# Applied Machine Learning
# Lecture 10: Unsupervised learning

**Selpi (selpi@chalmers.se)**

The slides are further development of Richard Johansson's slides

February 25, 2020

# Overview

Different learning approaches

Unsupervised learning

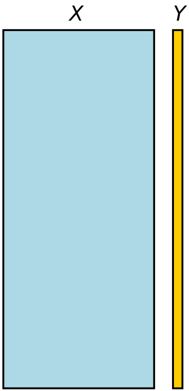
Clustering

modeling distributions

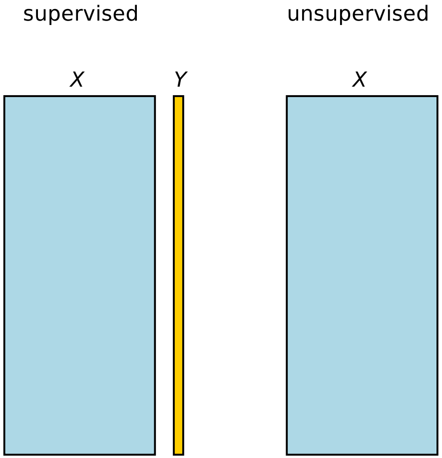
dimensionality reduction

# Different learning approaches based on supervision

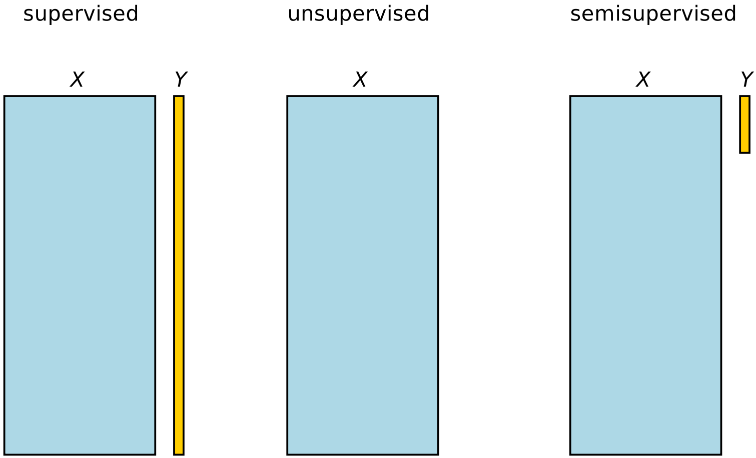
supervised



# Different learning approaches based on supervision



# Different learning approaches based on supervision



# Overview

Different learning approaches

Unsupervised learning

Clustering

modeling distributions

dimensionality reduction

# Unsupervised learning

- Why do this?
  - hand-labeled data might not be available or expensive and time-consuming to produce
- Goal:
  - to discover structure in the data to summarise, explore, visualise, and understand data

# Applications of unsupervised learning

**Cyber Security**

https://www.technologyreview.com/s/612427/the-rare-form-of-machine-learning-that-can-spot-hackers-who-have-already-broken-in/

When Darktrace deploys its software, it sets up physical and digital sensors around the client's network to map out its activity. That raw data is funneled to over 60 different unsupervised-learning algorithms that compete with one another to find anomalous behavior.

**Recommender Systems**

**Market segmentations**

# Main types of unsupervised approaches

- grouping the data: **clustering**
- modeling the **statistical distribution** of the data
- **dimensionality reduction**: projecting a high-dimensional dataset to a lower dimensionality

# Overview

Different learning approaches

Unsupervised learning

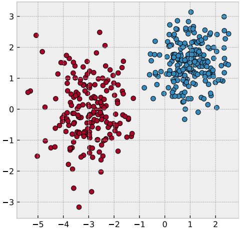
Clustering

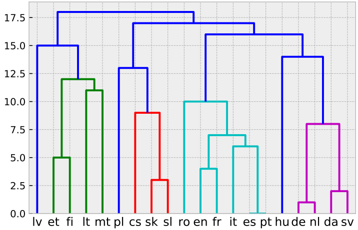
modeling distributions

dimensionality reduction

# clustering

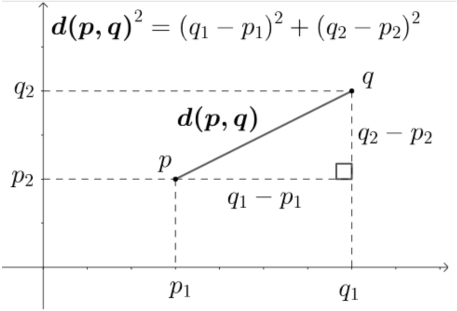
- **clustering** describes the data by forming **groups** (partitions) or **hierarchies**



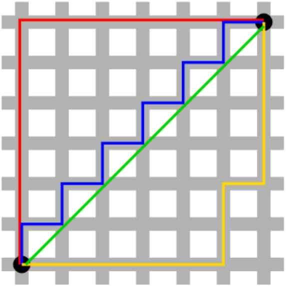
# What do we need to be able to cluster data?

- Euclidean distance




- Manhattan distance

$$d_1(\mathbf{p},\mathbf{q}) = \|\mathbf{p}-\mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|$$



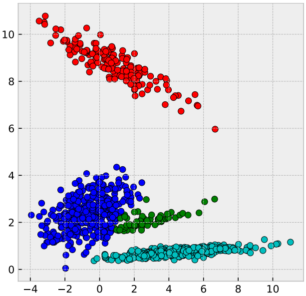
# Different clustering methods

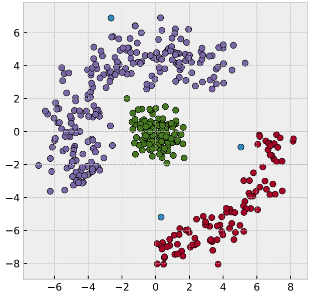
- Partitional
  - K-means
  - Graph-based
  - EM
- Hierarchical
  - Single linkage
  - Average linkage
  - Median linkage
  - Complete linkage
  - Centrod linkage
  - Ward's method
- ...
- Neural network

# in scikit-learn

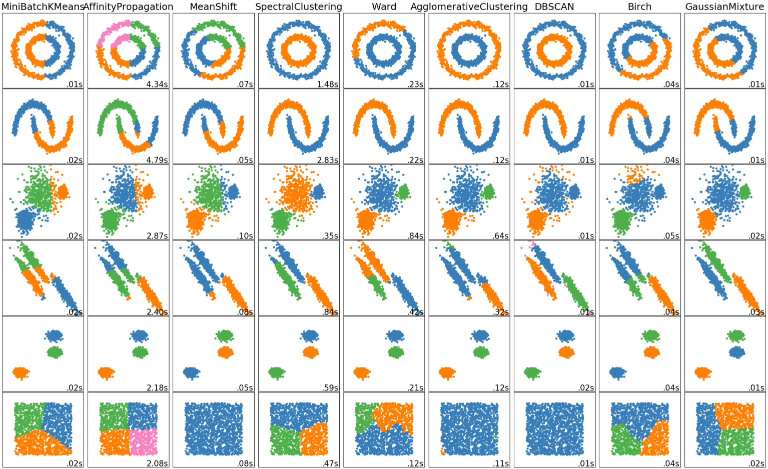
http://scikit-learn.org/stable/modules/clustering.html

# flat clustering

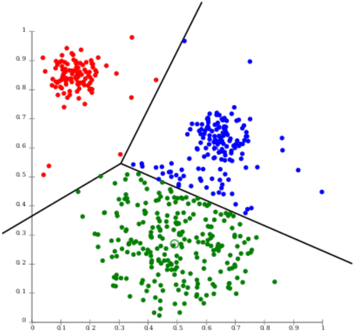


# flat clustering methods: overview



# K-means (1)



- each cluster is represented by its **centroid** $\boldsymbol{\mu}_i$
- minimize the within-cluster sum of squares:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \boldsymbol{\mu}_i\|^2$$

Randomly initialise k cluster centroids
**repeat until converged**:
  assign cluster to each training data, based on distance
  move each centroid to the centre of its cluster

# K-means (2)

- How to determine number of clusters?
  - Elbow method, trial and error, use domain knowledge

# K-means (2)

- How to determine number of clusters?
  - Elbow method, trial and error, use domain knowledge
- Advantages
  - simple, easy to understand, fast, robust enough, good for distinct clusters

# K-means (2)

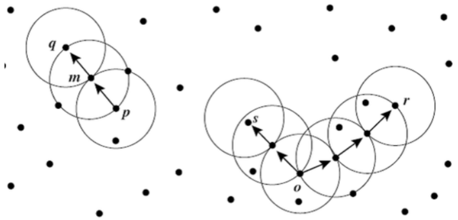
- How to determine number of clusters?
  - Elbow method, trial and error, use domain knowledge
- Advantages
  - simple, easy to understand, fast, robust enough, good for distinct clusters
- Disadvantages
  - Rely on k, can have local minima, depends on initialisation, it clusters even a uniform data, cannot handle outlier, sensitive to scalling

# K-means (2)

- How to determine number of clusters?
  - Elbow method, trial and error, use domain knowledge
- Advantages
  - simple, easy to understand, fast, robust enough, good for distinct clusters
- Disadvantages
  - Rely on k, can have local minima, depends on initialisation, it clusters even a uniform data, cannot handle outlier, sensitive to scalling

- See Demo!

# DBSCAN

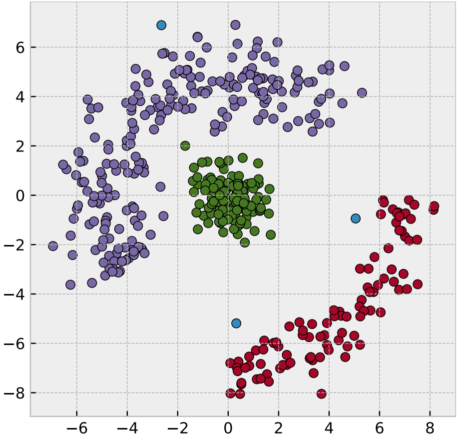



[source]

Parameters: eps and min_samples
1) Randomly select a point P
2) Retrieve all points directly density-reachable from P w.r.t. eps (radius to search for neighboors).

- If P is a core point, a cluster is formed. Find recursively all its density conected points and assign them to the same cluster as P
- If P is not a core point, iterate through the remaining unvisited points in the dataset

# DBSCAN example




See demo!

# Discussion

- why is clustering hard?

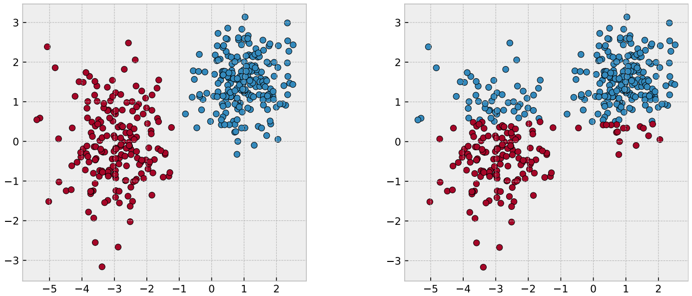
# Discussion

- why is clustering hard?
  - Difficult to interpret and evaluate the results

# Evaluating clustering algorithms

- how could we evaluate the result produced by a clustering algorithm?
  - Check Silhouette score
  - Compare with gold standard data (if any)

# Evaluating flat clustering (1): internal consistency



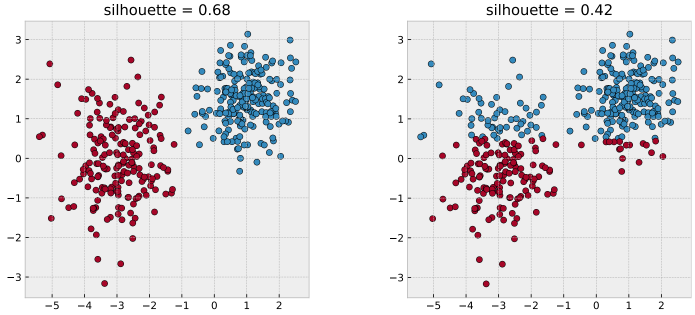
# Cluster evaluation: the silhouette score

- Measures how close each object is to its own cluster compared to other clusters.
- For each data point $x_i$, the **silhouette score** is defined

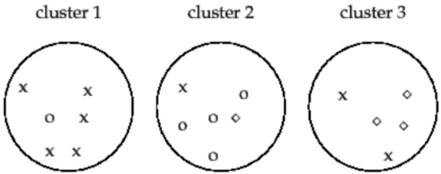
$$s_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

  where
  - $a_i$ is the average distance to other members in the **same** cluster
  - $b_i$ the minimal average distance to **another** cluster
- then we take the average over all the data
- in scikit-learn: sklearn.metrics.silhouette_score

# example

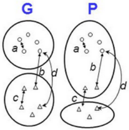


# evaluating flat clustering (2): comparing to a gold standard



# evaluating flat clustering (2): comparing to a gold standard

- scikit-learn contains several evaluation metrics for this scenario
  - http://scikit-learn.org/stable/modules/classes.html#clustering-metrics
  - http://scikit-learn.org/stable/modules/clustering.html#clustering-evaluation
- for instance, the *adjusted Rand score*:



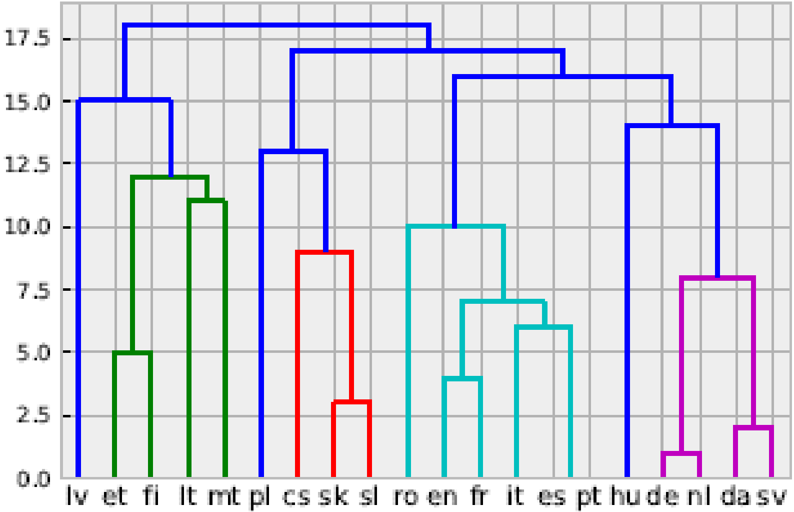

Agreement: $a, d$
Disagreement: $b, c$

$$RI(P,G) = \frac{a+d}{a+b+c+d}$$

$$ARI = \frac{RI - E(RI)}{1 - E(RI)}$$

[source]

# hierarchical clustering

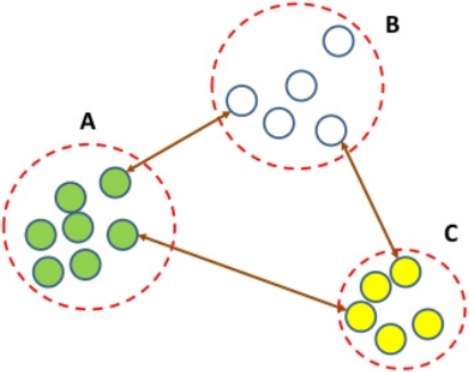



See demo!

# Single linkage clustering

## Single-linkage clustering (SLINK)

- distance between two clusters is determined by a *element pair*, that are closest to each other

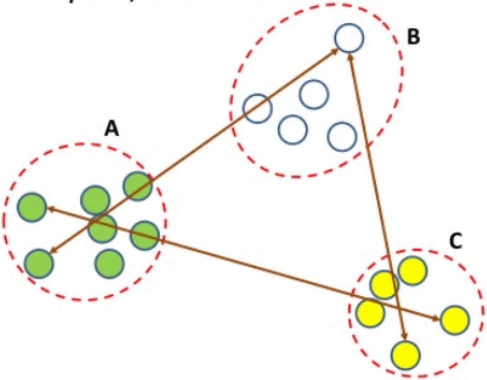



[source]

# Complete linkage clustering

## Complete-linkage clustering (CLINK)

- distance between two clusters is determined by a *element pair*, that are farthest to each other



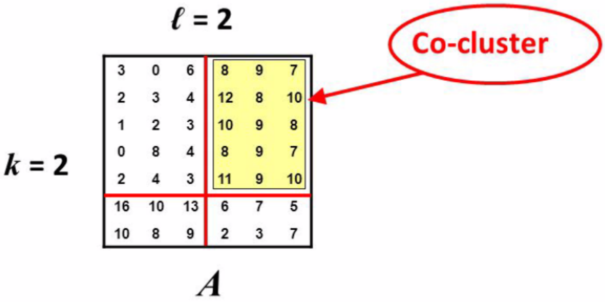

[source]

# Co-clustering

## Co-Clustering



- **Co-Clustering:** Cluster rows and columns of $A$ simultaneously:

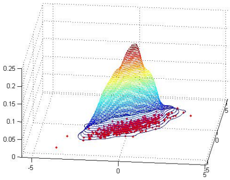



[source]

# Overview

# modeling distributions

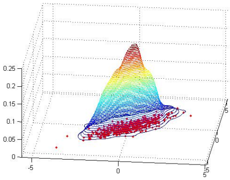
- we are given a dataset and now we need to
  - visualize the distribution
  - find the most probable region in the data
  - randomly generate new synthetic data from the same distribution
  - are there any exceptional data points?
  - for some new data point x, does it seem plausible that it comes from the same distribution?




[source]

# modeling distributions

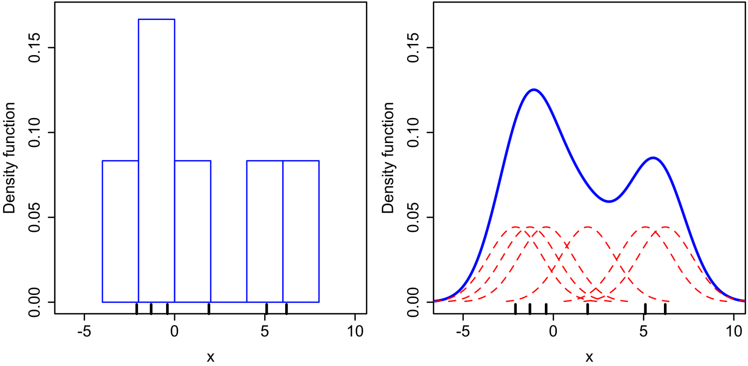
- we are given a dataset and now we need to
  - visualize the distribution
  - find the most probable region in the data
  - randomly generate new synthetic data from the same distribution
  - are there any exceptional data points?
  - for some new data point x, does it seem plausible that it comes from the same distribution?




[source]

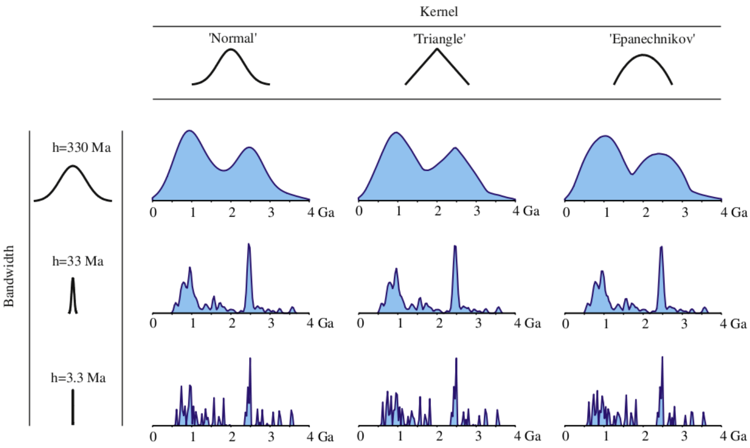
- we need to model the statistical **distribution** of the data
  - how would we solve this using an approach we've seen in a basic stats class?
  - ... and what are the limitations of that method?

# kernel density estimation




[source]

# kernel density estimation (2)




[source]

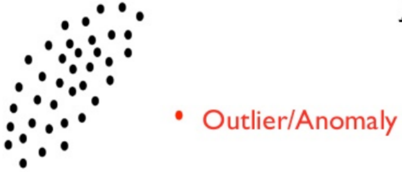
# in scikit-learn

sklearn.neighbors.KernelDensity

# outliers

"An **outlier** is an **observation** in a data set which appears to be **inconsistent** with the remainder of that set of data."

Johnson 1992

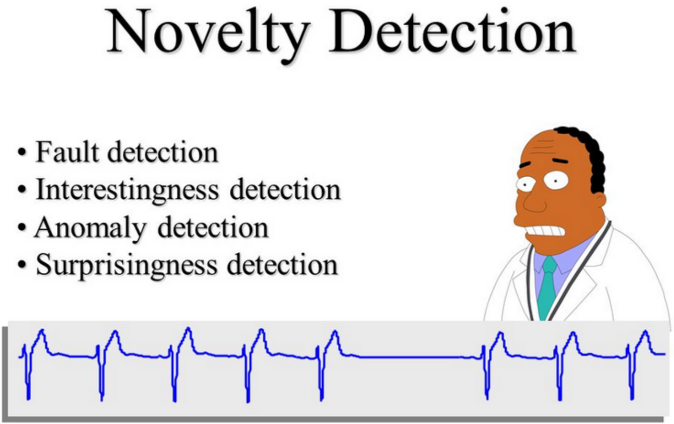



"An **outlier** is an **observation** which **deviates** so much from the other observations as to **arouse suspicions** that it was generated by a **different mechanism**."

Hawkins 1980

[source]

# novelty detection, anomaly detection, ...




[source]

Applications: fraud detection, cyber security, manufacturing processes, monitoring machine.

# Anomaly detection vs supervised

- Anomaly detection: Very small number of positive examples, large number of negative examples, future anomalies are unlikely to be similar to the training set

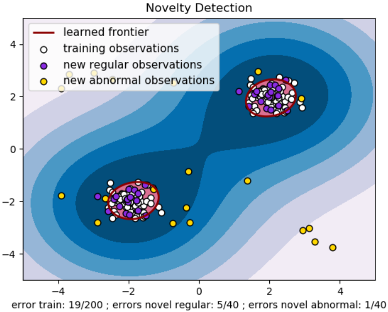
- Supervised: Large number of positives and negatives future examples are likely to be similar to training set.

# in scikit-learn

http://scikit-learn.org/stable/modules/outlier_detection.html

# one-class SVM

- the **one-class SVM** tries to find a decision boundary that encloses the training data (except a fraction $\nu$)




- it can be used for novelty and outlier detection

[source]

# one-class SVM (formally)

$$\min_{w,\,\xi_i,\,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \xi_i - \rho$$

subject to:

$$\begin{aligned} (w \cdot \phi(x_i)) \geq & \rho - \xi_i \qquad && \text{for all } i = 1, \dots, n \\ \xi_i \geq & \, 0 && \text{for all } i = 1, \dots, n \end{aligned}$$

# Anomaly detection - choosing features

- Plot histogram of data, see if the data looks gaussian. If not gaussian, do transformation to make the data look gaussian (e.g., take log(x) and plot the data again).
- log(x) can be changed with other function. The function that makes the data more Gaussian can be used as a feature

# evaluation of anomaly detection systems

- how do you think it should be done?

# Overview

Different learning approaches

Unsupervised learning

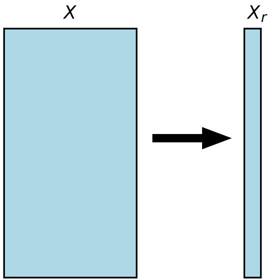
Clustering

modeling distributions

dimensionality reduction

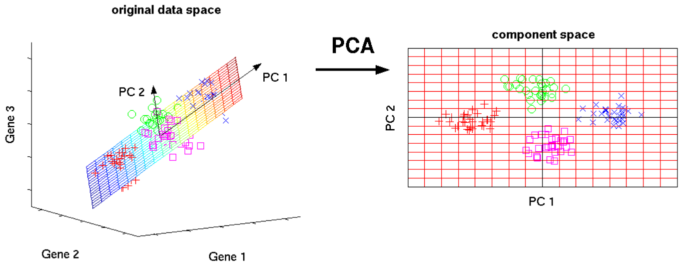
# dimensionality reduction

- reducing a high-dimensional dataset to a low-dimensional one



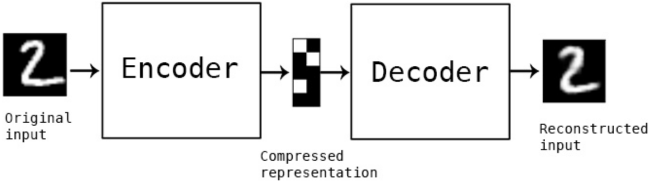

- **why?**
  - visualizing, understanding
  - reducing the need for storage
  - making supervised/unsupervised algorithms run faster
  - making supervised/unsupervised learning easier

# Principal Component Analysis




[source]

# autoencoders: dimensionality reduction using a NN




- for several Keras implementations, see

  `https://blog.keras.io/building-autoencoders-in-keras.html`

# Review from today's lecture

- Characterise essential differences between supervised, unsupervised, semi-supervised learning approaches
- Discuss several unsupervised learning methods
- Compare and evaluate different clustering methods
- Compare and evaluate different distance measures

# Next two lectures

- 28 Feb: Ethics in Machine Learning
  - Will be given by Vilhelm Verendel (https://www.chalmers.se/en/staff/Pages/vilhelm-gustav-verendel.aspx)
- 3 March: Dealing with time series data