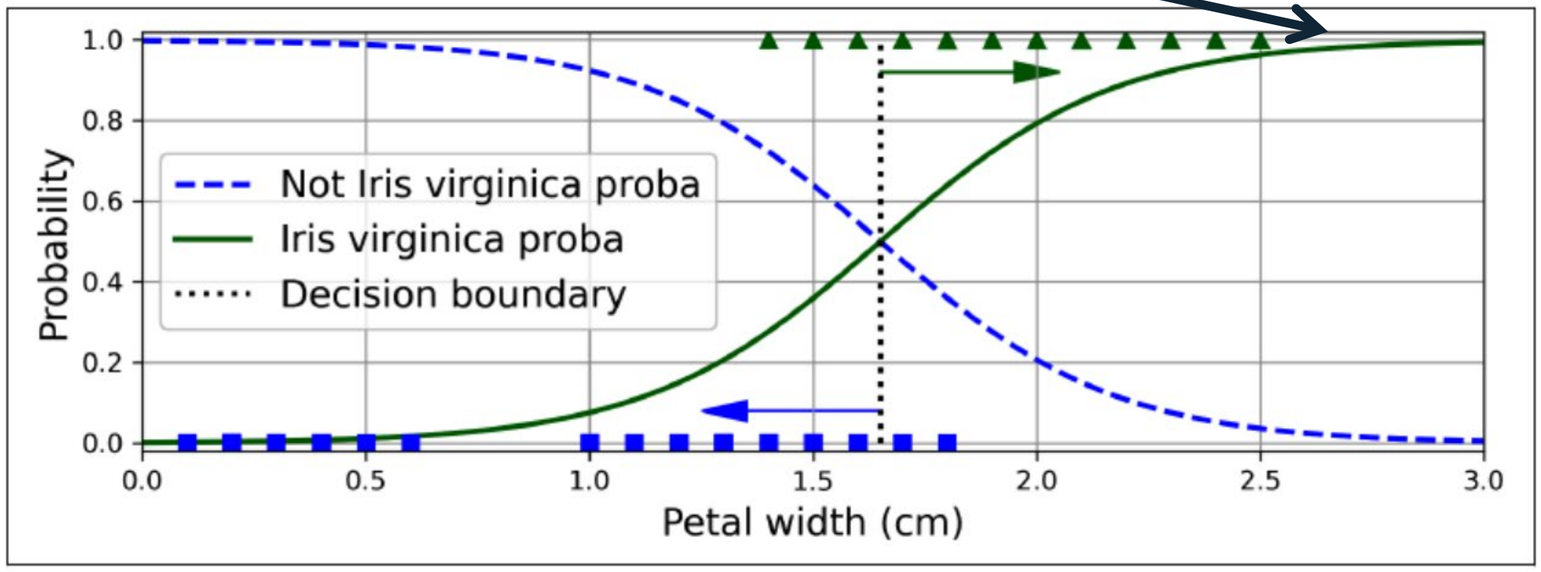


- **At the boundary:** No change in probability.
- **Away from the boundary:** Change in probability is more rapid.
- **Furthest from the boundary:** probability plateaus as it approaches 0 or 1.

from GeronHands on Machine Learning

Logistic regression: Decision boundary

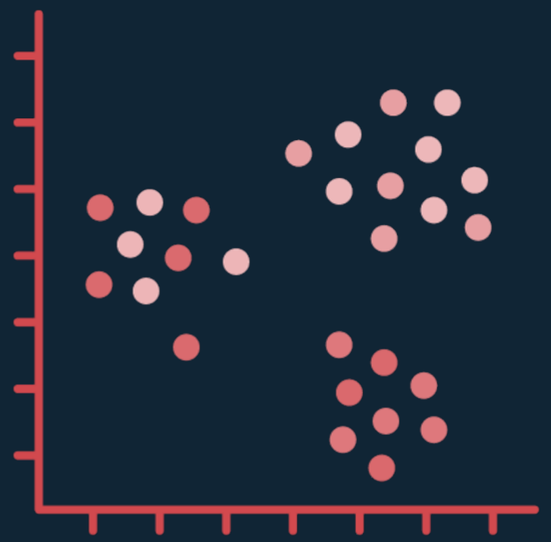
- **Probability prediction:** Graph below illustrates how the predicted probability changes with petal width.
- The intersection at p=0.5 is the decision boundary.
- This graph shows the rate of change near the decision boundary is the largest (probabilities plateau as you move further away).



from GeronHands on Machine Learning

This week

1. **Clustering**
   - K-Means
   - Gaussian Mixture Models
2. **Dimensionality Reduction**
   - PCA, tSNE
   - Other feature selection methods
3. **Data visualisations using unsupervised learning approaches**



Clustering

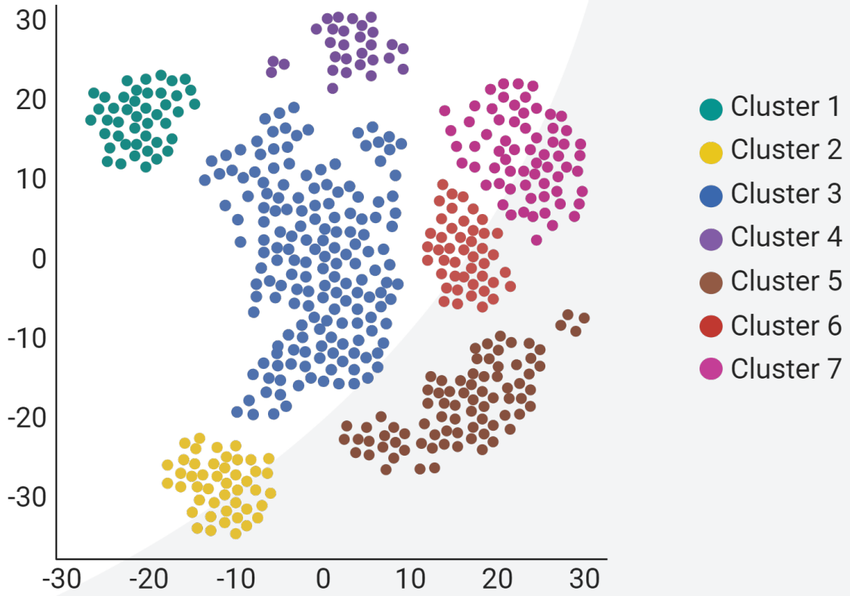
What is unsupervised learning?

Unsupervised learning: In the context of machine learning, **unsupervised** learning means:

- Training data includes ***does not include*** labels or values where we know the ground truth of the thing we are trying to predict

Common applications of unsupervised learning include:

- **Clustering**
  - Marketing, sales etc.
- **Dimension Reduction**
  - Visualisation
  - Pre-processing
- **Anomaly detection**
  - Fraud detection
  - Security
  - Pathologies in medical data



Clustering

A **type of unsupervised learning approach**, which aims to:

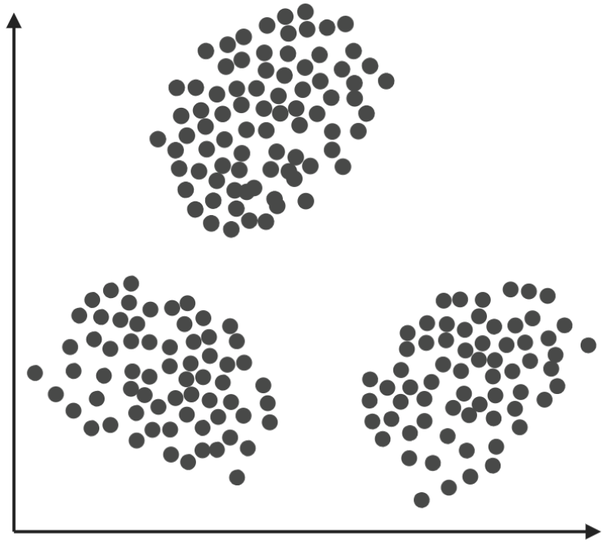
- Organise unlabelled data into similar groups called ‘clusters’.
- Assume no *a priori knowledge* about the grouping of the data instances given.
- Due to historical reasons, clustering is sometimes considered synonymously with unsupervised learning.

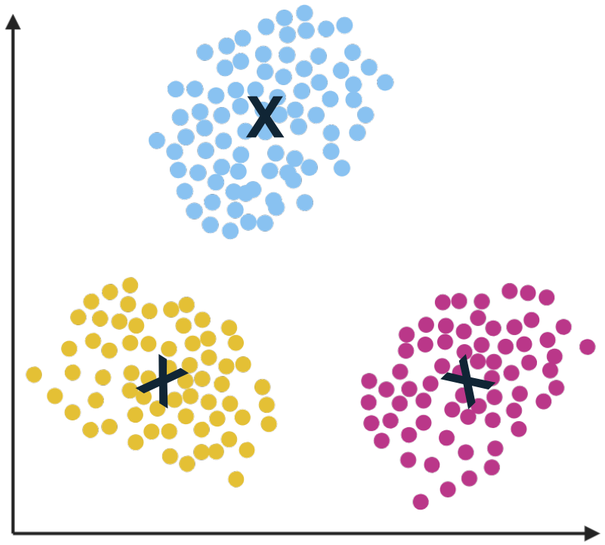
Types of algorithms include:

- **Partitioning Clustering:** K-Means, K-Medoids
- **Hierarchical Clustering:** Agglomerative, Divisive
- **Density-Based Clustering:** DBSCAN, OPTICS
- **Model-Based Clustering:** Gaussian Mixture Models

K-Means

- **K-Means** is a partitioning clustering algorithm that aims to divide n data points into k clusters.
- Each cluster is represented by the centroid (mean) of the data points within the cluster.





X = centroid
n = # data points
k = # clusters

THE UNIVERSITY of ADELAIDE
150 YEARS

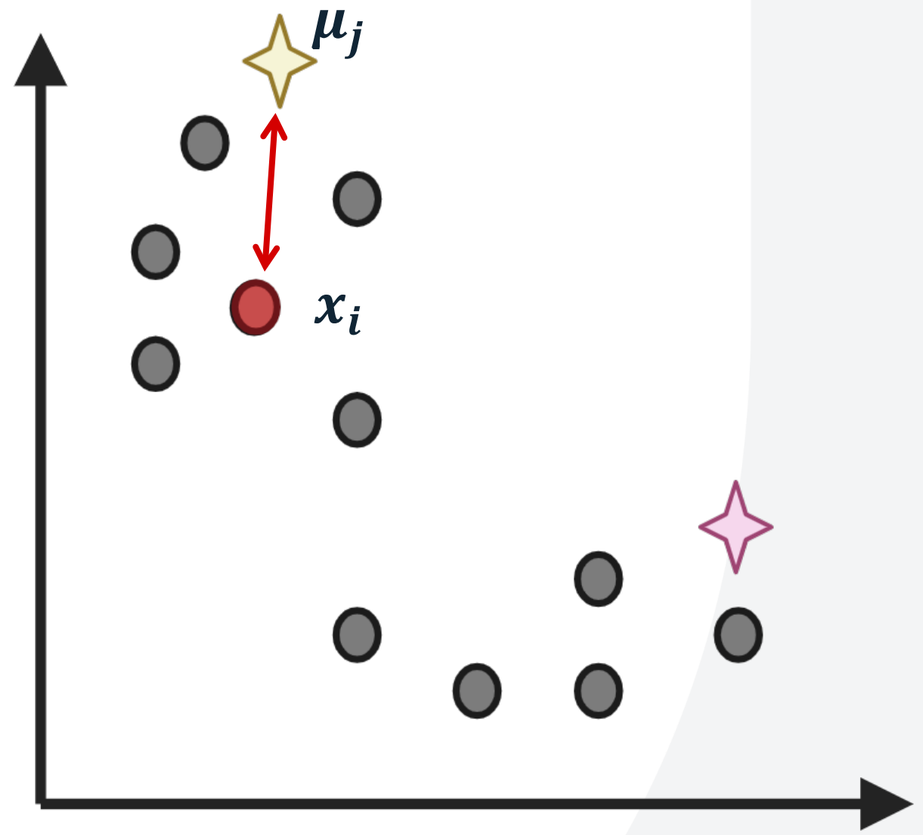
K-Means

- **Algorithm steps**

1. **Initialisation:** Randomly select k initial centroids (initial guesses for the centers of each cluster).

2. **Assignment:** Assign each data point, x_i to the nearest centroid by minimising the squared Euclidean distance to the centroid μ_j (index of specific cluster):

$$\|x_i - \mu_j\|^2$$



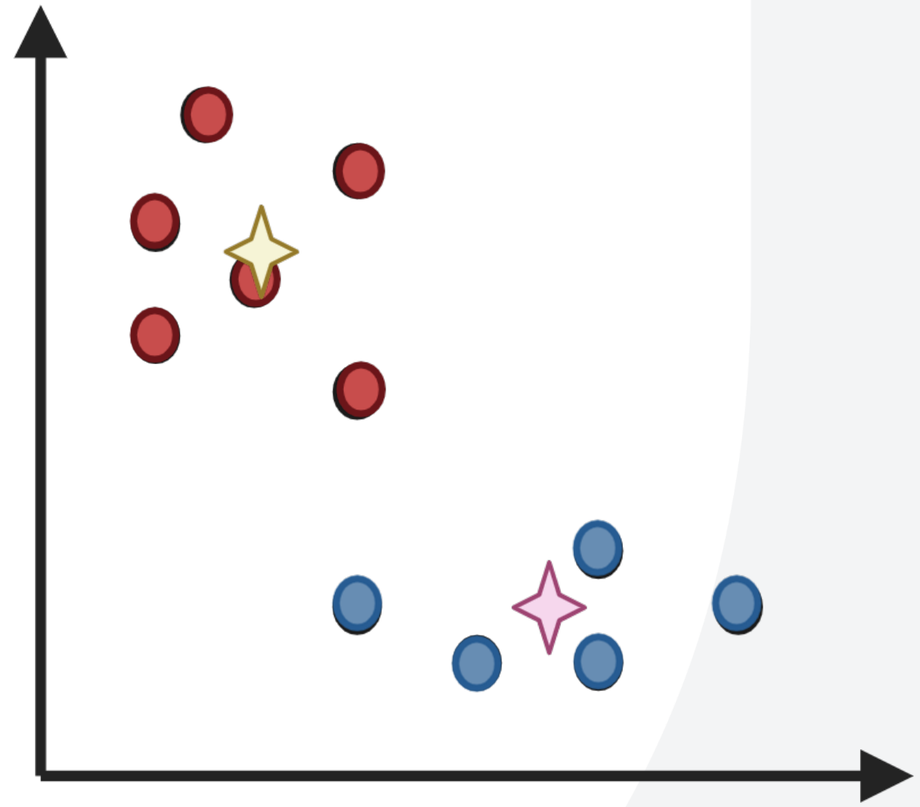
K-Means

1. **Update:** Recalculate centroids by taking the mean of all points in each cluster:

$$\boldsymbol{\mu_j} = \frac{\mathbf{1}}{|\boldsymbol{C_j}|} \sum_{\boldsymbol{x_i} \in \boldsymbol{C_j}} \boldsymbol{x_i}$$

1. **Iteration:** Repeat assignment and update steps until convergence by minimising the cost function (sum of squared distances):

$$J = \sum_{i=1}^{k} \sum_{x_i \in C_i} \left\| \boldsymbol{x_i} - \boldsymbol{\mu_j} \right\|^2$$



THE UNIVERSITY of ADELAIDE

150 YEARS

K-Means

Critical questions:

- **How do we choose the appropriate value of k?**
  - *Elbow Method:* Plot the sum of squared distances (inertia) for different k values and look for the 'elbow point' where the rate of decrease sharply slows.
  - *Silhouette Score:* A measure of how similar a data point is to its own cluster compared to other clusters. Use average silhouette score for different k values and choose the k with the highest score.

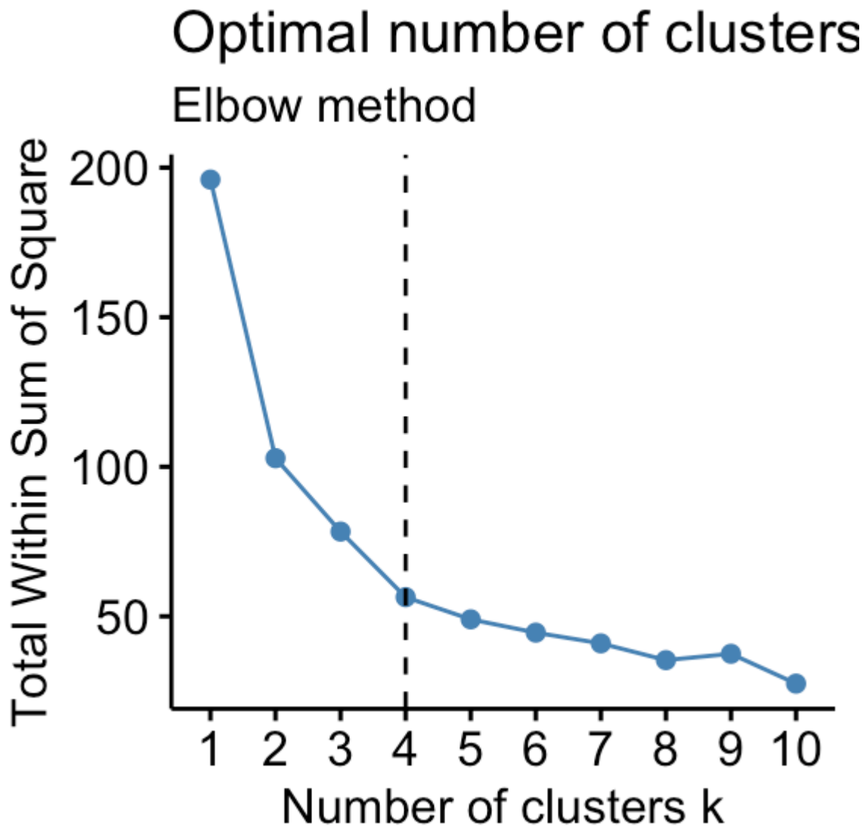
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

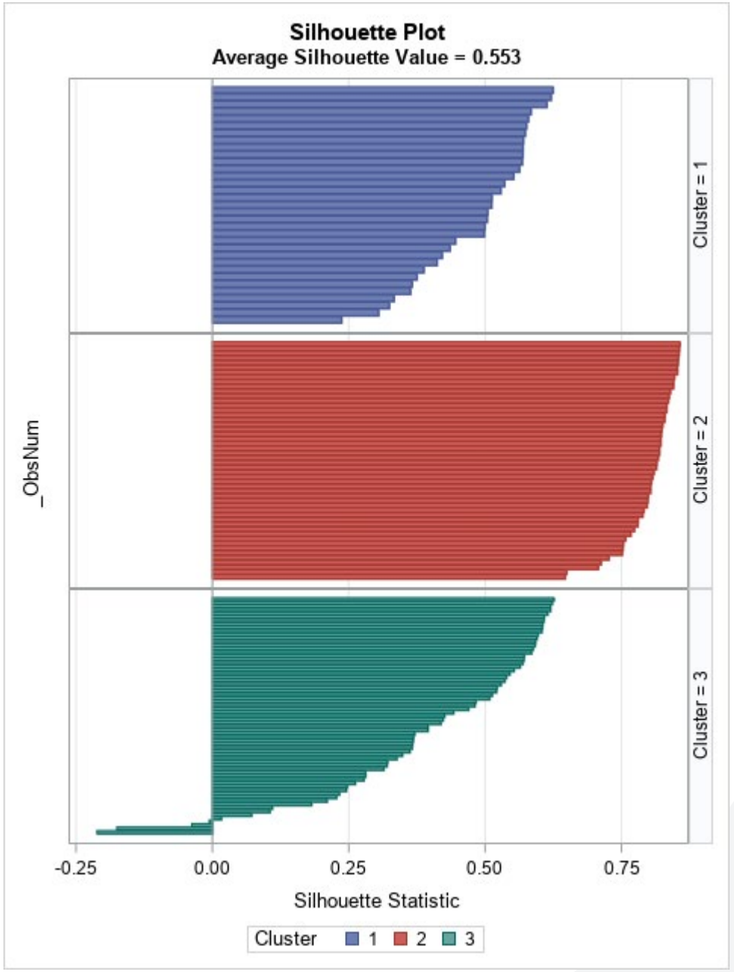
$a(i)$: Average distance of i to other points in own cluster.
$b(i)$: Average distance of i to other points in nearest cluster.

  - *Cross-Validation:* Use cross-validation techniques to evaluate the performance of different k values.

THE UNIVERSITY of ADELAIDE 150 YEARS

K-Means





https://blogs.sas.com/content/iml/2023/05/17/compute-silhouette-sas.html
https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/

THE UNIVERSITY of ADELAIDE
150 YEARS

K-Means

Critical questions:

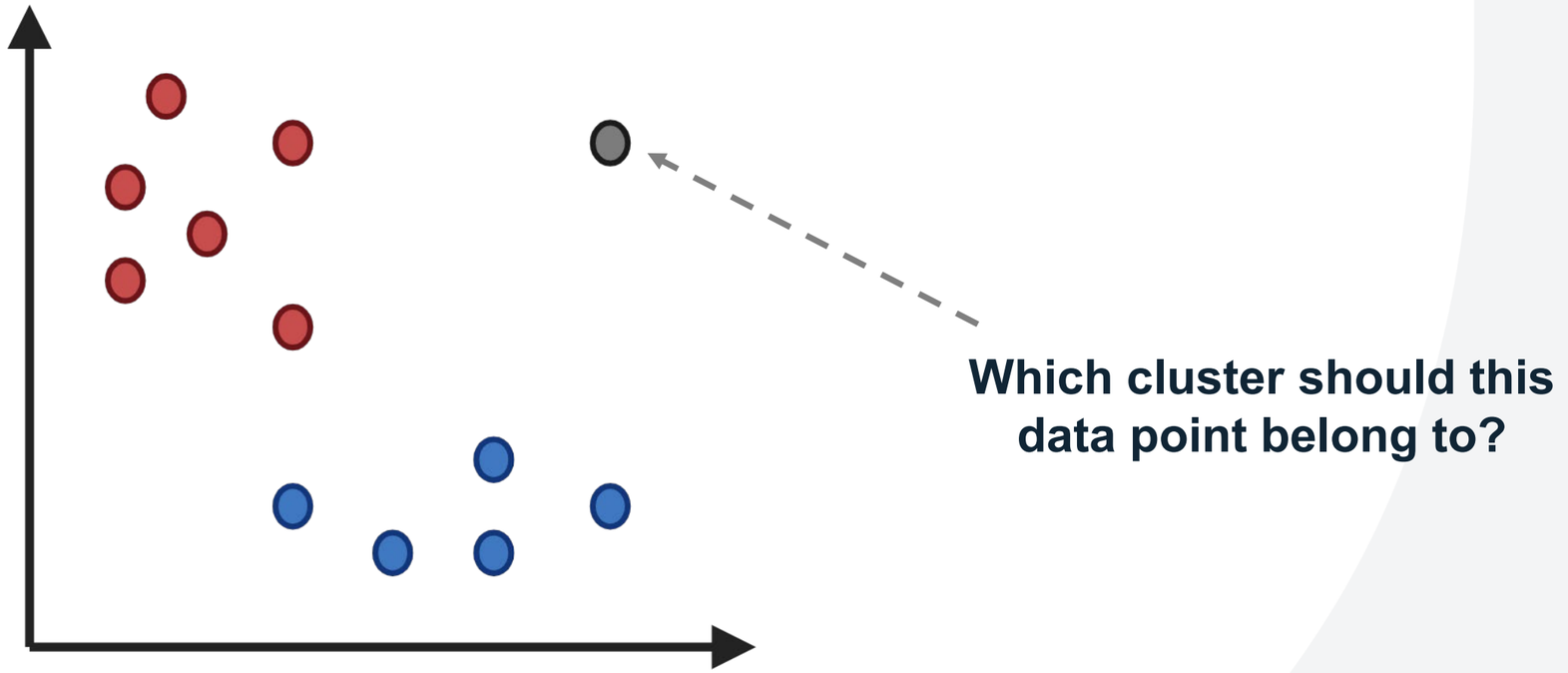
- **What about bad initialisations?**
  - *Random Restarts:* Perform multiple runs with different random initialisations. This can help increase chances of finding global minimum.
  - *K-means++:* Use K-Means++ for better initial centroid selection.
    - For each data point x, compute distance $\boldsymbol{D(x)}$ to the nearest centroid.
    - Select the next centroid with probability proportional to $\boldsymbol{D(x)^2}$.

Comparison between K-means++ and K-means:

- K-Means may lead to poor clustering (due to getting stuck in local minima) and slow convergence.
- K-Means++ spreads out centroids - more spread out initially, which helps in covering the entire data space better.
- K-Means++ uses diverse starting points - mean that the clusters can form around different regions of the data, reducing the chances of poor clustering.

What about more difficult data points?

Traditional clustering (like K-Means) can struggle with points that ***do not clearly belong to any cluster*** (outliers).

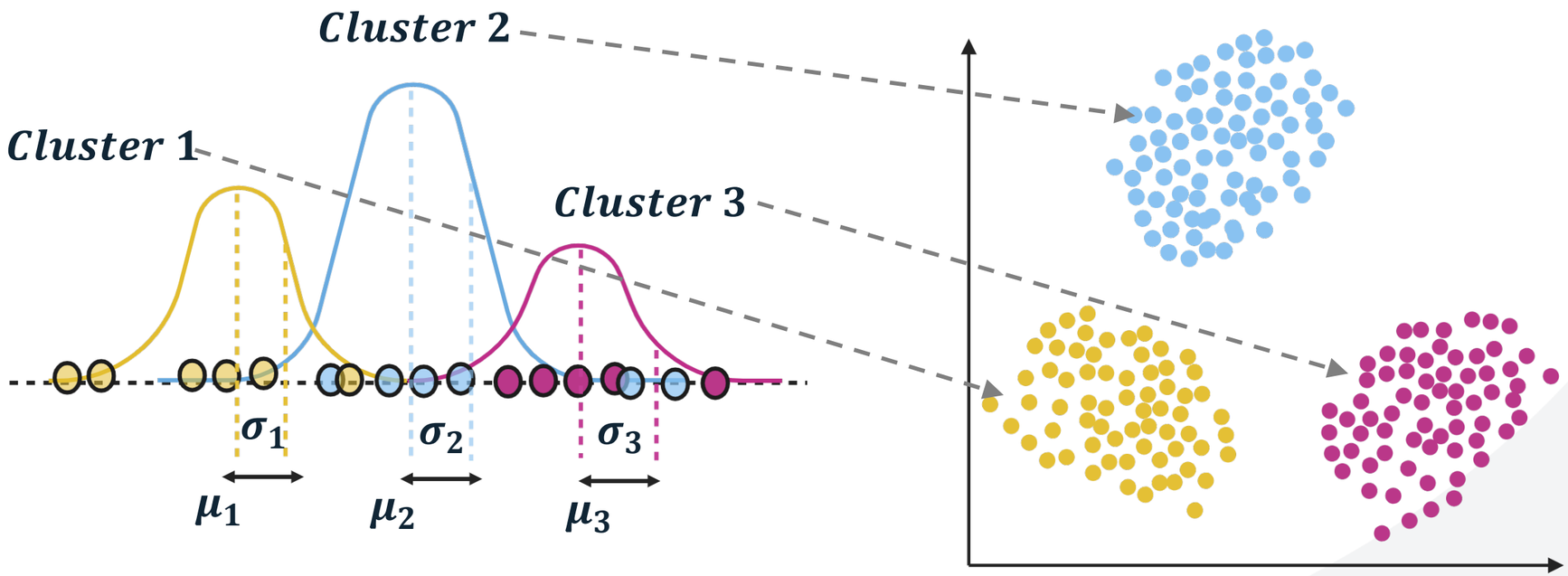


- *Density-Based Clustering:* Methods like DBSCAN can identify outliers based on data point density.
- *Model-Based Clustering:* Gaussian Mixture Models (GMM) use probabilistic models to determine cluster membership and handle clusters of various shapes and sizes.

Gaussian Mixture Models

Gaussian Mixture Models (GMM) use probabilistic models to determine cluster membership and handle clusters of various shapes and sizes.

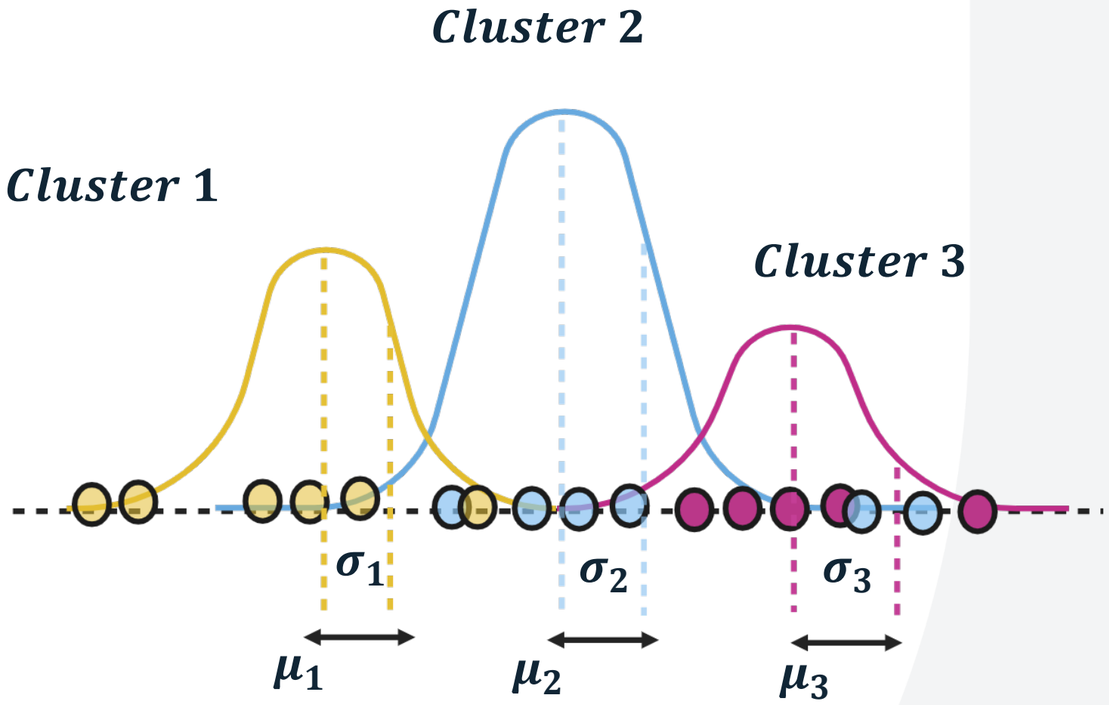
- Aims to estimate the underlying distribution that generated the data.
- Assumes data is generated from a mixture of Gaussian distributions, each representing a cluster.
- Can identify which cluster a point belongs to based on probability.



Gaussian Mixture Models

Components of GMM:

- *Mean (μ):* Centre of the Gaussian distribution.
- *Covariance (Σ):* Spread and shape of the Gaussian distribution.
- *Mixing Coefficient (π):* Weight of each Gaussian component in the mixture, representing the proportion of the dataset that belongs to that component.
  - Likelihood that randomly chosen data point was generated by the k-th Gaussian component.



Mixing Coefficient (π):

If $\pi 1 = 0.3$, $\pi 2 = 0.5$, and $\pi 3 = 0.2$ in a 3-component GMM, it suggests that 30% of the data points are expected to come from the first Gaussian, 50% from the second, and 20% from the third.

Gaussian Mixture Models

- **Algorithm steps**

1. **Initialisation:** Initialise the parameters (mean, covariance, and mixing coefficients) of the Gaussian components.

2. **Expectation-Maximisation (EM) Algorithm:**

   - *E-Step:* Calculate the responsibility (γ_{ik}) of each Gaussian component (k) for each data point (x_i):

$$\gamma_{ik} = \frac{\pi_k N(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_i \mid \mu_j, \Sigma_j)}$$

π_k: Mixing coefficient for Gaussian component k.

$N(x_i \mid \mu_k, \Sigma_k)$: Probability density of x_i under the k-th Gaussian component with mean μ_k and covariance Σ_k.

Gaussian Mixture Models

- *M-Step:* Update the parameters $\boldsymbol{\mu_k}$ (mean), $\boldsymbol{\Sigma_k}$ (covariance) and $\boldsymbol{\pi_k}$ (mixing coefficients) of the $\boldsymbol{k}$ -th Gaussian component to maximise the likelihood.

Mixing coefficient"

$$\boldsymbol{\pi_k = \frac{N_k}{N}}$$

Mean:

$$\boldsymbol{\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik}\, x_i}$$

Covariance:

$$\boldsymbol{\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik}\, (x_i - \mu_k)(x_i - \mu_k)}$$

$\boldsymbol{N_k = \sum_{j=1}^{K} \gamma_{ik}}$ is the effective number of points assigned to component $\boldsymbol{k}$.

$\boldsymbol{N}$: Total number of data points.

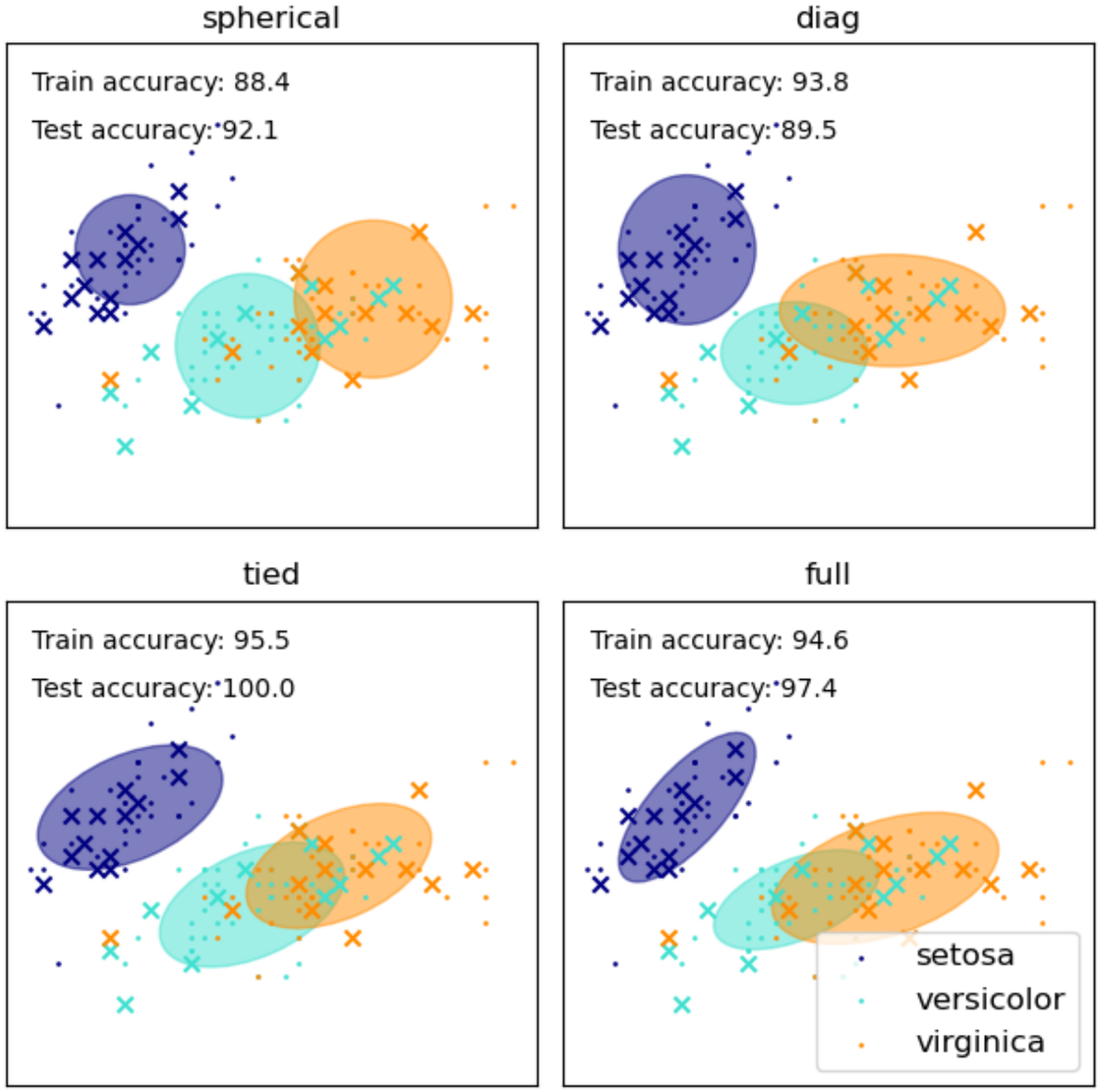
$\boldsymbol{N_k = \sum_{j=1}^{K} \gamma_{ik}}$ is the effective number of points assigned to component $\boldsymbol{k}$.

$\boldsymbol{N}$: Total number of data points.

$\boldsymbol{\gamma_{ik}}$: responsibility ($\boldsymbol{\gamma_{ik}}$) of each Gaussian component ($\boldsymbol{k}$) for each data point ($\boldsymbol{x_i}$):

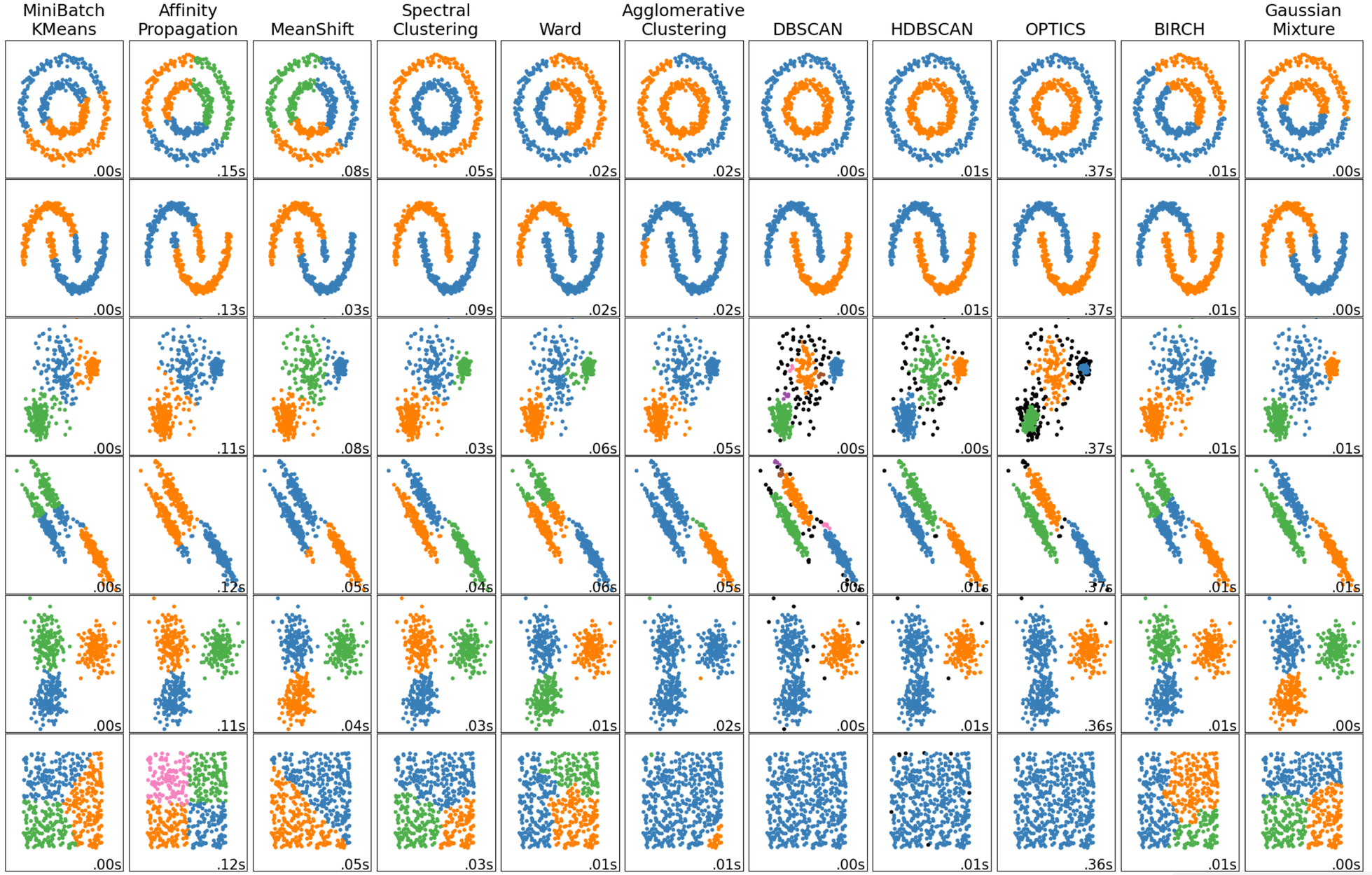
3. Convergence: Repeat the E-Step and M-Step until parameters stabilise.

Gaussian Mixture Models



- **Spherical:** Each component has its own variance but is spherical (same variance in all directions).
- **Diagonal:** Each component has its own diagonal covariance matrix (variance differs along each axis).
- **Tied:** All components share the same covariance matrix (same shape and orientation for all components).
- **Full:** Each component has its own full covariance matrix (different shapes and orientations).

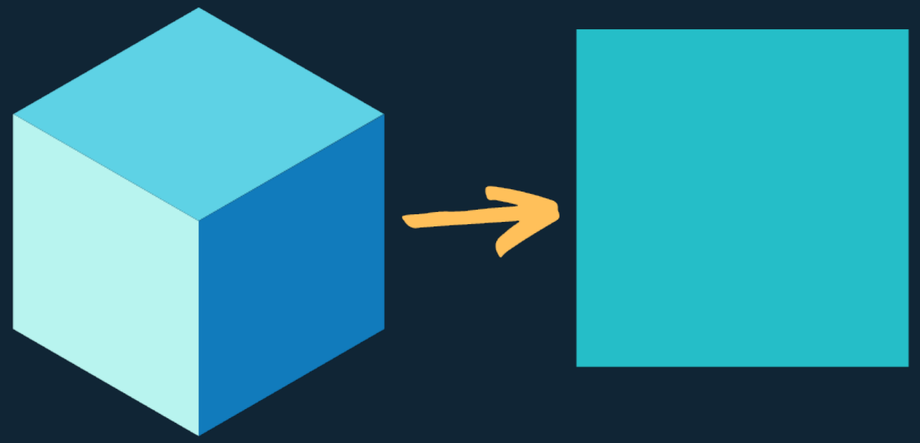
https://scikit-learn.org/stable/modules/mixture.html



https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

THE UNIVERSITY of ADELAIDE

150 YEARS



Dimensionality Reduction

Dimensionality Reduction

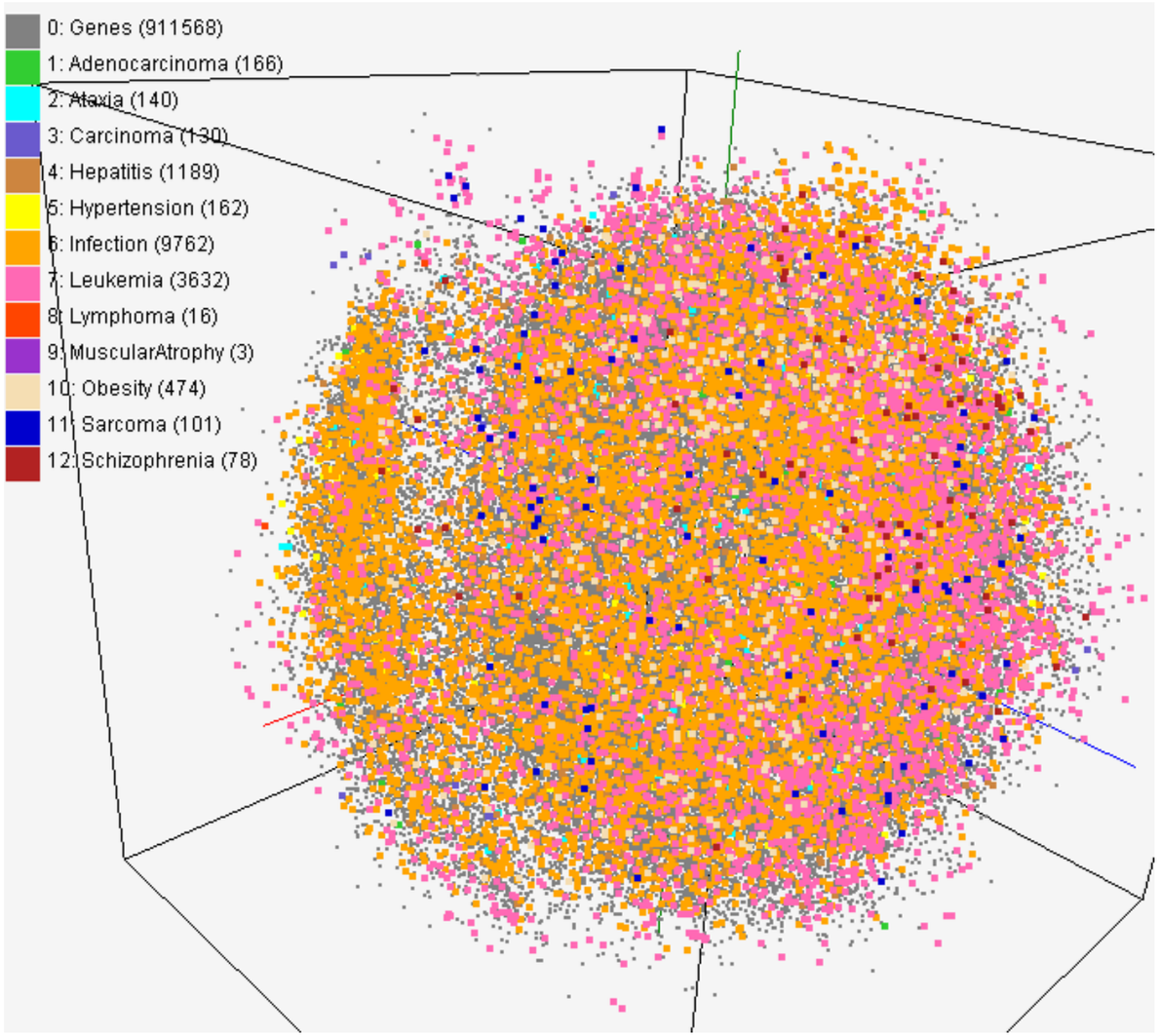
High dimensionality: high-dimensional data refers to datasets with a large number of features or attributes.

- *Genomics:* Thousands of genes, millions of DNA polymorphisms.
- *Image Processing:* Each pixel in an image represents a dimension. For example, a 256x256 RGB image has 196,608 features (256x256x3).

Challenges:

- **Computationally Expensive:** Requires more memory and computational power.
- **Complexity:** Higher complexity can make models harder to interpret.
- **Overfitting:** Models may perform well on training data but poorly on unseen data.
- **Visualisation:** Visualising data with more than three dimensions is not straightforward.
- **Distance metrics:** Traditional distance metrics (i.e Euclidean) can become less meaningful.

Dimensionality Reduction



- High-dimensional data, including genes and various medical conditions.
- Each point represents a data sample, with different colours indicating different conditions.

https://deepai.org/machine-learning-glossary-and-terms/high-dimensional-data

THE UNIVERSITY of ADELAIDE
150 YEARS

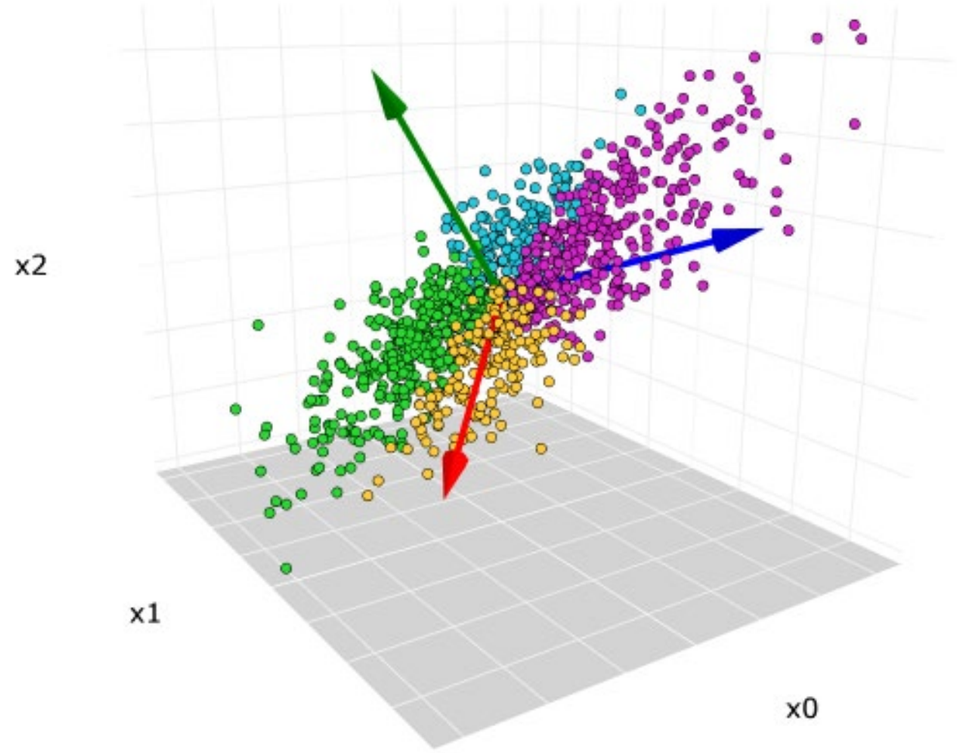
Dimensionality Reduction

- **Dimensionality reduction:** a technique used to reduce the number of features (and hence dimension) in a dataset while preserving as much information as possible.
  - **Purpose:** Simplify data, reduce computational cost, remove noise, and visualise high-dimensional data.
  - Many dimensionality reduction techniques are inherently unsupervised, as they do not require labelled data to identify the underlying structure of the dataset.
  - Often used in conjunction with clustering algorithms to enhance performance by simplifying the dataset and making underlying patterns more apparent.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA): Transforms features into a set of linearly uncorrelated variables (***principal components)***, ordered by the amount of variance captured.

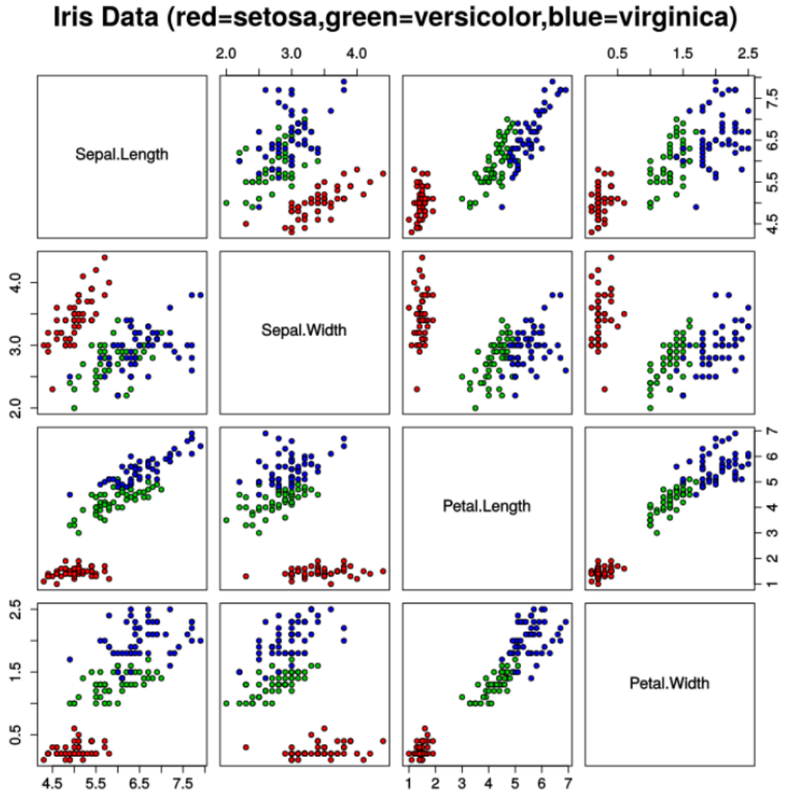
- Too many features can slow down predictions.
- Redundant features may represent the same information.
- *Uses:* Data compression, visualisation, noise reduction, pre-processing.

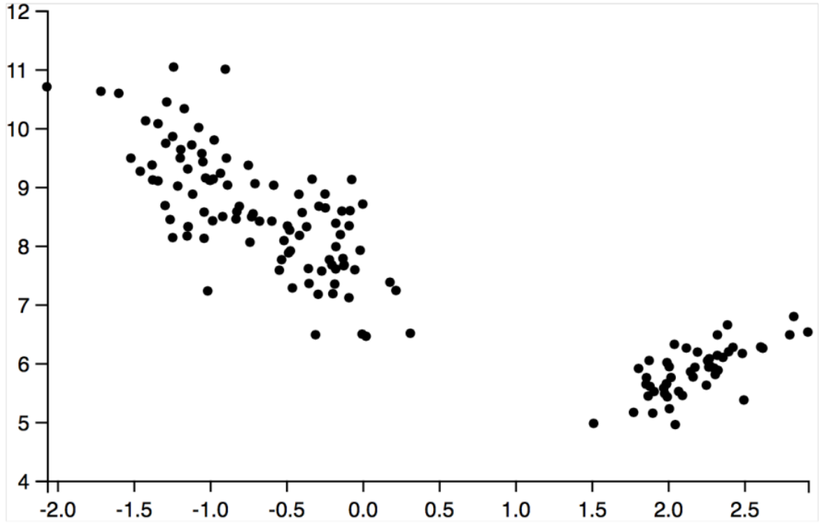


https://towardsdatascience.com/principal-component-analysis-pca-explained-visually-with-zero-math-1cbf392b9e7d

Principal Component Analysis (PCA)

Principal Component Aanalysis (PCA): Transforms features into a set of linearly uncorrelated variables (***principal components)***, ordered by the amount of variance captured.



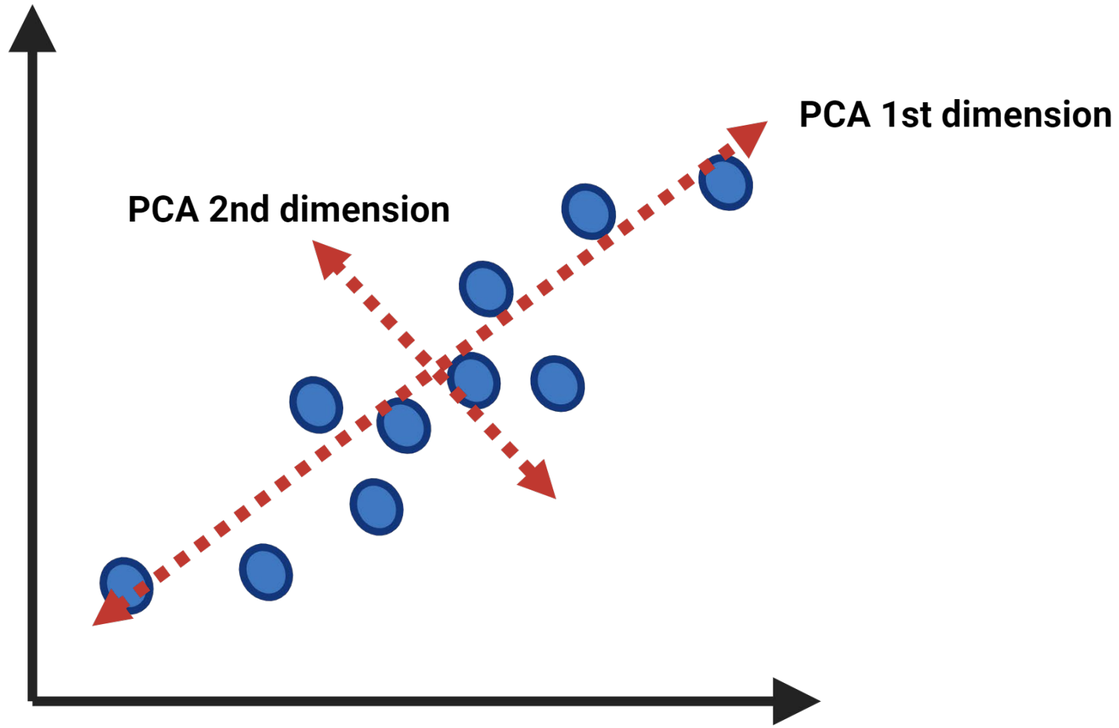


Goal: Map high-dimensional data onto lower-dimensional data in a manner that preserves distances/similarities

https://course.ccs.neu.edu/ds4420sp20/slides/08-ds4420-s2020-dimensionality-reduction1.pdf

Linear Dimensionality Reduction with PCA

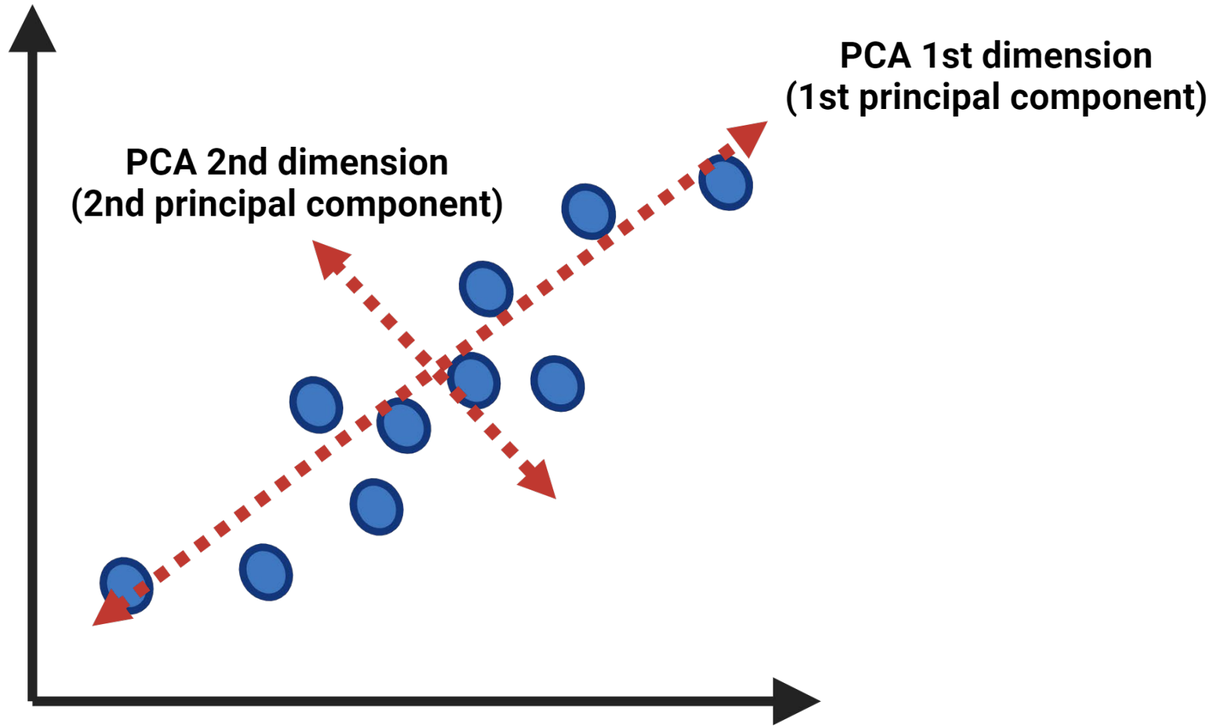
Idea: Project high-dimensional vector onto a lower-dimensional space by capturing variance



PCA finds the principal components that capture the most variance (dispersion or spread) in the data.

Linear Dimensionality Reduction with PCA

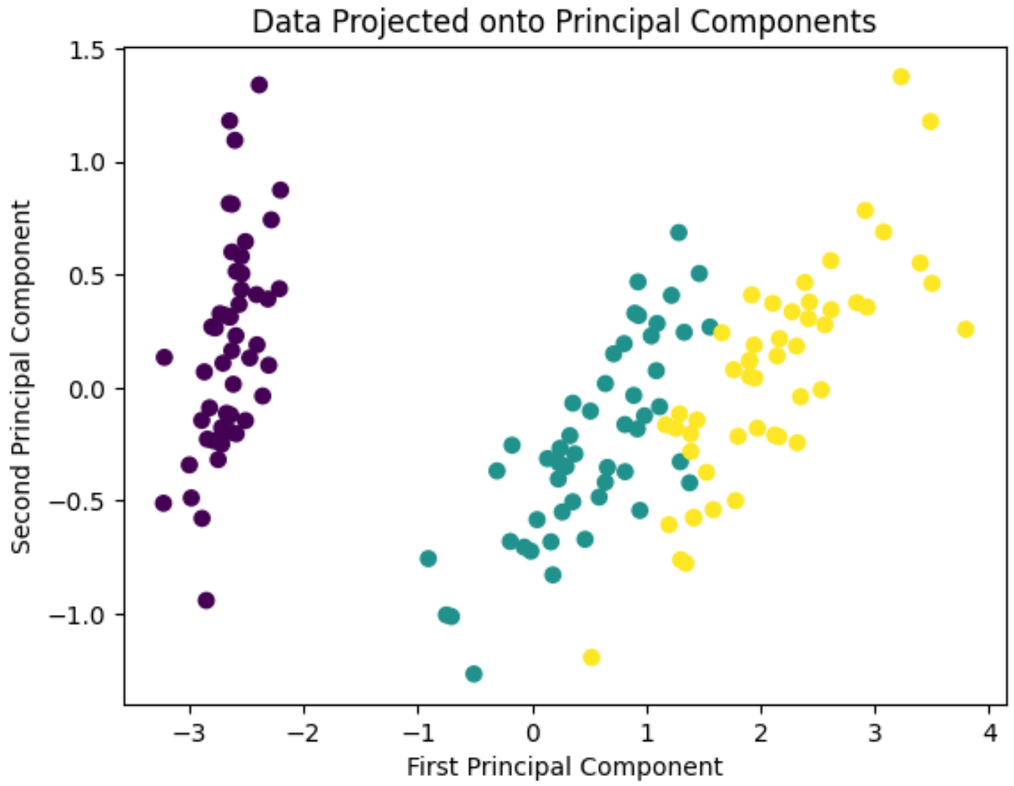
Idea: Project high-dimensional vector onto a lower-dimensional space by capturing variance



Principal components are the new axes (directions) along which the variance in the data is maximised. Ensures most significant features (in terms of data spread) are retained in the reduced dataset.

Linear Dimensionality Reduction with PCA

Idea: Project high-dimensional vector onto a lower-dimensional space by capturing variance



Here, each data point is represented in terms of its coordinates along the principal components rather than the original features.

Data is then projected onto these components (new axes), reducing the dimensionality while preserving as much information as possible.

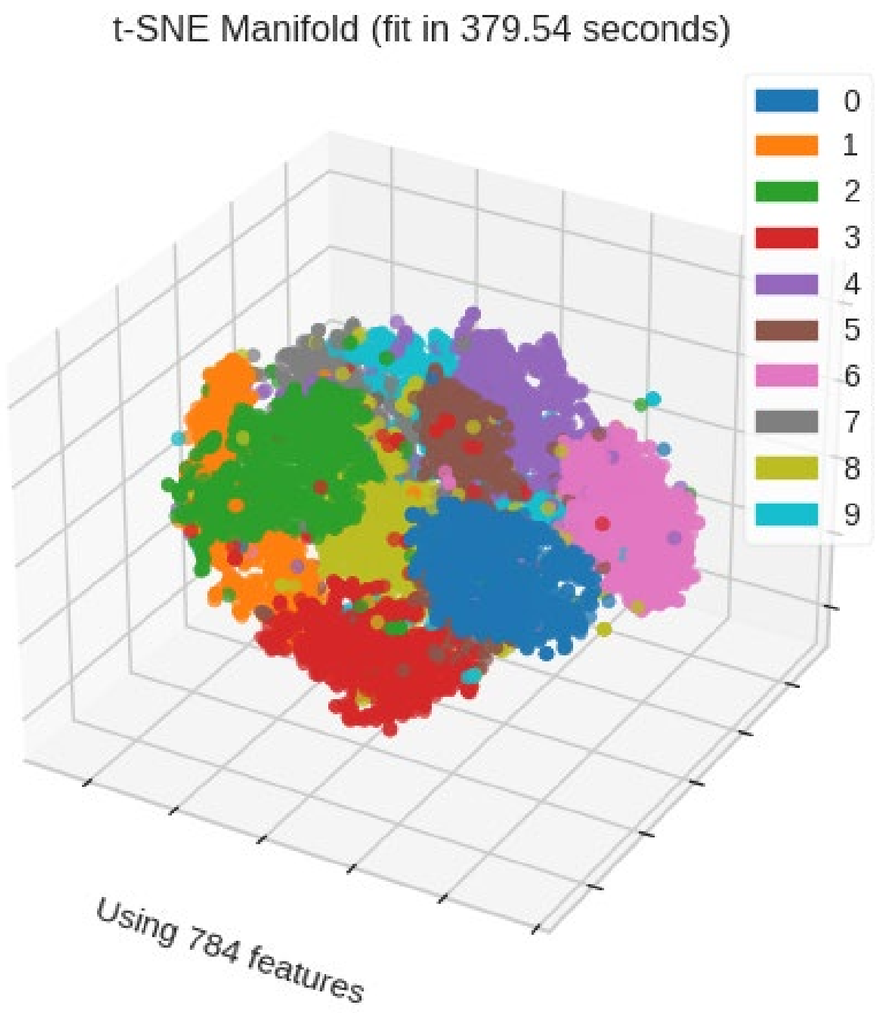
PCA with scikit-learn

1. Create PCA object.
2. Specify number of components.
3. Fit the PCA model.
4. Transform the data.

```
from sklearn.decomposition import PCA

# Create PCA object
pca = PCA(n_components=2)

# Fit the PCA model
pca.fit(data)

# Transform the data
data_transformed = pca.transform(data)
```

t-SNE

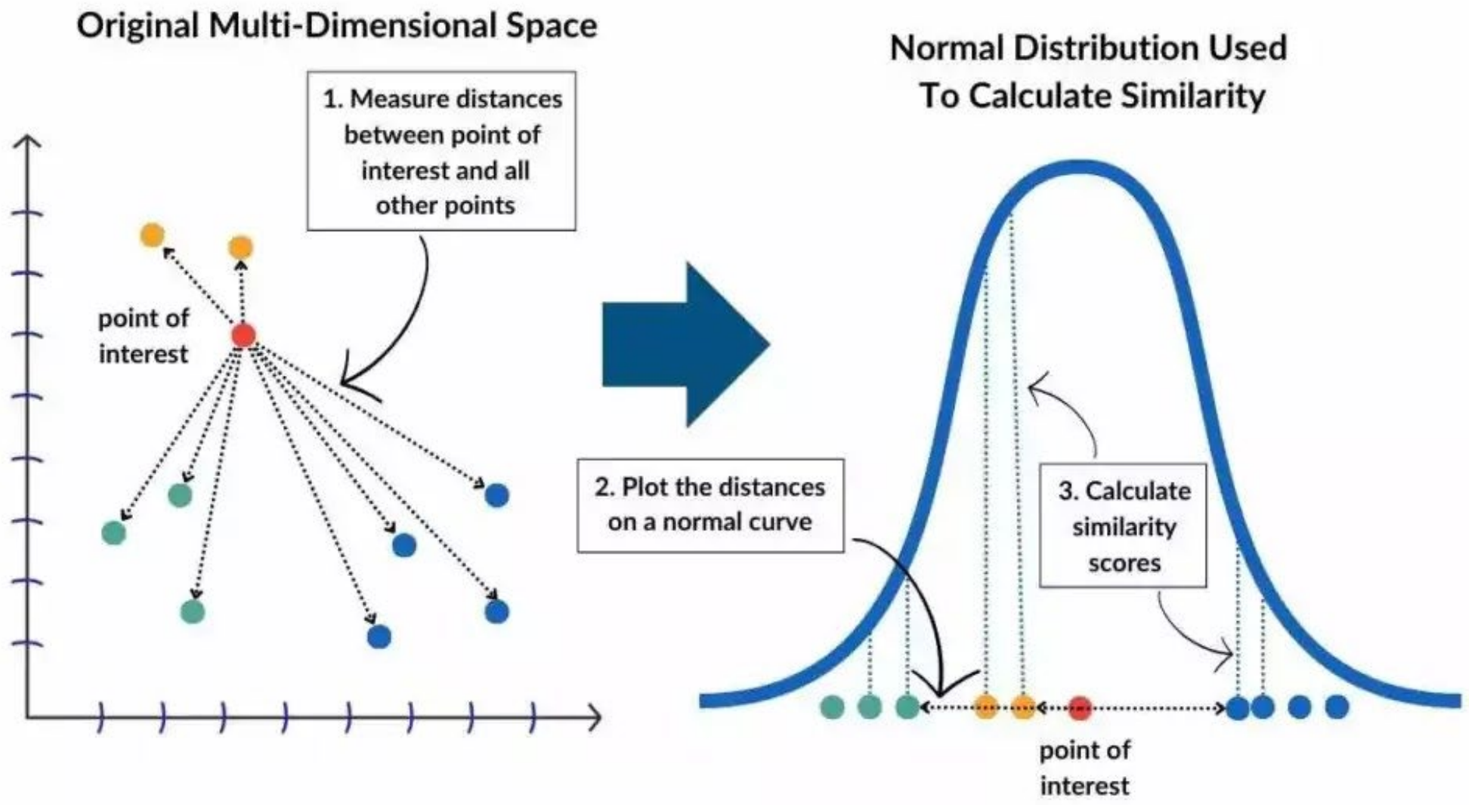
- **t-Distributed Stochastic Neighbor Embedding (t-SNE):** Focuses on preserving the local structure of the data, meaning similar data points in high-dimensional space remain close to each other in lower-dimensional representation.
- Unlike PCA, t-SNE is a non-linear dimensionality reduction technique. This means it can capture more complex relationships between data points.



https://rukshanpramoditha.medium.com/t-sne-visualization-with-yellowbrick-a-fast-and-easy-method-a6e60ba3d838

t-SNE

Idea: Project high-dimensional data into a lower-dimensional space while preserving local relationships between data points.

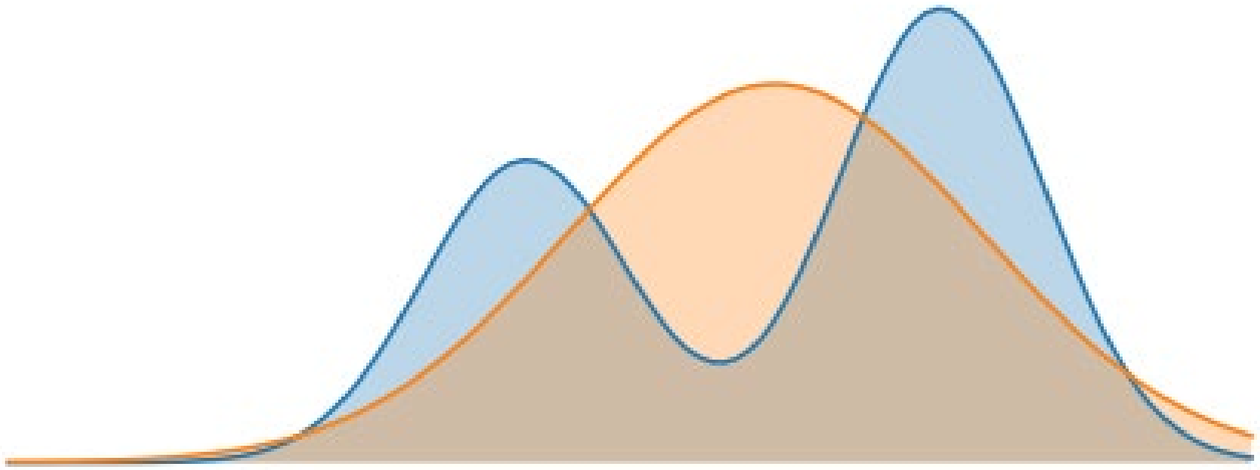


While original algorithm uses Euclidean distance as the base of its similarity metric, this can be modified.

Calculate pairwise similarities (i.e distances) between data points in high-dimensional space and convert similarities into probability distributions.

t-SNE

Idea: Project high-dimensional data into a lower-dimensional space while preserving local relationships between data points.

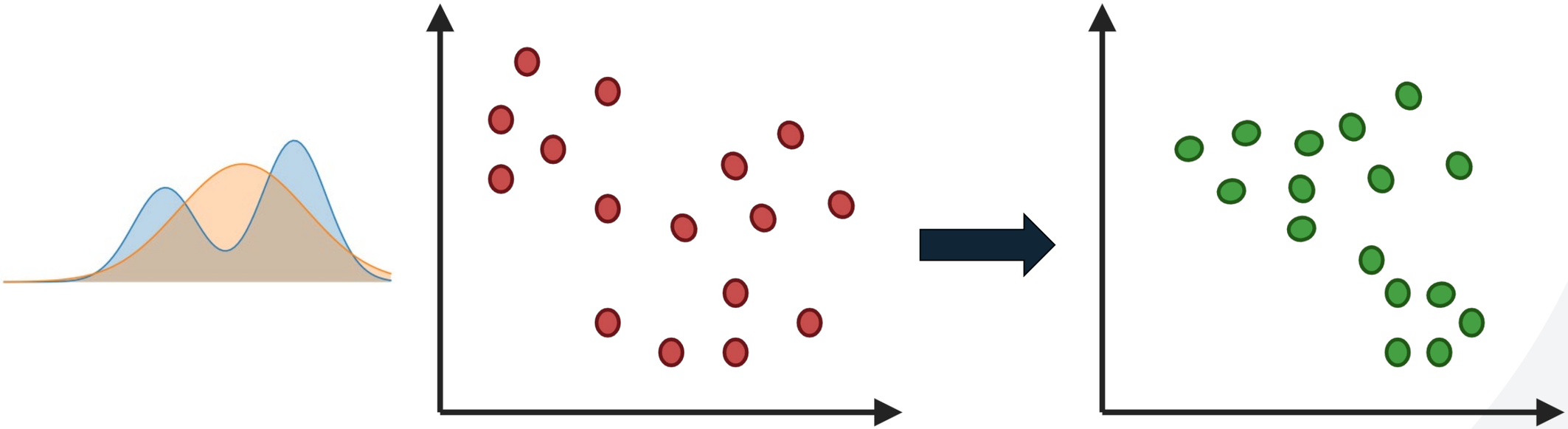


Kullback-Leibler (KL) divergence: A way to measure how different two probability distributions are from each other

Minimise KL-divergence between high-dimensional and low-dimensional distributions.

t-SNE

Idea: Project high-dimensional data into a lower-dimensional space while preserving local relationships between data points.



Optimise the positions of data points in the lower-dimensional space.

Interactive t-SNE playground:
https://wedadanbtawi95.github.io/tsne/

THE UNIVERSITY of ADELAIDE
150 YEARS

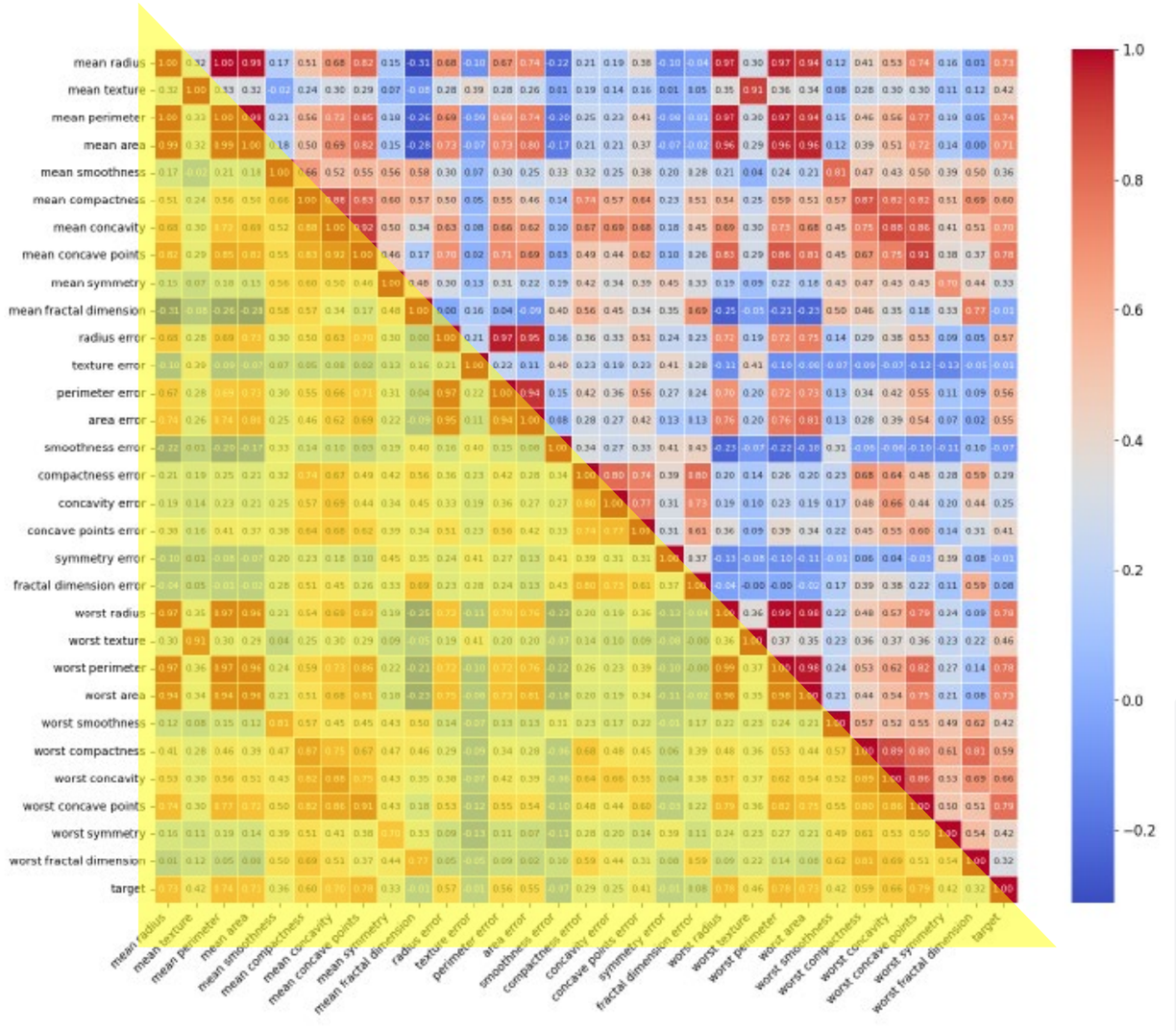
Dimension reduction through feature selection

Other dimension reduction methods exist, by way of feature selection, i.e:

- **Filter methods:** Correlation Coefficient, Chi-squared Test, Variance Threshold.
- **Wrapper methods:** Recursive Feature Elimination (RFE), Forward Selection, Backward Elimination.
- **Embedded methods:** L1 (Lasso) Regularisation, Decision Trees.

Other Methods: Correlation

- **Heatmap** or **correlation matrix** displaying correlation coefficients between each feature, as well as with the target.
- Both sides of the diagonal show the same information, so we can just look at one half of the triangle.

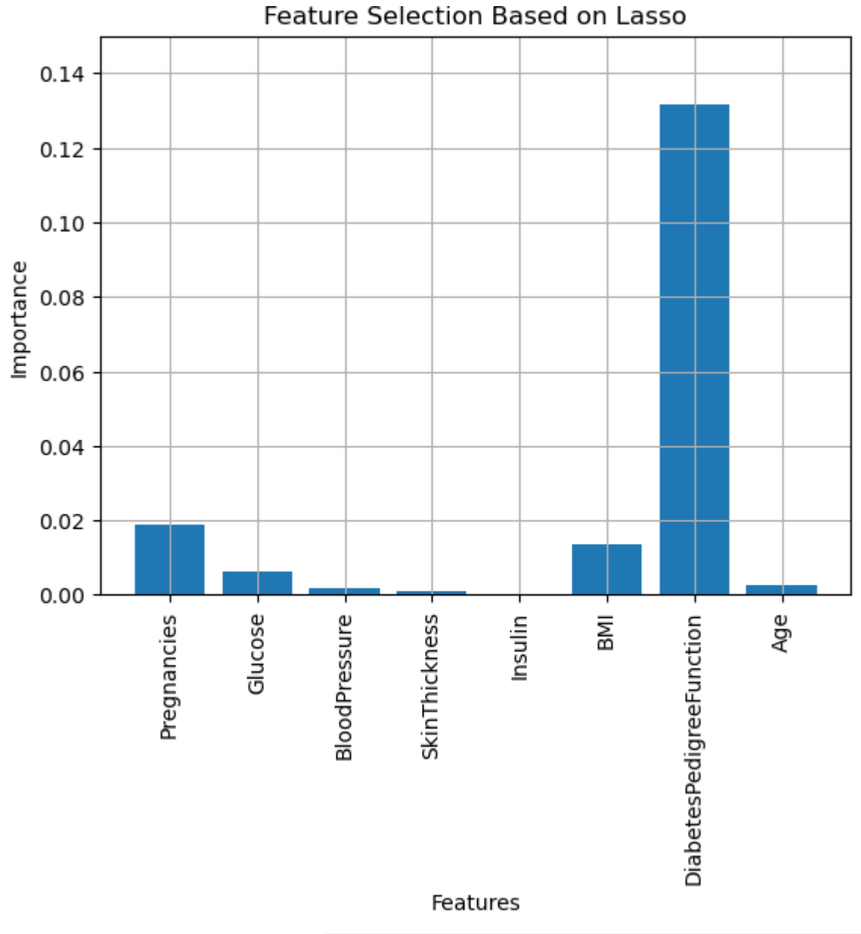


THE UNIVERSITY of ADELAIDE
150 YEARS

Other Methods: L1 Regularisation

L1 or lasso regularisation can also help with feature selection, and sometimes considered an ***embedded*** method as it is implemented during training.

- *Mechanism:* Adds an L1 penalty term to the cost function, which is proportional to the absolute value of the coefficients of the features.
- *Effect:* Encourages the model to have sparse coefficients, i.e., most of the coefficients are zero, and only a few features are used.



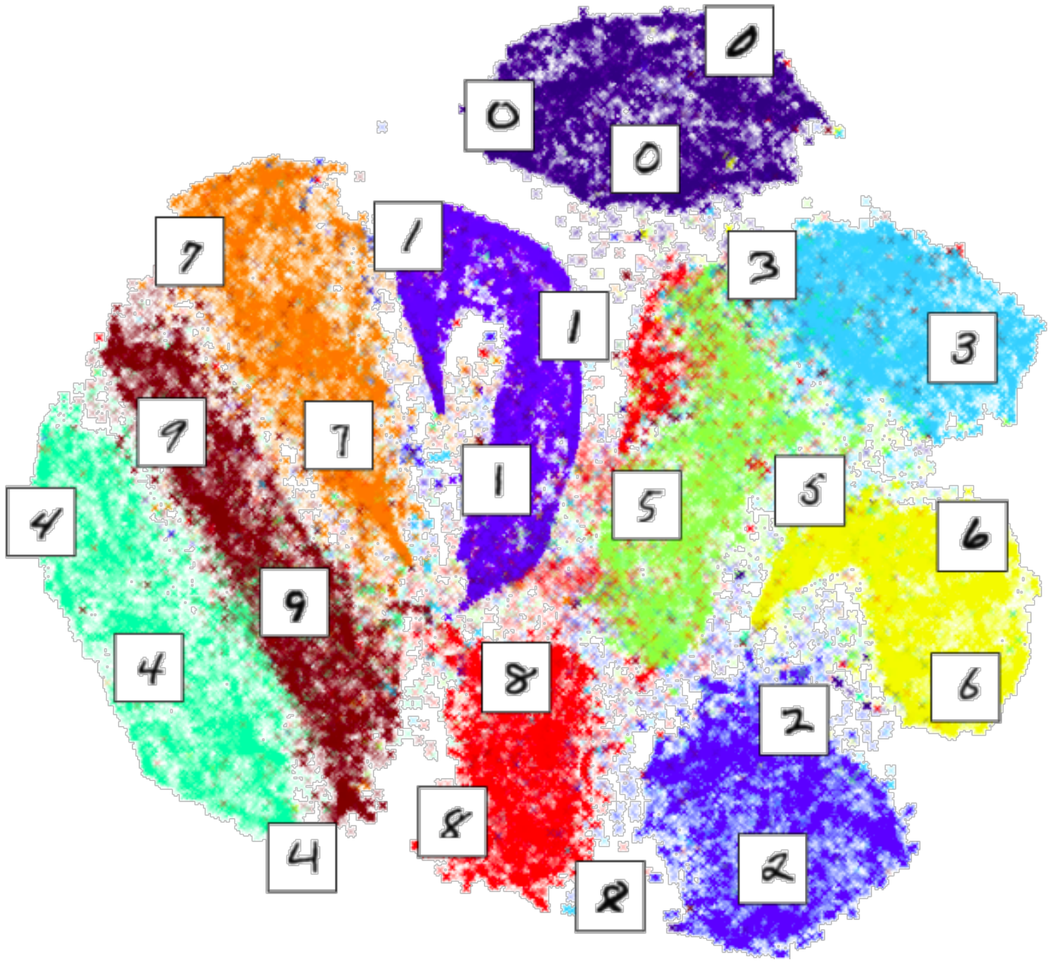
THE UNIVERSITY of ADELAIDE
150 YEARS

Data Visualisations

Data visualisations using unsupervised learning

- Unsupervised learning approaches, such as t-SNE, are powerful tools for visualising high-dimensional data.
- These techniques help uncover hidden structures and patterns in the data by projecting it into a lower-dimensional space, making it easier to interpret and analyse complex datasets.
    - i.e. often reveal **natural clusters** and **patterns** that are not immediately apparent in the high-dimensional space.

Data visualisations using unsupervised learning



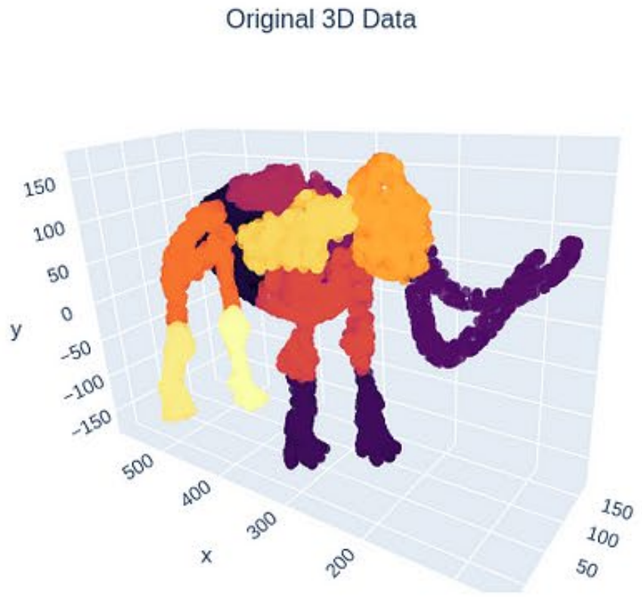
- Clusters identified by t-SNE from classifying written numerical digits → Reveals patterns in the underlying data.
- Note: t-SNE focuses on preserving the local structure of the data, but may not preserve global structure.

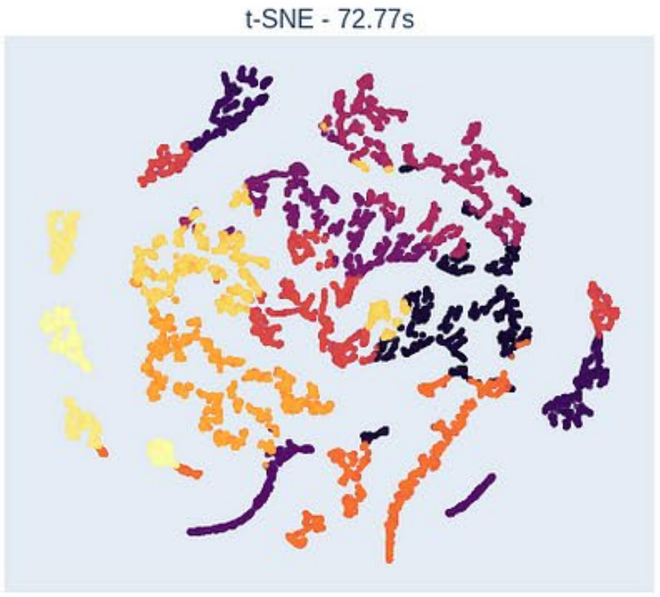
THE UNIVERSITY of ADELAIDE

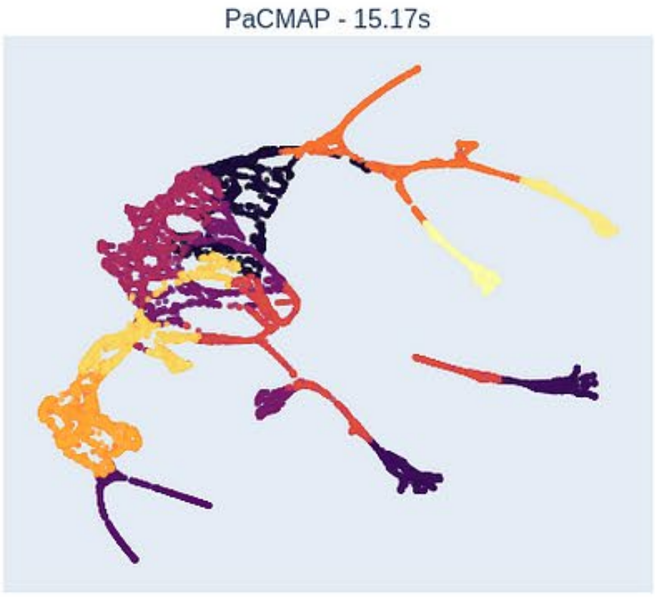
150 YEARS

UMAP

Uniform Manifold Approximation and Projection (UMAP): Similar to t-SNE but faster and often preserves more of the global structure of the data, making it suitable for large datasets and revealing overarching patterns.







Summary

1. **Clustering**
   - K-Means
   - Gaussian Mixture Models
2. **Dimensionality Reduction**
   - PCA, tSNE
   - Other feature selection methods
3. **Data visualisations using unsupervised learning approaches**

Questions?

dhani.dharmaprani@adelaide.edu.au