Bits of Probability, Statistics and Machine Learning

Marco Cogoni
CRS4

23-24/5/2018

Who I am

- Degree and Ph.D. in Physics
- Background in theoretical/computational physics
  - Collective phenomena in complex systems: spin glasses, etc
  - Materials science: modeling of semiconductor growth and behavior
  - Turbulent flows
  - Crystal structures of ice
  - Optimization of complex solar plants
  - Optimization of network structures: communication, transportation,...
  - Analysis of industrial and medical data for classification and prediction
  - Analysis of “entrepreneurial data”

CRS4

---

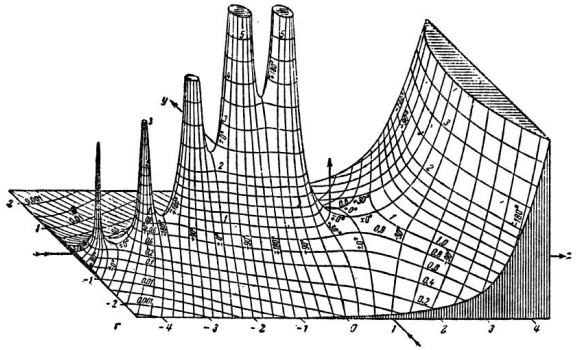
http://www.crs4.it/research/

http://www.crs4.it/research/data-intensive-computing/

*The **Data-Intensive Computing group** at CRS4 strives to address the challenge of **extracting useful information from mountains of data**. The technology permeating every modern activity creates disparate data sources that provide heterogenous points of view on any given process. Our work advances the state-of-the-art methods and technologies to collect these points of view and integrate them into a single understandable perspective. **Our applications span from the collection and distributed analysis of large clinical and industrial process datasets to sophisticated tracking of provenance information in health-related and biological studies**.*

They all have a common denominator: scalable and interoperable computing solutions.

Common themes



- Perform experiments (real or simulated)
- Gather (large) amounts of data
- Create a (rough) model of the system
- Extract as much information as possible from the data:
  - time series analysis
  - network analysis
  - statistics, machine learning, ...
- Improve your model of reality/abstract system
- *Think of a new experiment!*

brief probability and statistics recap

Classical definition of probability

From **Laplace**'s *Théorie analytique des probabilités*:

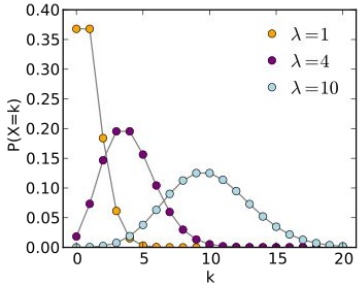
*“**The probability of an event is the ratio of the number of cases favorable to it, to the number of all cases possible** when nothing leads us to expect that any one of these cases should occur more than any other, which renders them, for us, equally possible.”*

Random variables

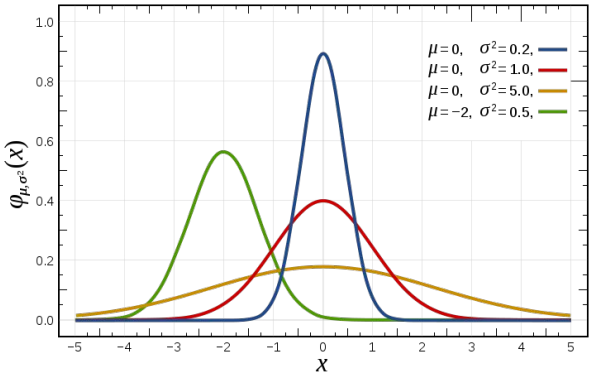
- A random variable, random quantity, aleatory variable, or stochastic variable is a variable whose possible values are outcomes of some “random process”
- The **outcome depends on some physical process** that is not well understood. For example, when tossing a fair coin, the final outcome of heads or tails depends on a (too complex to forecast) physical phenomenon
- To find a truly **random** process we should look at quantum mechanics: atoms, nuclei, elementary particles...

Probability distribution of random variables

- Discrete probability distributions
  - variable may assume a countable number of values
  - values are usually integers



- Continuous probability distributions
  - variable values fill a continuous domain



Expected value of a random variable

$$E[X] = x_1 p_1 + x_2 p_2 + \ldots + x_k p_k$$

$$p_1 + p_2 + \ldots + p_k = 1$$

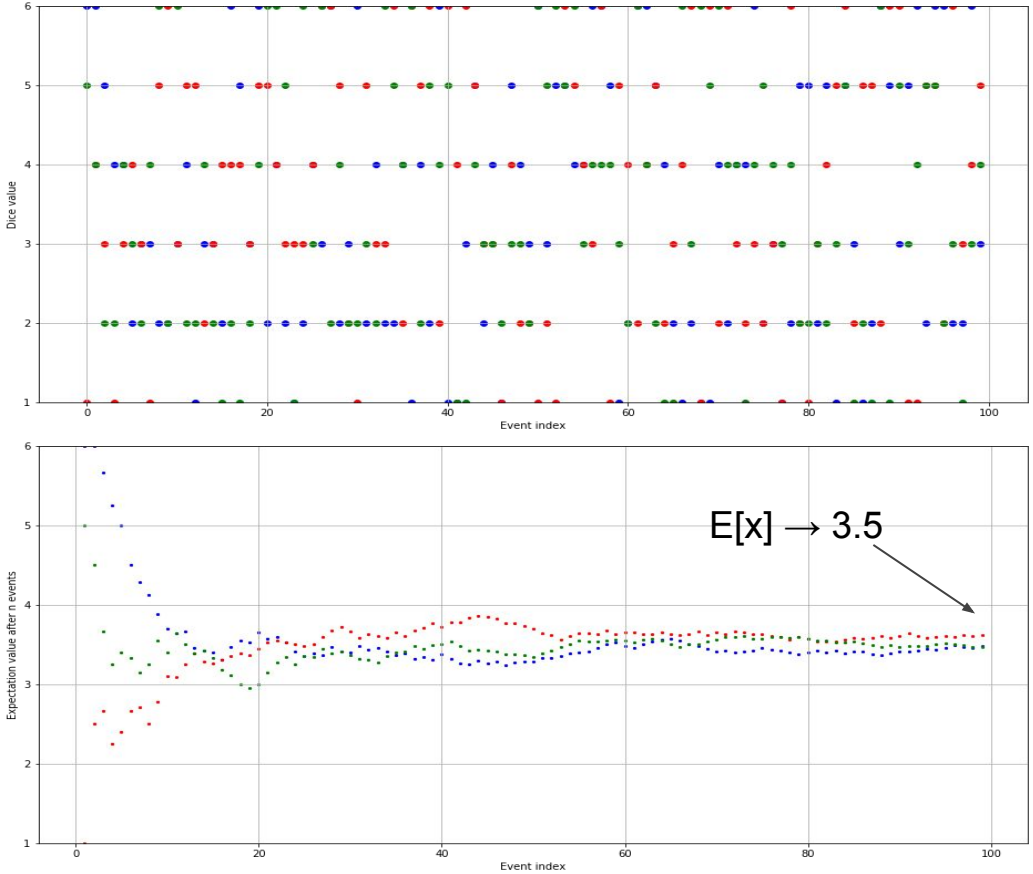
- If all outcomes of a random variable are **equiprobable**, *the expected value is the simple average of the outcomes.*

$$p_1 = p_2 = \ldots = p_k$$

- If the outcomes are **not equiprobable**, then the simple average must be replaced with the weighted average, which takes into account the fact that some outcomes are more likely than the others.

Expected value for dice

- Take a **fair** 6-faced die
- Perform 3 **equivalent** experiments:
  - launch the die 100 times for each experiment
- Plot the **expected value** as statistics is acquired



Expected value may be a outside of the variable permitted domain!

Law of large numbers

The law of large numbers states that **the sample average of a sequence of independent and identically distributed random variables converges towards their common expectation**, provided that the expectation value is finite.

$$\overline{X_n} = \frac{1}{n}\sum_{k=1}^{n} X_k$$

Central limit theorem

The theorem states that **the average of many independent and identically distributed (generic shape) random variables with finite variance tends towards a normal distribution:**

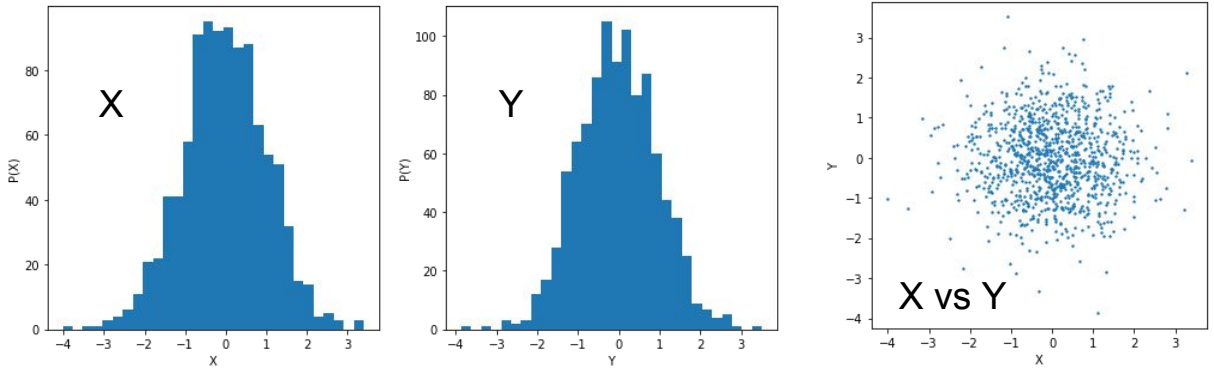
$$Z_n = \frac{\sum_{i=1}^{n} (X_i - \mu)}{\sigma\sqrt{n}}$$

NB: *for some random variables of the* ***heavy tail*** *and* ***fat tail*** *variety,* ***it works very slowly*** *or may not work at all*

Independent and identically distributed random variables

Two random variables X and Y are **independent** and **identically distributed** (iid) if they have the same probability distribution and they are mutually independent

$$P(X = x, Y = y) = P(X = x)P(Y = y), \forall x, y$$

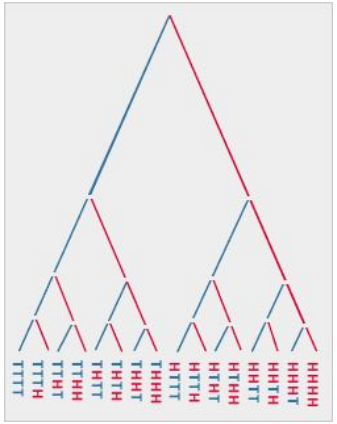


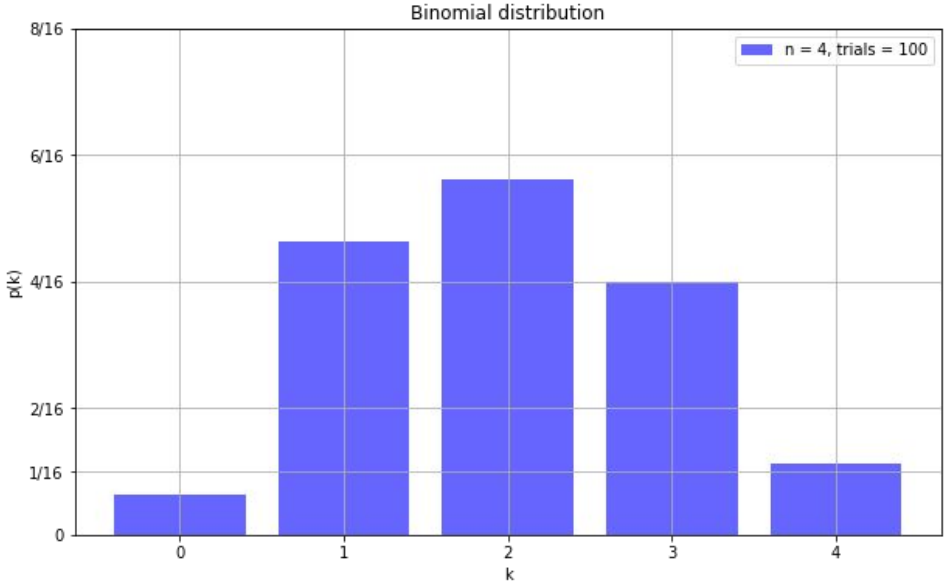
Canonical heads or tails example

Given a sequence of ***n*** heads (H) or tails (T) from a fair coin (p=p(T)=0.5), how often do we expect to find ***k*** times H?

Binomial distribution
(discrete)

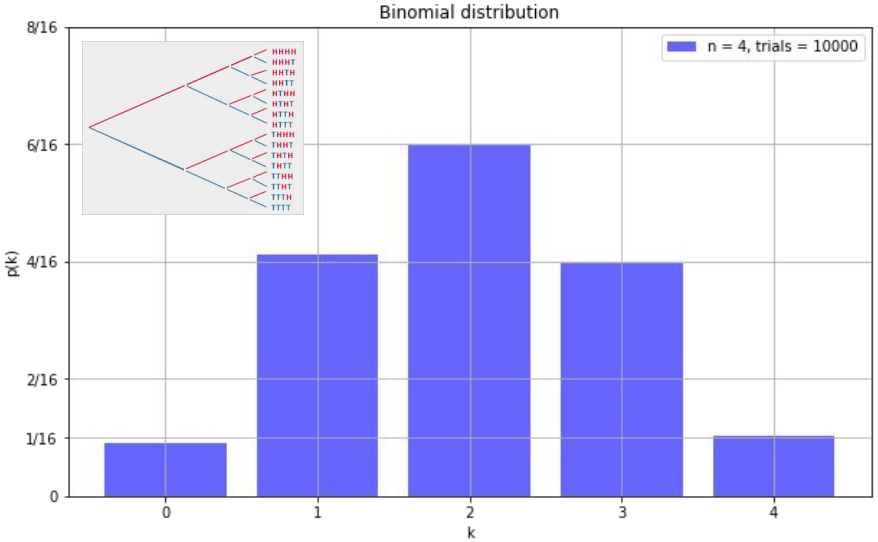
$$\binom{n}{k} p^k (1-p)^{n-k}$$

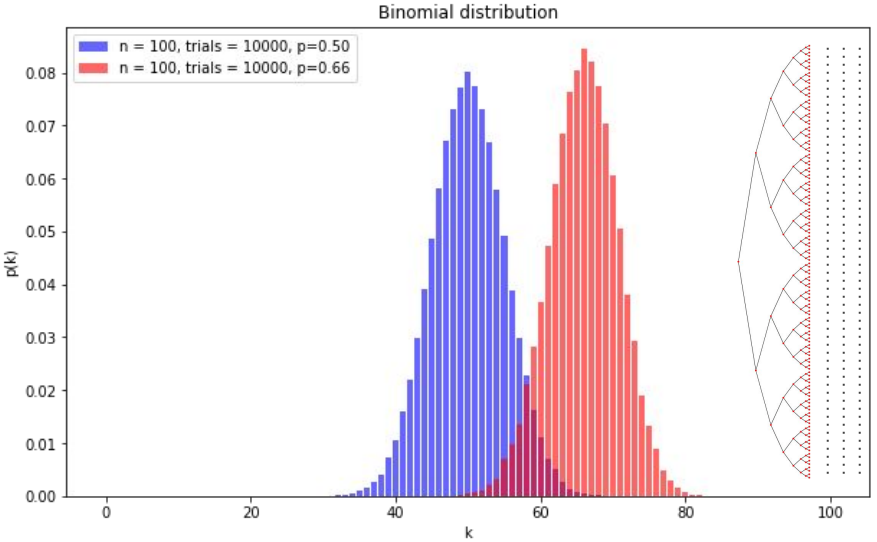




Canonical heads or tails example

Let's play **10.000** heads or tails matches, each containing **n=4** or **n=100** coin tosses



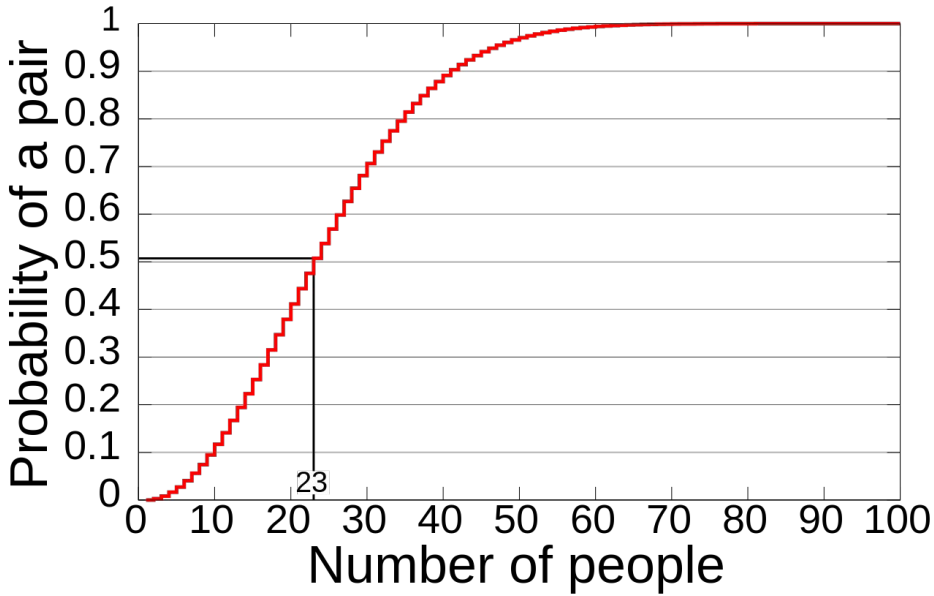


The Birthday problem

How likely is it that **two of us in this room share the same birthday**?

$$\bar{p}(n) = 1 \times \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \cdots \left(1 - \frac{n-1}{365}\right) =$$

$$= \frac{365 \times 364 \cdots (365 - n + 1)}{365^n} = \frac{365!}{365^n (365 - n)!}$$



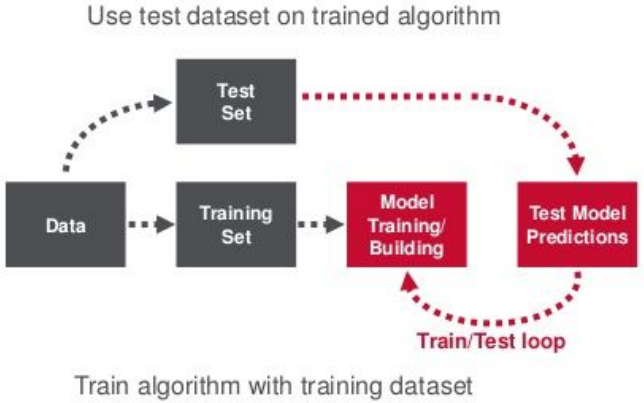
what is machine learning

Machine Learning

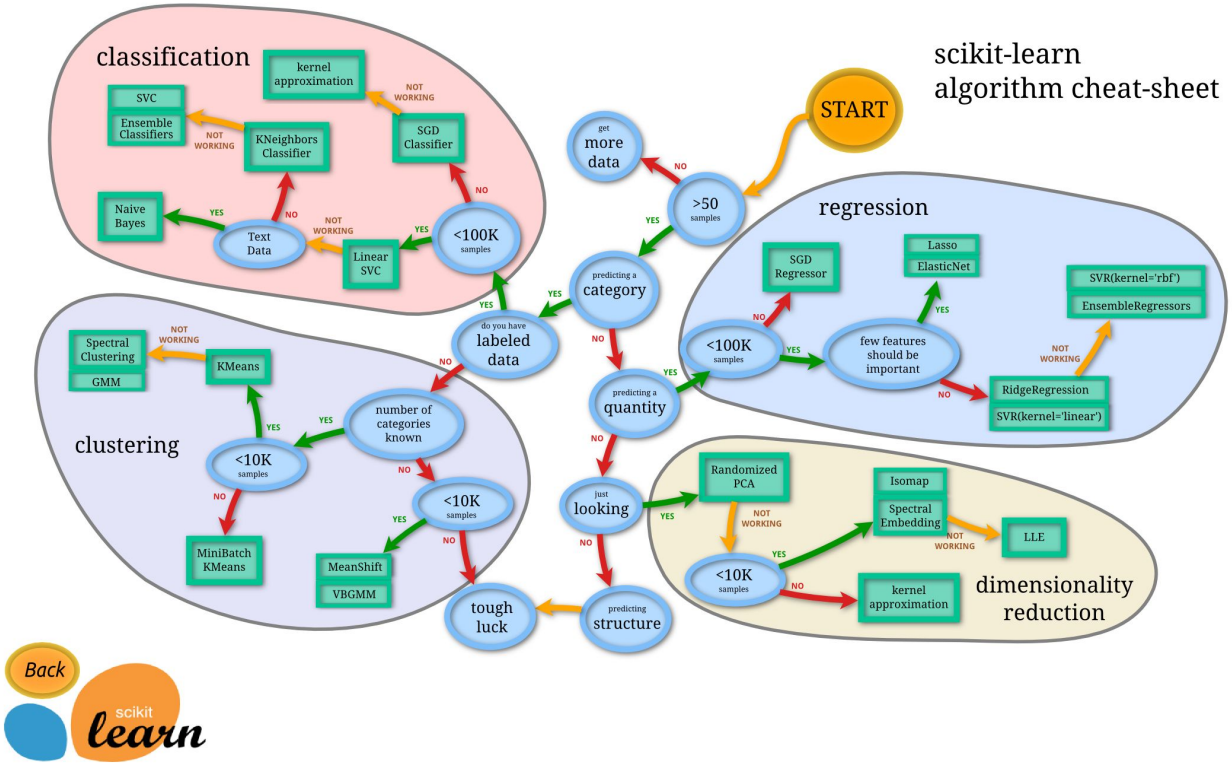
---

from Wikipedia:

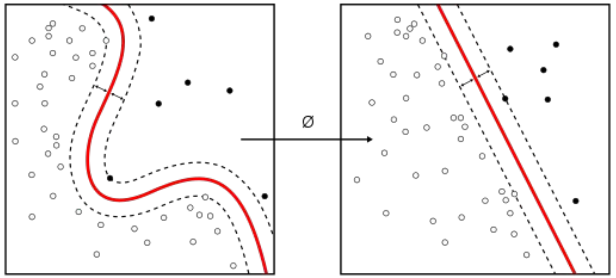
- Machine learning is **a field of computer science that uses statistical techniques** to give computer systems the ability to "learn" (e.g., progressively improve performance on a specific task) with data, without being explicitly programmed.



A Machine Learning taxonomy



Machine Learning tasks

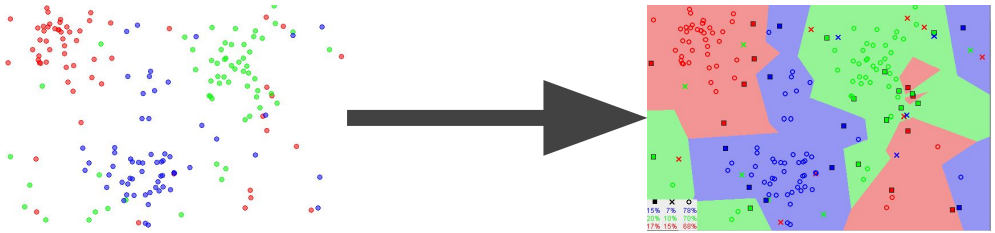


Machine learning tasks are typically classified into two broad categories:

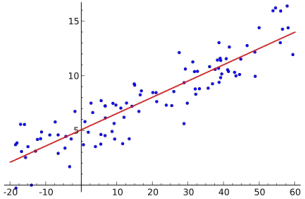
- **Supervised learning:** *The computer is presented with example inputs and their desired outputs*
- **Unsupervised learning:** *No labels are given to the learning algorithm, leaving it on its own to find structure in its input*

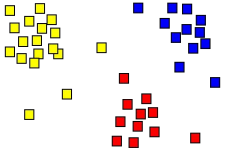
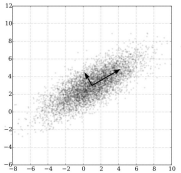
Machine Learning applications

- **Classification:** inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. *Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are "spam" and "not spam".*



Machine Learning applications



- **Regression:** similar to classification but outputs are continuous rather than discrete.
- **Clustering:** a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.
- **Dimensionality reduction:** simplifies inputs by mapping them into a lower-dimensional space.

linear regression

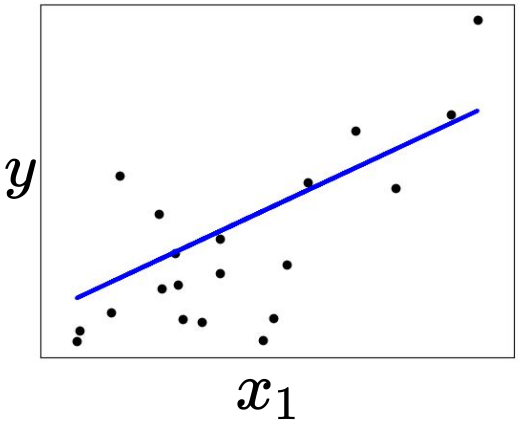
Linear models

A set of methods to perform regression in which **the target value Y is expected to be a linear combination of the input variables X.** The approximating function will be a line in 2D, and a hyperplane in general.

$$y(w, x) = w_0 + w_1 x_1 + \ldots + w_p x_p$$

offset: w_0

weights: $w = (w_1, \ldots, w_p)$

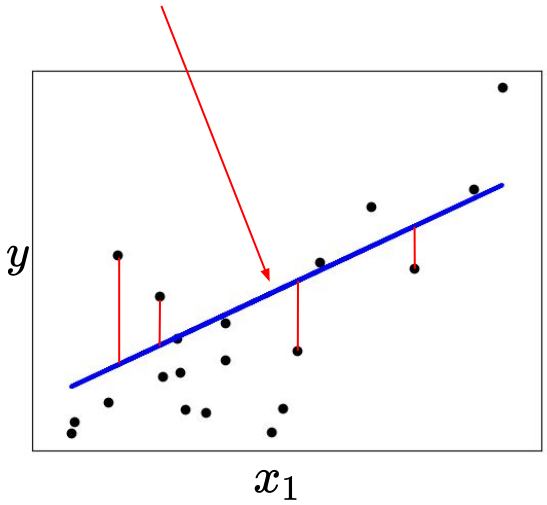


Linear models: Ordinary Least Squares

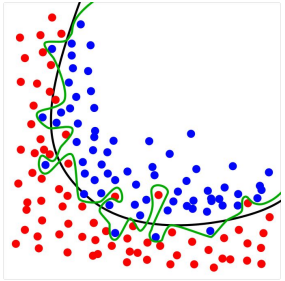
The simplest method to learn the model parameters is to **minimize the sum of the squared differences for each data point.**

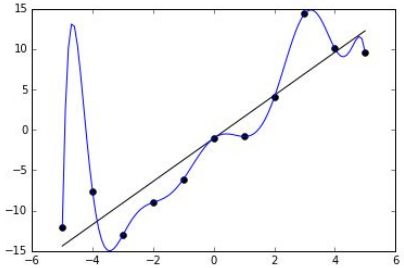
This method works well in 2D, but **may become highly sensitive to random errors in data points** when data columns are not independent for higher dimensional datasets.

$$min_w \|Xw - y\|_2^2$$



Overfit / underfit



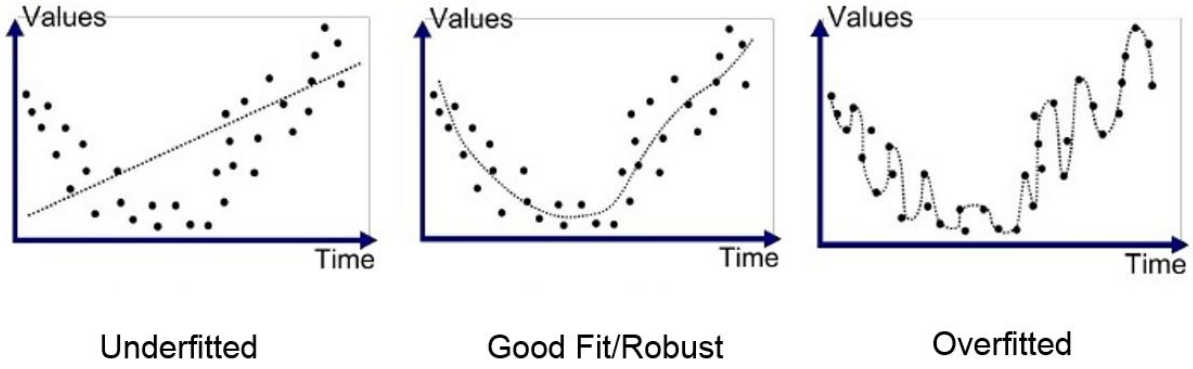


- An **overfitted** model is a **statistical model that contains more parameters than can be justified** by the data.
  - The essence of **overfitting** is to have unknowingly extracted some noise as if represented underlying model structure.

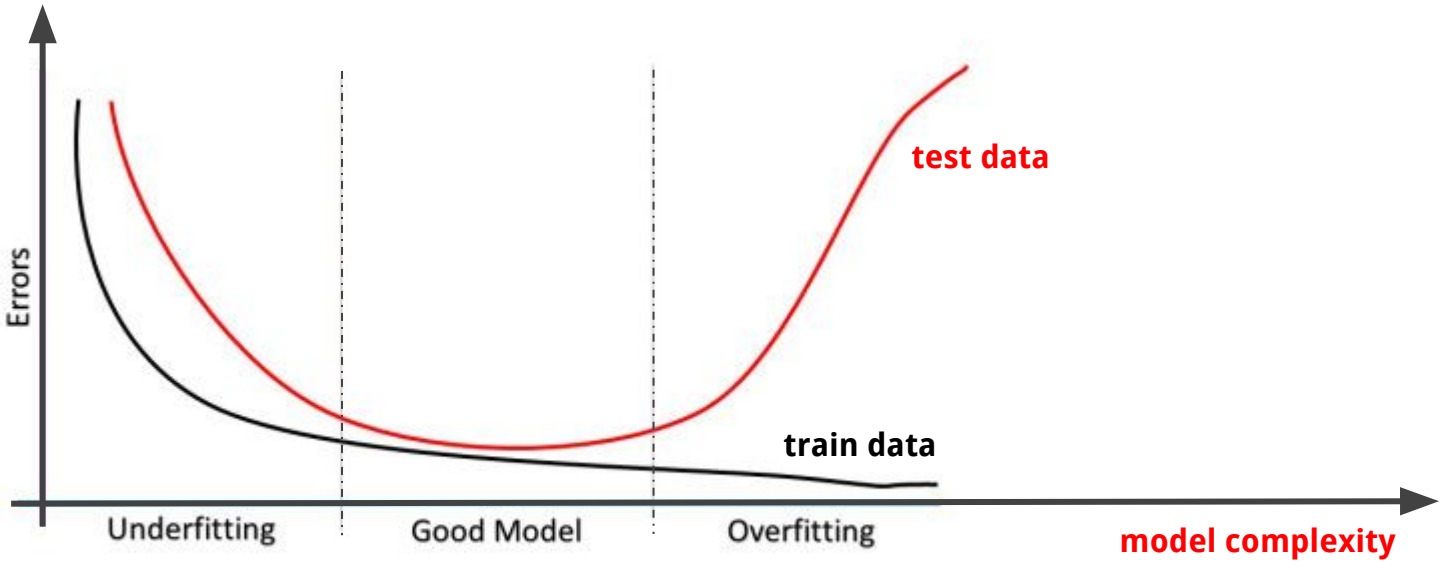
- **Underfitting** occurs when a statistical model cannot adequately capture the underlying structure of the data.
  - Underfitting occurs when fitting a linear model to non-linear data.
  - This model will have poor **predictive** performance.

Overfit / underfit

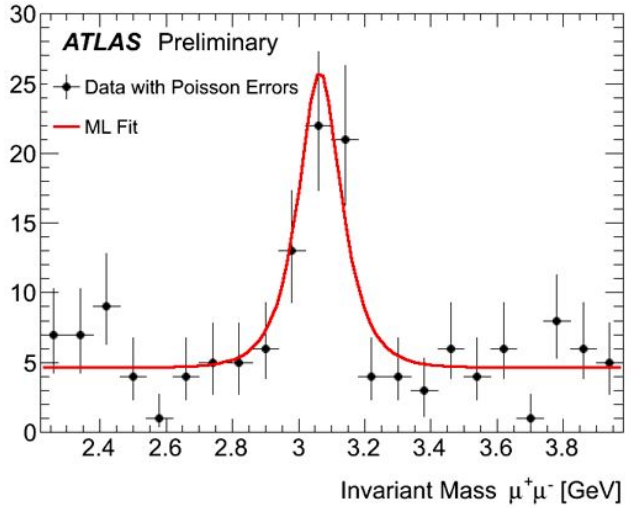
- An **overfitted** model generally presents a **very good score on training data**, but it **behaves poorly on new data.**
- An **underfitted** model is usually recognizable by **low scores both for training and test** data.

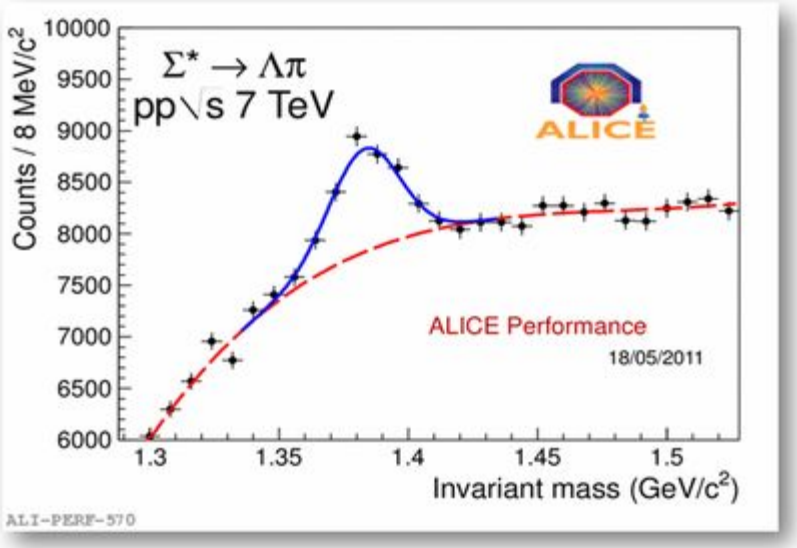


Overfit / underfit



Overfit / underfit: relevance for applications





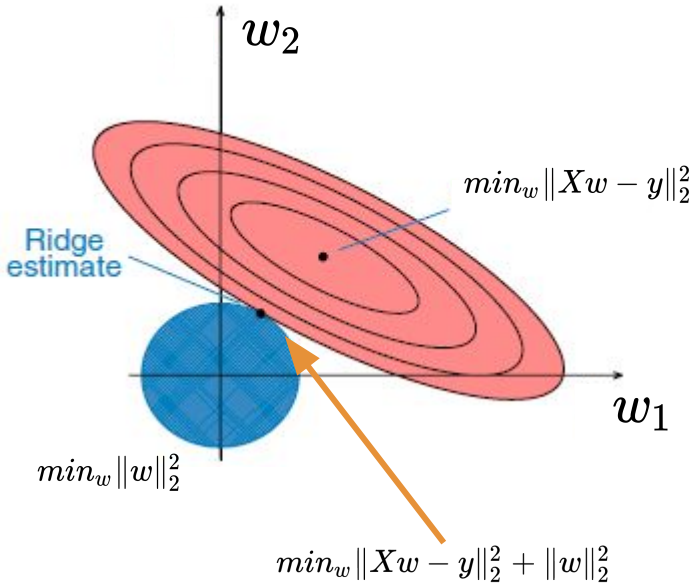
Linear models: Ridge regression

The **formula is the same one used for ordinary least squares**.

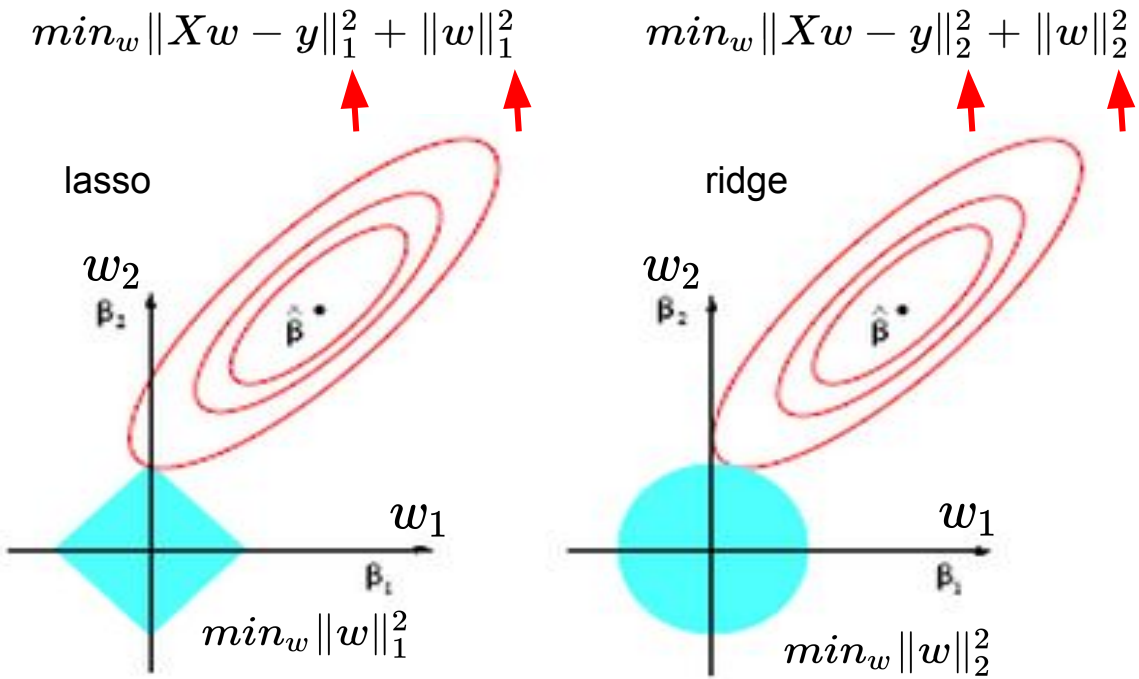
- In ridge regression the coefficients (w) are chosen to well fit training data, but also an additional constraint: the magnitude of coefficients should be smallest;
- Intuitively, this means each feature should have as little effect on the outcome as possible (small slope), while still predicting well. *This constraint is an example of what is called regularization (Tychonov).*

Linear models: Ridge regression

- We are trying to **minimize the ellipse size and circle simultaneously** in the ridge regression.
- The ridge estimate is given by the point at which the ellipse and the circle touch.
- **Poorer** results on training set, **better** on test set



Linear models: Lasso regression

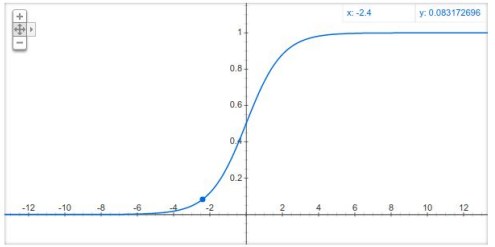


Logistic regression (binary classification)

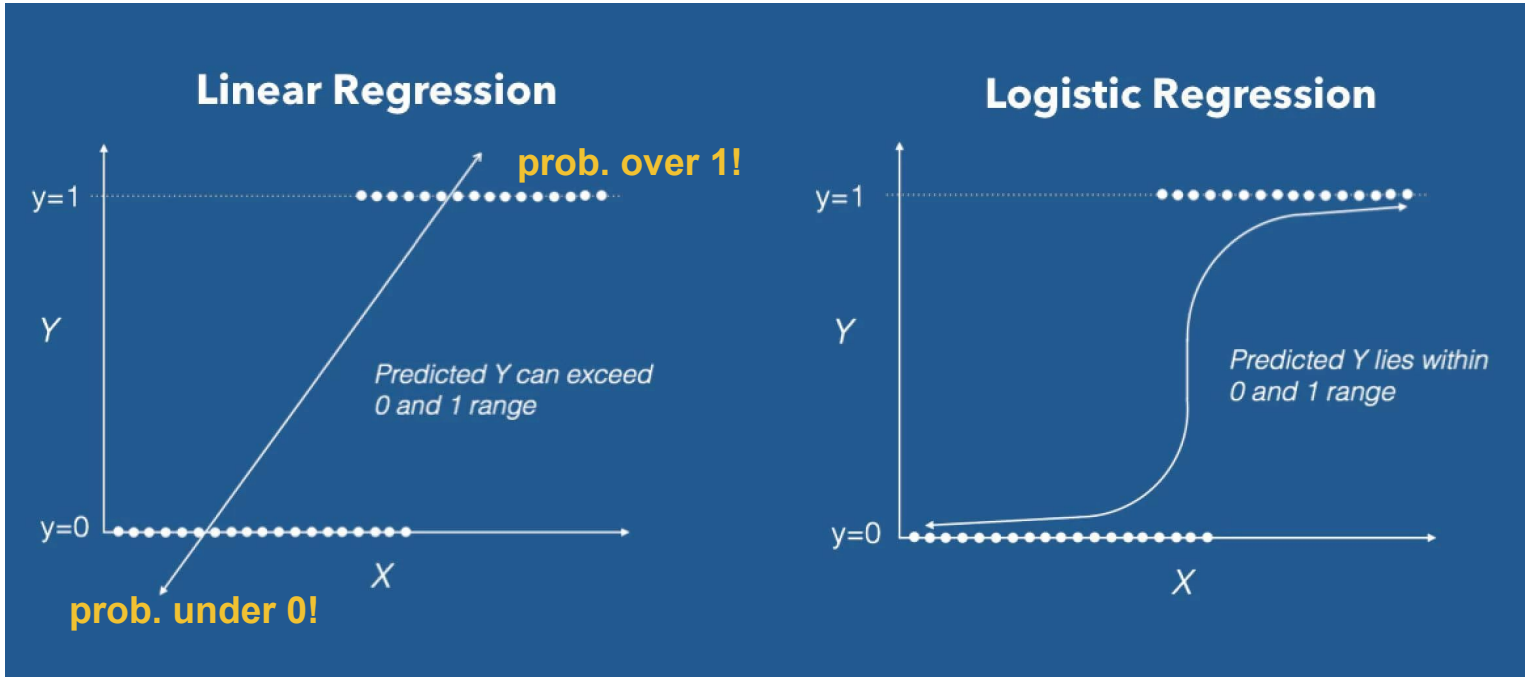
Suppose you want to **model the probability of some event** to happen (or not) given some continuous variable ***p*(*X*) = *Pr*(*Y*=1|*X*)**:

- let **X represent the continuous** variable
- let **Y represent the outcome** of the event
- linear model: $p(X)=\beta_0+\beta_1 X$
  - *doesn't work well because it extends to infinity*
- so, **let's use a sigmoid** for a better fit:

$$p(X) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$



Logistic regression (binary classification)



Logistic regression (binary classification)

Basically **it's a linear model** (with the usual two parameters for 1D) **but with a transformation of the Y** coordinate

$$p(X) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

The odds ratio is just the ratio of likelihoods for the event happening and not happening

$$odds = \frac{p(X)}{1-p(X)}$$

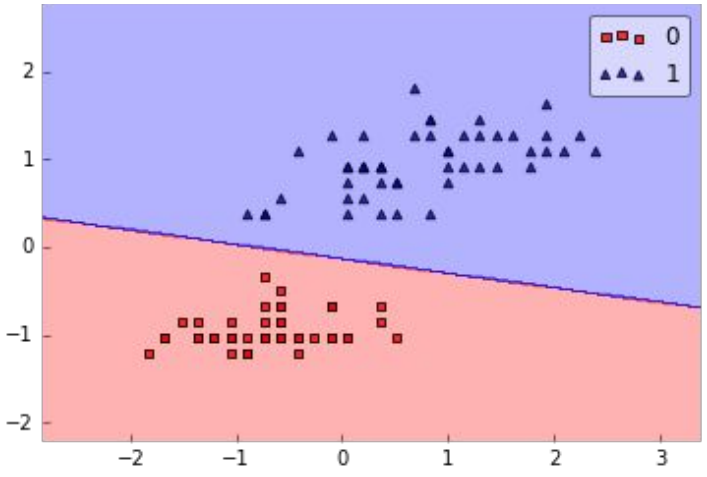
odds ratio

$$odds = e^{\beta_0+\beta_1 x}$$

Logistic regression (binary classification)

- Logistic regression **may be used as a classifier** by setting a threshold on p(X)
- The sigmoid may live in any number of dimensions just like a hyperplane for linear regressions

$$p(X) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$



dimensionality reduction

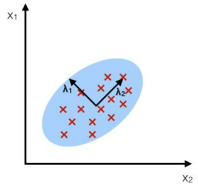
Dimensionality reduction

Dimensionality reduction is the process of reducing the number of random variables representing some phenomenon by **obtaining a set of “principal” variables.**

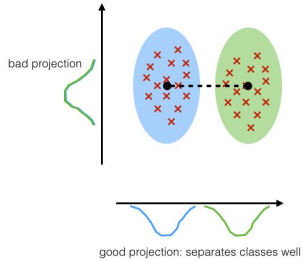
Principal means that we make the hypothesis that **some of the input variables are more important than others** to understand the phenomenon and create a model that can be “explained”.

Usually this equates to thinking of few variables living in a high dimensional space: *think of a crumpled sheet of paper*

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation



Dimensionality reduction

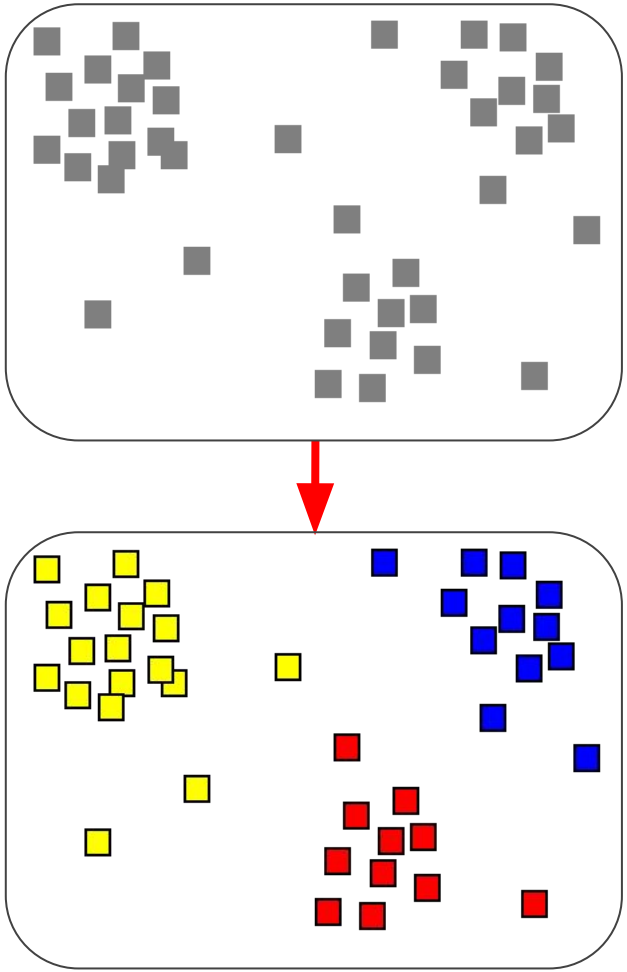
- **Principal component analysis (PCA):**
  - The main **linear** technique for dimensionality reduction, principal component analysis, **performs a linear mapping of the data to a lower-dimensional space** in such a way that the variance of the data in the low-dimensional representation is maximized.
- **Linear discriminant analysis (LDA):**
  - It is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition and machine learning to **find a linear combination of features that characterizes or separates two or more classes** of objects or events.

clustering

Clustering methods

Clustering is similar to classification but:

- you just have one data set (no train/test data);
- you don’t know the labels in advance: you assign them;
- you (usually) don’t build a model.



Clustering methods

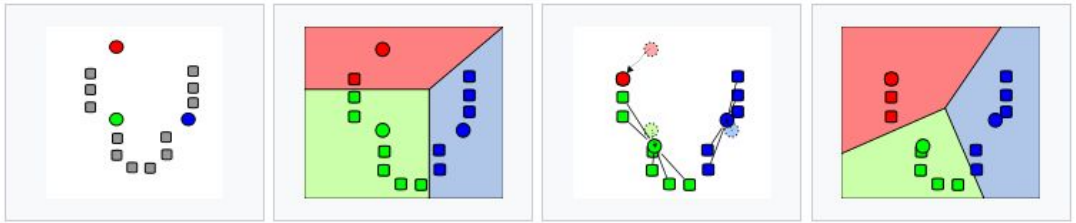
From Wikipedia:

- *Connectivity models*: for example, hierarchical clustering builds models based on distance connectivity.
- *Centroid models*: for example, the k-means algorithm represents each cluster by a single mean vector.
- *Distribution models*: clusters are modeled using statistical distributions, such as multivariate normal distributions used by the expectation-maximization algorithm.
- *Density models*: for example, DBSCAN and OPTICS defines clusters as connected dense regions in the data space.
- *Subspace models*: in biclustering (also known as co-clustering or two-mode-clustering), clusters are modeled with both cluster members and relevant attributes.
- *Group models*: some algorithms do not provide a refined model for their results and just provide the grouping information.
- *Graph-based models*: a clique, that is, a subset of nodes in a graph such that every two nodes in the subset are connected by an edge can be considered as a prototypical form of cluster. Relaxations of the complete connectivity requirement (a fraction of the edges can be missing) are known as quasi-cliques, as in the HCS clustering algorithm.
- *Neural models*: the most well known unsupervised neural network is the self-organizing map and these models can usually be characterized as similar to one or more of the above models, and including subspace models when neural networks implement a form of Principal Component Analysis or Independent Component Analysis.

k-means clustering

---

k-means clustering aims to partition n observations into **k clusters** in which **each observation belongs to the cluster with the nearest mean**, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.



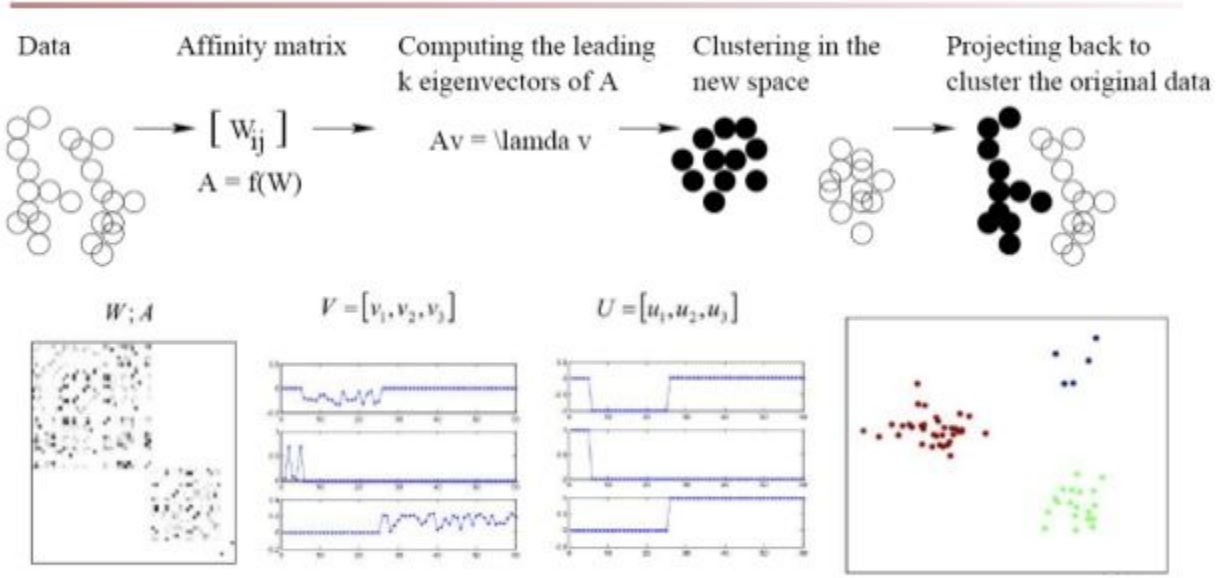
1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3. The centroid of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

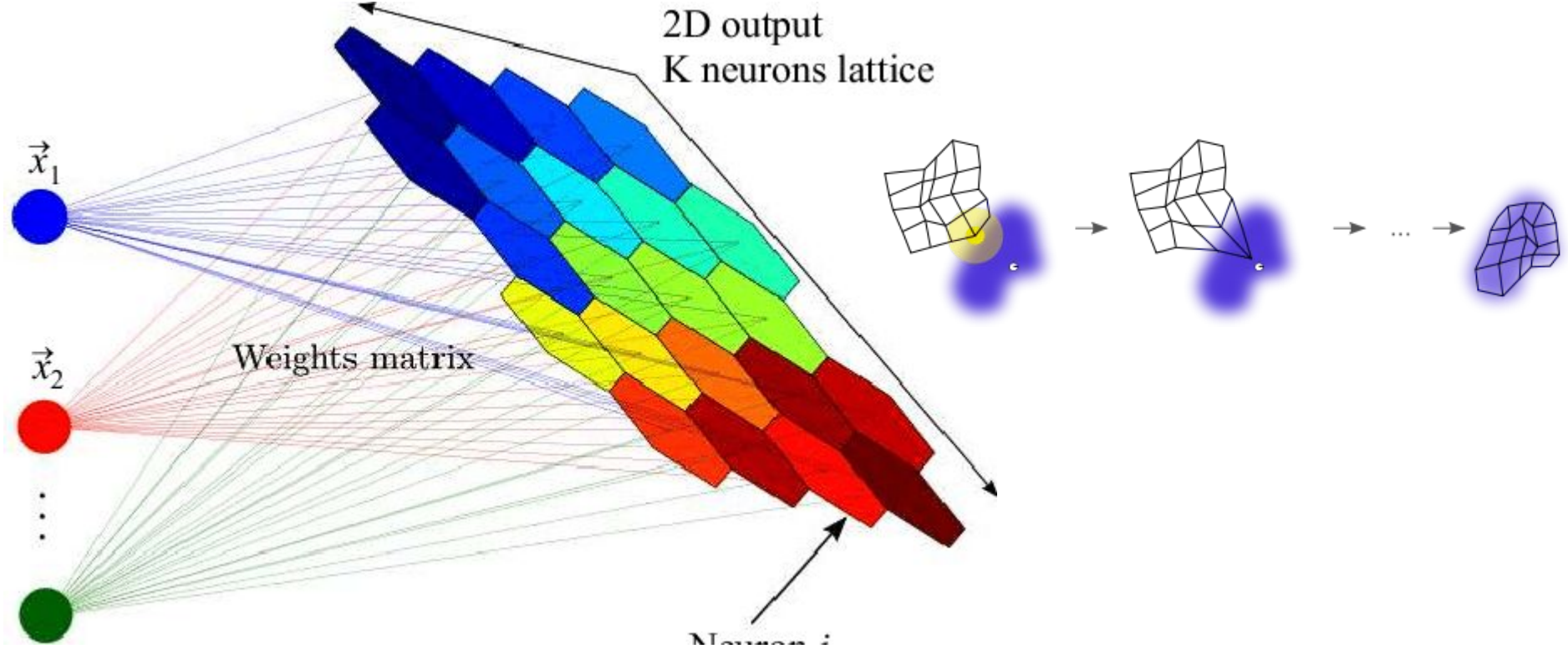
Spectral Clustering: Illustration and Comments



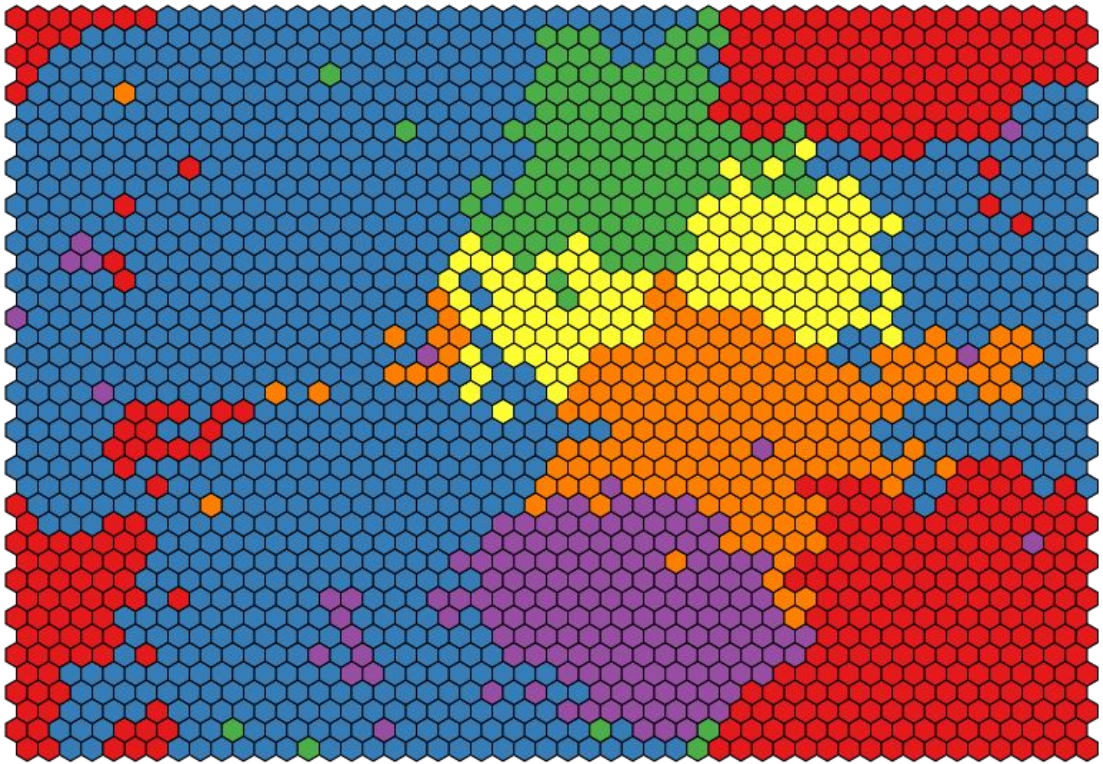
- Spectral clustering: Effective in tasks like image processing
- Scalability challenge: Computing eigenvectors on a large matrix is costly
- Can be combined with other clustering methods, such as bi-clustering

unsupervised ml: SOM

Self-Organizing Map



SOM



SOM

SOM limitations:

- does not build a generative model for the data
- relies on a predefined distance in feature space (a problem shared by most clustering algorithms)
- slow training, hard to train against slowly evolving data
- not so intuitive : neurons close on the map (topologically) may be far away in feature space
- does not behave so gently when using categorical data
- no generally admitted rule of thumb for the various parameters (map size, neighbouring function, time evolution of the learning rate ...)

next gen clustering: t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

Gaussian short-tailed

Raw, high-D data

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)},$$

Kullback-Leibler divergence

$$KL(P||Q) = \sum_{i\neq j} p_{ij}\log\frac{p_{ij}}{q_{ij}}$$

t-Student heavy-tailed

Projected, low-D data

$$q_{ij} = \frac{(1+\|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k\neq m}(1+\|\mathbf{y}_k - \mathbf{y}_m\|^2)^{-1}}$$

The minimization of the Kullback–Leibler divergence with respect to the points Y is performed using gradient descent. The result of this optimization is a map that reflects the similarities between the high-dimensional inputs well.

t-Distributed Stochastic Neighbor Embedding (t-SNE)

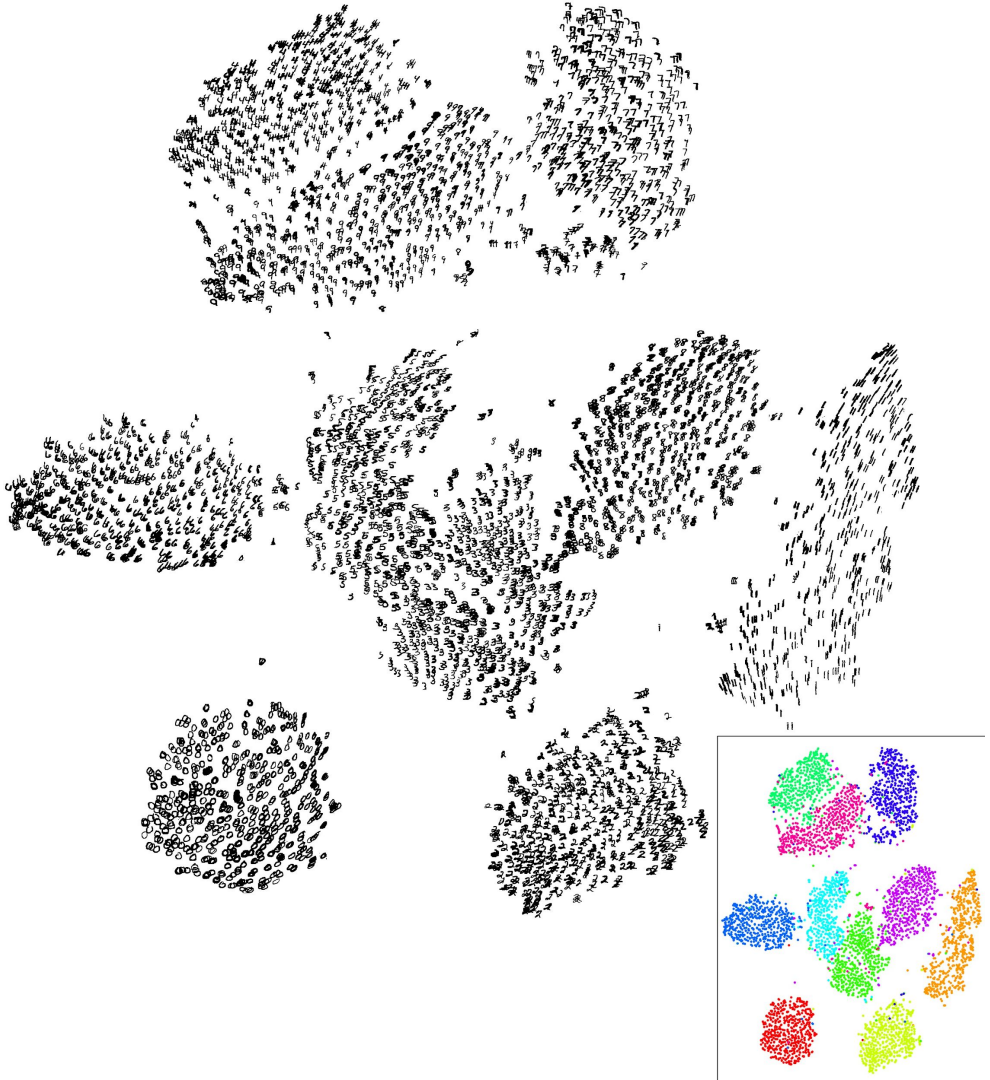
---

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. The technique can be implemented via Barnes-Hut approximations, allowing it to be applied on large real-world datasets. We applied it on data sets with up to 30 million examples. The technique and its variants are introduced in the following papers:

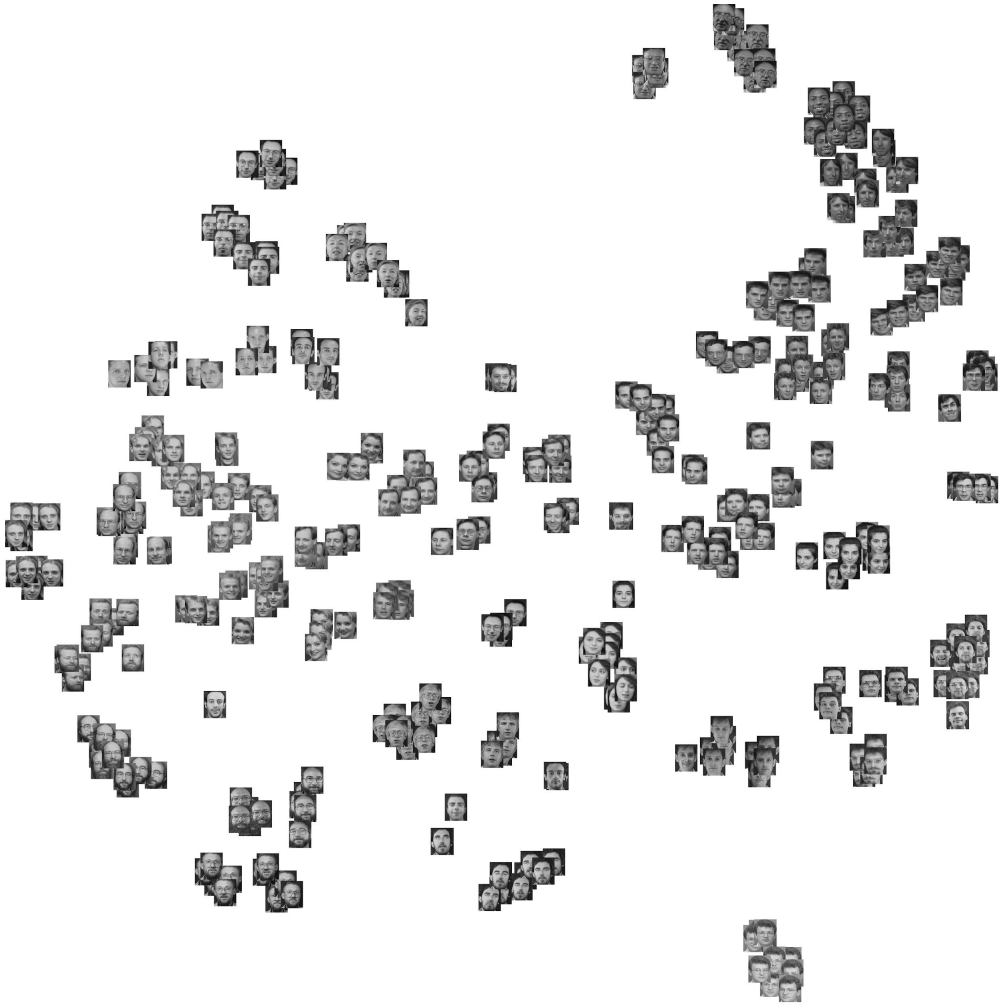
- L.J.P. van der Maaten. **Accelerating t-SNE using Tree-Based Algorithms**. *Journal of Machine Learning Research* 15(Oct):3221-3245, 2014.
- L.J.P. van der Maaten and G.E. Hinton. **Visualizing Non-Metric Similarities in Multiple Maps**. *Machine Learning* 87(1):33-55, 2012.
- L.J.P. van der Maaten. **Learning a Parametric Embedding by Preserving Local Structure**. In *Proceedings of the Twelfth International Conference on Artificial Intelligence & Statistics (AI-STATS), JMLR W&CP* 5:384-391, 2009.
- L.J.P. van der Maaten and G.E. Hinton. **Visualizing High-Dimensional Data Using t-SNE**. *Journal of Machine Learning Research* 9(Nov):2579-2605, 2008.

https://lvdmaaten.github.io/tsne/

t-SNE: MNIST digits dataset



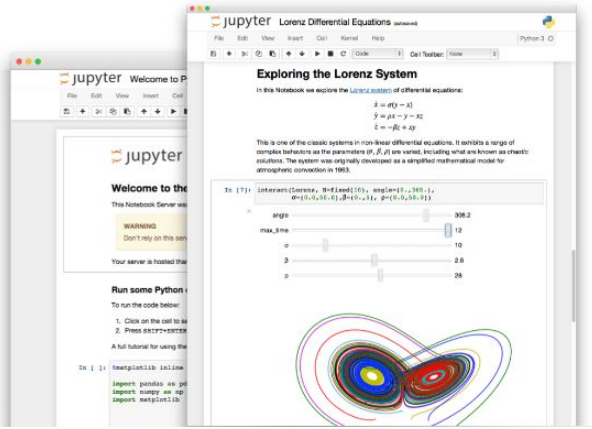
t-SNE: Olivetti faces dataset



t-SNE: Google img dataset

interactive environment for machine learning and some minimal bibliography

Interactive environment: http://jupyter.org



The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Try it in your browser | Install the Notebook

Language of choice

The Notebook has support for over 40 programming languages, including Python, R, Julia, and Scala.

Share notebooks

Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter Notebook Viewer.

Interactive output

Your code can produce rich, interactive output: HTML, images, videos, LaTeX, and custom MIME types.

Big data integration

Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore that same data with pandas, scikit-learn, ggplot2, TensorFlow.

http://jupyter.org

Currently in use at

Books and links

- *Swaroop C. H.* **A byte of Python** python.swaroopch.com
- *Allen Downey* **Think Python** greenteapress.com/wp/think-python/
- *Allen Downey* **Think Stats** greenteapress.com/thinkstats
- *Allen Downey* **Think Bayes** greenteapress.com/wp/think-bayes

- *Jake VanderPlas* **Python Data Science Handbook**
- *Rachel Schutt - Cathy O'Neil* **Doing Data Science**
- *A. Müller - S. Guido* **Intr. to Machine Learning with Python**
- *Drew Conway - John M. White* **Machine Learning for Hackers**

data science/big data

Hype around big data/data science

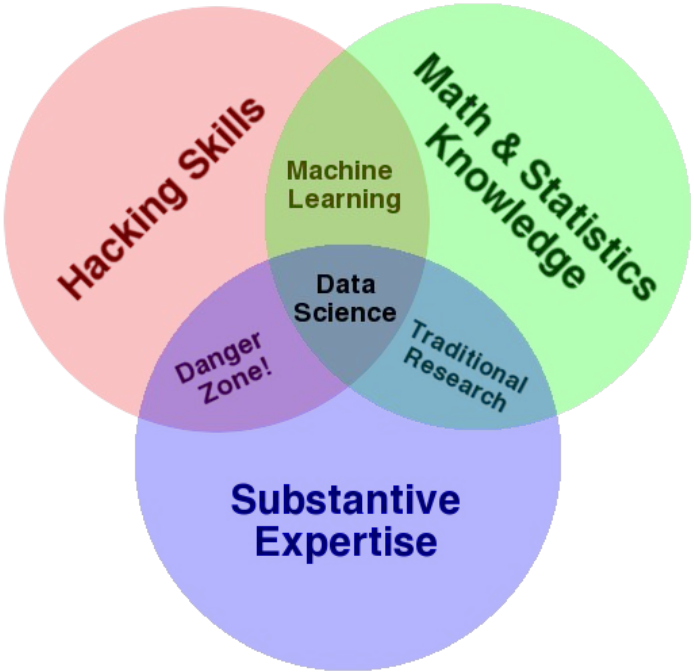
What is “Big Data” anyway? What does “data science” mean? What is the relationship between Big Data and data science? Is data science the science of Big Data? Is data science only the stuff going on in companies like Google and Facebook and tech companies? Why do many people refer to Big Data as crossing disciplines (astronomy, finance, tech, etc.) and to data science as only taking place in tech? Just how big is big?

Hype around big data/data science

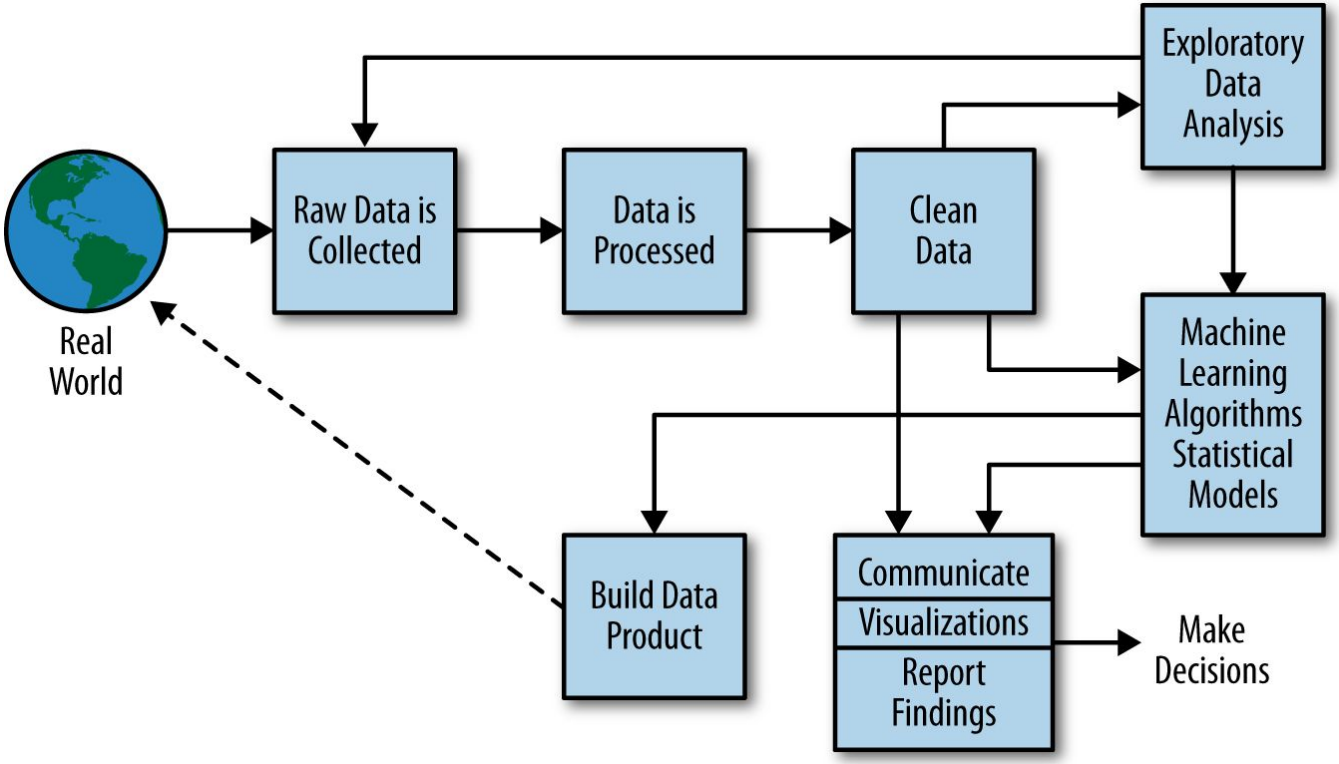
From the way the media describes it, machine learning algorithms were just invented last week and data was never “big” before Google.

This is simply not true. Many of the methods and techniques we’re using today are part of the evolution of everything that’s come before.

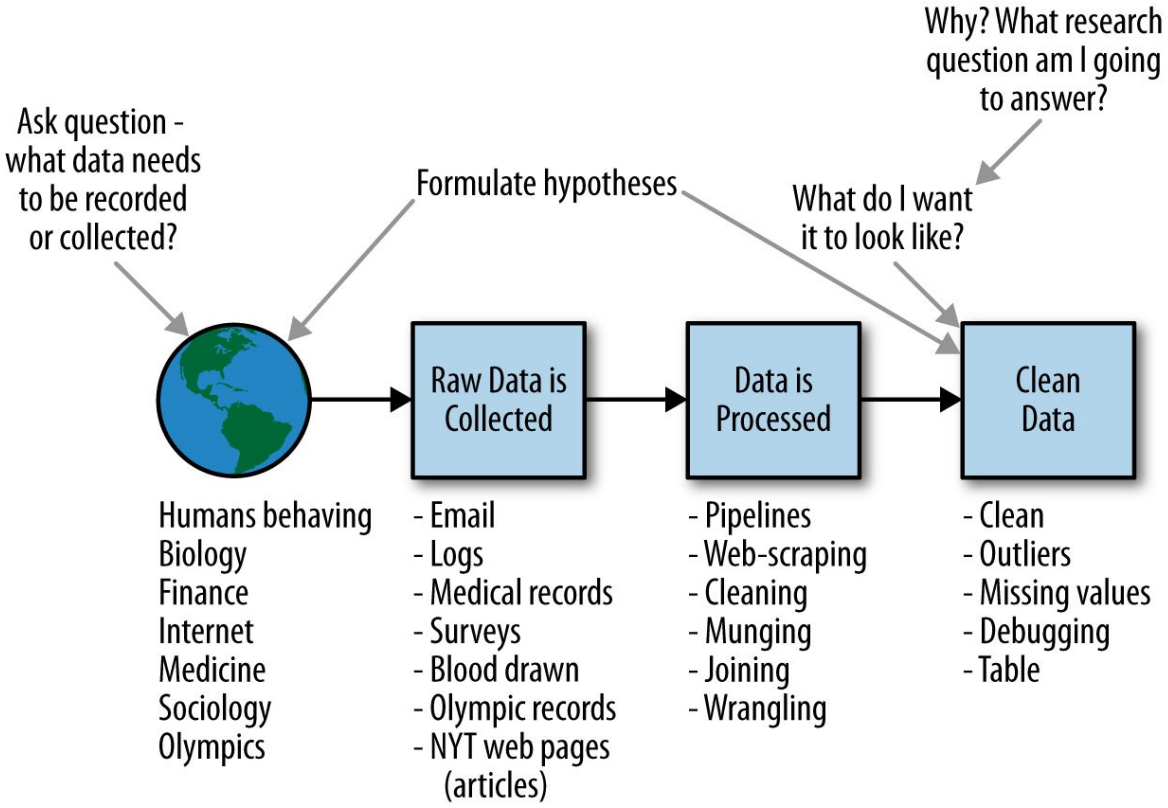
Data science Venn diagram (Drew Conway)



Data science process



The typical day of a data scientist



The Scientific Method, revisited

Ask a question.

- Do background research
- Construct a hypothesis
- Test your hypothesis by doing an experiment
- Analyze your data and draw a conclusion
- Communicate your results

Synthetic datasets

---

http://scikit-learn.org/stable/datasets/index.html

https://deeplearning4j.org/opendata

https://www.kaggle.com/datasets

make money with ds

Kaggle competitions

---

https://www.kaggle.com/competitions

a few real world examples

Case study #1: the rainfall database

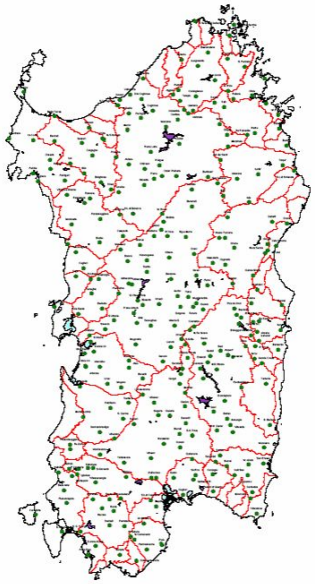
- Consider a large regional database containing:
  - daily temperature (min, max) and rainfall data
- The spatial resolution is a few km (typically every small village has a recording station)
- The database is roughly a hundred years long
- There are missing data
- The data is stored on paper that have been scanned and put in server
- The server gives one year data at a time per meteo station

Case study #1: how do we proceed?

- Is it big data? Does it matter? Estimate the hardware needs for storage/analysis.
- What kind of knowledge can we hope to extract?
- Which methods do we use?
  - data filtering? timeseries analysis? clustering? network analysis? dimensionality reduction? classification?
- Which models do we need?
  - linear models? logistic regression?
- What can we hope to predict after the analysis?

rainfall data

Ubicazione della stazioni



Idrografico di Cagliari
Stampa di controllo di Pluviometria (dati giornalieri) - Del 08/07/2011

950 - DESULO

1922	GEN	FEB	MAR	APR	MAG	GIU	LUG	AGO	SET	OTT	NOV	DIC
Giorni	Pioggia	Pioggia	Pioggia	Pioggia	Pioggia	Pioggia	Pioggia	Pioggia	Pioggia	Pioggia	Pioggia	Pioggia
1	10,0	40,1	—	5,0	—	—	—	—	—	—	10,0	—
2	—	5,0	—	2,0	—	—	—	—	—	—	15,0	—
3	5,2	4,0	—	4,0	10,0	—	—	—	—	—	17,0	—
4	0,3	8,0	—	7,0	24,0	5,0	—	—	10,0	—	20,0	—
5	0,3	2,0	—	—	—	—	—	—	10,0	—	30,0	—
6	4,0	2,0	—	22,0	—	—	—	—	—	—	5,0	—
7	0,2	8,0	—	—	—	—	—	—	—	—	7,0	—
8	0,3	5,0	—	—	—	—	—	—	—	—	2,0	—
9	20,1	0,2	—	—	—	—	—	—	—	—	1,0	12,5
10	12,1	—	—	—	—	17,0	—	—	—	—	—	—
11	—	—	—	—	—	—	—	—	5,0	—	12,0	—
12	—	—	—	—	—	—	—	—	5,0	—	—	—
13	—	—	—	—	—	—	—	—	5,0	—	—	—
14	—	—	—	—	—	—	—	—	10,0	—	—	—
15	—	—	42,0	—	—	1,0	—	—	—	—	—	—
16	5,0	—	11,0	—	—	—	—	—	—	—	—	—
17	—	5,1	5,0	—	—	—	—	—	—	—	—	—
18	—	—	—	15,0	—	—	—	—	—	—	—	17,5
19	0,3	10,2	—	17,0	—	—	—	—	—	—	—	15,0
20	0,9	15,1	—	23,0	—	10,0	—	—	—	—	—	—
21	0,2	—	—	25,0	—	5,0	—	—	—	6,0	5,0	15,0
22	—	0,2	—	—	—	—	—	1,0	—	4,0	—	12,5
23	0,1	10,2	10,0	1,0	—	—	—	—	—	10,0	—	10,0
24	—	10,3	26,0	7,0	10,0	—	—	—	—	15,0	—	17,5
25	—	—	34,0	15,0	—	—	—	—	—	15,0	—	—
26	0,2	—	48,0	—	—	—	—	—	—	13,0	—	10,0
27	0,7	—	52,0	—	—	—	—	—	—	16,0	5,0	—
28	25,0	—	30,0	—	—	—	—	—	—	1,0	2,0	5,0
29	40,0		30,0	—	—	—	—	—	10,0	0,5	—	—
30	0,4		15,0	—	—	—	—	—	5,0	5,0	—	—
31	10,0		8,0		—		—	—		1,1		7,5
totali	135,3	125,4	311,0	143,0	44,0	38,0	—	1,0	60,0	86,6	131,0	122,5
gg pio	9,0	13,0	12,0	12,0	3,0	5,0	—	1,0	8,0	10,0	13,0	10,0

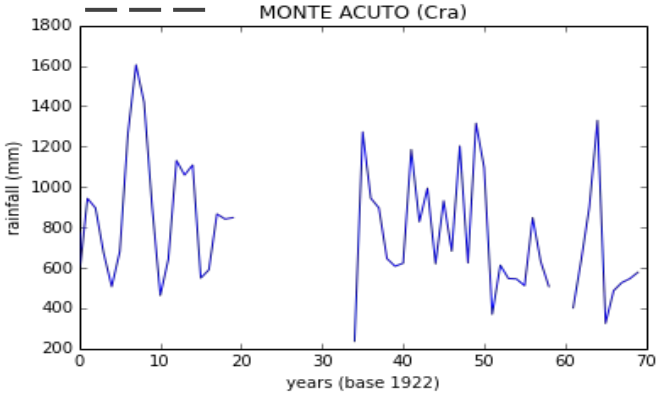
rainfall data

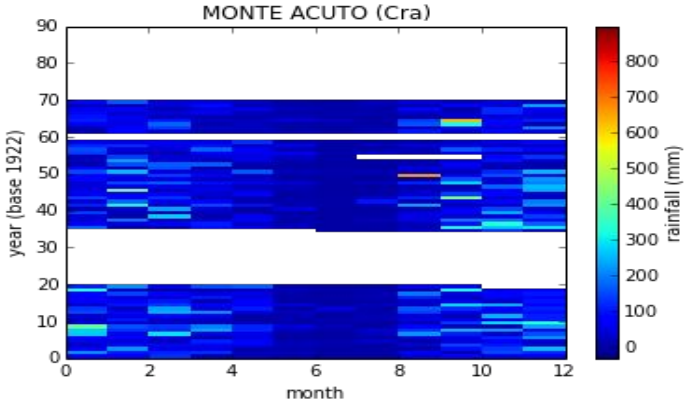
REGIONE AUTONOMA DELLA SARDEGNA

ELENCO DELLE STAZIONI CON COORDINATE E SENSORI

codice	Stazione nome breve	longitudine	latitudine	quota	Sensori
720	Abbasanta	1484600	4441710	317,000	P-PR-T
1780	Aggius	1505450	4530900	514,000	P-PR-T
1730	Aglientu	1509510	4547450	490,000	VR-P-PR-T
1535	Ala' dei Sardi	1527880	4500030	663,000	VR-P-PR-T
580	Ales	1484720	4401710	167,000	P-PR-T
1270	Alghero	1441800	4490000	7,000	P-PR-T
1060	Allai	1488480	4423060	50,000	P-PR-T
670	Arborea	1464300	4402175	7,000	P
1630	Ardara	1484670	4498070	297,000	VR-P-PR-T
1340	Argentiera	1428100	4510140	17,000	P-PR-T
2360	Armungia	1532550	4374880	366,000	P-PR-T
2430	Arqueri'	1531420	4407820	934,000	P-PR-T
1870	Arzachena	1531210	4547940	81,000	P-PR-T
2240	Arzana	1545170	4418760	674,000	P-PR-T
1000	Austis	1507650	4435710	737,000	P-PR-T
2480	B Mandara	1536350	4425950	812,000	P-PR-T
2500	B Muggeris	1536450	4422750	820,000	P-PR-T
470	Bacu Abis	1454000	4343800	60,000	P-PR-T
2370	Ballao	1530980	4377850	100,000	VR-P-PR-T
1420	Bancali	1449610	4509010	74,000	P-PR-T
590	Baradili	1491190	4396630	158,000	VR-P-PR-T
2260	Barisardo	1555070	4410470	50,000	P-PR-T
210	Barrali	1508488	4370560	132,000	P-PR-T
1820	Bassacutena	1521970	4550980	69,000	P-PR-T

rainfall data analysis





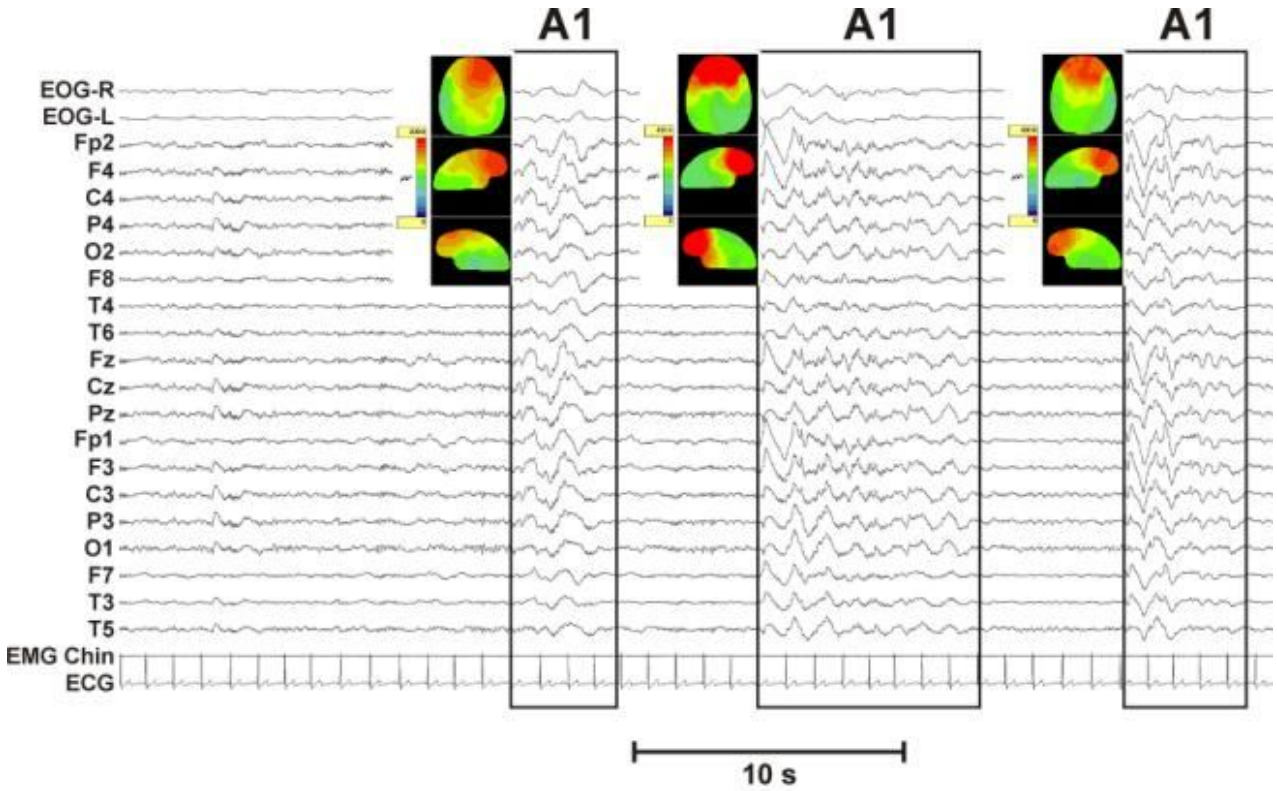
Case study #2: nocturnal EEG data

- Consider a large database containing:
  - EEG data acquired with ~20 channels at 256 bit/s for ~8 hours
  - Number of patients: ~20
  - Number of controls: ~10
  - Patients suffer from nocturnal epilepsy
- Some patients may have multiple recordings
- Some patients may be under anti-epilectic drugs
- The data is stored on DVDs and several computers
- Data was recorded with multiple formats
- Some recordings are noisy due to the environment...

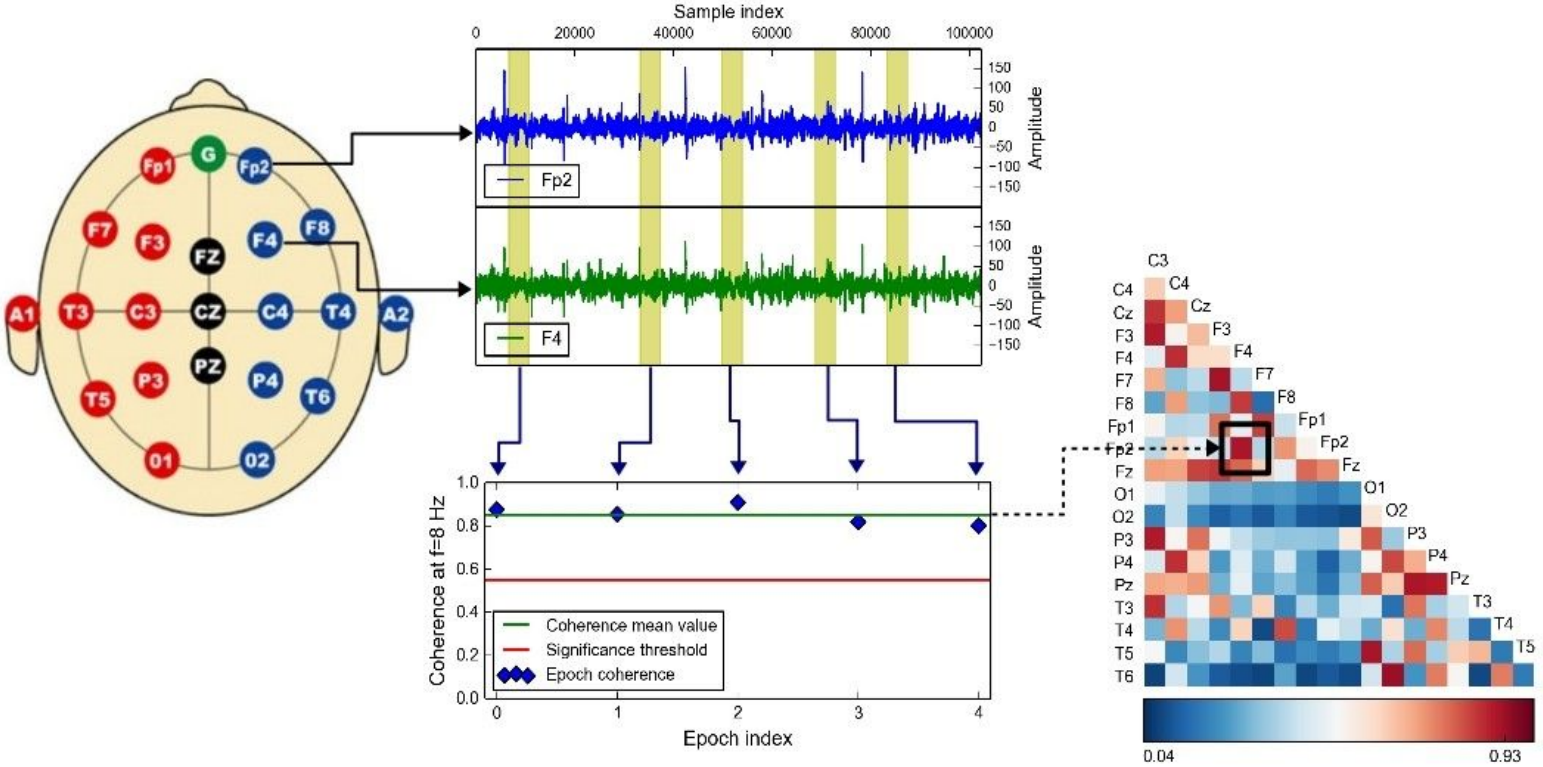
Case study #2: how do we proceed?

- Is it big data? Does it matter? Estimate the hardware needs for storage/analysis.
- What kind of knowledge can we hope to extract?
- Which methods do we use?
  - data filtering? timeseries analysis? clustering? network analysis? dimensionality reduction? classification?
- Which models do we need?
  - linear models? logistic regression?
- What can we hope to predict after the analysis?

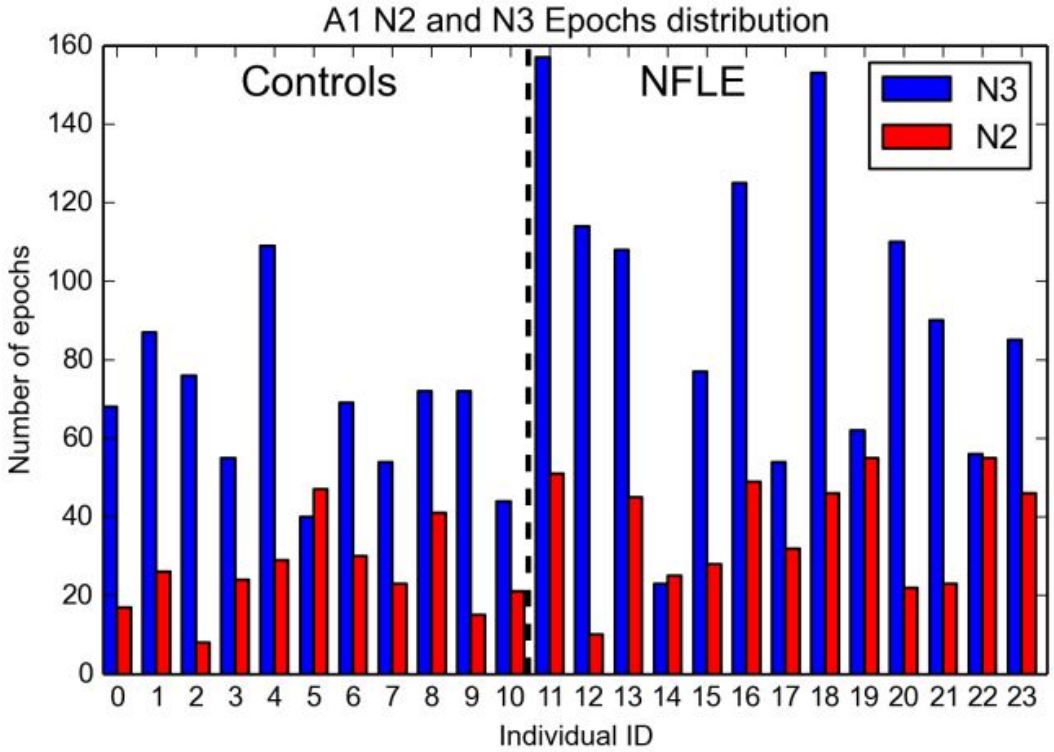
NFLE EEG data analysis



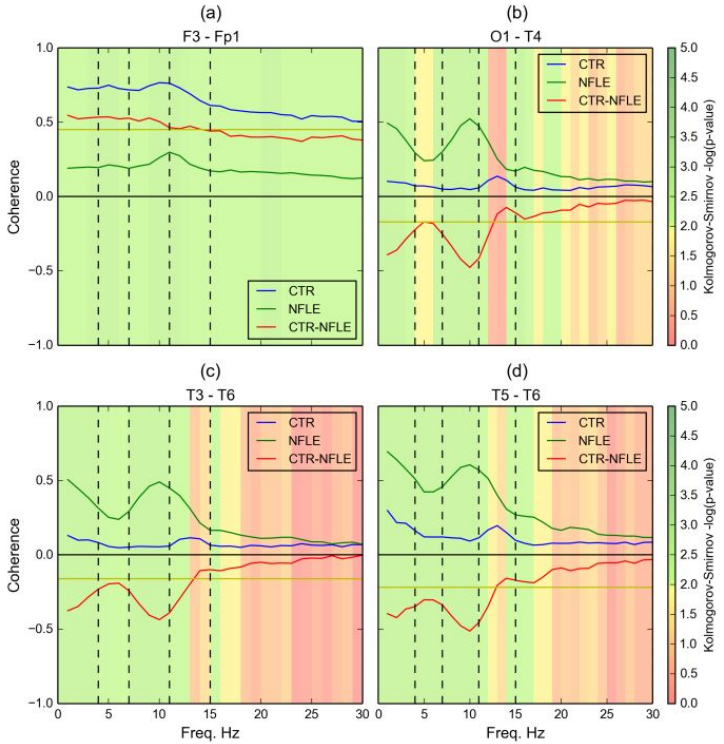
NFLE EEG data analysis

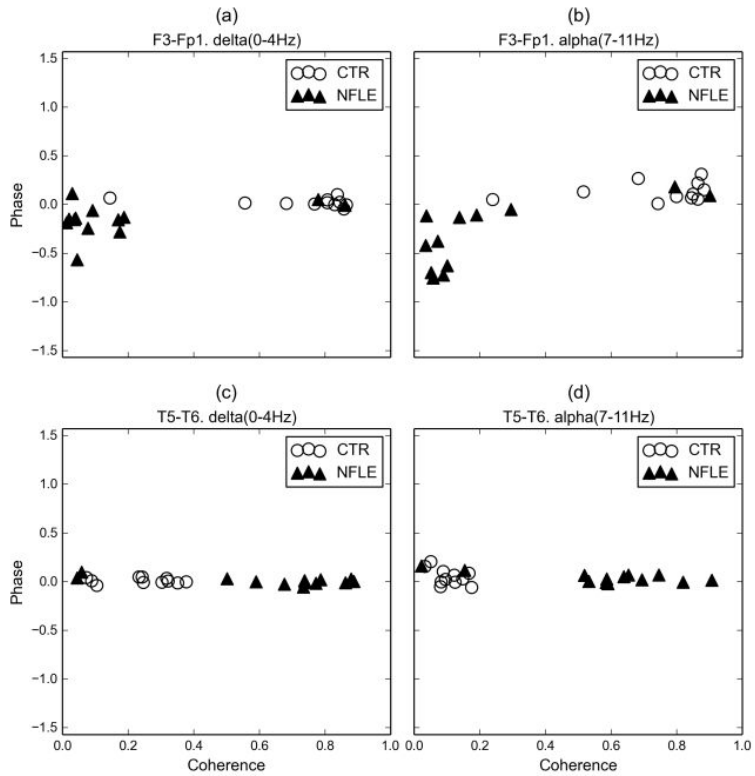


NFLE EEG data analysis

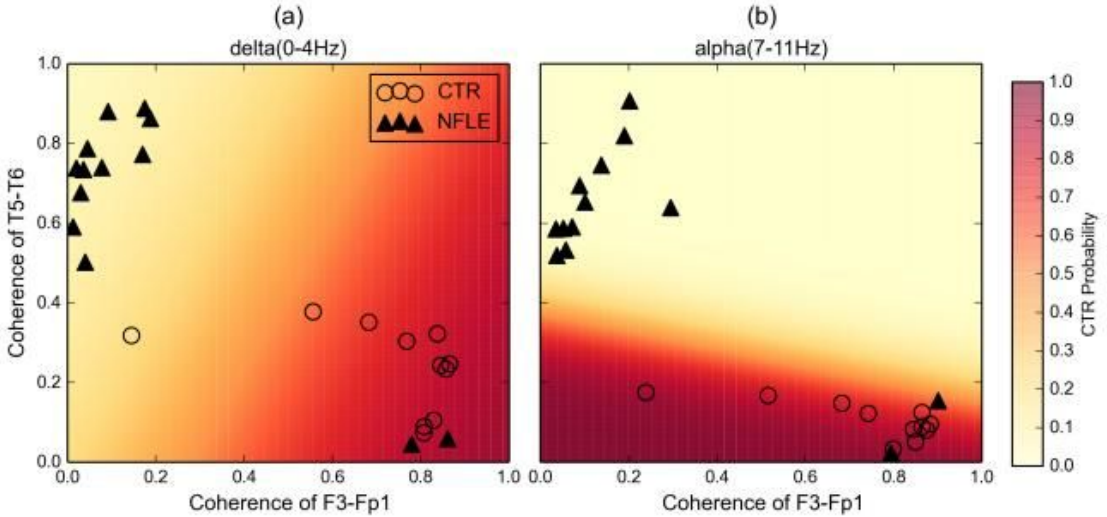


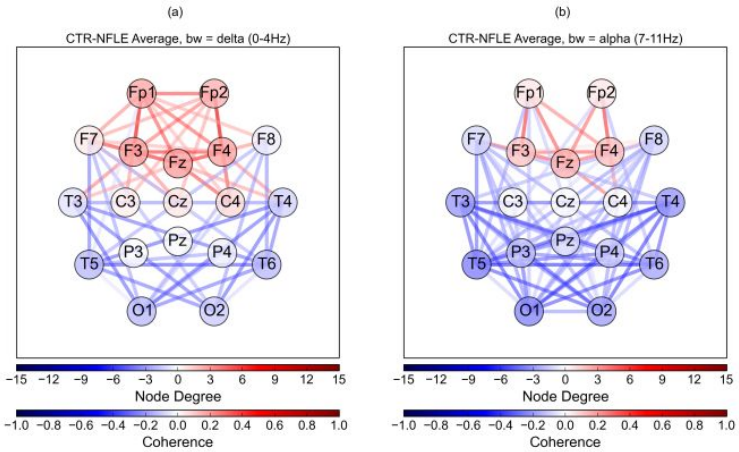
NFLE EEG data analysis





NFLE EEG data analysis: main results





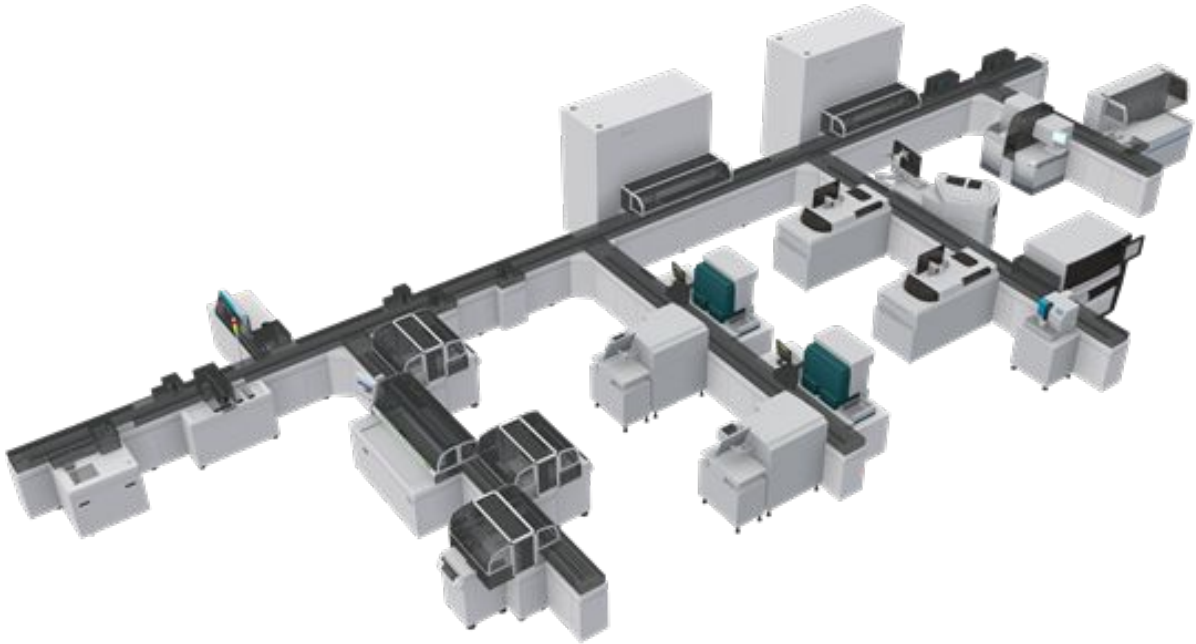
Case study #3: fully-automated clinical analysis plant

- Consider a large database containing:
  - Automation data acquired from hundreds of nodes, several times per s
  - Potentially continuous stream of data
  - ~20.000 tubes analyzed per day, each tube undergoes many analyses
  - Data come from: analytic nodes, automation track, motors, conveyor belts, etc
- Tubes come into batches, randomly inserted
- Some events have the same time tag (time resolution lack)
- If too many tubes are added, the system may collapse
- Who owns the plant does not cooperate with you :)

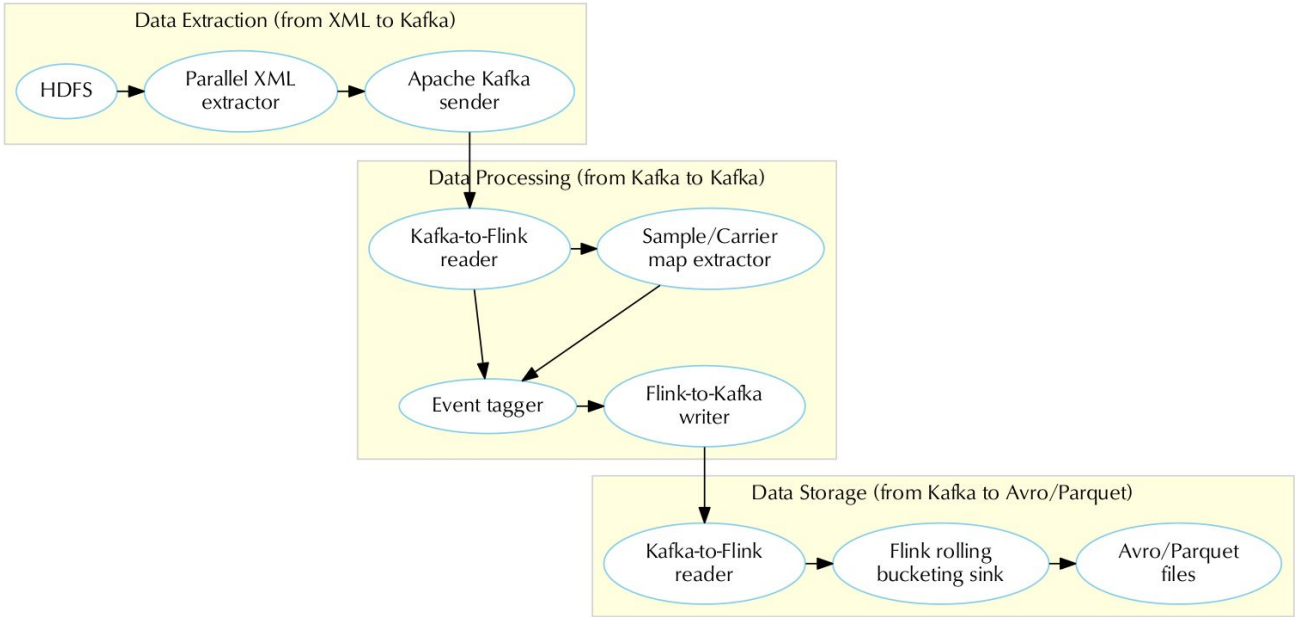
Case study #3: how do we proceed?

- Is it big data? Does it matter? Estimate the hardware needs for streaming/storage/analysis.
- What kind of knowledge can we hope to extract?
- Which methods do we use?
  - data filtering? timeseries analysis? clustering? network analysis? dimensionality reduction? classification?
- Which models do we need?
  - linear models? logistic regression?
- What can we hope to predict after the analysis?
- Do we need real-time analytics tools?

fully-automated clinical processing plant



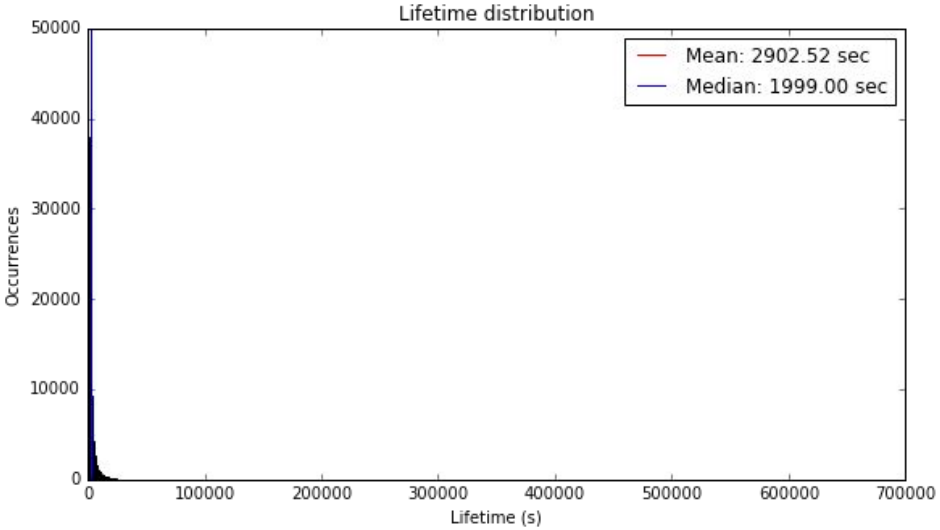
clinical processing plant: data infrastructure

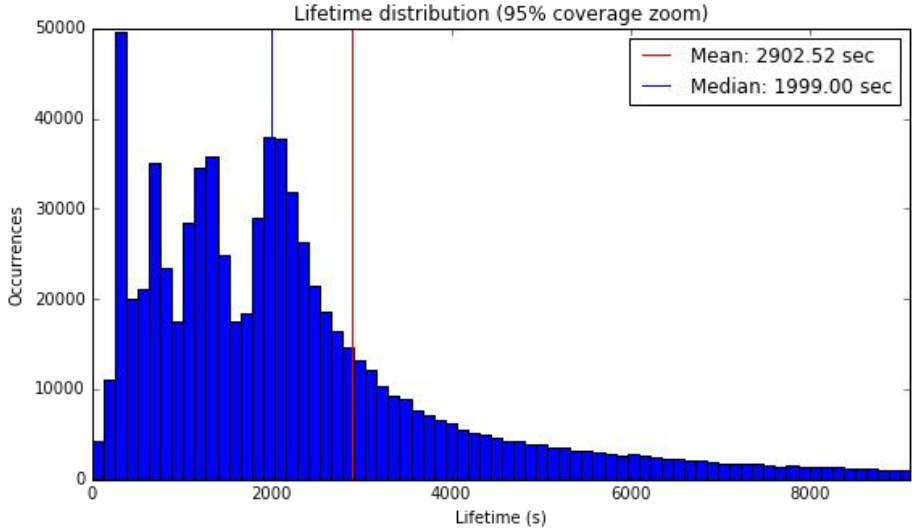


clinical processing plant: typical data content...

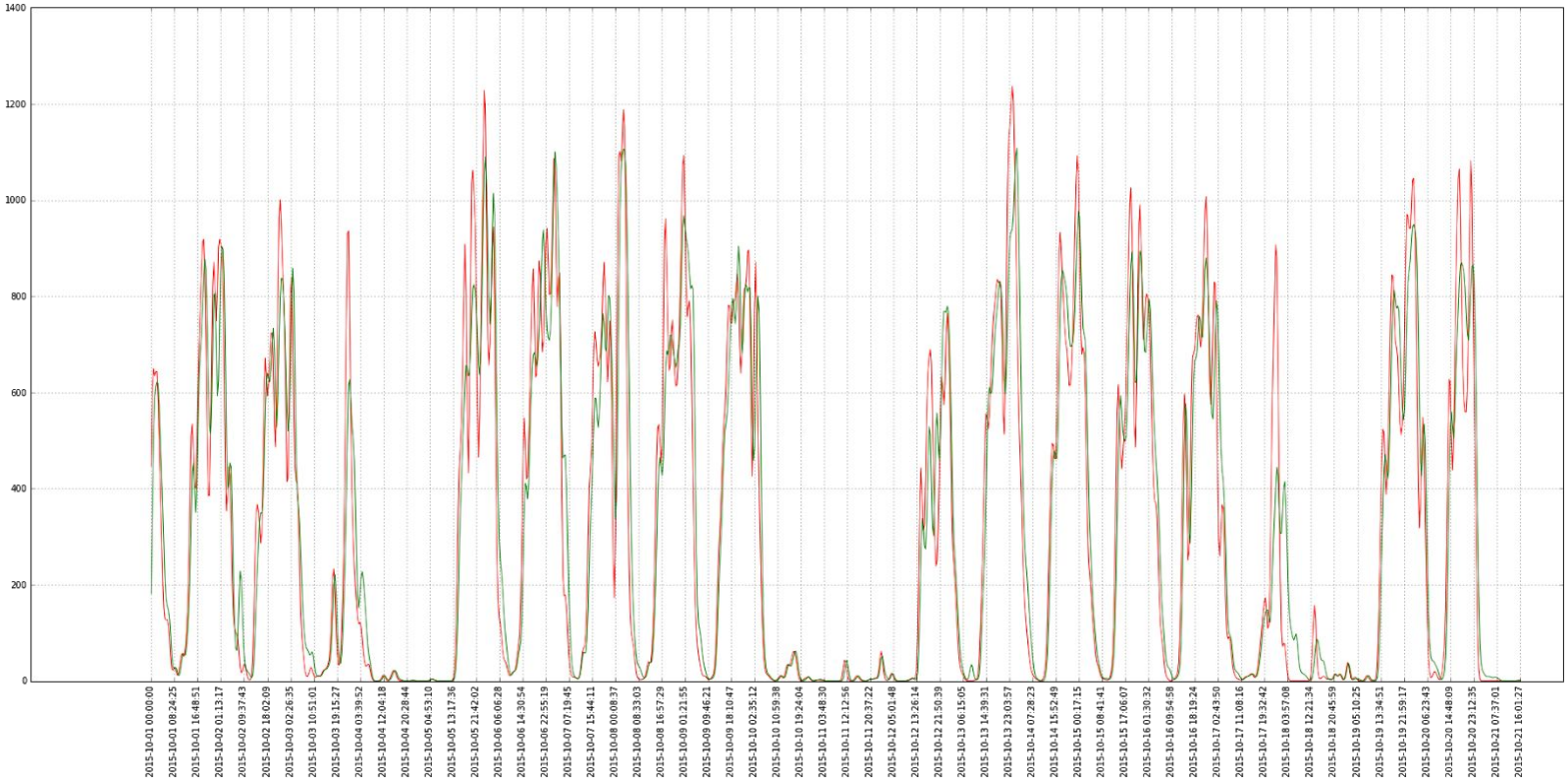
SID	start_ts	stop_ts	status	trace
WC892691L01	1446821935	1446824298	A	u'05:FIM:SAMPLE-DETECTED:1446821935'...
WC806112L02	1446608302	1446625857	A	u'02:FIM:SAMPLE-DETECTED:1446608302'...
WC022765M01	1447284463	1447287096	A	u'09:FIM:SAMPLE-DETECTED:1447284463'...
WC319468L10	1445064990	1445066023	A	u'10:FIM:SAMPLE-DETECTED:1445064990'...
WC026879M01	1447268142	1447268444	A	u'12:IOM:SAMPLE-DETECTED:1447268142'...
WC115843L02	1444438410	1444438830	A	u'29:AQM:RETURNED:1444438410'...
WC427796L03	1445445828	1445447002	A	u'05:FIM:SAMPLE-DETECTED:1445445828'...
WC858376L03	1446734619	1446736440	A	u'10:FIM:SAMPLE-DETECTED:1446734619'...
WC060120M11	1447351136	1447353467	A	u'11:FIM:SAMPLE-DETECTED:1447351136'...
WC970125K05	1444102394	1444103709	A	u'02:FIM:SAMPLE-DETECTED:1444102394'...
WC386425M01	1448339044	1448340997	A	u'05:FIM:SAMPLE-DETECTED:1448339044'...
WC460913L01	1445565425	1445568261	A	u'10:FIM:SAMPLE-DETECTED:1445565425'...
WC218361L01	1444842791	1444851365	A	u'06:FIM:SAMPLE-DETECTED:1444842791'...

clinical processing plant: tubes lifetime within the system

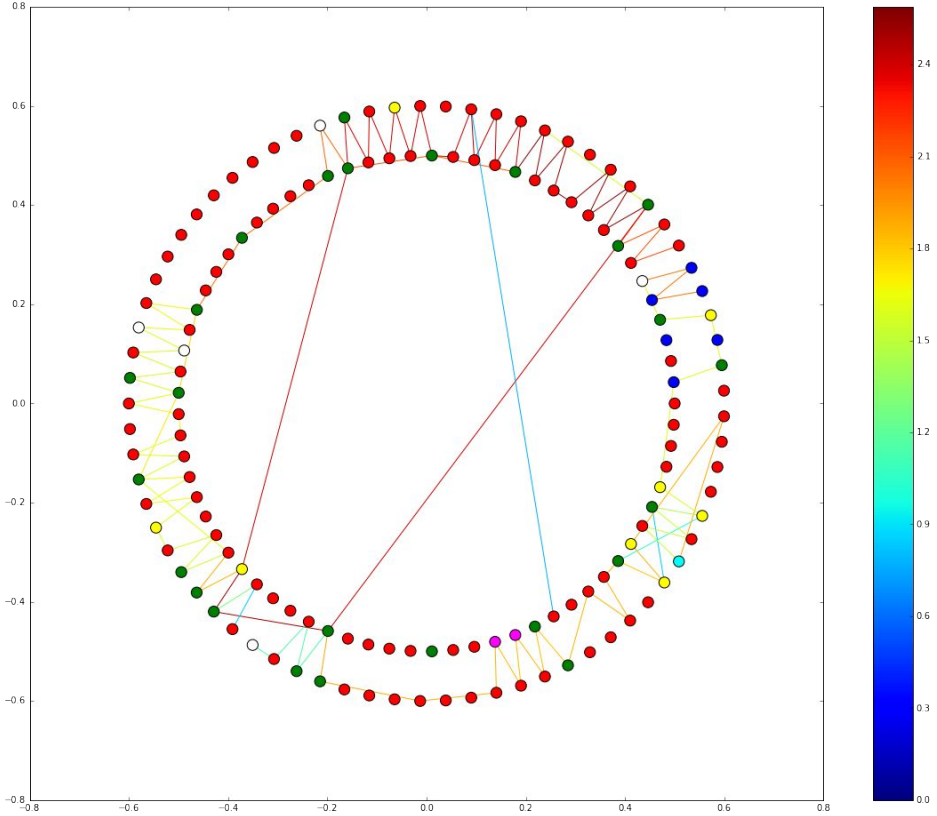




clinical processing plant: tubes in and out (3 weeks data)



clinical processing plant: topology reconstruction



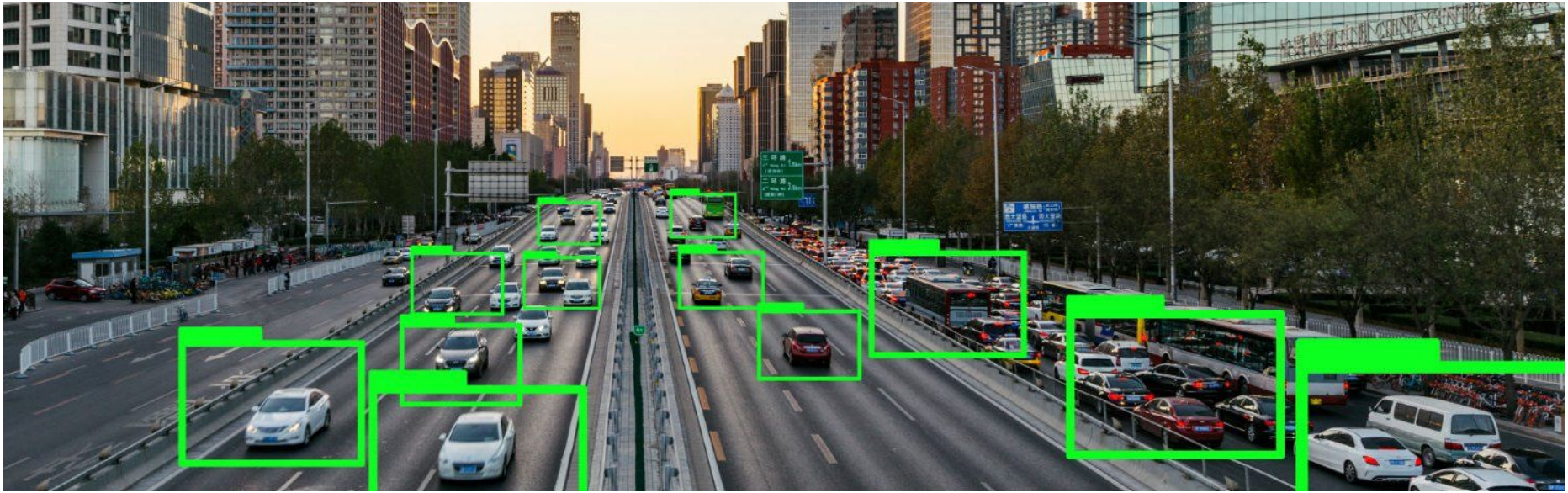
Case study #4: city traffic data

- Consider a large database containing:
  - Automation data acquired from hundreds of traffic monitoring stations, time resolution ~1 s
  - Potentially continuous stream of data
  - Data may also come from: Google Maps (collective), meteo (!), buses with GPS, taxis, etc
- Which other data could be relevant to predict traffic?
- Data may be from many cities with its specific features
- Optimizing the system in some points is always good?
- Who runs the city doesn’t want to test your ideas :)

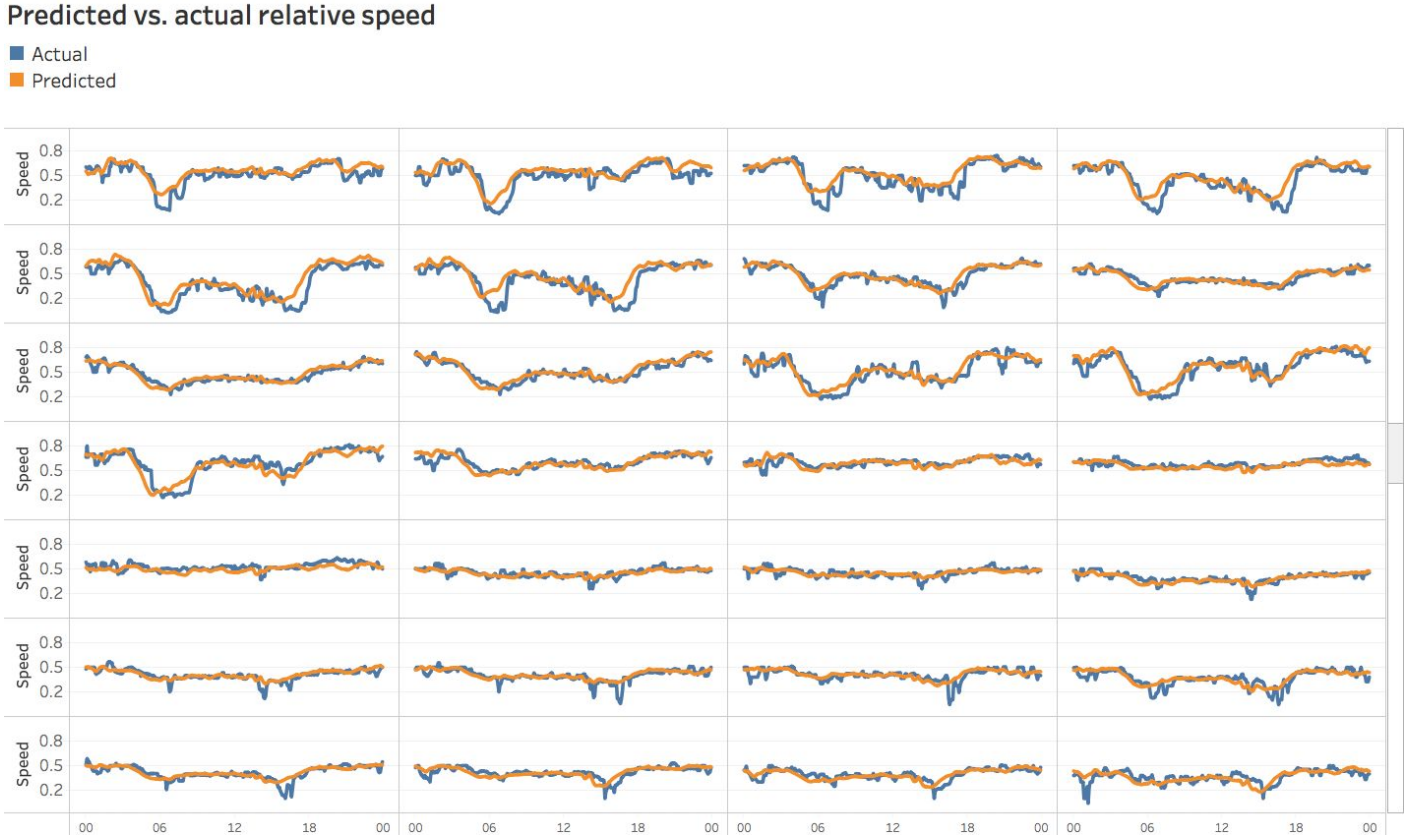
Case study #3: how do we proceed?

- Is it big data? Does it matter? Estimate the hardware needs for streaming/storage/analysis.
- What kind of knowledge can we hope to extract?
- What can we improve?
- Which methods do we use?
  - data filtering? timeseries analysis? clustering? network analysis? dimensionality reduction? classification?
- Which models do we need?
  - linear models? logistic regression?
- What can we hope to predict after the analysis?
- Do we need real-time analytics tools?

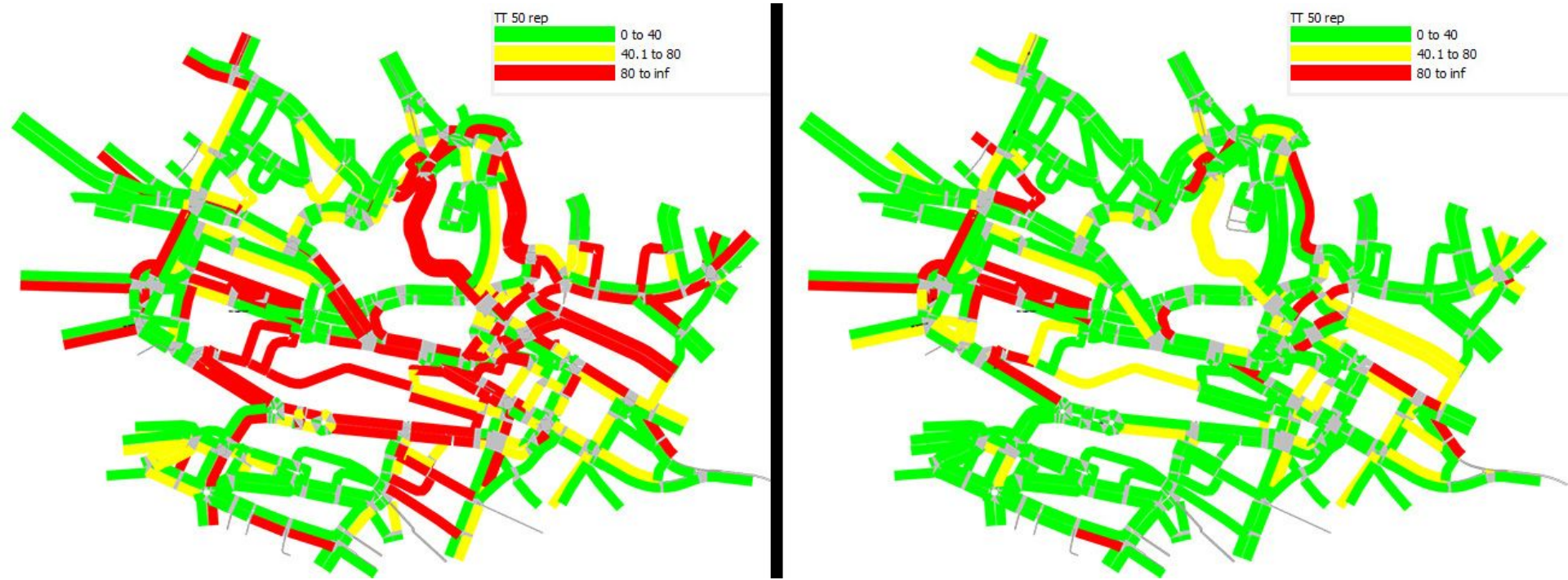
City traffic



City traffic: predictions



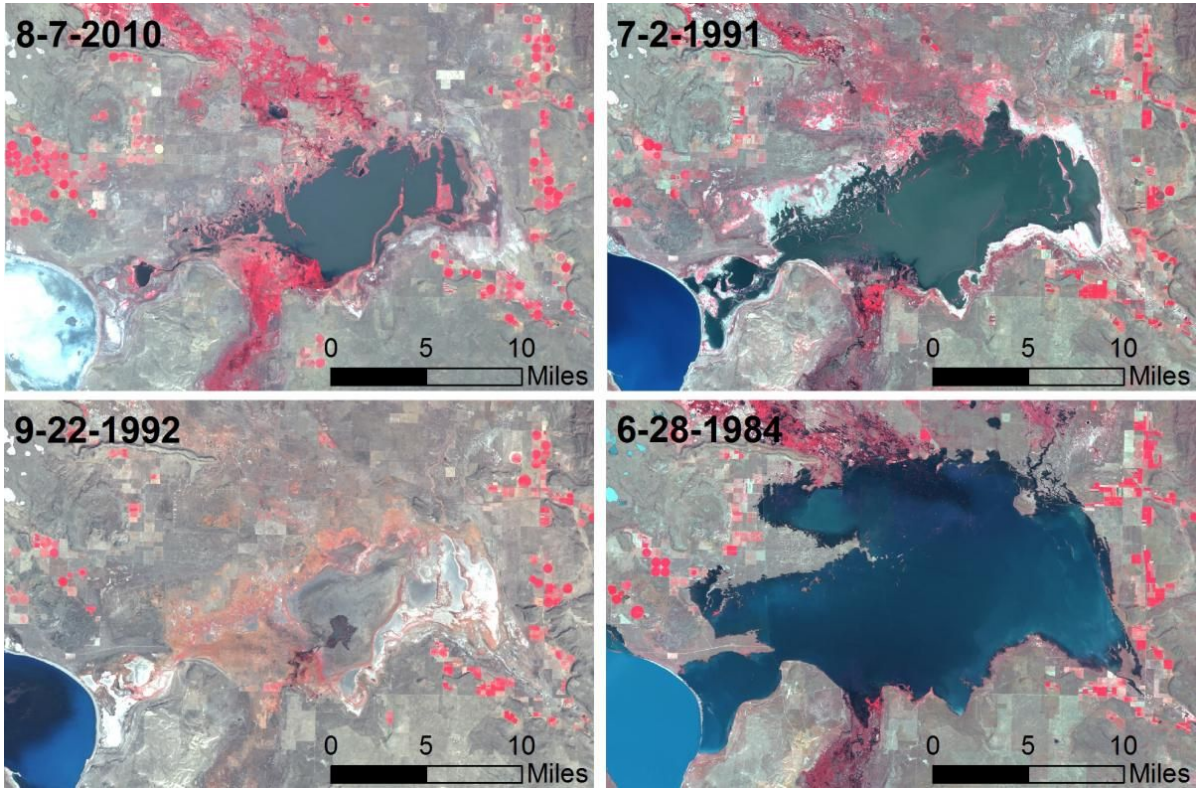
city traffic: better traffic lights



Case study #5: multi-spectral satellite imagery data

- Consider a large database containing:
  - aerial, very high resolution imagery of a large piece of land
  - each image contains several layers each coming from different sensor
  - you may have the same land photographed several times over time
- You're given a set of features we'd like to detect on the pictures
- What we want: for each pixel, the probability of it belonging to each class of features: a car, a house, a river, grass, forest, a road, etc
- Or maybe: detect illegal buildings, check water levels, estimate crop growth/illnesses, air pollution, etc!

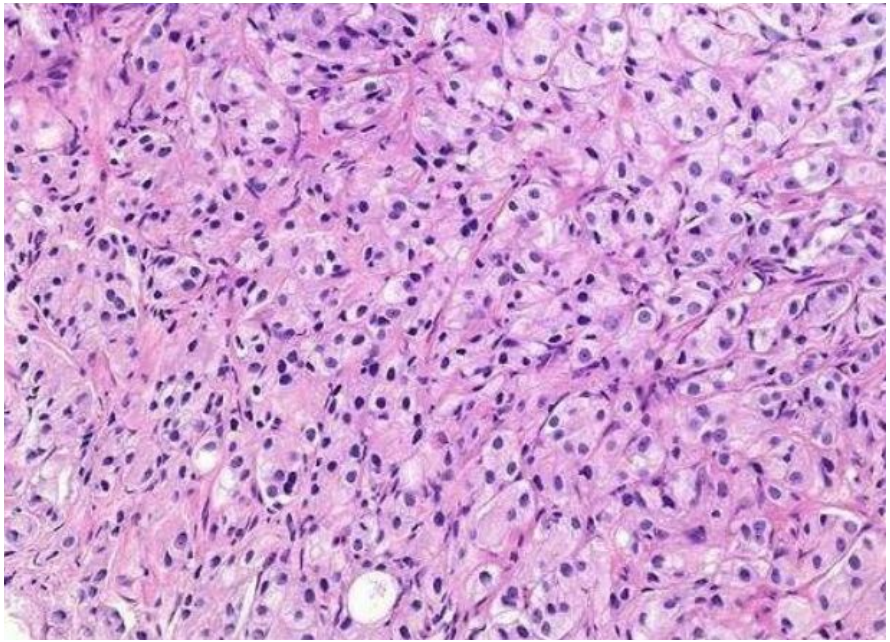
Satellite imagery

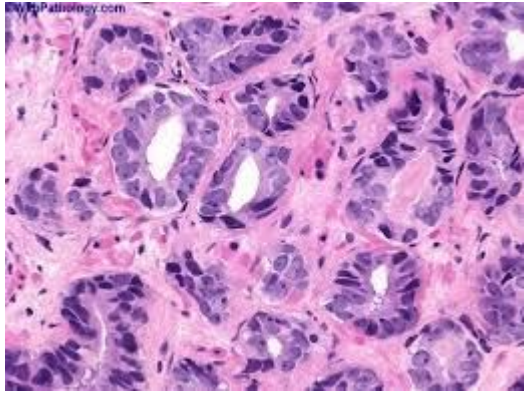


Case study #6: digital pathology

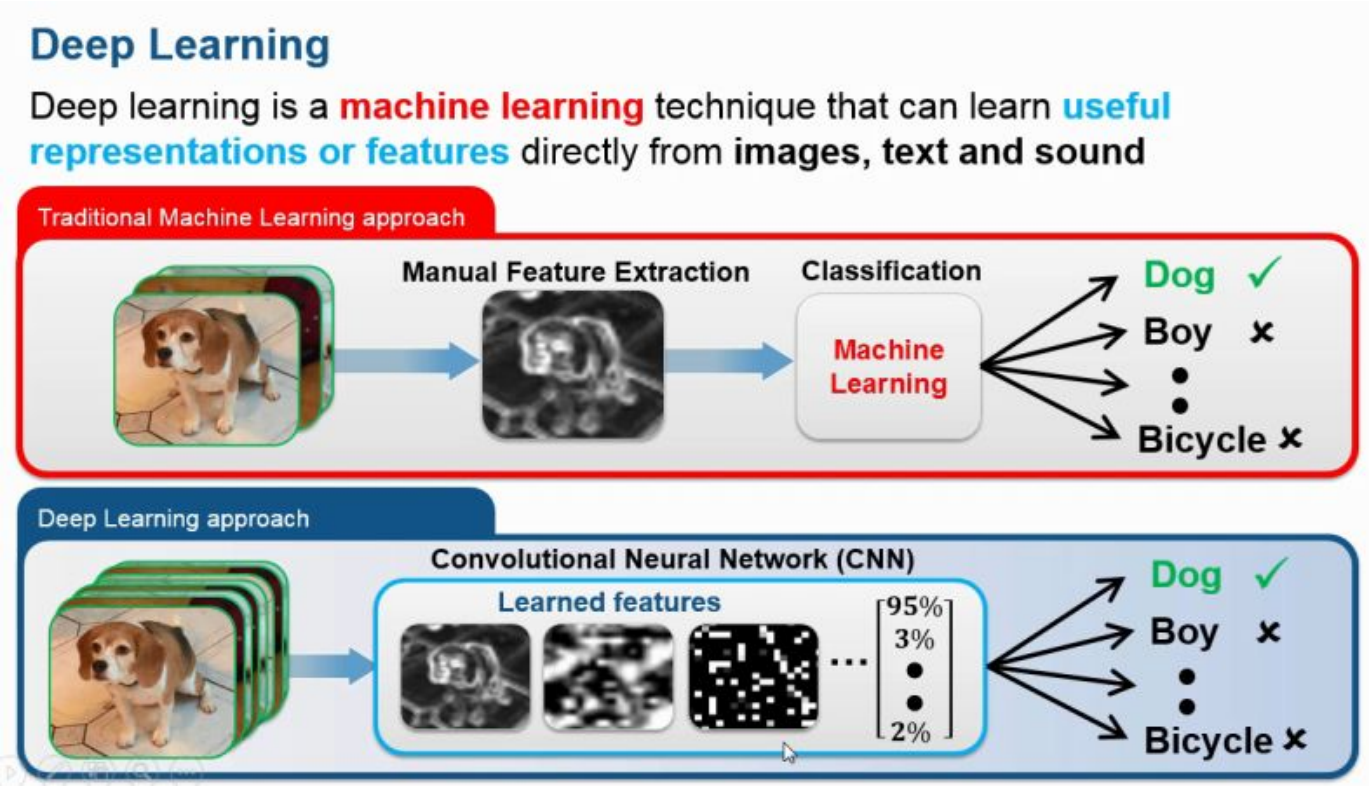
- Consider a VERY large database containing:
  - very high resolution imagery of biological tissue
  - each image is classified by more than one pathologist
    - tissue with cancer cells
    - healthy tissue
    - a lot of white space!
- What we want: for each pixel, the probability of it belonging to two main classes: cancer tissue (several levels), healthy tissue

digital pathology: typical specimens





digital pathology: machine learning or deep learning?



“Entrepreneurial” datasets

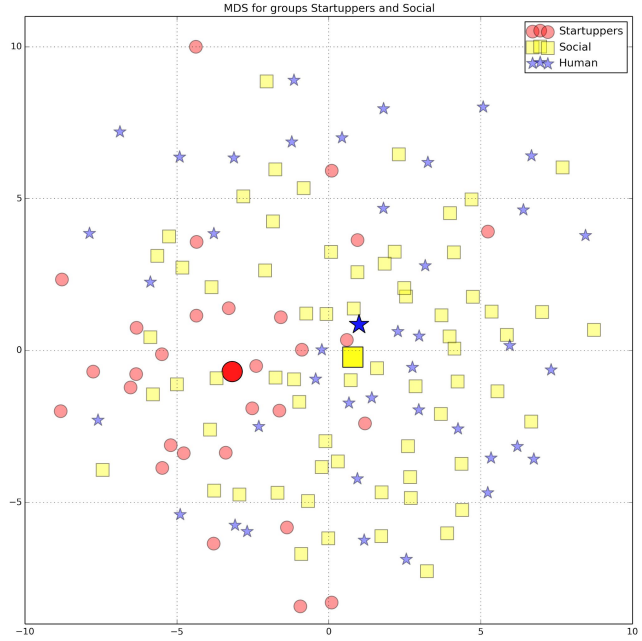
Table 1. Descriptive Statistics for Demographic data

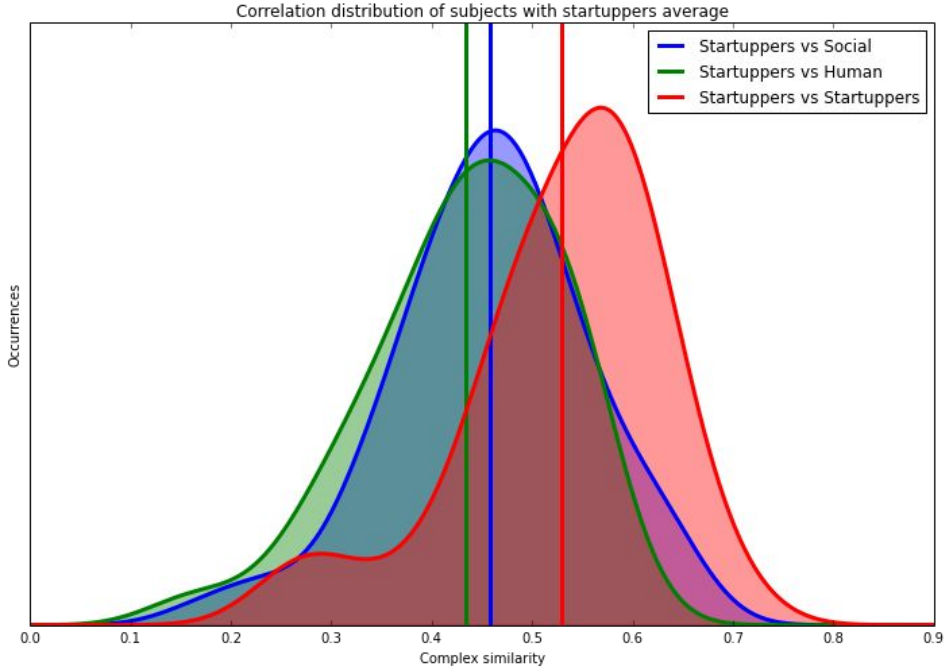
		Age		Experience	
Groups	Participants	Mean	SD	Mean	SD
Start-uppers	29	33	6	4	4
Social sciences students	66	25	5	-	-
Human sciences students	40	27	5	-	-

Table 2. Concepts

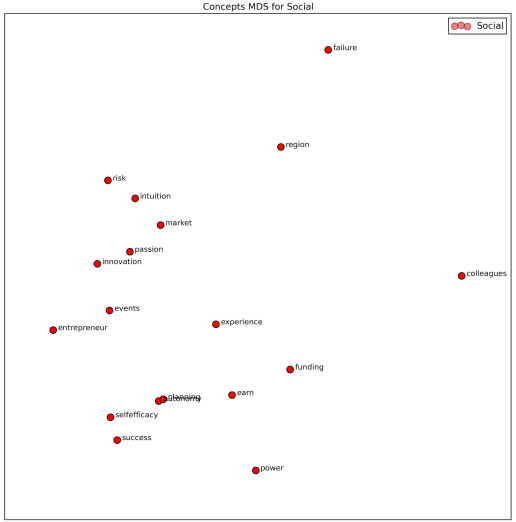
Theoretical factors	**Concepts and Description**	**Abbreviation**
Focal concept	Entrepreneur	entrepreneur
Decision making	To rely on your own experience	experience
	Letting your intuition guide you	intuition
	Planning activities	planning
	Modifying events to create new opportunities	events
Motivation	Achieving personal success	success
	Being self-confident	self-efficacy
	Taking risks	risk
	Earning money	earn
	Being passionate	passion
	Being autonomous	autonomy
	Gaining power	power
	Failure	failure
Context	Market	market
	Innovation	innovation
	Having funds available	funds
	Having colleagues entrepreneurs	colleague
	Region	region

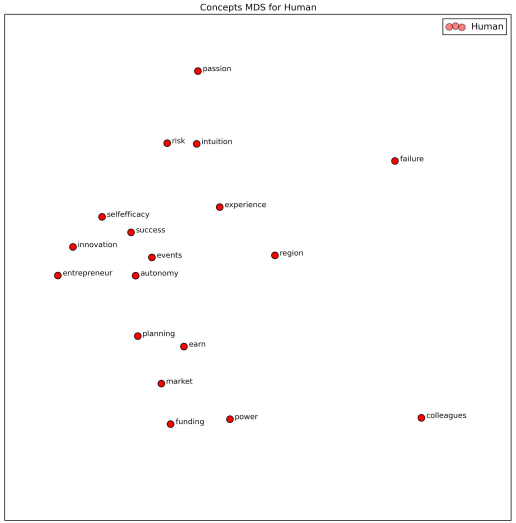
“Entrepreneurial” datasets: PCA and similarity





“Entrepreneurial” datasets: MDS





Is it Science or just data science?

---

data science:

- results are usually valid only locally, for a specific population and timeframe, etc
- you don’t usually expect to “establish the truth”

but:

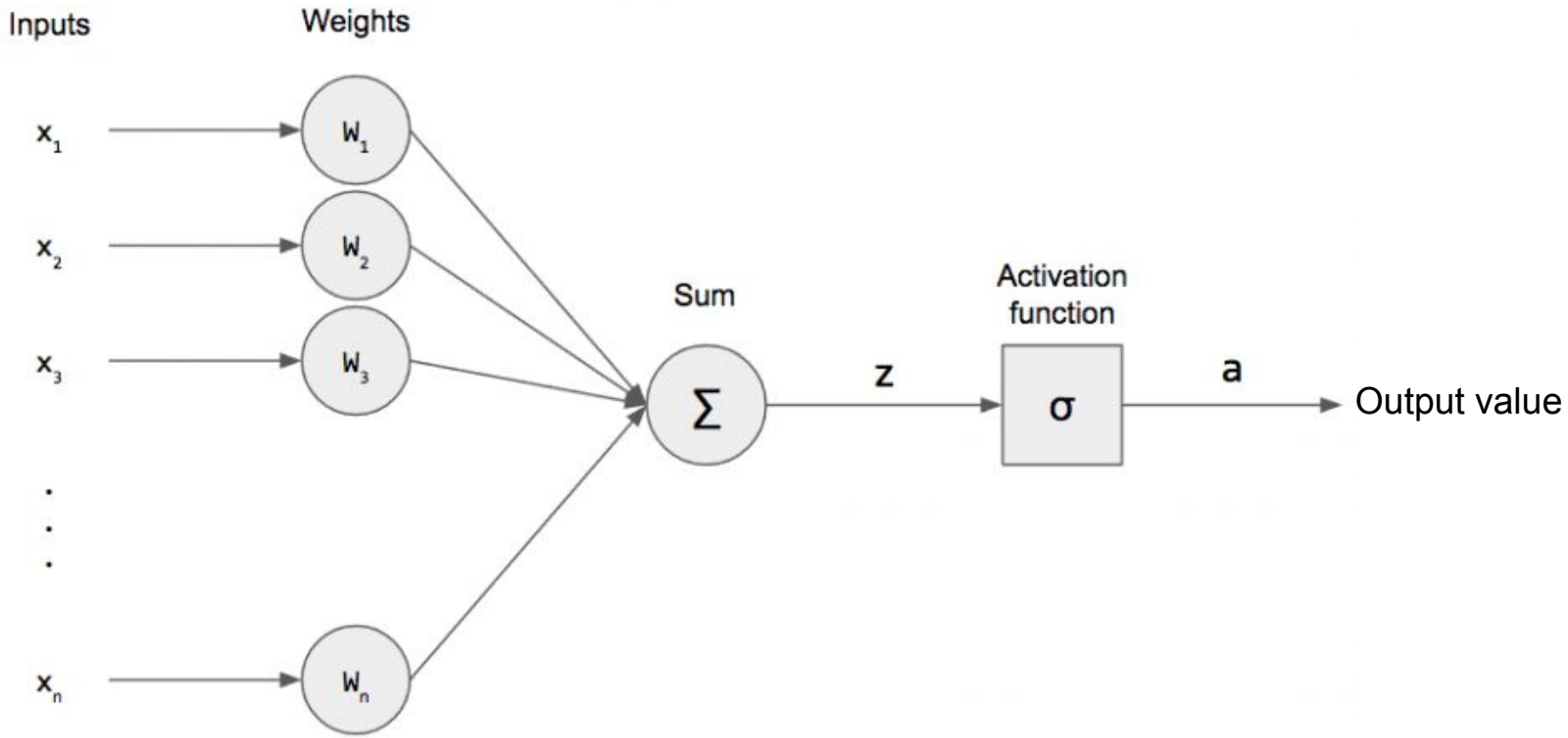
- you get paid (usually) to perform the analyses
- your results let somebody else make more money / improve some collective services (transports, etc.)

“Every science that needs to be called a science is probably not science” cit.

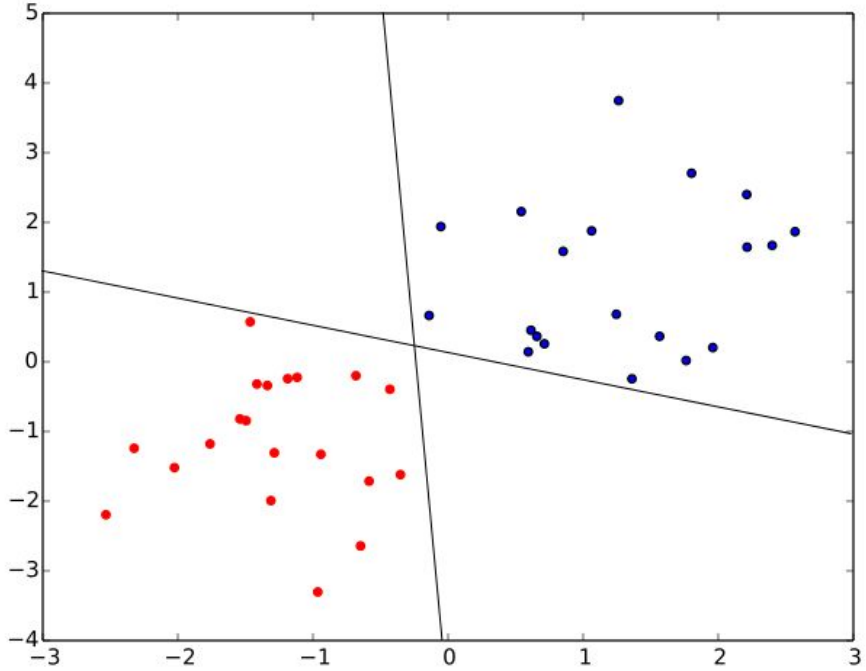
slight detour: optimization

machine learning -> deep learning

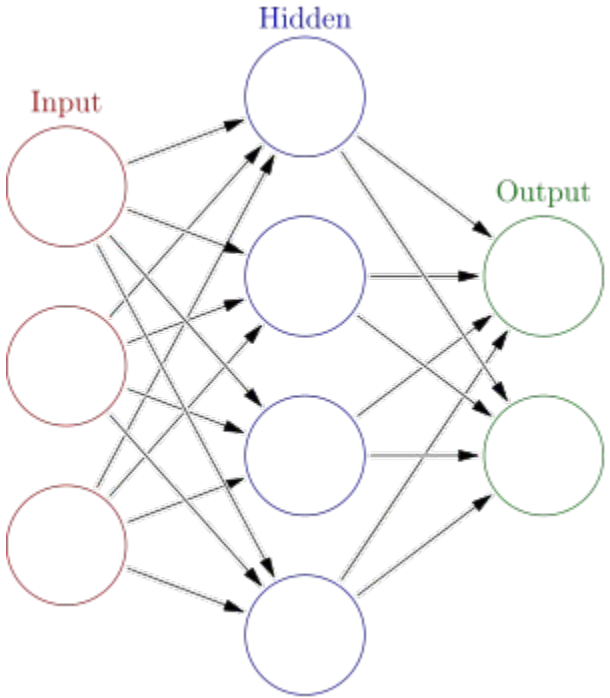
First steps beyond ML: the Perceptron



Perceptron limits



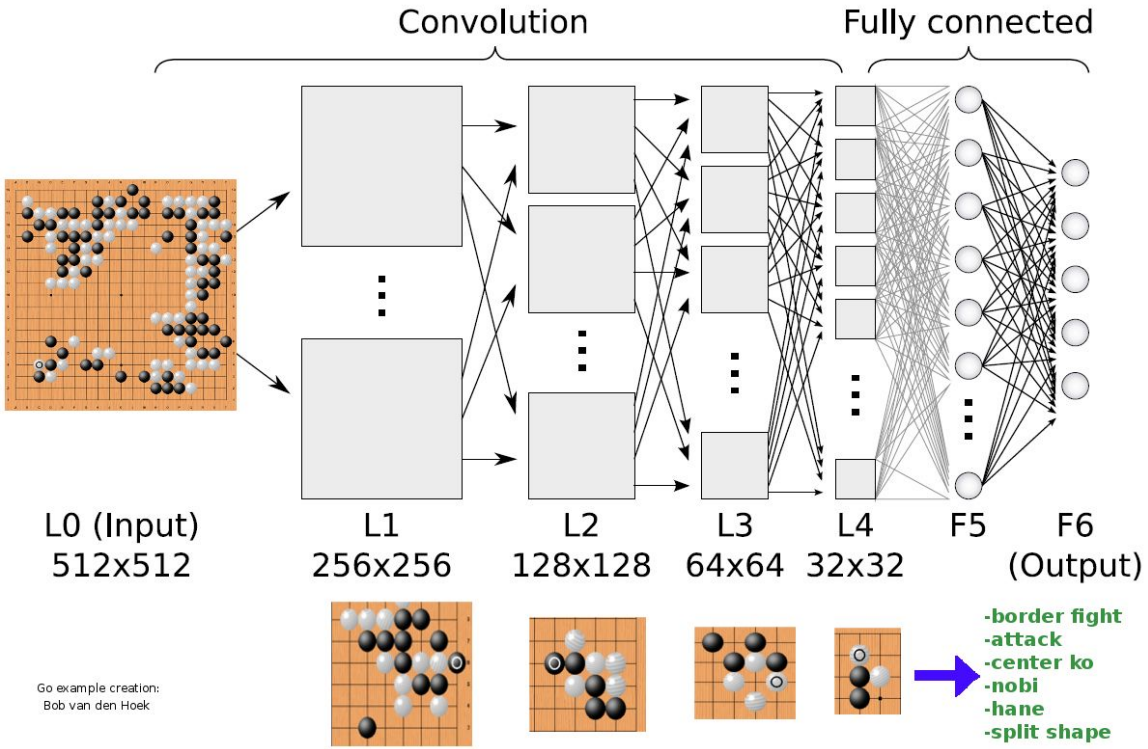
Artificial NN with multiple layers



Recent results

- YOLO NET: https://pjreddie.com/darknet/yolo/ (VIDEO)
- http://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/

Alpha Go (Google)



questions?