Introduction to Machine Learning
[Fall 2022]

Dimensionality Reduction

November 10, 2022

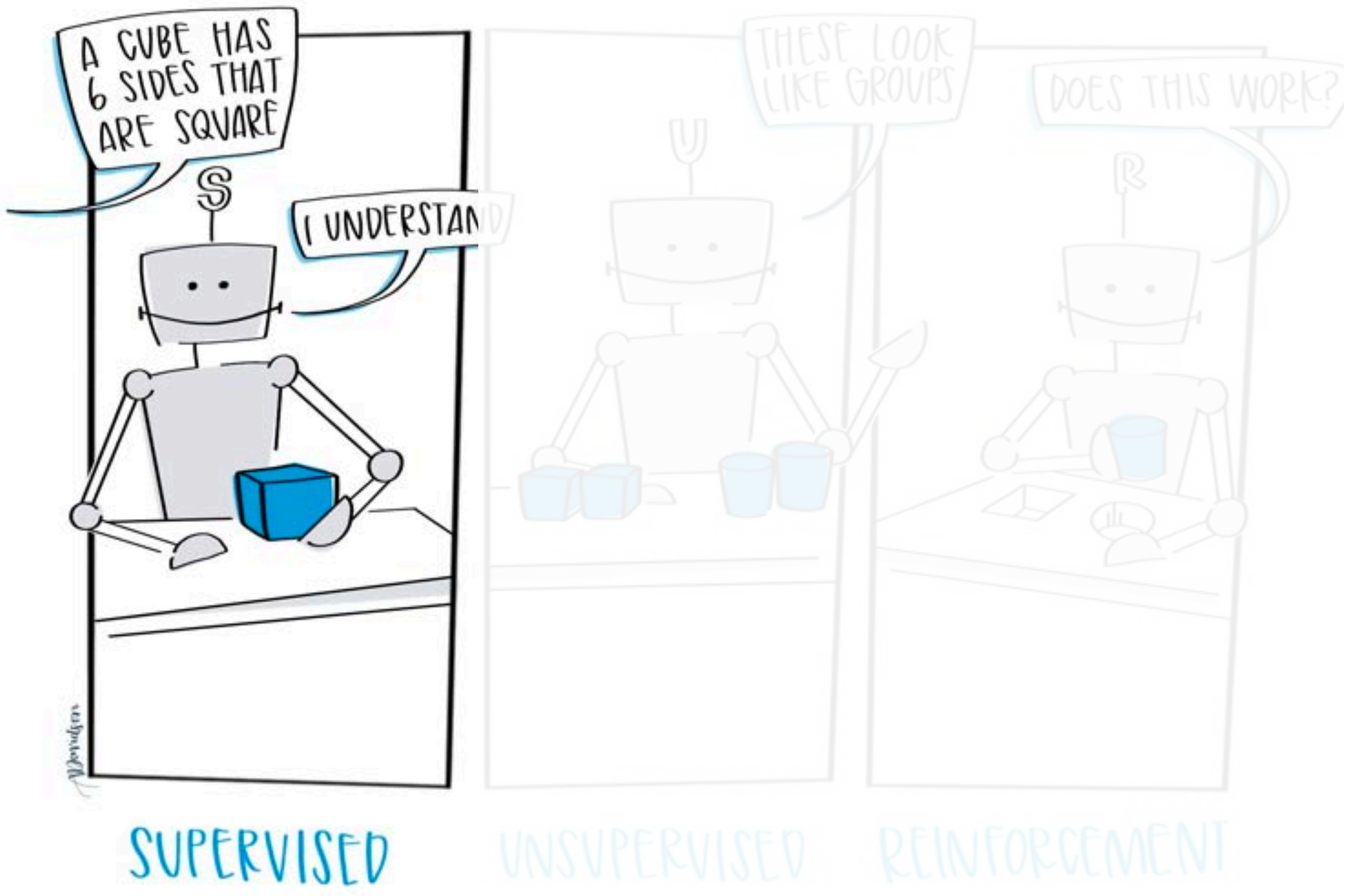
Lerrel Pinto

Logistics

- We are nearing the end of the class!
  - HW5 & HW6.
  - Capstone Project.
- Syllabus changes:
  - https://nyu-robot-learning.github.io/ml-class/docs/lectures
- Want something covered in later classes? Mention on #class-discussion

What we have covered so far

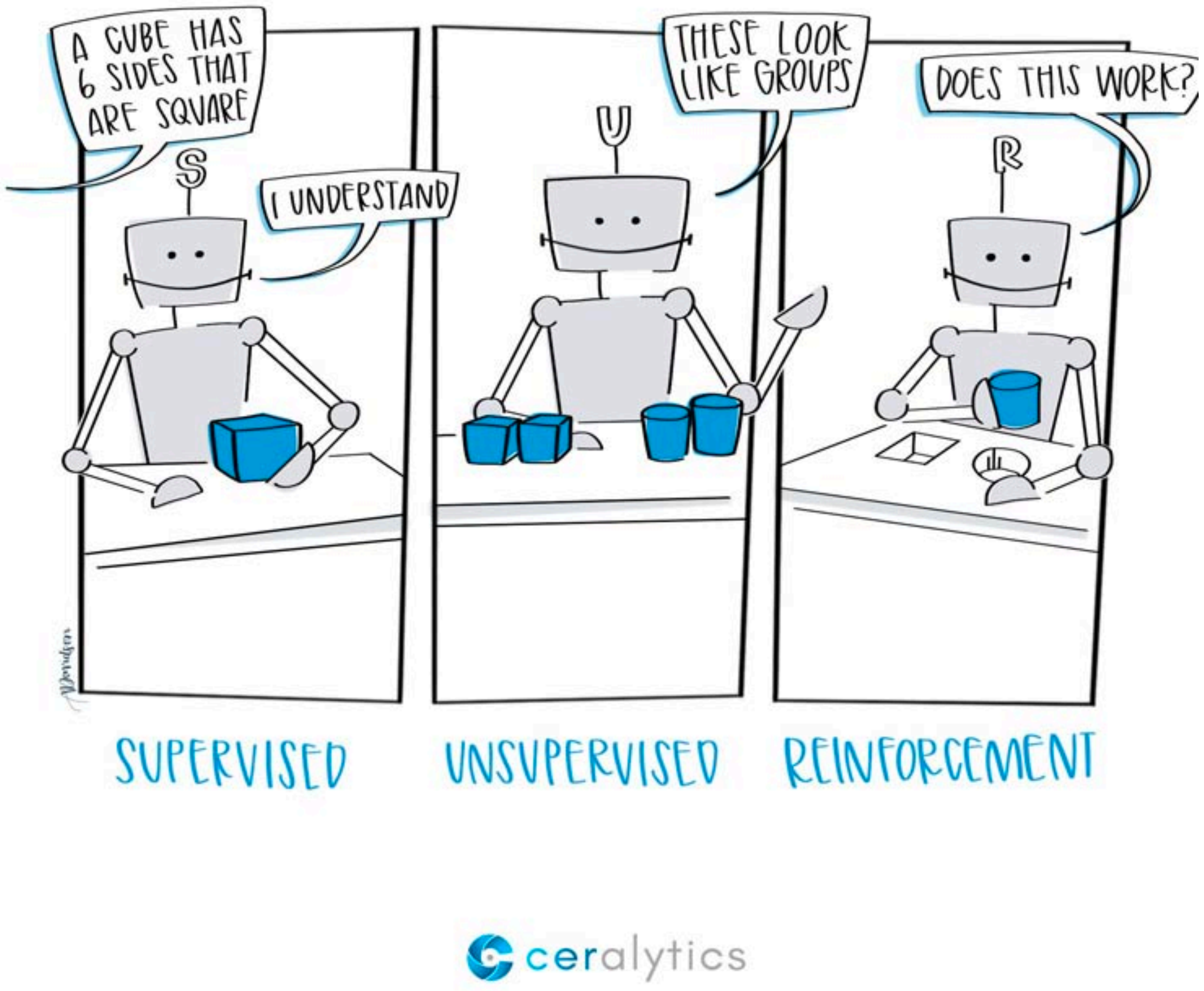
What we have covered so far



ceralytics

What we will cover in the remainder of the class

What we will cover in the remainder of the class



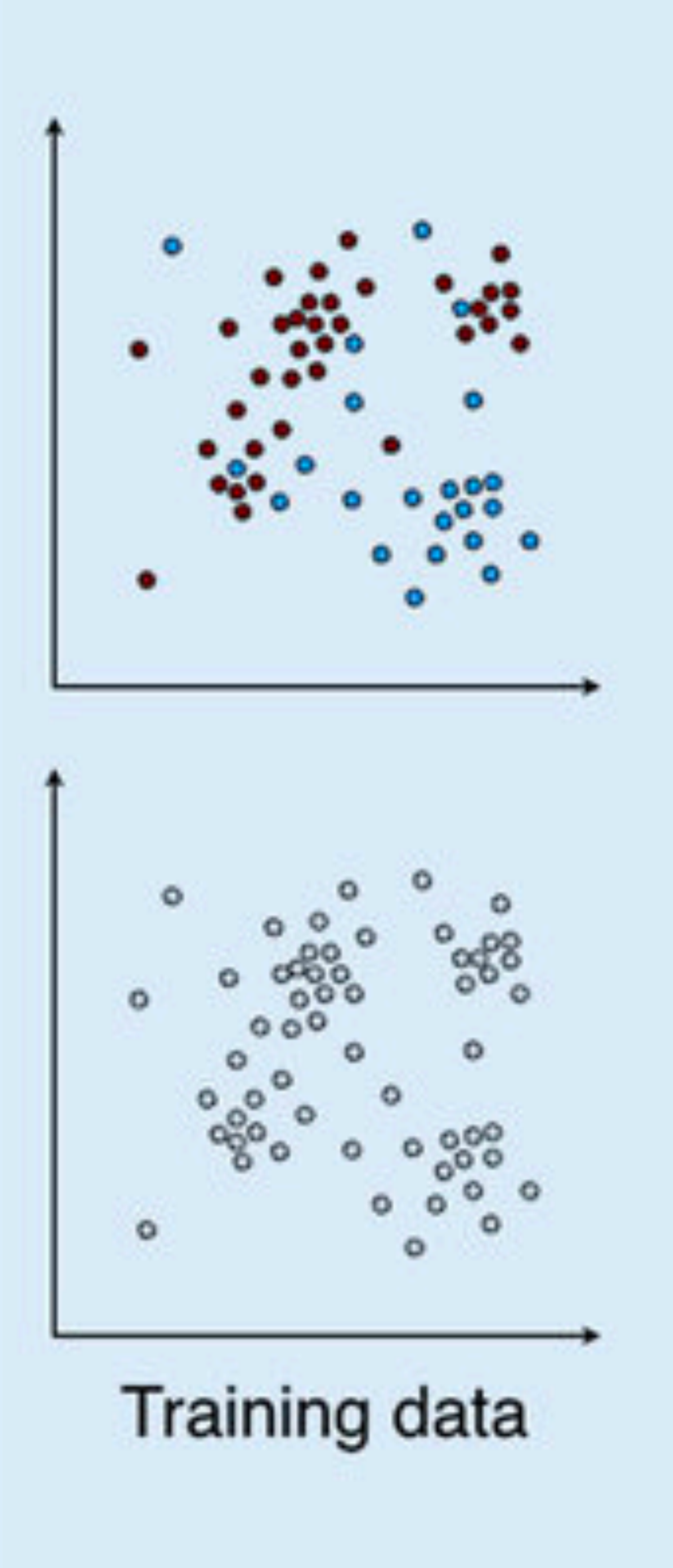
Unsupervised Learning

Credits: Langs et al. 2018

Unsupervised Learning

Supervised Learning

Unsupervised Learning

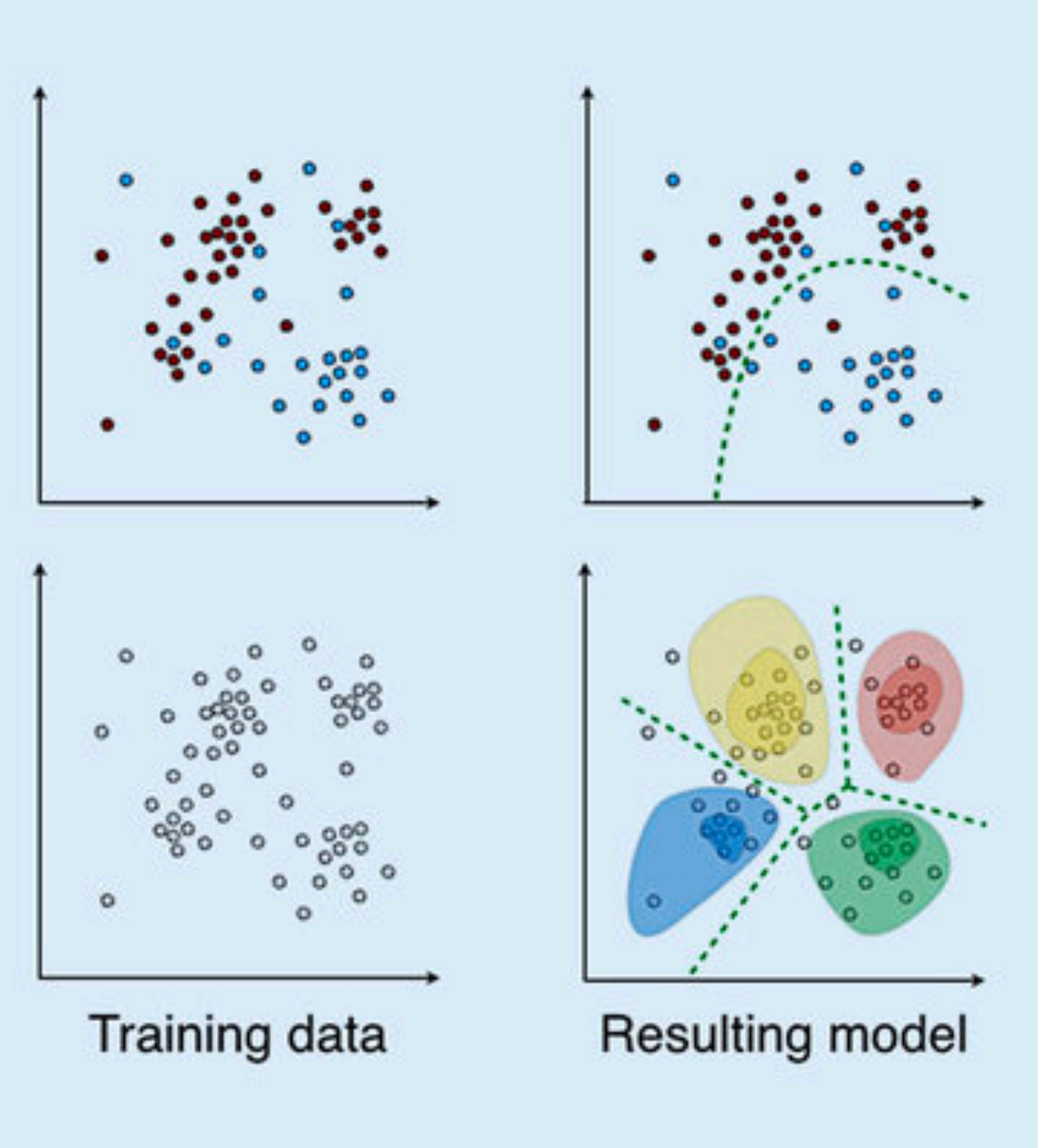


Credits: Langs et al. 2018

Unsupervised Learning

Supervised Learning

Unsupervised Learning



Credits: Langs et al. 2018

Dimensionality Reduction

Dimensionality Reduction

- High-dimensional data can be repetitive:
  - Document classification - Thousands of words per document

Credits: Aarti Singh

Dimensionality Reduction

- High-dimensional data can be repetitive:
  - Document classification - Thousands of words per document
  - Netflix / Amazon review - number of users * number of shows

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6
Tom	5	?	?	1	3	?
George	?	?	3	1	2	5
Susan	4	3	1	?	5	1
Beth	4	3	?	2	4	2

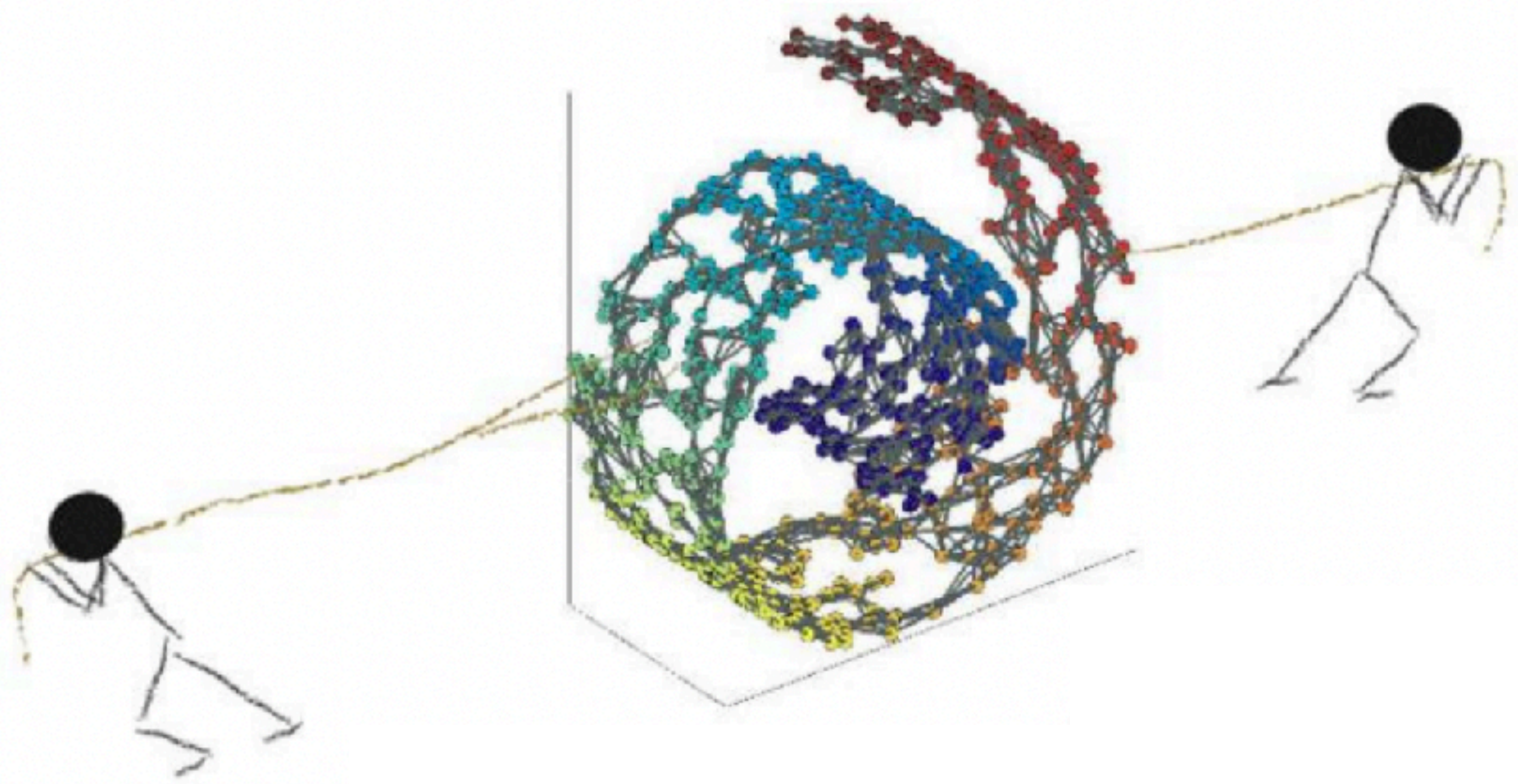
Credits: Aarti Singh

Dimensionality Reduction

- Curse of dimensionality:
  - Redundant / Irrelevant features degrade performance. (Signal to noise)
  - Difficult to interpret and visualize.
  - Computation becomes infeasible for some algorithms.
  - Hard to store high dimensional data.

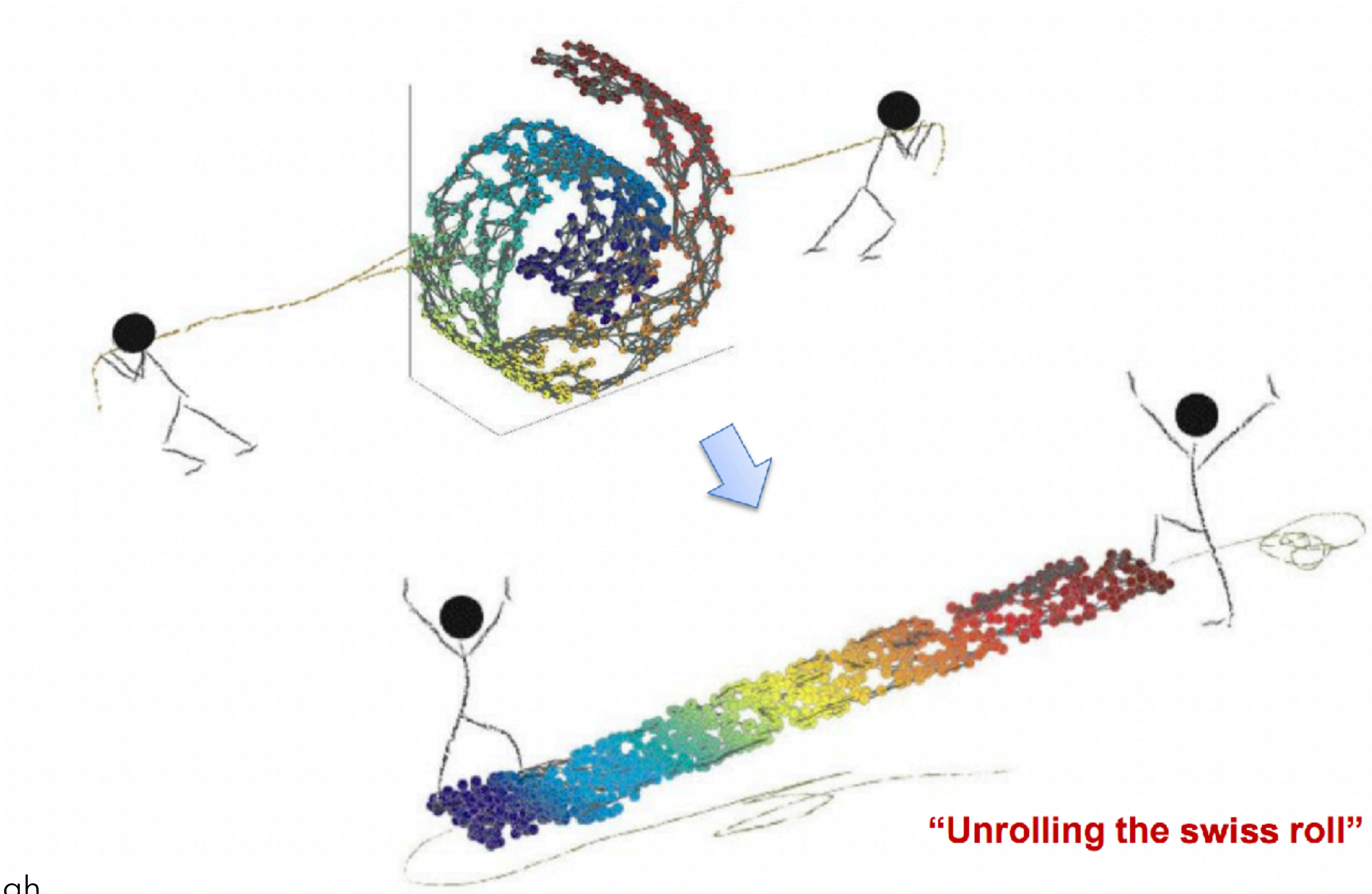
Credits: Aarti Singh, Jeff Howbert

Dimensionality Reduction



Credits: Aarti Singh

Dimensionality Reduction



Credits: Aarti Singh

Dimensionality Reduction

Credits: Aarti Singh, Jeff Howbert

Dimensionality Reduction

- Option 1: Feature selection
  - Cons: Requires domain knowledge.

Credits: Aarti Singh, Jeff Howbert

Dimensionality Reduction

- Option 1: Feature selection
  - Cons: Requires domain knowledge.
- Option 2: Model Regularization
  - Cons: Requires supervised learning

Credits: Aarti Singh, Jeff Howbert

Dimensionality Reduction

- Option 1: Feature selection
    - Cons: Requires domain knowledge.
- Option 2: Model Regularization
    - Cons: Requires supervised learning
- Option 3: Learn mapping from high-D to low-D space
    - Either linear or non-linear mapping

Credits: Aarti Singh, Jeff Howbert

Linear Dimensionality Reduction

Credits: Aarti Singh, Jeff Howbert

Linear Dimensionality Reduction

- Goal: given a dataset: $X \in \mathbb{R}^{d \times n}$, where $\vec{x} \in \mathbb{R}^d$ corresponds to a data point. Project $\vec{x} \rightarrow \vec{z}$, where $\vec{z} \in \mathbb{R}^k$ and $k < d$

Credits: Aarti Singh, Jeff Howbert

Linear Dimensionality Reduction

- Goal: given a dataset: $X \in \mathbb{R}^{d \times n}$, where $\vec{x} \in \mathbb{R}^d$ corresponds to a data point. Project $\vec{x} \rightarrow \vec{z}$, where $\vec{z} \in \mathbb{R}^k$ and $k < d$
  - Usually $k << d$
  - Example: Project 1000 word features to 3 dimensions (horror, fiction, reading level)

Credits: Aarti Singh, Jeff Howbert

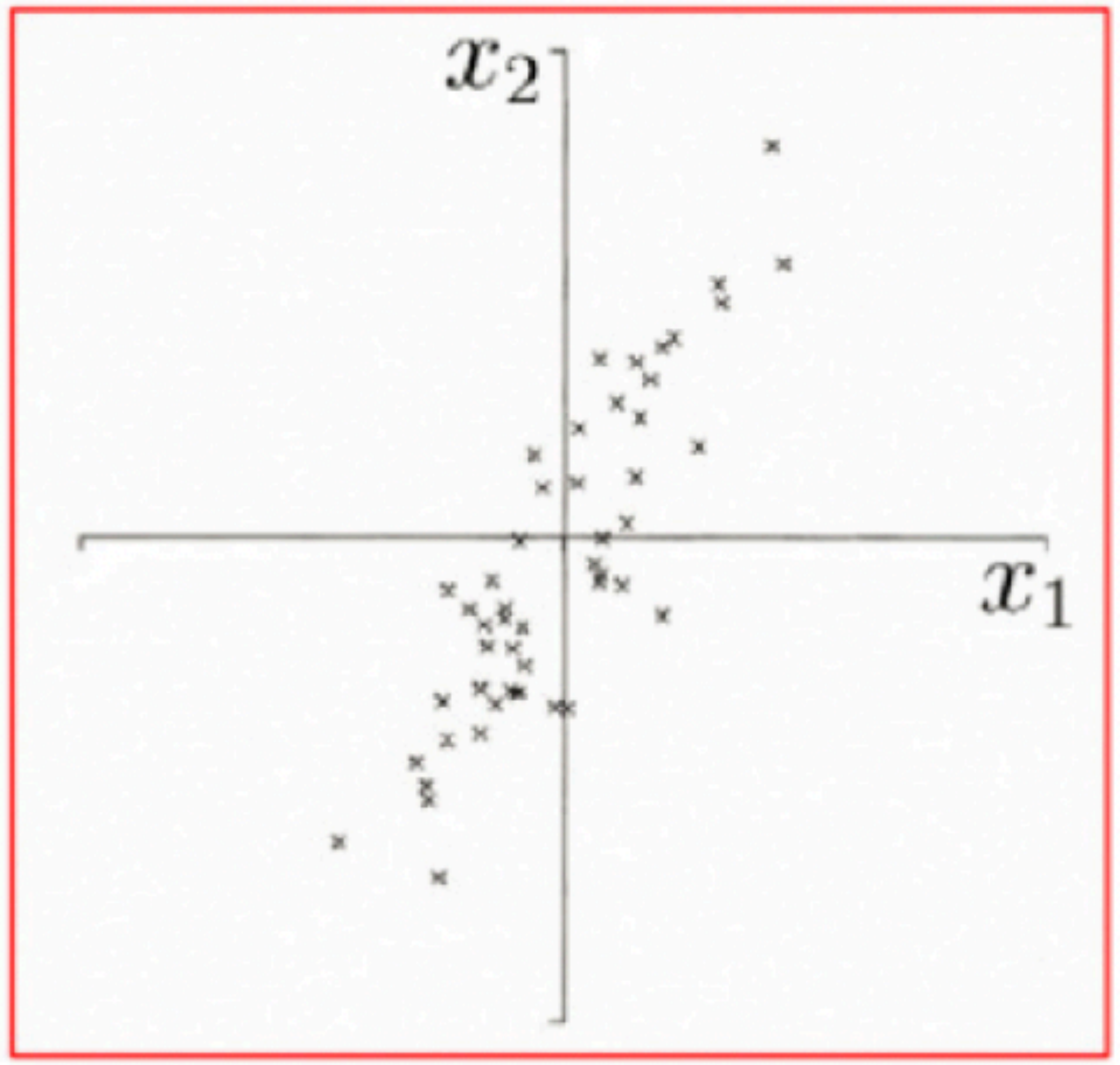
Linear Dimensionality Reduction

- Goal: given a dataset: $X \in \mathbb{R}^{d \times n}$, where $\vec{x} \in \mathbb{R}^d$ corresponds to a data point. Project $\vec{x} \rightarrow \vec{z}$, where $\vec{z} \in \mathbb{R}^k$ and $k < d$
  - Usually $k << d$
  - Example: Project 1000 word features to 3 dimensions (horror, fiction, reading level)
  - Linear: $\vec{z} = A\vec{x}$

Credits: Aarti Singh, Jeff Howbert

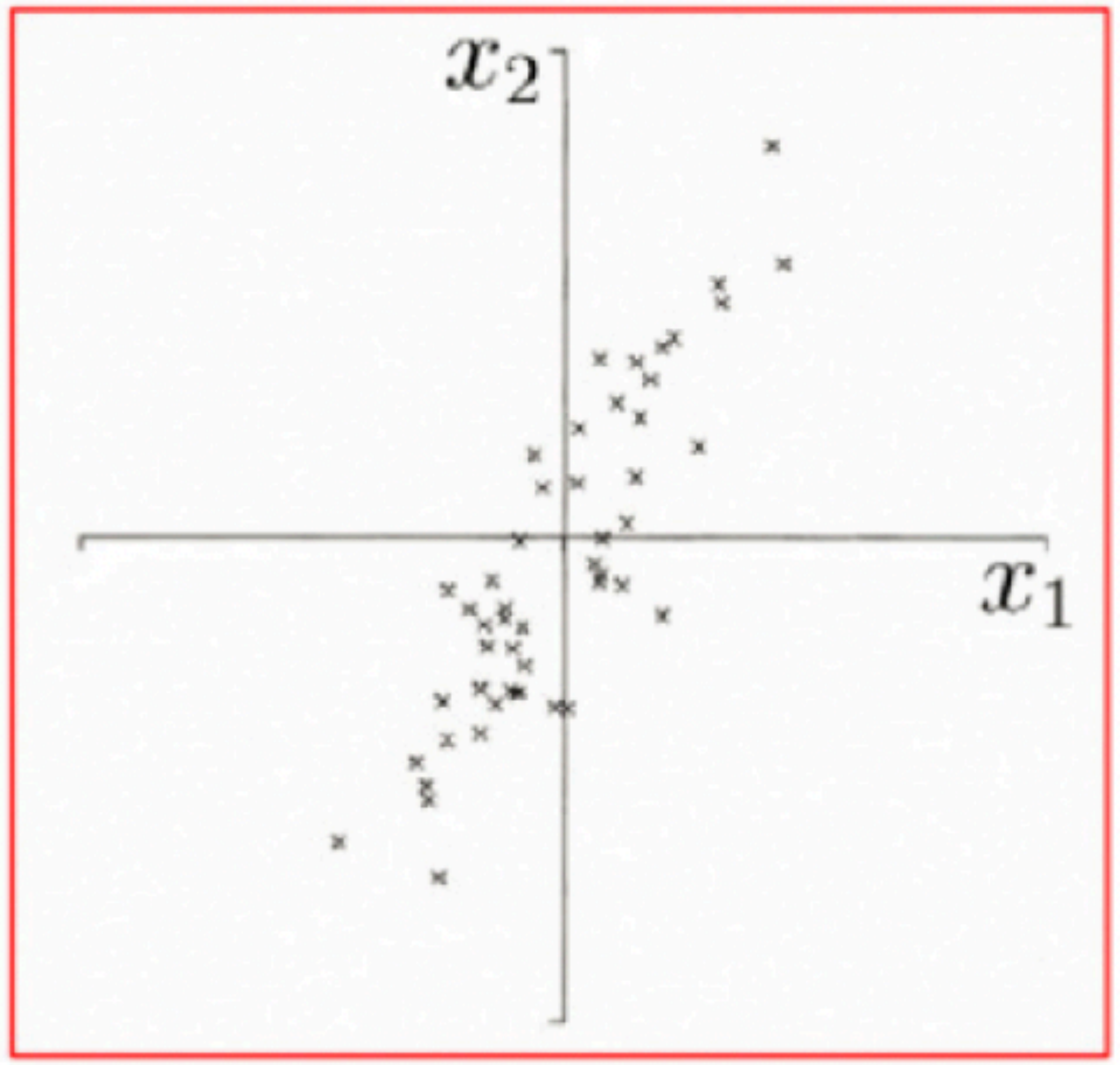
How to solve this? (Intuitively)

Principal Component Analysis (PCA)



Credits: Jeff Howbert

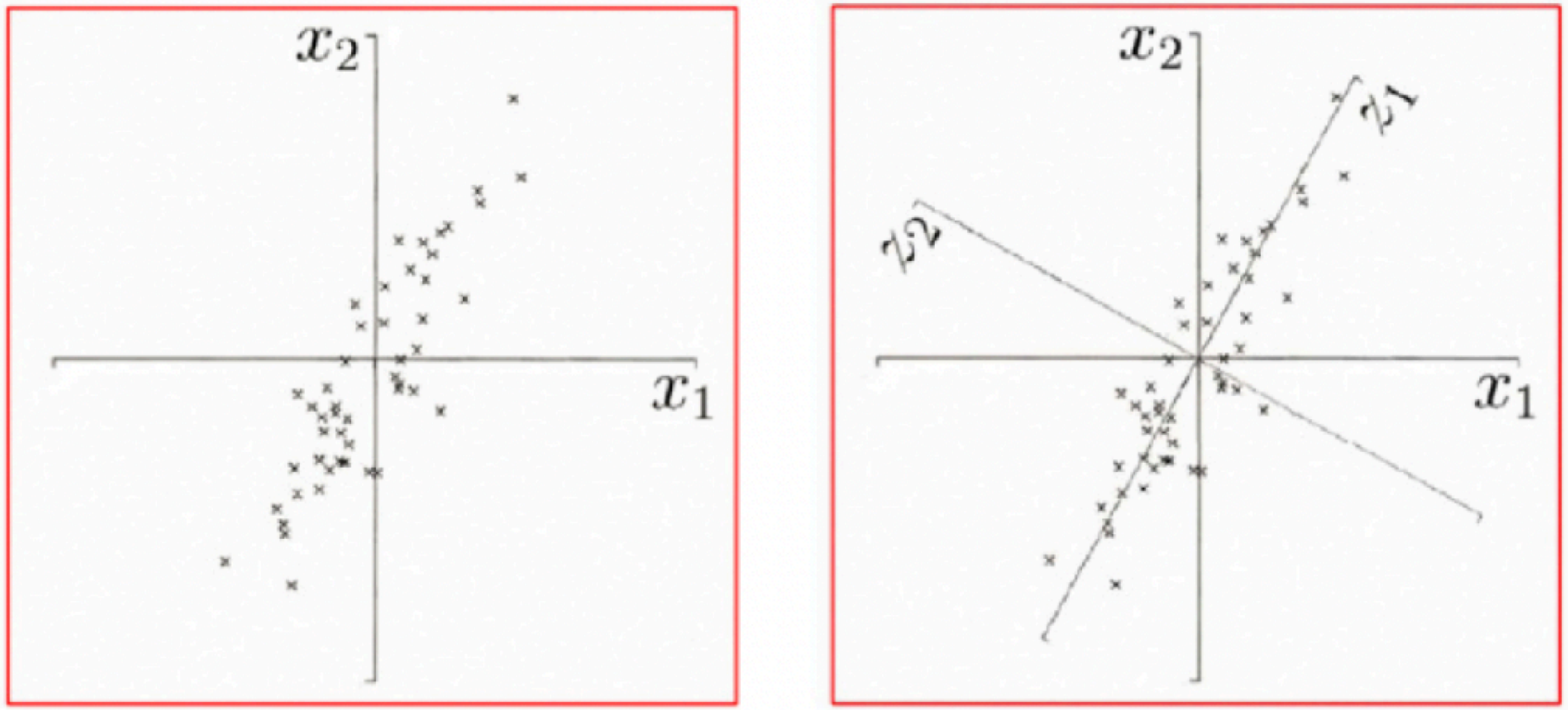
Principal Component Analysis (PCA)



- First PC: Projection direction that maximizes variance of data.
- Second PC: Orthogonal to first and maximizes variance of data.

Credits: Jeff Howbert

Principal Component Analysis (PCA)

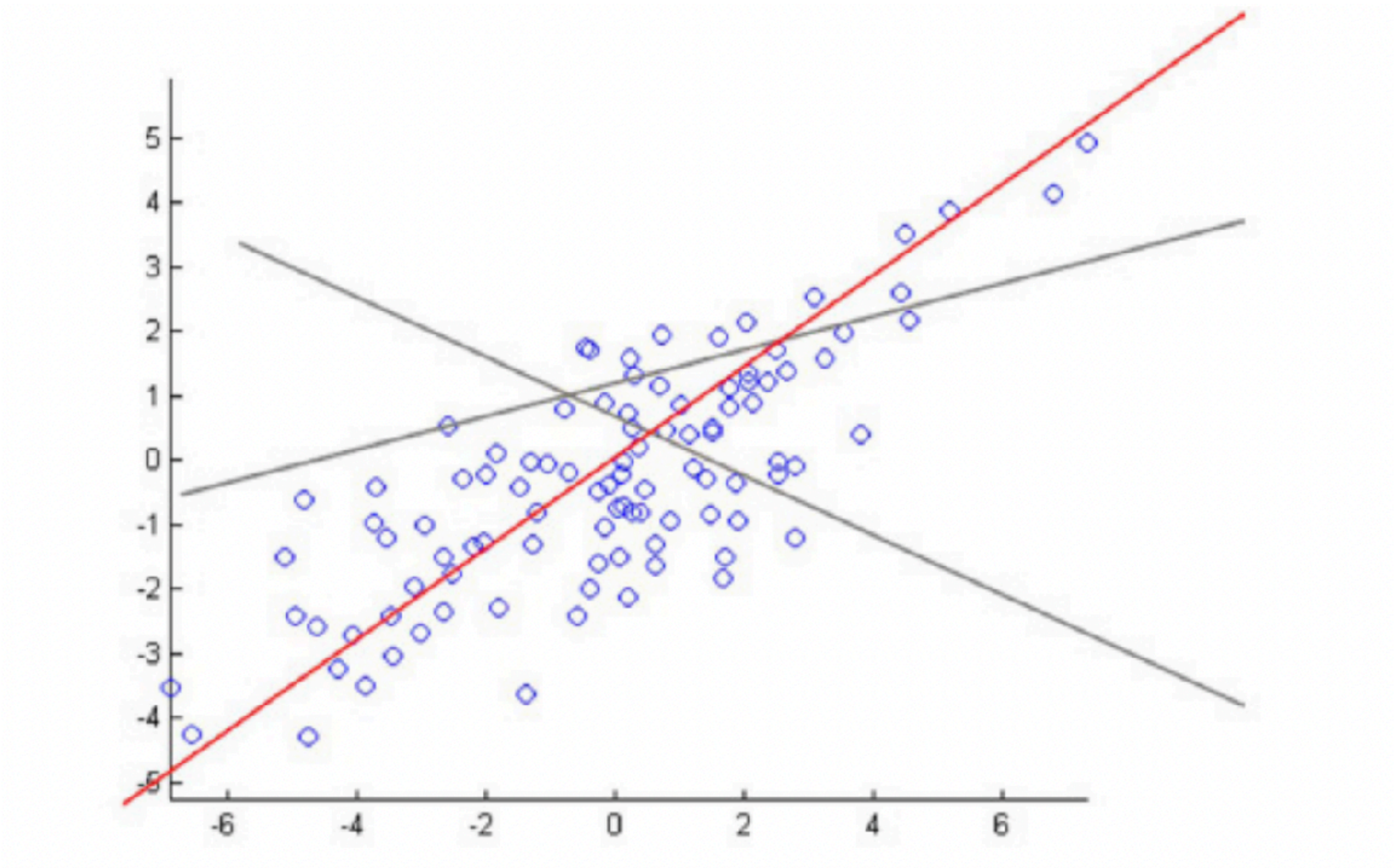


- First PC: Projection direction that maximizes variance of data.
- Second PC: Orthogonal to first and maximizes variance of data.

Credits: Jeff Howbert

Principal Component Analysis (PCA)

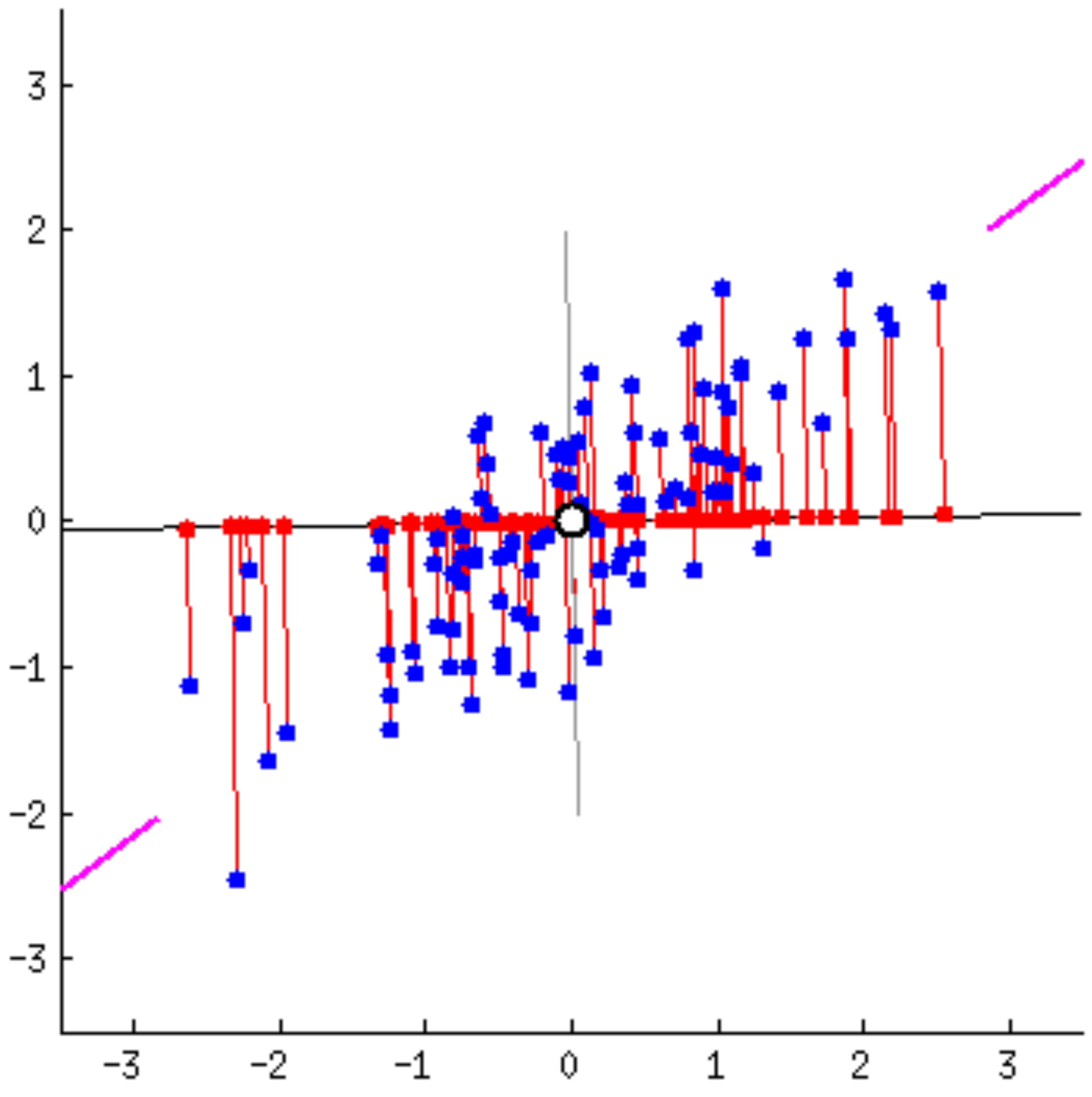
Step 1



Credits: Jeff Howbert

Principal Component Analysis (PCA)

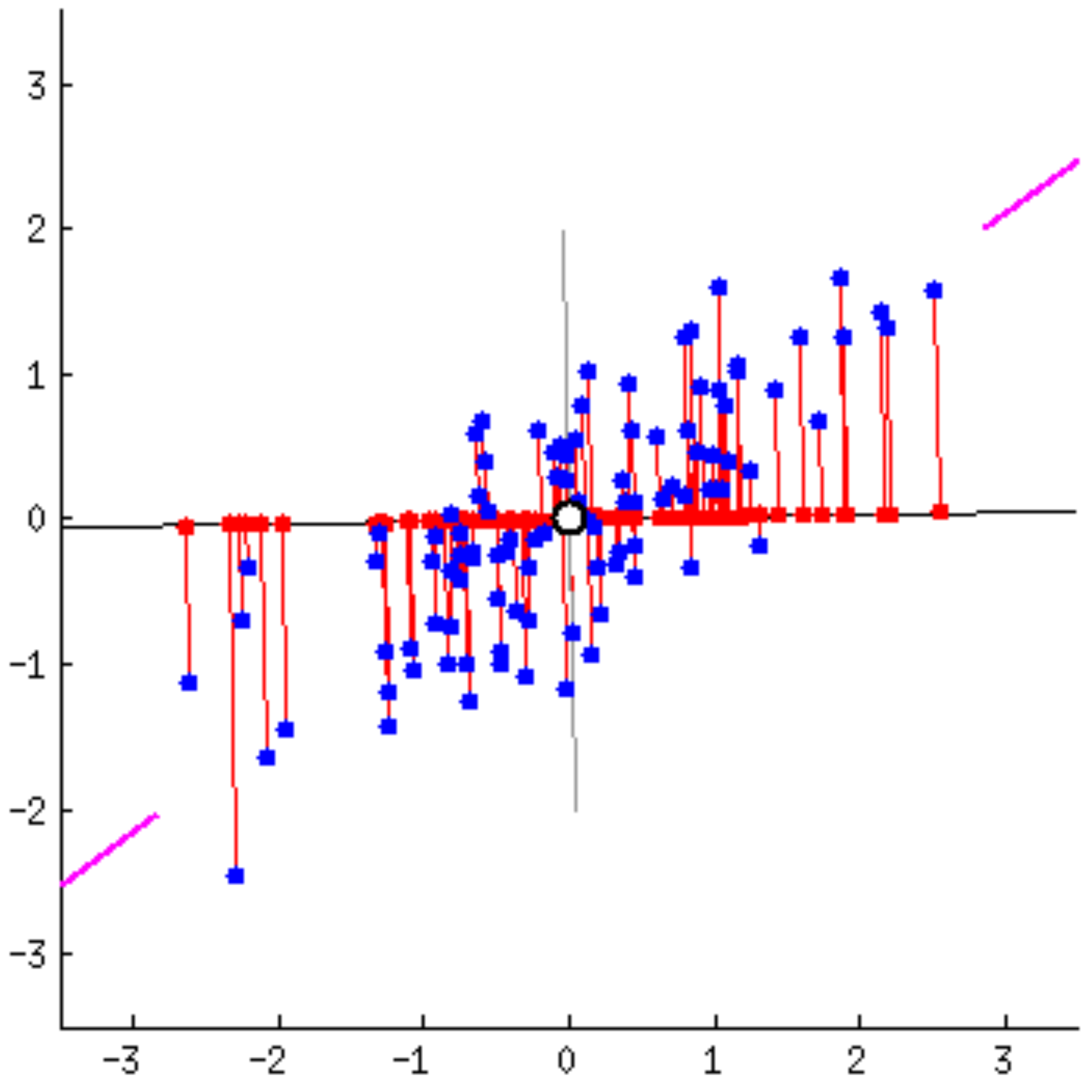
Step 1



Credits: https://builtin.com/data-science/step-step-explanation-principal-component-analysis

Principal Component Analysis (PCA)

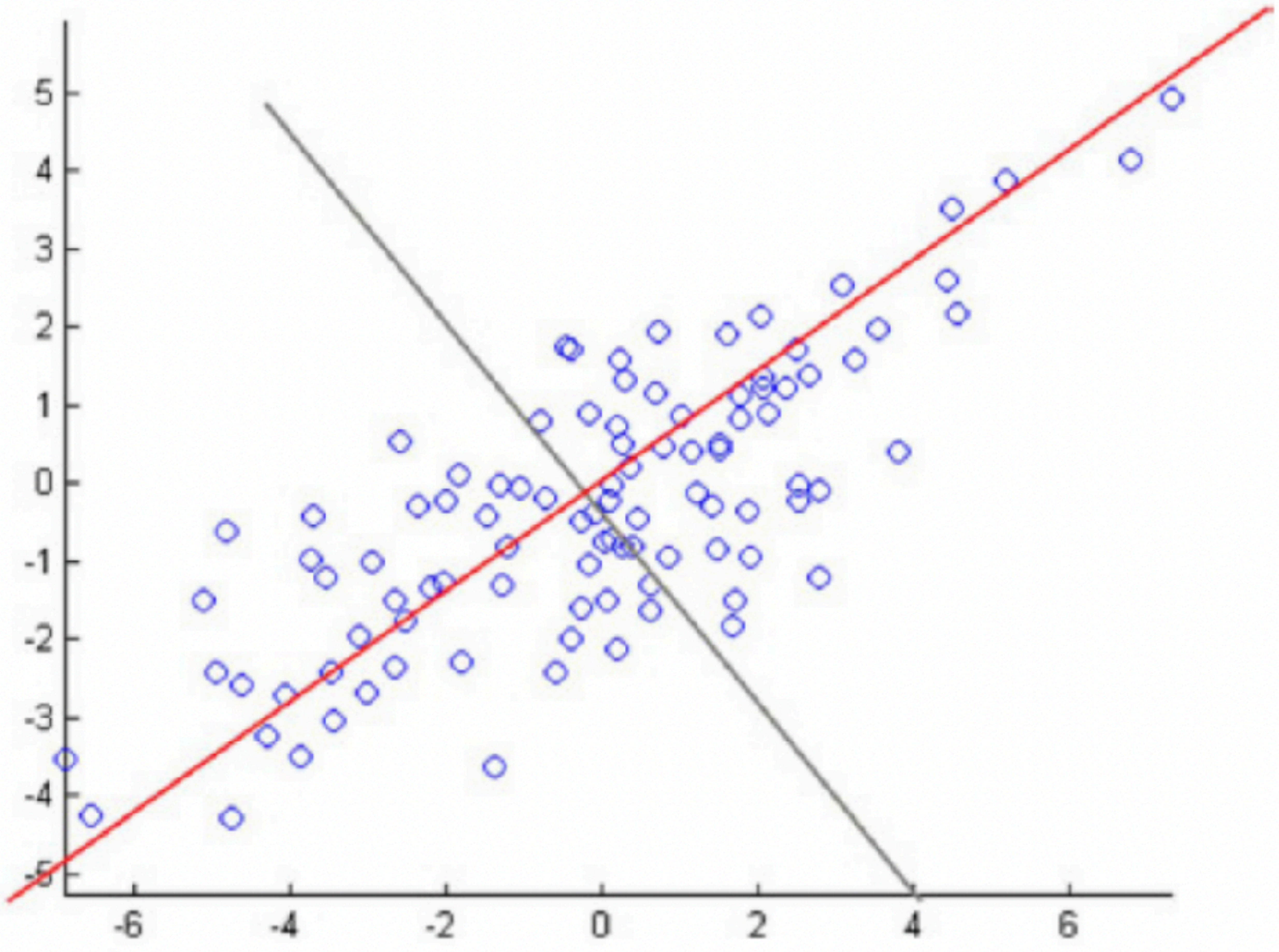
Step 1



Credits: https://builtin.com/data-science/step-step-explanation-principal-component-analysis

Principal Component Analysis (PCA)

Step 2



Credits: Jeff Howbert

Principal Component Analysis (PCA)

Interactive tool:

https://setosa.io/ev/principal-component-analysis/

How to do PCA Efficiently?

How to do PCA Efficiently?

- Step 1: Make the mean of the data zero (Standardize your data)

How to do PCA Efficiently?

- Step 1: Make the mean of the data zero (Standardize your data)
- Step 2: Compute the Covariance Matrix C

How to do PCA Efficiently?

- Step 1: Make the mean of the data zero (Standardize your data)
- Step 2: Compute the Covariance Matrix C
- Step 3: Calculate eigenvectors and eigenvalues of C
  - Eigenvector with largest eigenvalue $\lambda_1$corresponds to the 1st PC (why?)
  - Eigenvector with k^{th} largest eigenvalue λ_k corresponds to k^{th} PC
  - $\frac{\lambda_k}{\sum_i \lambda_i}$ = proportion of variance captured by k^{th} PC

Questions?