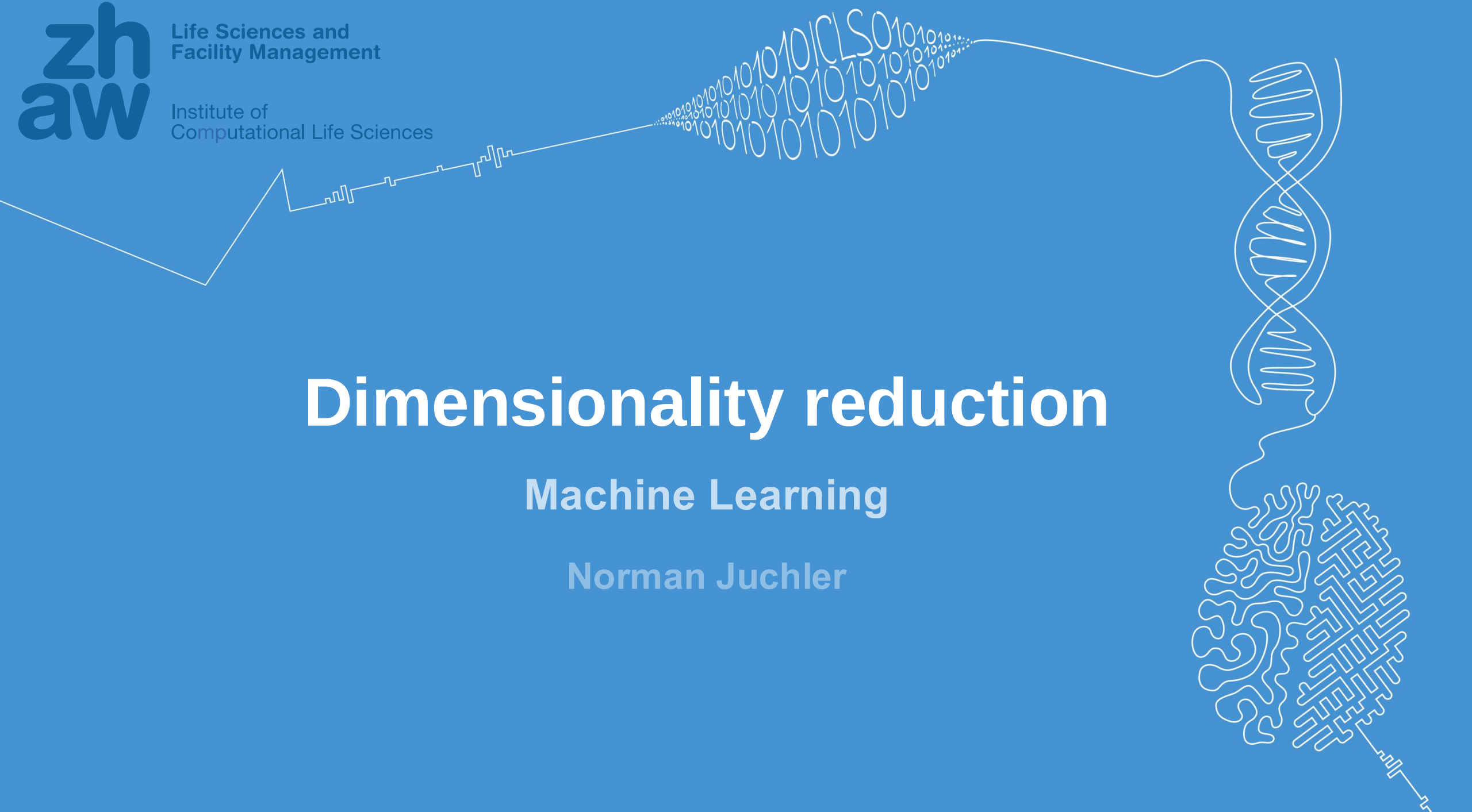
Life Sciences and
Facility Management

Institute of
Computational Life Sciences



Dimensionality reduction

Machine Learning

Norman Juchler

Learning objectives

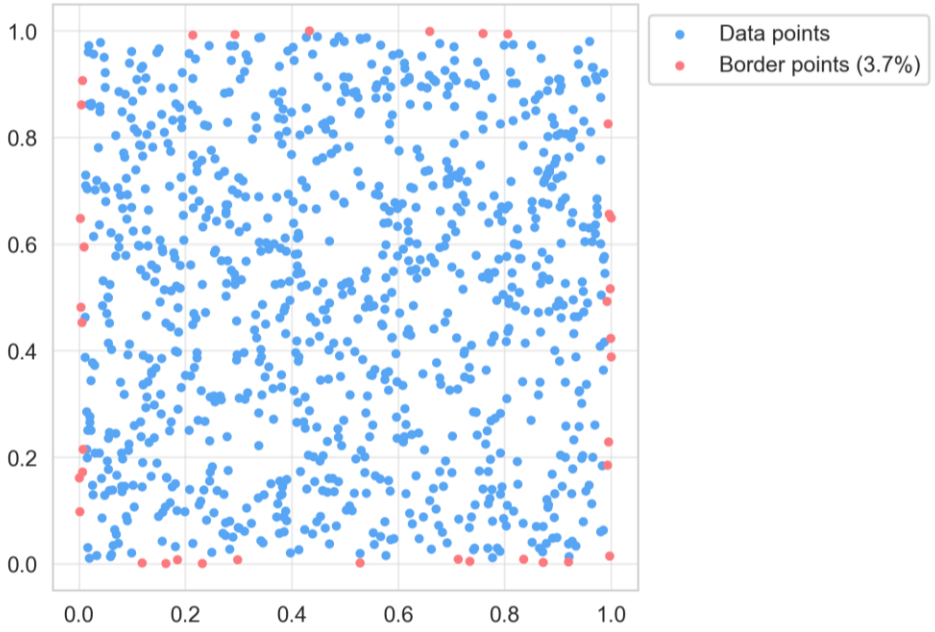
- Understand the need for dimensionality reduction
- Get to know some methods for dimensionality reduction
  - Feature selection
  - Linear discriminant analysis
  - Principal component analysis
  - t-SNE

Introduction

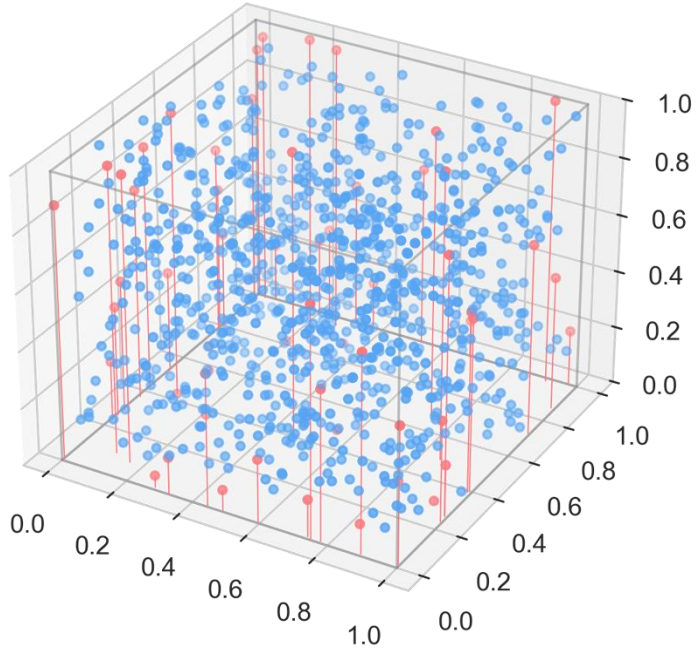
Recap: The curse of dimensionality

- Illustration of the problem:
  - Measure the fraction of points close to the border (ε<0.01)
  - Repeat for increasing dimensionality

2D sample (n=1000)

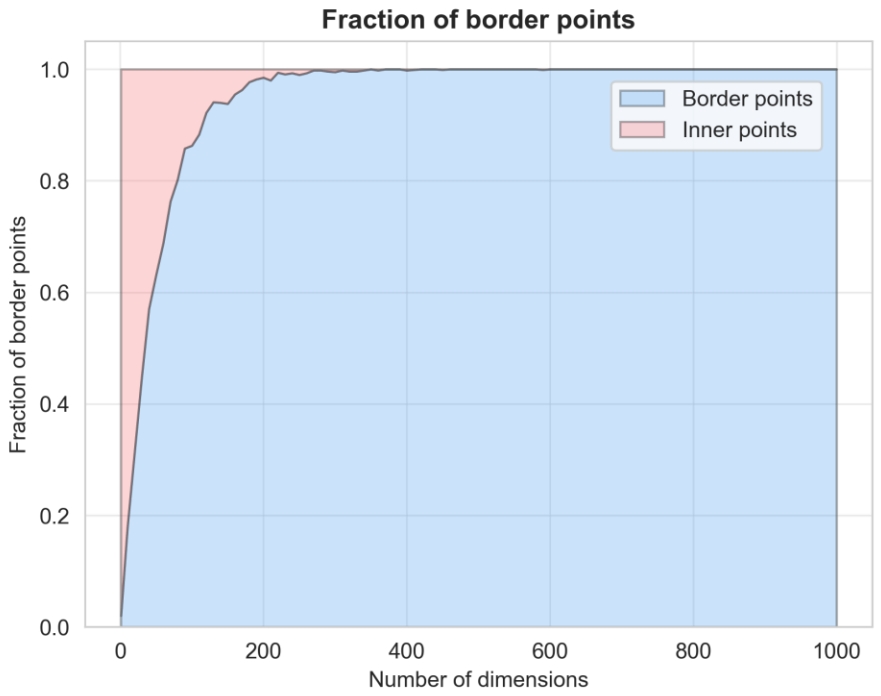


3D sample (n=1000)



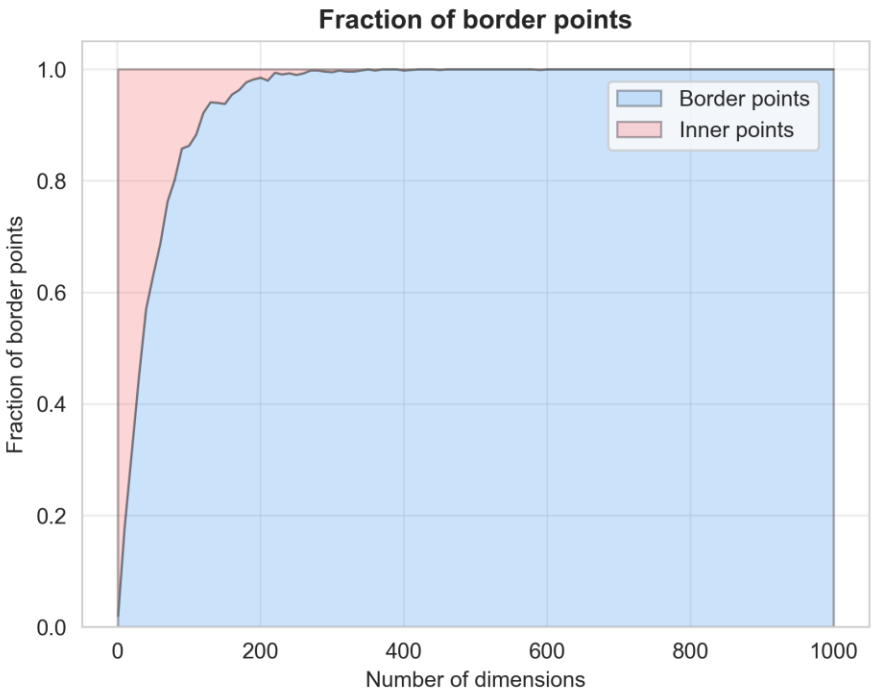
Recap: The curse of dimensionality

- Illustration of the problem:
  - Measure the fraction of points close to the border (ε<0.01)
  - Repeat for increasing dimensionality
- Observations:
  - The fraction of border points increase with the dimension
  - With large dimensions, almost all points are on the boundary



Recap: The curse of dimensionality

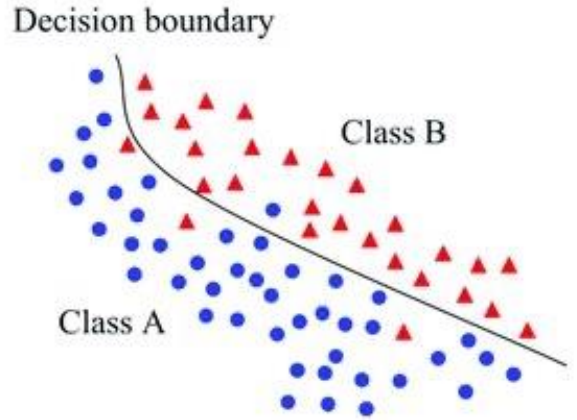
- High-dimensional data can cause problems
  - **Data sparsity**: data points are more spread out in high-dimensional spaces → it's more difficult to find patterns.
  - **Increased computational cost**: The amount of data needed to adequately represent the feature space increases exponentially with the dimensionality.
  - **Distance measures become less informative**: It becomes hard to distinguish close and distant points (e.g., relevant for kNN clustering).
  - **Overfitting risk**: Models may learn noise rather than true patterns, Random patterns may appear significant in high dimensions → poor generalization, overfitting
- Solutions:
  - Dimensionality reduction
  - Feature selection
  - More data...

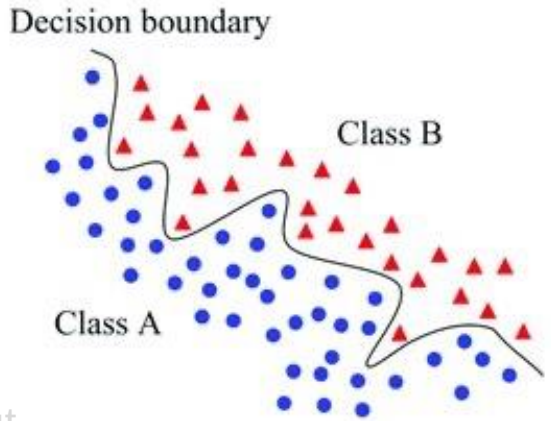


Model complexity vs. dimensionality

High dimensionality increases the risk of overfitting because

- **Data sparsity**: In high-dimensional spaces, data points become sparse and less representative
- **Increased model complexity**: More dimensions allow for more complex models that can fit the data better, but more complexity also increases the risk of overfitting
- **Higher parameter-to-sample ratio**: With many features but limited data, models have more parameters to tune than they can reliably estimate.





Examples of 2D classification without and with overfitting. Link

Motivation for dimensionality reduction

- **Reduce complexity**: Simplify models by working with fewer features, making them computationally efficient and easier to train.
- **Mitigate the curse of dimensionality**: High-dimensional data can lead to sparse, less meaningful patterns, negatively affecting model performance.
- **Enhance visualization**: Reduced dimensionality enables visualization of high-dimensional data in 2D or 3D to gain better insights.
- **Improve model performance**: Reduces noise and irrelevant features, which helps in building more robust and accurate models.
- **Facilitate interpretability**: Focuses on the most important patterns or features in the data, making results easier to understand.

Overview: Common dimensionality reduction methods

- **Feature selection**:
  Pick a subset of the original features according to some criterion
- **Linear discriminant analysis (LDA)**:
  Find a linear combination of features that separates classes
- **Principal component analysis (PCA)**:
  Find a linear combination of features that captures the most variance
- **t-distributed stochastic neighbor embedding (t-SNE)**:
  Non-linear dimensionality reduction that preserves local structure
- **Autoencoders**:
  Non-linear dimensionality reduction using neural networks that learn compressed representations of the data

Question:

If you think of the machine learning pipeline, after which step would you apply dimensionality reduction?

Feature selection

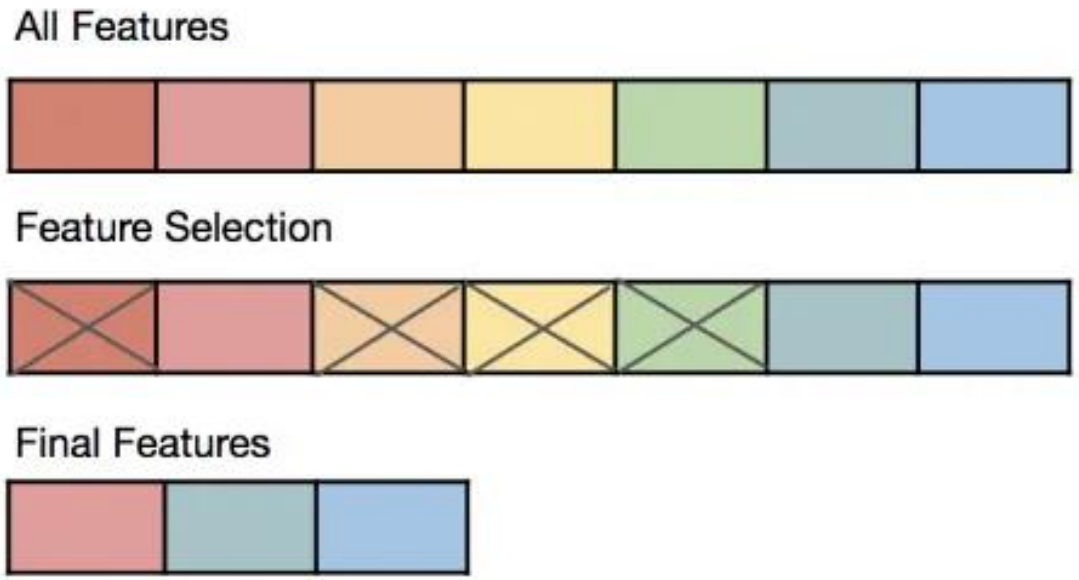
What is feature selection?

Central ideas:

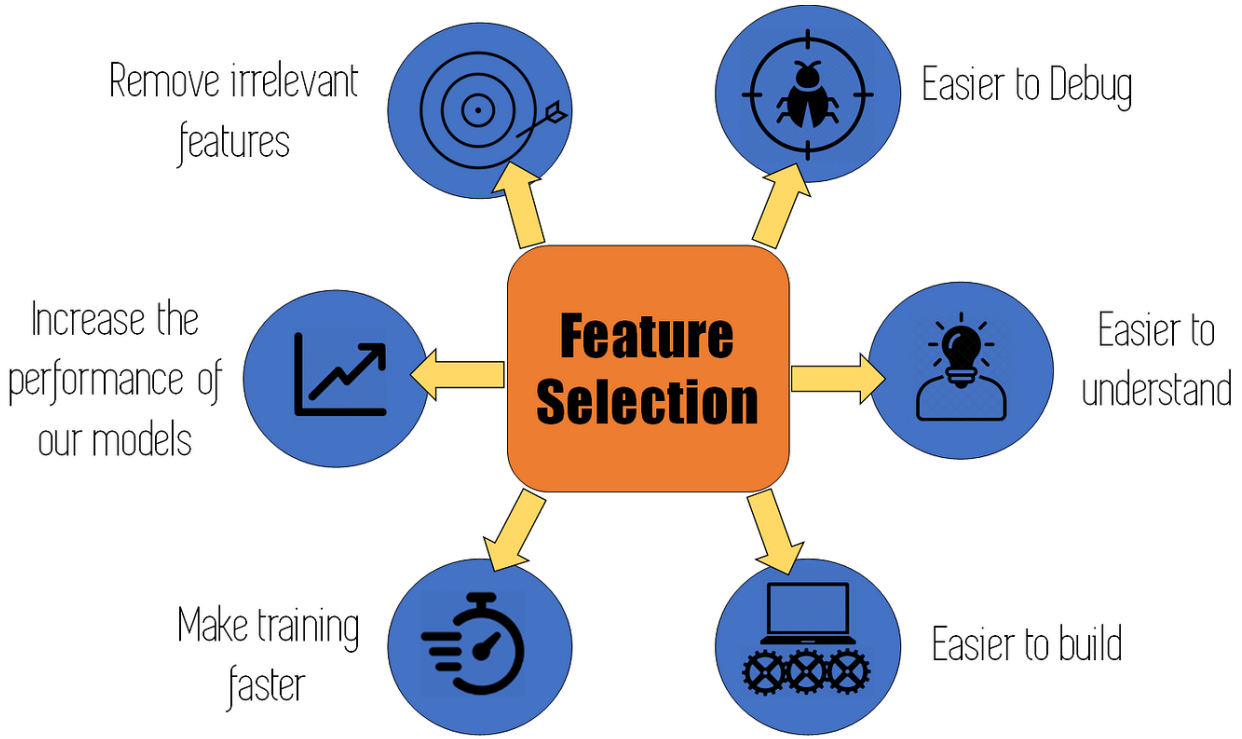
- Identify and retain the most relevant features from a dataset
- Discard irrelevant or redundant features

Effect:

- Remove entire columns from the feature matrix



Why to selection features?

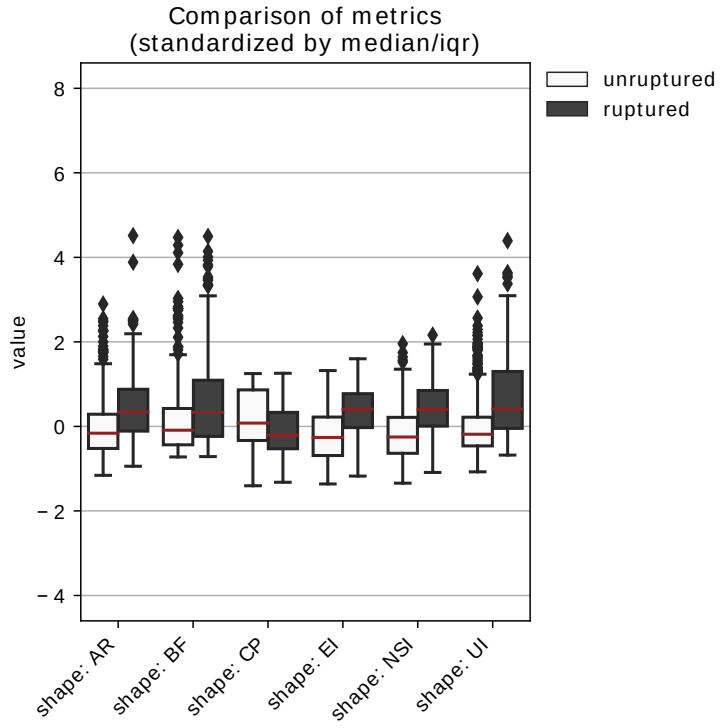


Reasons for feature selection. Source:: Link

Study question: Most of the reasons apply for dimensionality reduction in general. Which ones do not?

Some feature selection methods

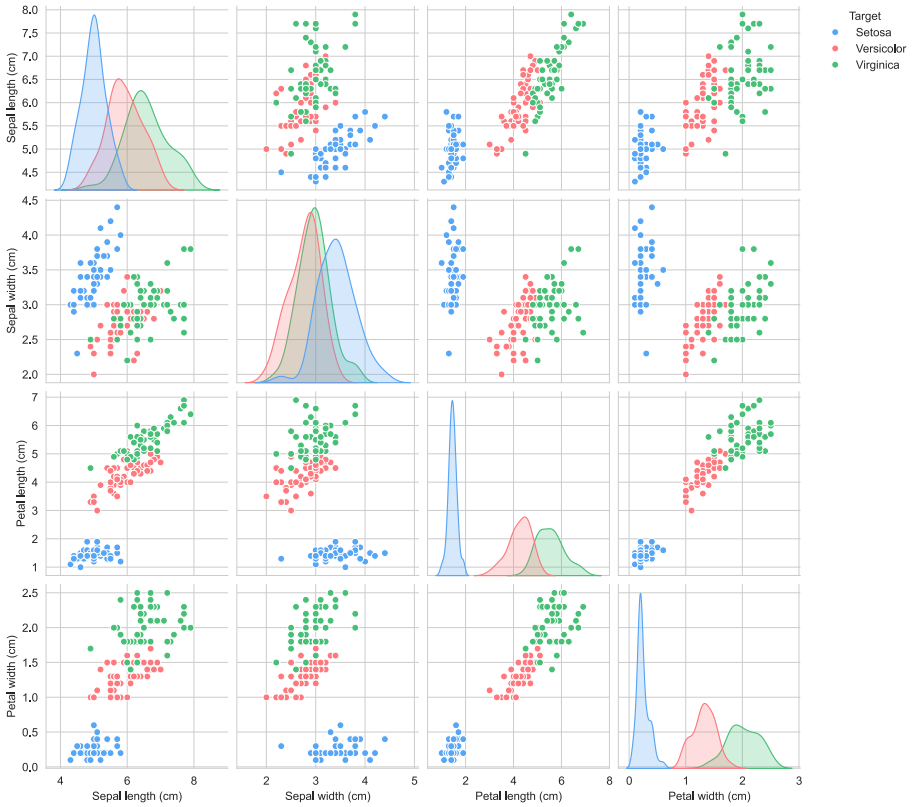
- Univariate analysis:
  - Compare each feature to the target variable and retain the most relevant ones



Example for univariate analysis.
Goal: Identify features with good univariate prediction performance.

Some feature selection methods

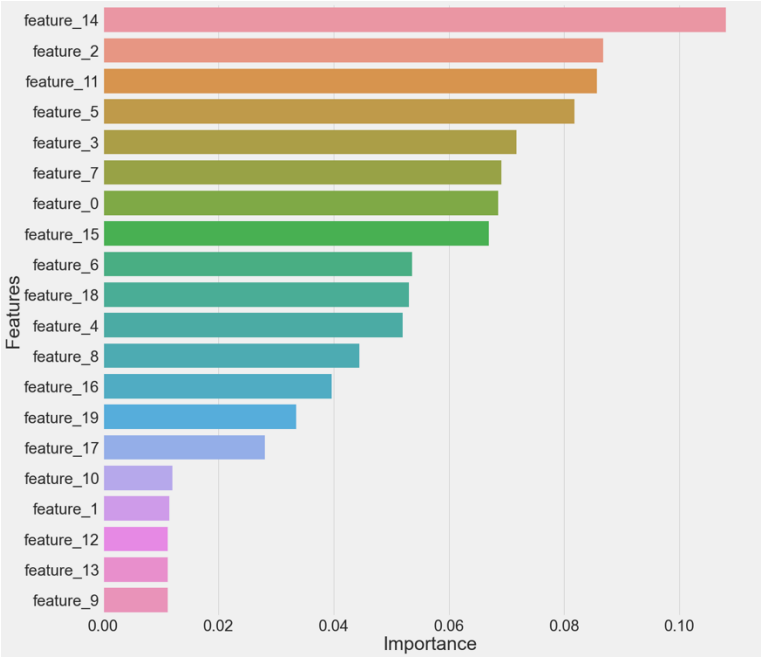
- Univariate analysis:
  - Compare each feature to the target variable and retain the most relevant ones
- Correlation analysis:
  - Identify and remove redundant features (which are strongly correlated with another feature



A pairplot can reveal redundancies between features.

Some feature selection methods

- Univariate analysis:
  - Compare each feature to the target variable and retain the most relevant ones
- Correlation analysis:
  - Identify and remove redundant features (which are strongly correlated with another feature
- Feature importance:
  - Identify the most important features using a machine learning model like random forests or gradient boostin



Visualization of the importance of different features in a random forest. Source

Some feature selection methods

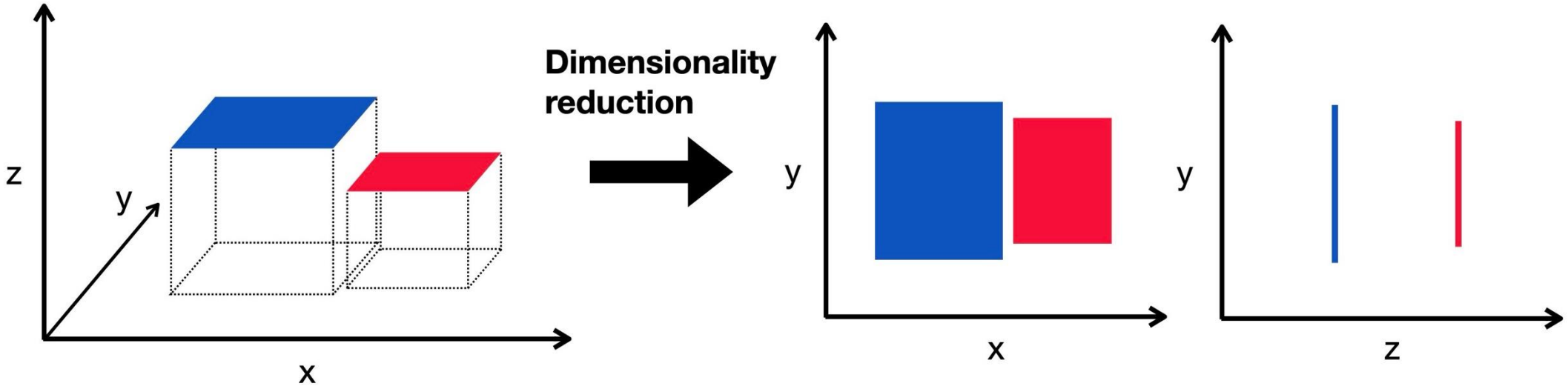
- Univariate analysis:
  - Compare each feature to the target variable and retain the most relevant ones
- Correlation analysis:
  - Identify and remove redundant features (which are strongly correlated with another feature
- Feature importance:
  - Identify the most important features using a machine learning model like random forests or gradient boosting
- Recursive feature elimination (RFE):
  - Recursively remove the least important features until the desired number of features is reached

The single steps for recursive feature.

Linear discriminant analysis (LDA)

Dimensionality reduction through projection

- Projection-based dimensionality reduction involves mapping high-dimensional data onto a lower-dimensional subspace while preserving as much of the data's structure as possible.

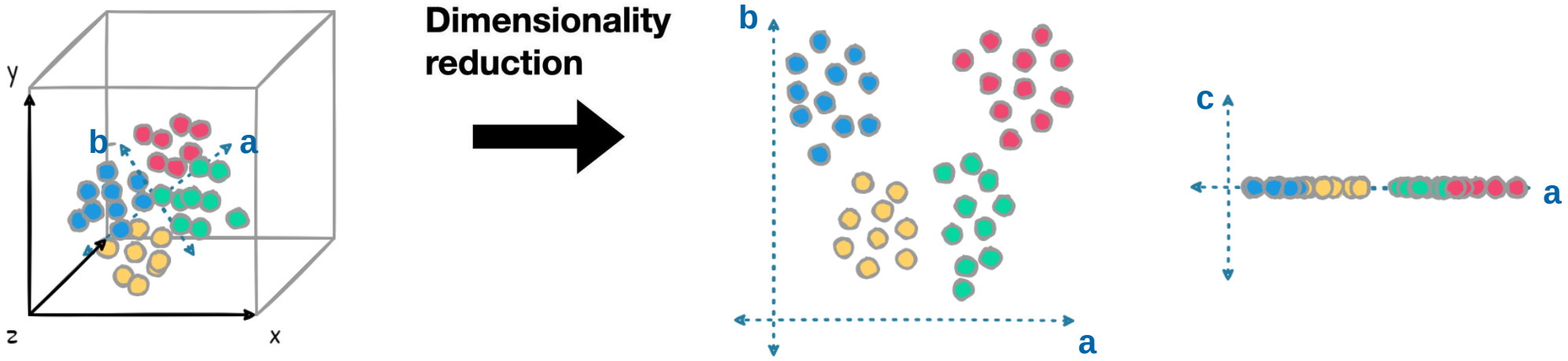


Two possible projections from a 3D space onto a 2D subspace.
Here, the 2D subspaces are the xy-plane and the yz-plane.

Source of illustration: Link

Dimensionality reduction through projection

- Projection-based dimensionality reduction involves mapping high-dimensional data onto a lower-dimensional subspace while preserving as much of the data's structure as possible.

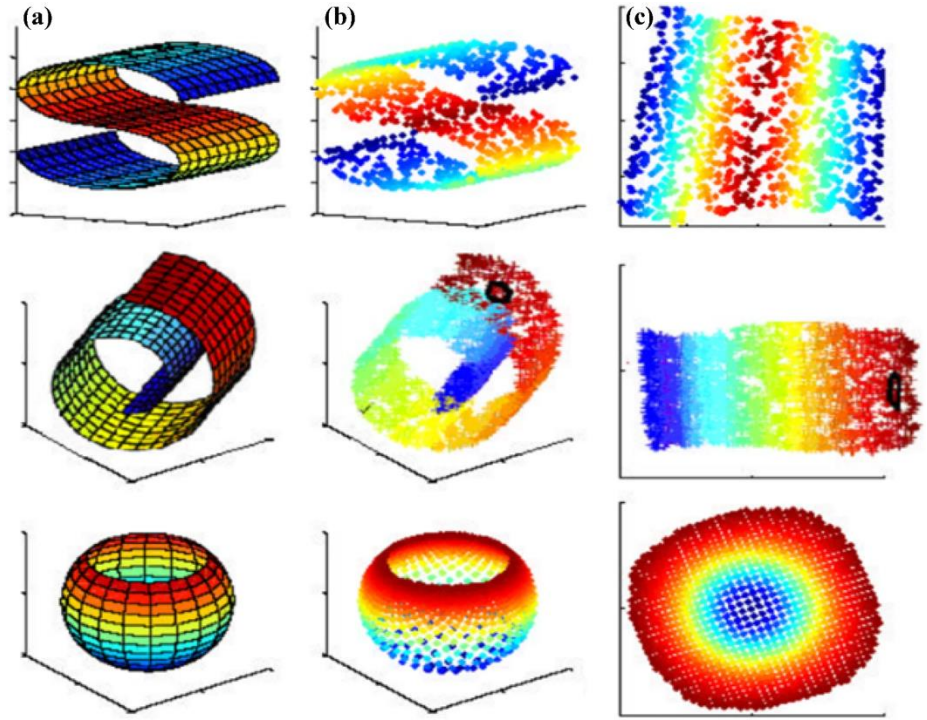


Two possible projections from a 3D space onto a 2D subspace. Here, the first subspace is the plane in which the data points lie (plane with the axes a-b). The second one is the plane a-c, which is perpendicular to the a-b plane.

Source of illustration: Link

Dimensionality reduction through projection

- We can generalize this idea:
  - ...from high-dimensional space with $D_{full} > 3$ dimensions
  - ...to low-dimensional space with D_{red} dimensions
  - ...using linear or non-linear transformations
- We'll skip the details here! The math is getting too hard... ☺



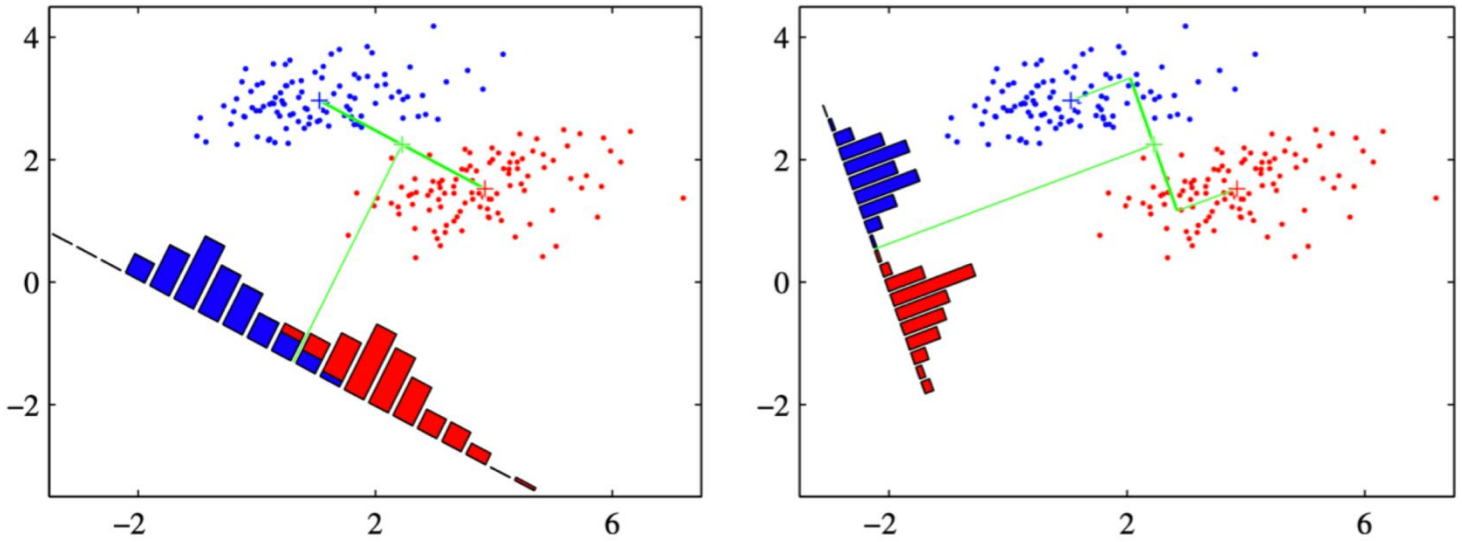
Examples of non-linear dimensionality reduction (using a method called LLE). Source: Link

Fisher's linear discriminant technique

Central idea:

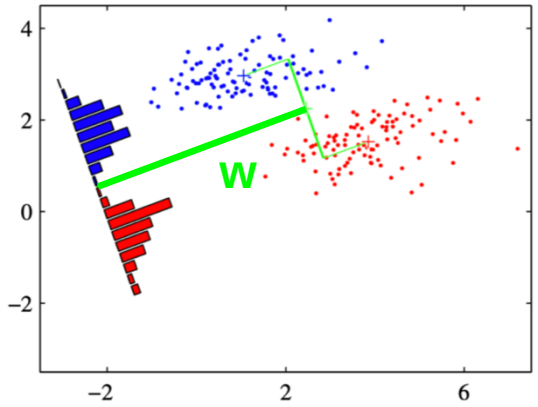
- Project the feature space onto a single discriminant dimension using a **linear transformation**.
- Aim to find the transformation that **maximizes the separation** between the classes.

Two projections from 2D onto a 1D subspace. The green lines show the direction of the projection. The histogram summarizes how the samples (blue and red dots) are projected onto this subspace. We can see that the projection on the right side is more favorable if we want to ensure a good separation between the two classes (blue and red dots).



Separation criteria

- **Assumption**: 2D binary classification problem
- **Aims**: Find a projection direction **w**, such that that the classes are separated as much as possible

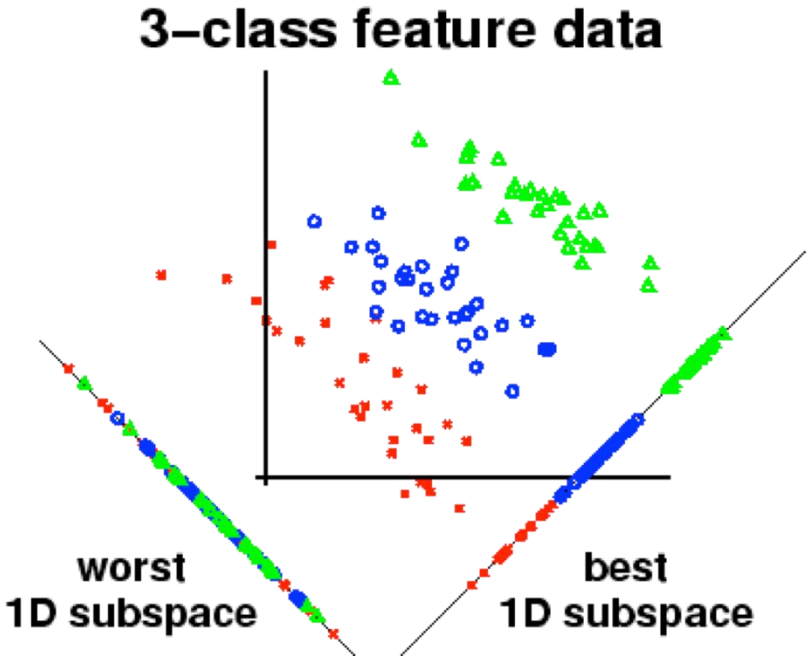


- **Fisher's approach**:
  - Maximize the distance between the projected class means (μ_1 and μ_2)
  - Minimize the spread within each class (within-class scatter matrices: Σ_1 and Σ_2)
- **Solution**:
  - All this can be formulated as an eigenvalue problem.
  - It can be shown that the optimal solution is given by: $\vec{w} = (\Sigma_1 + \Sigma_2)^{-1}(\vec{\mu}_2 - \vec{\mu}_1)$
  - μ_1, μ_2 are the class means, and Σ_1, Σ_2 are the within-class scatter matrices

$$\Sigma_k = \sum_{x \in C_k} (x - \mu_k)(x - \mu_k)^T$$

Linear discriminant analysis (LDA)

- LDA is a generalization of Fisher's linear discriminant to more than two classes.
- But adds the assumption that the independent variables are normally distributed.



- LDA is often used for dimensionality reduction before later classification.

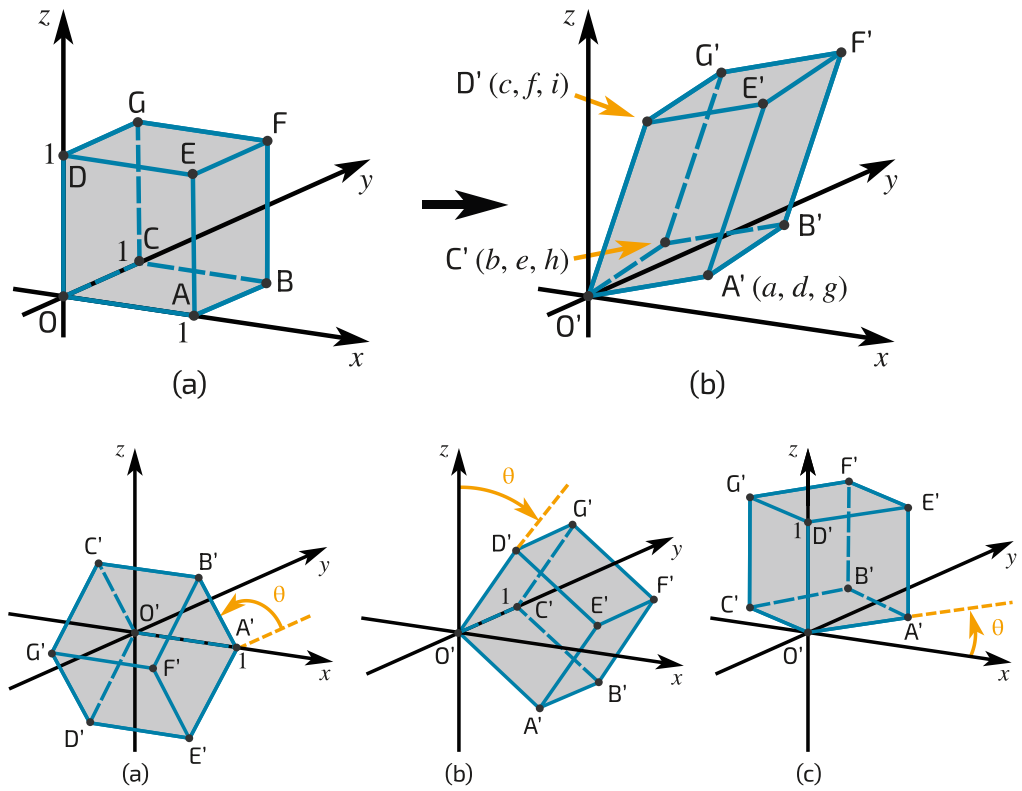
Principal Component Analysis (PCA)

Recap: Linear transformations

- Linear transformations map data points from one space to another by transforming the axes of the original space into new axes.
- A linear transformation (of vector spaces) can be represented mathematically by a matrix multiplication

$$\vec{x}' = A \cdot \vec{x}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$



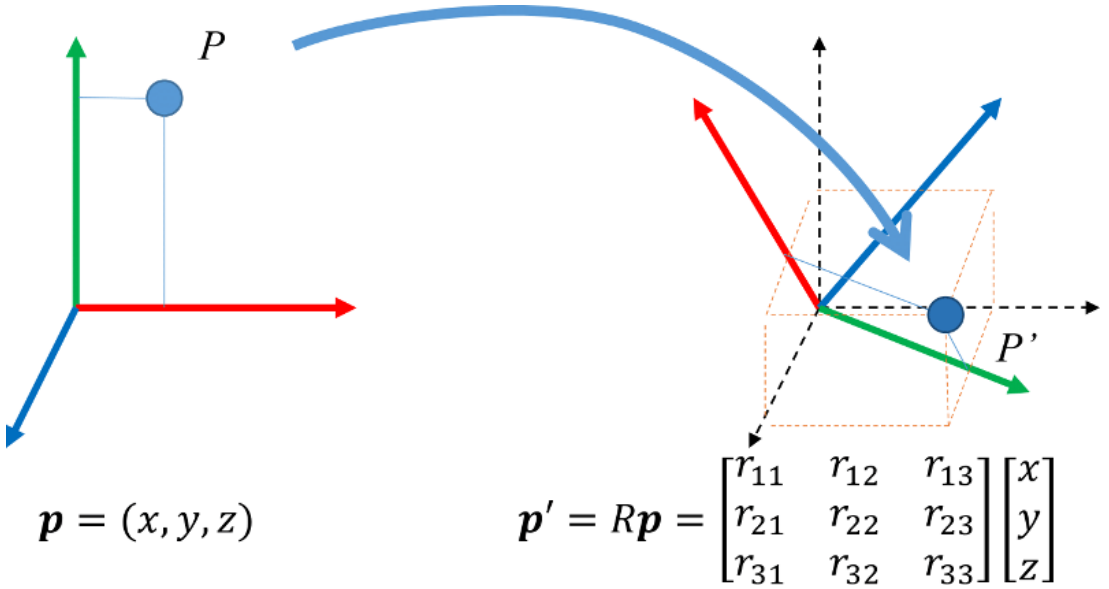
Examples of linear transformations in 3D. With a linear transformation, we can rotate, scale or shear the space. Note how the transformation acts on the coordinate axes. (Translation can be also taken care of, but requires a small trick)

Recap: Linear transformations

- Linear transformations map data points from one space to another by transforming the axes of the original space into new axes.
- A linear transformation (of vector spaces) can be represented mathematically by a matrix multiplication

$$\vec{x}' = A \cdot \vec{x}$$

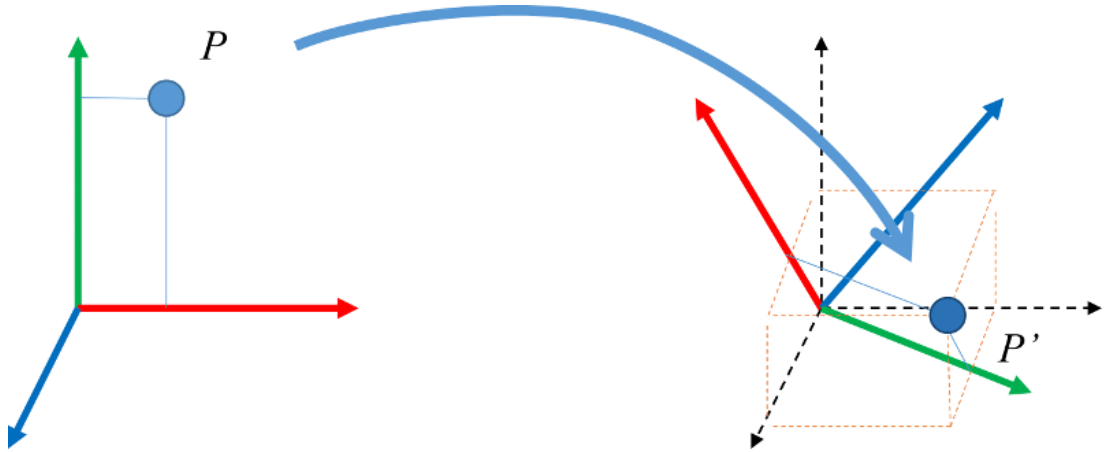
$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$



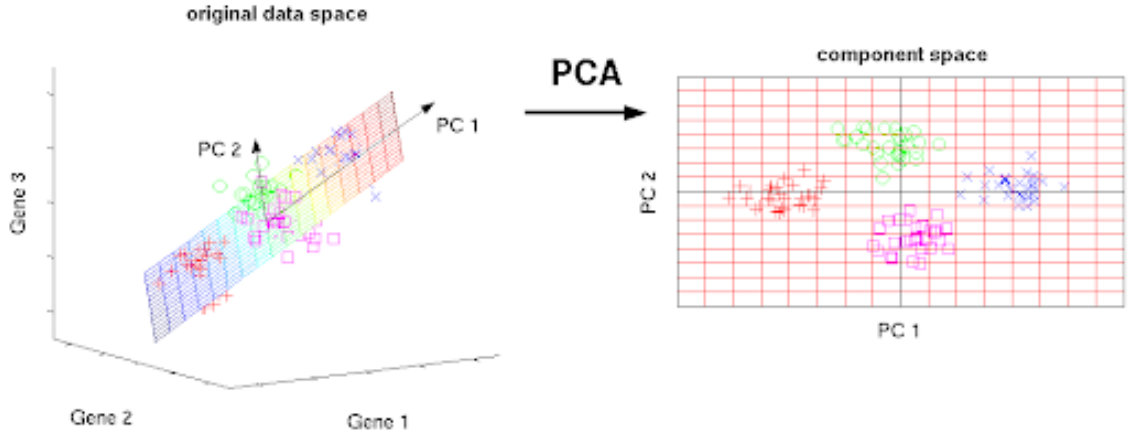
Example: After a rotation, the coordinates of the transformed point (relative to the original axes) are determined via a matrix multiplication with the rotation matrix R.

What is PCA?

- Another transformation-based dimensionality reduction technique
  - Linear transformation of the feature space ⇔ Map the coordinate axes to new directions
  - The new axes after transformation are called principal components (PCs)



- **Objective**: Find the transformation such that the components capture the most variance in the data



- **Result**: A list of principal components, ordered by the amount of variance they see in the data.

How to compute the PCA

- (...well, as always, we will just use a scikit-learn function...)

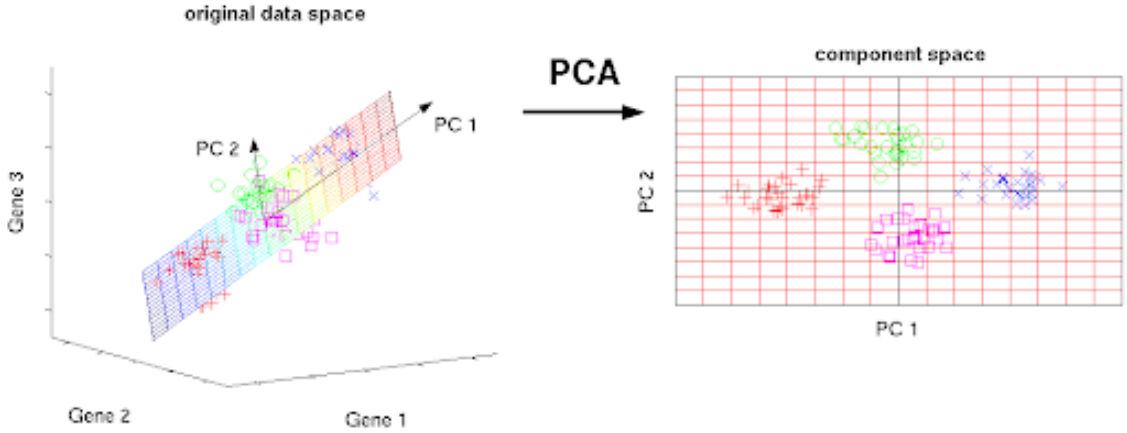
- **Steps**:
  - Standardize the data: ensure each feature has zero mean and unit variance to eliminate scale differences
  - Compute the covariance matrix (X is the standardized feature matrix)

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}^\top\mathbf{X}$$

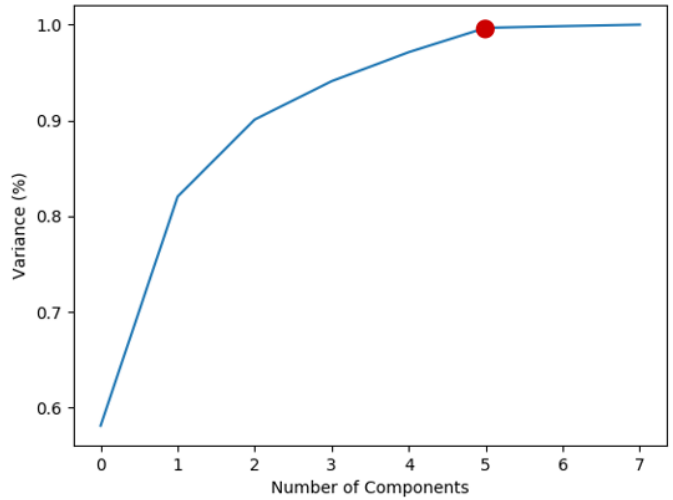
  - Perform an eigenvalue decomposition, i.e., compute:
    - Eigenvectors: Those are the principal components
    - Eigenvalues: Capture the variance captured per principal component
  - Sort principal components by their eigenvalue
  - Select the number of principal components to include
  - Project the feature space onto the resulting subspace

How is PCA used?

- The first few principal components often capture most of the **essential information**, allowing dimensionality reduction.
- Dimensions with little information can be excluded

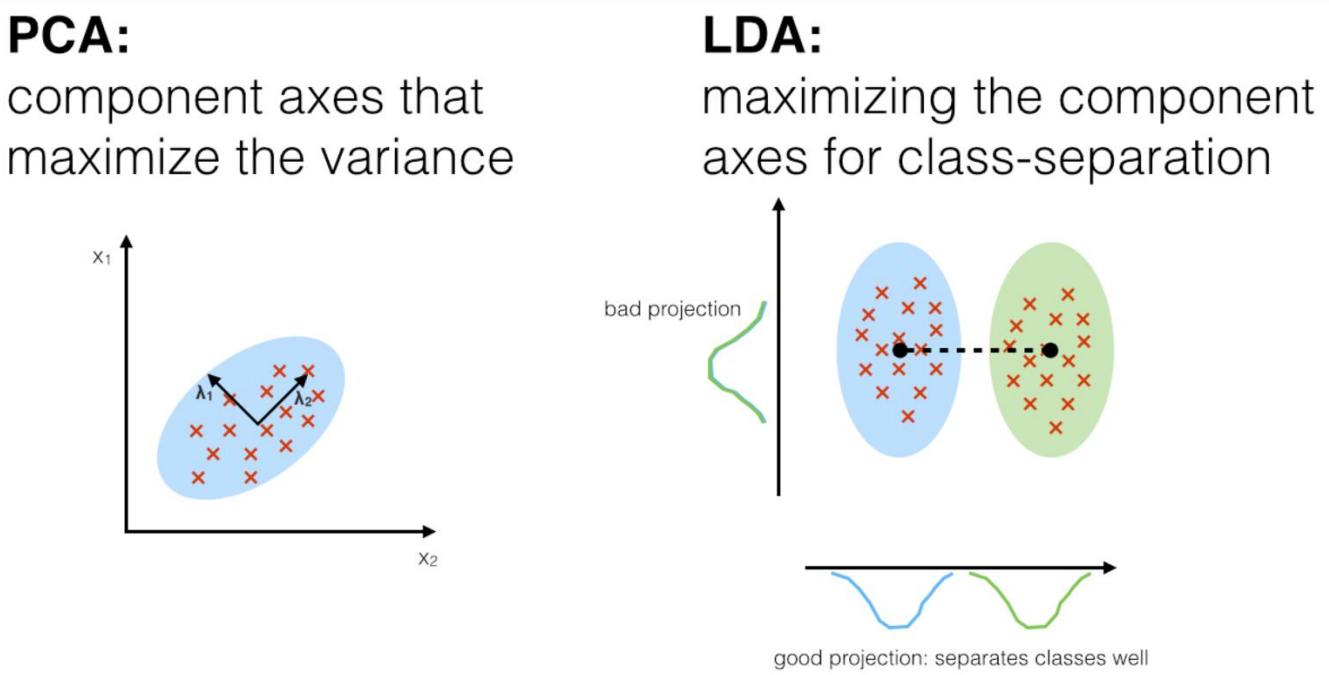


- **How many dimension to retain?**
  - Plot the cumulative explained variance (see plot on the right) against the number of PCs
  - Keep adding PCs until the cumulative explained variance surpasses a threshold
  - It is common to set a threshold of 95% or 99%
  - (See notebook for details!)



LDA versus PCA

- Principal Component Analysis (PCA) uses variance to separate dimensions, it is an example of unsupervised learning (no labels required).
- LDA uses label information to separate distributions of different classes (an example of supervised learning).



LDA versus PCA

- Principal Component Analysis (PCA) uses variance to separate dimensions, it is an example of unsupervised learning (no labels required).
- LDA uses label information to separate distributions of different classes (an example of supervised learning).

Aspect	PCA	LDA
Objective	Maximize variance in data	Maximize class separability
Supervised/Unsupervised	Unsupervised	Supervised (requires labels)
Output	Components ranked by variance	Linear discriminants for class separation
Focus	General data structure	Class discrimination
Applications	Data compression, noise reduction	Classification, pattern recognition

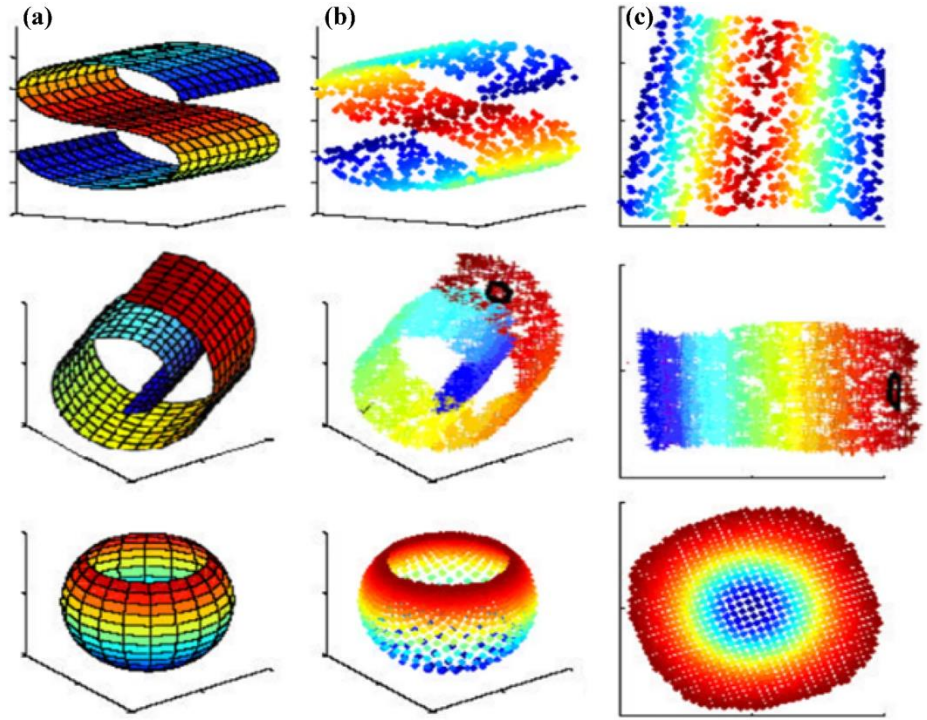
Non-linear techniques

Non-linear dimensionality reduction techniques

- Sometimes, linear transformations are not sufficient

Popular Non-Linear Methods

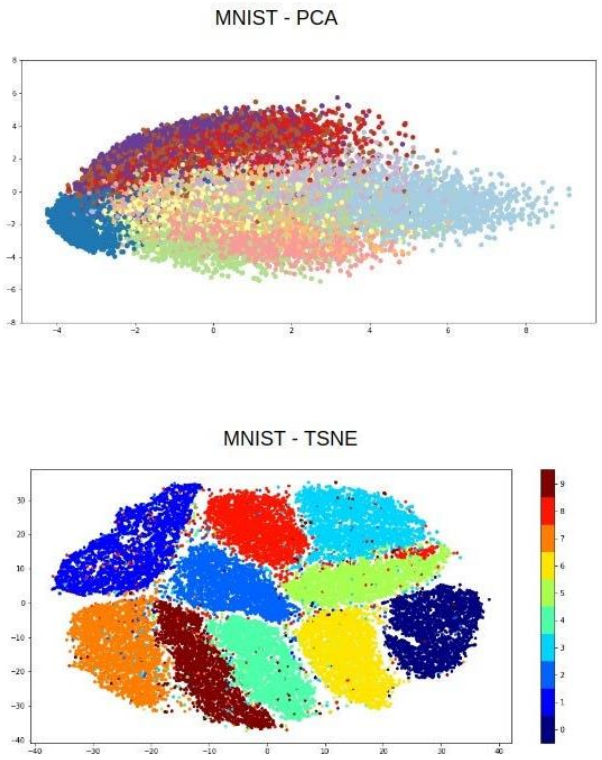
Technique	Key Feature	Applications
t-SNE	Preserves local relationships for visualization	Clustering visualization
UMAP	Balances local and global structure; faster than t-SNE	Large-scale data visualization
Isomap	Preserves geodesic distances along the data manifold	Unveiling intrinsic manifolds
LLE (Locally Linear Embedding)	Retains local neighborhood relationships	Modeling curved manifolds



Examples of non-linear dimensionality reduction (using a method called LLE). Source: Link

Example: t-SNE

- t-SNE: t-distributed stochastic neighbor embedding
- The method is widely used for exploring data but not for downstream machine learning tasks.
- Core ideas:
  - Map data points into a low-dimensional space while preserving the local structure (e.g., clusters or neighborhoods).
  - High-dimensional similarities are modeled using Gaussian distributions.
  - Low-dimensional similarities are modeled using t-distributions.
  - The mapping minimizes the difference (Kullback-Leibler divergence) between these distributions.
- Advantages:
  - Very useful for visualization of high-dimensional spaces
  - May reveal natural clusters or groupings.



Example: MNIST dataset reduced to 2 dimensions. Upper plot: Using PCA, lower plot: Using t-SNE. Link

Feature selection	**Dimensionality reduction**
Retains original features.	Creates new features by transformation.
Improves interpretability.	May reduce interpretability.
Examples: RFE, Lasso.	Examples: PCA, t-SNE.

Summary

Further ~~reading~~ watching

- StatQuest: PCA main ideas in 5 minutes
- StatQuest: Principal component analysis (22 min)
- StatQuest: PCA - practical tips (8 min)
- Mathematische Grundlagen von D. Jung erklärt: Link