# Introduction générale au machine learning

L'IA, un outil de compétitivité pour les entreprises

Formation DIRECCTE

Aurélien Garivier

# Contenu de la séance

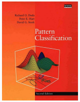
Nous allons voir

- en quoi consiste l'approche "machine learning" pour la résolution de problèmes
- quels sont les grands types de problème résolubles par apprentissage
- quel type de données les algorithmes d'apprentissage savent traiter, et comment parfois s'y ramener
- quels sont les grandes familles d'algorithmes d'apprentissage et quels outils permettent de les mettre en œuvre

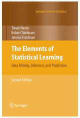
Nous n'allons pas voir :

- l'ensemble des techniques de ML
- la théorie du ML et les problèmes de recherche les plus récents
- comment utiliser des GPU et autres architectures massivement parallèles

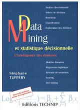
# Bibliographie - Ressources




Pattern Classification (2001) - Wiley Interscience, *R. Duda, P. Hart, D. Stork*




The Elements of Statistical Learning (2001) - Springer, *T. Hastie, R. Tibshirani, J. Friedman*
Disponible en ligne : `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`




Data Mining - Technip, *S. Tufféry*

Cours en ligne de Andrew Ng (Stanford): `https://www.coursera.org/course/ml`

WikiStat `http://wikistat.fr/`
dont sont issus certains de ces slides !

scikit learn `http://scikit-learn.org`

UCI Machine Learning Repository
Base de données de benchmarking: `http://archive.ics.uci.edu/ml/`

# Outline

# Artificial Intelligence (AI): Definition

**Intelligence exhibited by machines**

- emulate cognitive capabilities of humans
  (big data: humans learn from abundant and diverse sources of data).
- a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".

**Ideal "intelligent" machine =**
flexible rational agent that perceives its environment and takes actions that maximize its chance of success at some goal.

**Founded on the claim that human intelligence**
"can be so precisely described that a machine can be made to simulate it."

# Artificial Intelligence: Tension

**Operational goals**

- Autonomous robots for not-too-specialized tasks
- In particular, vision + understand and produce language

**Tension between operational and philosophical goals**

- As machines become increasingly capable, facilities once thought to require intelligence are removed from the definition. For example, optical character recognition is no longer perceived as an exemplar of "artificial intelligence"; having become a routine technology.
- Capabilities still classified as AI include advanced Chess and Go systems and self-driving cars.

# Machine Learning (ML): Definition

**Arthur Samuel (1959)**
Field of study that gives computers the ability to learn without being explicitly programmed

**Tom M. Mitchell (1997)**
A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.

# ML: Learn from and make predictions on data

- Algorithms operate by building a model from **example** inputs in order to make data-driven **predictions or decisions**...
- ...rather than following strictly static program instructions: useful when designing and programming explicit algorithms is unfeasible or poorly efficient.

**Within Data Analytics**

- Machine Learning used to devise complex models and algorithms that lend themselves to **prediction** - in commercial use, this is known as *predictive analytics*.
- www.sas.com: "Produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical **relationships and trends** in the data.
- evolved from the study of pattern recognition and computational learning theory in artificial intelligence.

# Machine Learning: Typical Problems

- spam filtering, text classification
- optical character recognition (OCR)
- search engines
- recommendation platforms
- speach recognition software
- computer vision
- bio-informatics, DNA analysis, medicine
- etc.

For each of this task, it is possible but very inefficient to write an explicit program reaching the prescribed goal.

It proves much more succesful to have a machine infer what the good decision rules are.

# Related Fields

- **Computational Statistics**: focuses in prediction-making through the use of computers together with statistical models (ex: Bayesian methods).
- **Statistical Learning**: ML by statistical methods, with statistical point of view (probabilistic guarantees: consistency, oracle inequalities, minimax)
  $\rightarrow$ more focused on *correlation*, less on *causality*
- **Data Mining** (unsupervised learning) focuses more on exploratory data analysis: discovery of (previously) unknown properties in the data. This is the analysis step of Knowledge Discovery in Databases.
- Importance of **probability**- and **statistics**-based methods $\rightarrow$ **Data Science** (Michael Jordan)
- Strong ties to **Mathematical Optimization**, which delivers methods, theory and application domains to the field

# What is Data?

Data (here): digital content used for learning.

- images (objects, satelite, hand-written text, hyperspectral, etc.)
- times series (finance, earthquakes activity, ECG, etc.)
- internet traces (marketing, social network)
- texts
- audio (signal)
- behaviour: game, robot activity
- ...

# Qu'est-ce qu'une (très grande) masse de données ?

VLDB

Big Data

XLDB

Very Big Data

Data Deluge

Massive Data

Data Masses

**Data inflation** 2

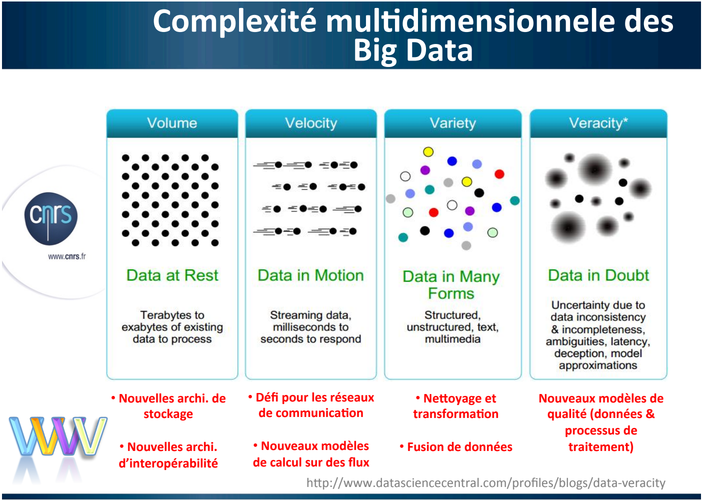
| Unit | Size | What it means |
|---|---|---|
| Bit (b) | 1 or 0 | Short for "binary digit", after the binary code (1 or 0) computers use to store and process data |
| Byte (B) | 8 bits | Enough information to create an English letter or number in computer code. It is the basic unit of computing |
| Kilobyte (KB) | 1,000, or $2^{10}$, bytes | From "thousand" in Greek. One page of typed text is 2KB |
| Megabyte (MB) | 1,000KB; $2^{20}$ bytes | From "large" in Greek. The complete works of Shakespeare total 5MB. A typical pop song is about 4MB |
| Gigabyte (GB) | 1,000MB; $2^{30}$ bytes | From "giant" in Greek. A two-hour film can be compressed into 1-2GB |
| Terabyte (TB) | 1,000GB; $2^{40}$ bytes | From "monster" in Greek. All the catalogued books in America's Library of Congress total 15TB |
| Petabyte (PB) | 1,000TB; $2^{50}$ bytes | All letters delivered by America's postal service this year will amount to around 5PB. Google processes around 1PB every hour |
| Exabyte (EB) | 1,000PB; $2^{60}$ bytes | Equivalent to 10 billion copies of *The Economist* |
| Zettabyte (ZB) | 1,000EB; $2^{70}$ bytes | The total amount of information in existence this year is forecast to be around 1.2ZB |
| Yottabyte (YB) | 1,000ZB; $2^{80}$ bytes | Currently too big to imagine |

Source: *The Economist*

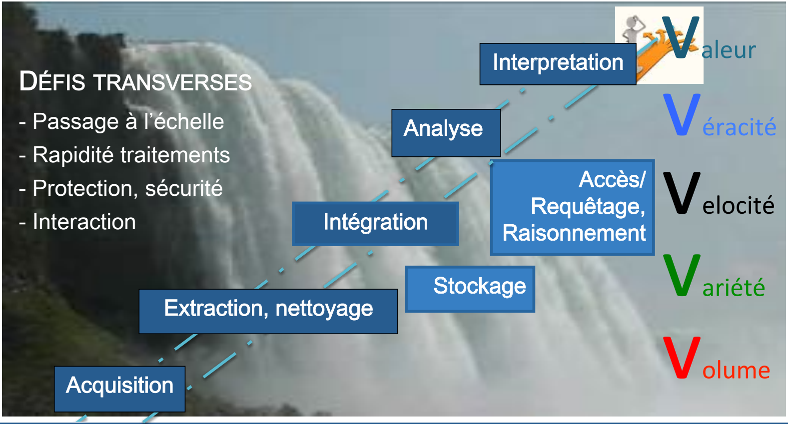
The prefixes are set by an intergovernmental group, the International Bureau of Weights and Measures. Yotta and Zetta were added in 1991; terms for larger amounts have yet to be established.

Grandes Conf du domaine: VLDB, XLDB, ICDE, EDBT, …

[Src: Bouzeghoub, *Mastodons: Une approche interdisciplinaire des Big Data*]



# Défis accompagnant les chgts



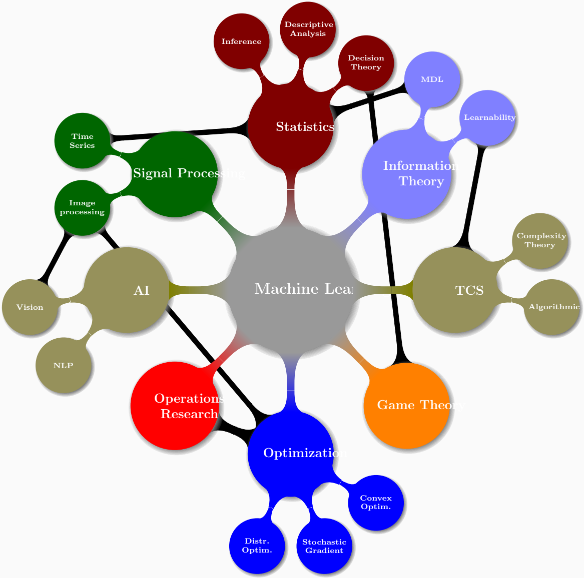

*repenser les outils algorithmiques et mathématiques*

*inspired by "Big Data and Its Technical Challenges, Communications of the ACM, July 2014, vol 57, n°7", © H.V. Jagadish et all.*

# Machine Learning and Statistics

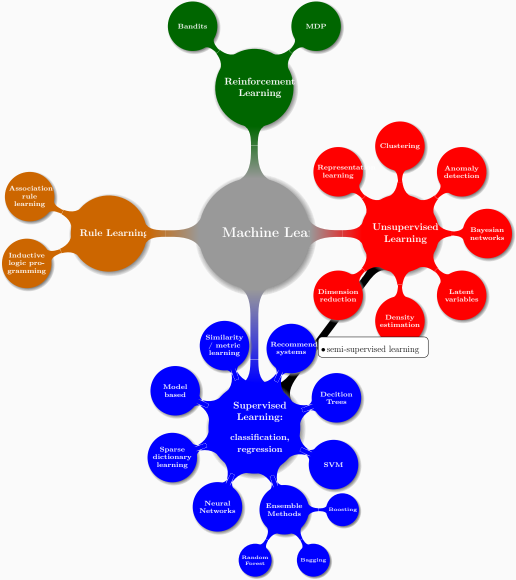
- Data analysis (inference, description) is the goal of statistics for long.
- Machine Learning has more **operational** goals (ex: consistency is important the statistics literature, but often makes little sense in ML).
  Models (if any) are *instrumental*
  Ex: linear model (nice mathematical theory) vs Random Forests.
- Machine Learning/big data: no seperation between statistical modelling and optimization (in contrast to the statistics tradition).
- In ML, data is often here before (unfortunately)
- No clear separation (statistics evolves as well).

# ML and its neighbors



# Outline

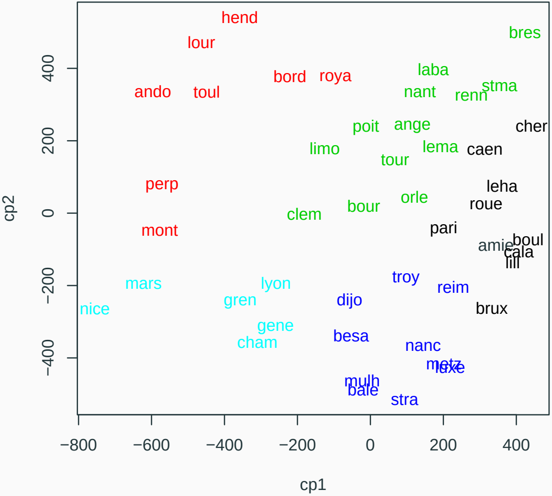
# What ML is composed of



# Unsupervised Learning

- (many) observations on (many) individuals
- need to have a simplified, structured overview of the data
- *taxonomy*: untargeted search for *homogeneous clusters* emerging from the data
- Examples:
    - customer segmentation
    - image analysis (recognizing different zones)
    - exploration of data

# Example



hend
lour
bres
laba
bord
roya
ando
toul
nant
renn
stma
poit
ange
cher
limo
lema
caen
tour
perp
leha
orle
roue
bour
clem
pari
mont
amie
boul
cala
lill
troy
mars
lyon
reim
dijo
gren
nice
brux
gene
besa
cham
nanc
metz
luxe
mulh
bale
stra
−800
−600
−400
−200
0
200
400
cp1
cp2

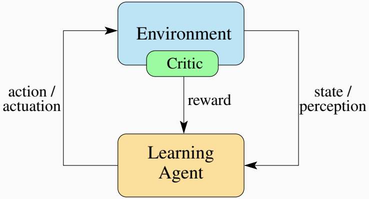
# Supervised Learning

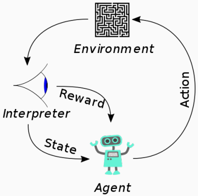
- observations = pairs $(X_i, Y_i)$
- goal = learn to *predict* $Y_i$ *given* $X_i$
- regression (when $Y$ is continuous)
- classification (when $Y$ is discrete)
- statistical technique: linear models

# Example: Character Recognition

| | |
|---|---|
| Input space $\mathcal{X}$<br>Output space $\mathcal{Y}$<br>Joint distribution $P(x, y)$ | $64 \times 64$ images<br>$\{0, 1, \ldots, 9\}$<br>? |
| Prediction function $h \in \mathcal{H}$<br>Risk $R(h) = P(h(X) \neq Y)$ | |
| Sample $\{(x_i, y_i)\}_{i=1}^n$<br>Empirical risk<br>$\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{h(x_i) \neq y_i\}$ | MNIST dataset |
| Learning algorithm<br>$\phi_n : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{H}$<br>Expected risk $R_n(\phi) = \mathbb{E}_n[R(\phi_n)]$ | NN,boosting... |
| Empirical risk minimizer<br>$\hat{h}_n = \mathrm{argmin}_{h \in \mathcal{H}} \hat{R}_n(h)$<br>Regularized empirical risk minimizer<br>$\hat{h}_n = \mathrm{argmin}_{h \in \mathcal{H}} \hat{R}_n(h) + \lambda C(h)$ | |

# Reinforcement Learning




[Src: https://en.wikipedia.org/wiki/Reinforcement_learning]

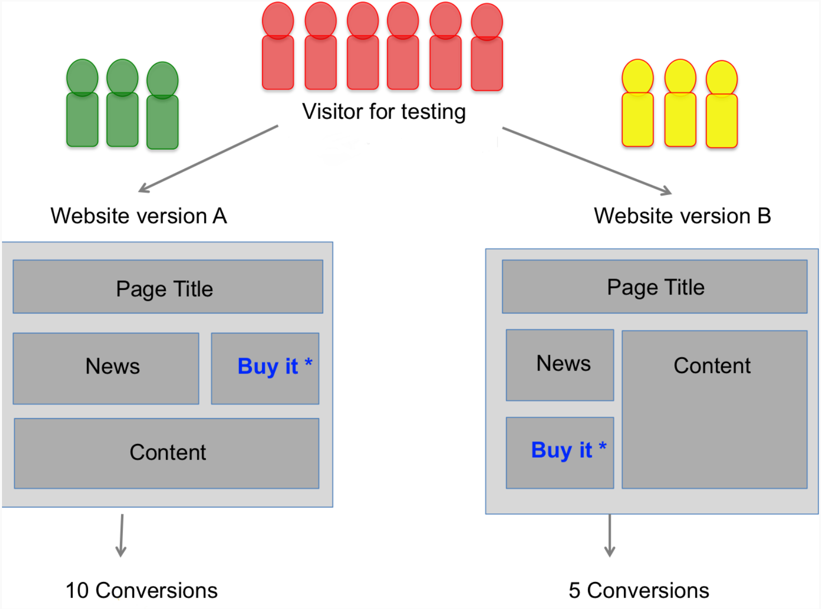
- area of machine learning inspired by behaviourist psychology
- how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.
- Model: random system (typically : Markov Decision Process)
    - agent
    - state
    - actions
    - rewards
- sometimes called approximate dynamic programming, or neuro-dynamic programming

# Example: the Retail Store Management Problem

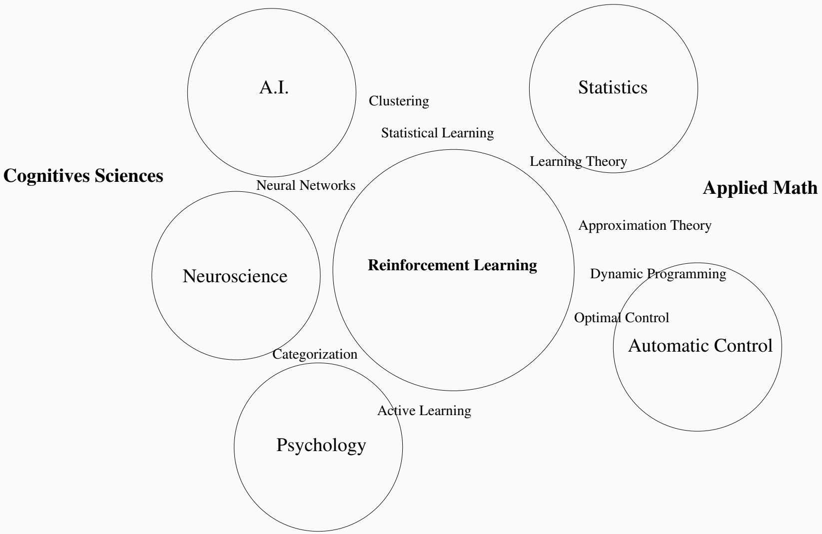
At each month $t$, a store contains $x_t$ items of a specific goods and the demand for that goods is $D_t$. At the end of each month the manager of the store can order $a_t$ more items from his supplier. Furthermore we know that:

- The cost of maintaining an inventory of $x$ is $h(x)$.
- The cost to order $a$ items is $C(a)$.
- The income for selling $q$ items is $f(q)$.
- If the demand $D$ is bigger than the available inventory $x$, customers that cannot be served leave.
- The value of the remaining inventory at the end of the year is $g(x)$.
- Constraint: the store has a maximum capacity $M$.

# Example: A/B testing



# Reinforcement Learning and the others



# Outline

# ML Data

$n$-by-$p$ matrix $X$

- $n$ examples = points of observations
- $p$ features = characteristics measured for each example

Questions to consider:

- Are the features centered?
- Are the features normalized? bounded?

In `scikitlearn`, all methods expect a 2D array of shape $(n, p)$ often called

```
X (n_samples, n_features)
```

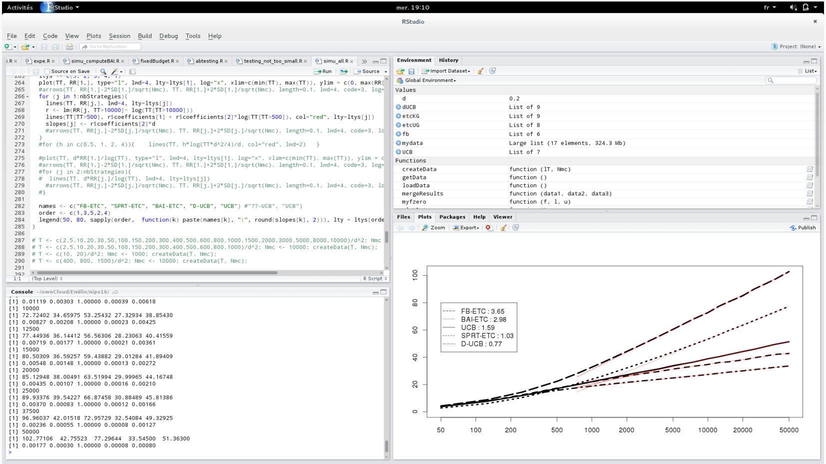
# Data repositories

- Inside R: package `datasets`
- Inside scikitlearn: package `sklearn.datasets`
- UCI Machine Learning Repository
- Challenges: Kaggle, etc.

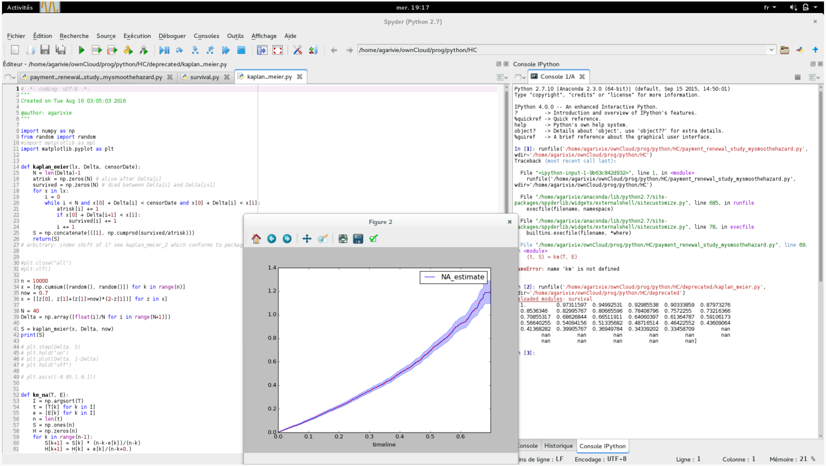
## The big steps of data analysis

1. Extracting the data to expected format
2. Exploring the data
    - detection of outliers, of inconsistencies
    - descriptive exploration of the distributions, of correlations
    - data transformations
3. Random partitioning of the data: (see also: cross-validation)
    - learning sample
    - validation sample
    - test sample
4. For each algorithm: parameter estimation using training and validation samples
5. Choice of final algorithm using testing sample, risk estimation

# Machine Learning tools: R

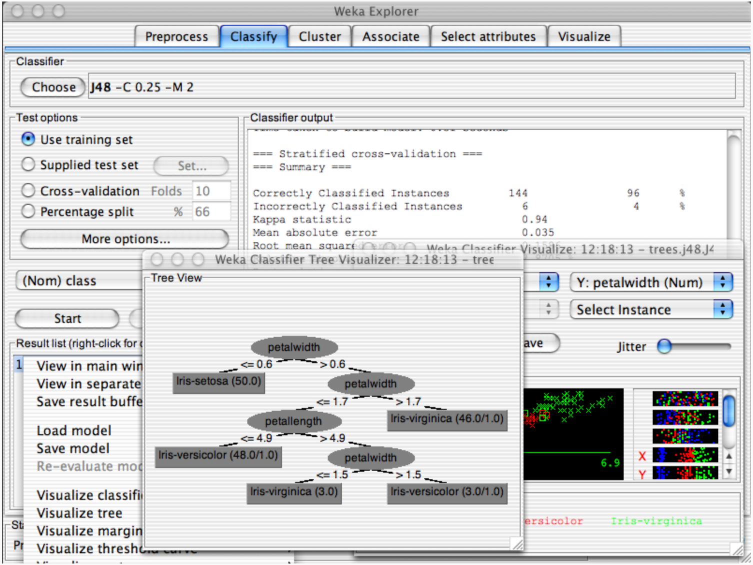


# Machine Learning tools: python



# scikitlearn: `http://scikit-learn.org/stable/index.html`

# Knime, Weka and co: integrated environments

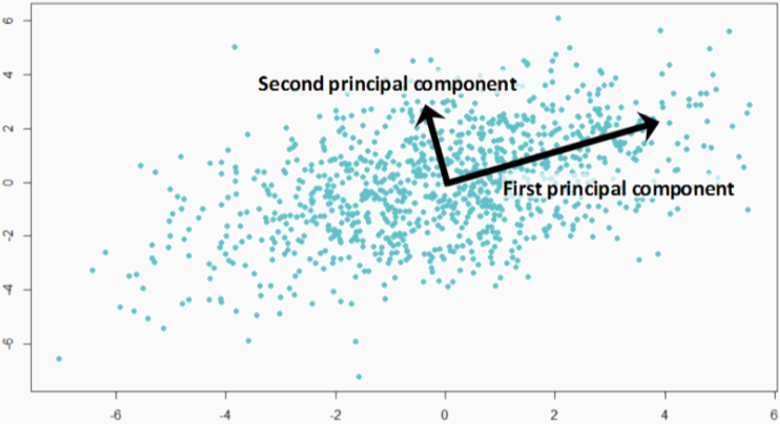


# Outline

# Unsupervised methods

- PCA
- K-means
- Hierarchical Methods
- Others:
  - Spectral clustering
  - t-SNE
  - ...

# PCA: visualization




Src: [https://techannouncer.com/global-pca-unit-market-2017-adelte-airmak-industries-amss-ltd-cavotec-airport-division-ciat-effeti/]

# PCA algorithm

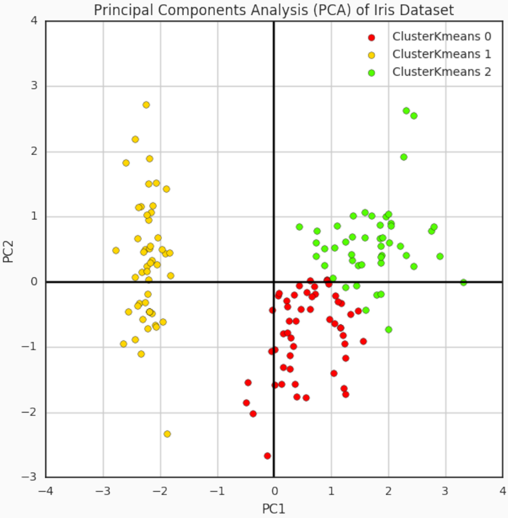
## PCA

- Center all variables
- Compute the $p \times p$ empirical covariance matrix $X^T X$.
- Compute the components $W_d =$ the $d$ first eigenvectors of $X^T X$ in decreasing order of the eigenvalues
- Return the projection of $X$ onto the $d$ first components $T_d = X\,W_d$.

Then:

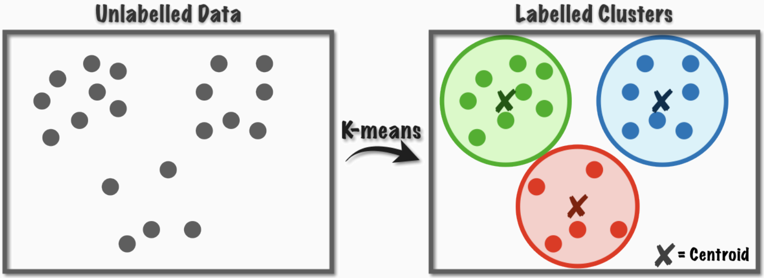
- either vizualize clusters (2d or 3d plots)
- or use another clustering algorithm on the lower-dimensionnal data $T_d$ (*dimension reduction*)

# Example: IRIS




[Src: `https://www.kaggle.com/bburns/iris-exploration-pca-k-means-and-gmm-clustering`]

# Model-free clustering: K-means




Src: `towardsdatascience.com`

# K-means target

- Observations $X_1, \ldots, X_n$ in $\mathbb{R}^p$;
- Objective function: for candidate cluster centers $\mu = (\mu_1, \ldots, \mu_K)$ and cluster assignations $z = (z_1, \ldots, z_n)$:

$$L(\mu, z) = \sum_{k=1}^{K} \sum_{i:z_i=k} \|X_i - \mu_k\|^2 = \sum_{i=1}^{n} \sum_{k=1}^{k} \mathbb{1}\{z_i = k\} \|X_i - \mu_k\|^2$$

- If $S_k = \{i : z_i = k\}$,

$$L(\mu, z) = \sum_{k=1}^{K} |S_k| \, \mathbb{V}ar[S_k]$$

- Minimizing $L$ is equivalent to minimizing pairwise deviations in the clusters:

$$\mathrm{argmin}_{\mu,z} L(\mu, z) = \mathrm{argmin}_{\mu,z} \sum_{k=1}^{K} \frac{1}{|S_k|} \sum_{i,j \in S_k} \|X_i - X_j\|^2$$

# Lloyd's algorithm

- For a fixed $\mu$, optimizing in $z$ is easy: choose $z_i = \mathrm{argmin}_k \|X_i - \mu_k\|$
- BUT optimizing in $\mu$ is NP-hard!

**k-means**

- randomly initialize $\theta_0$
- compute Lloyd's iterations until convergence:
  - *membership variables* $z_i^j = \mathrm{argmin}_k \|X_i - \mu_k^j\|$
  - *updated cluster weights* $N_k^j = \sum_{i=1}^{n} \mathbb{1}\{z_i^j = k\}$
  - *updated cluster means* $\mu_k^{j+1} = \dfrac{\sum_{i : z_i^j = k} X_i}{N_k^j}$
- start again (a few times) to look for a better local optimum

# (Agglomerative) Hierarchical Cluster Analysis

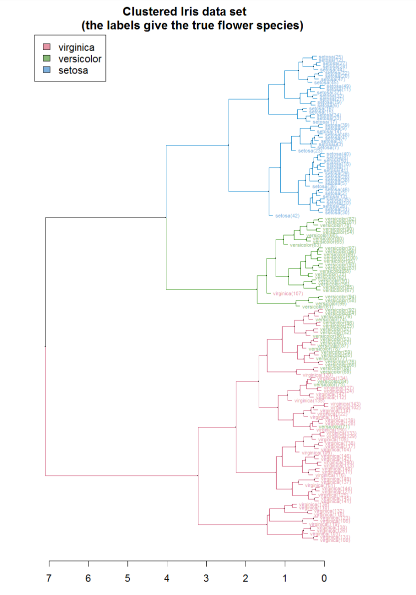
- greedy bottom-up algorithm
- requires a distance (idssimilarity) between observations $\|x - x'\|$
- choice of *distance between clusters:*
  - complete linkage: $d(A, B) = \max \left\{ \|x - x'\| : x \in A,\, x' \in B \right\}$
  - single linkage: $d(A, B) = \min \left\{ \|x - x'\| : x \in A,\, x' \in B \right\}$
  - average linkage distance: $d(A, B) = \dfrac{1}{|A|\,|B|} \displaystyle\sum_{x \in A} \sum_{x' \in B} \|x - x'\|$
  - Ward distance for Euclidian mean: $d(A, B) = \dfrac{|A|\,|B|}{n\big(|A| + |B|\big)} \left\|\bar{A} - \bar{B}\right\|$
  - sum of intra-cluster variance
  - etc.

# HCA algorithm

**HCA**

- Initialization: all observations are clusters $\{X_1\}, \ldots, \{X_n\}$
- As long as there are at least two clusters:
  - add a link between two clusters with smallest distance
  - merge them for the next iterations
- Return the *dendrogram* = hierarchy of clusters

# HCA: Dendrogram

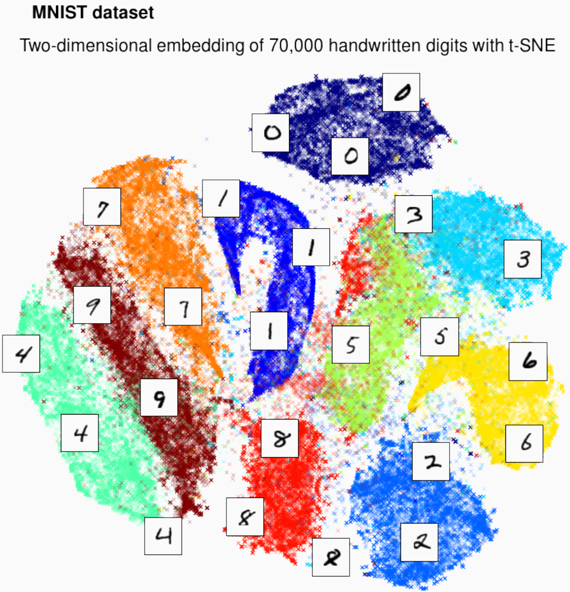



Author: Talgalili

`https://commons.wikimedia.org/wiki/File:`

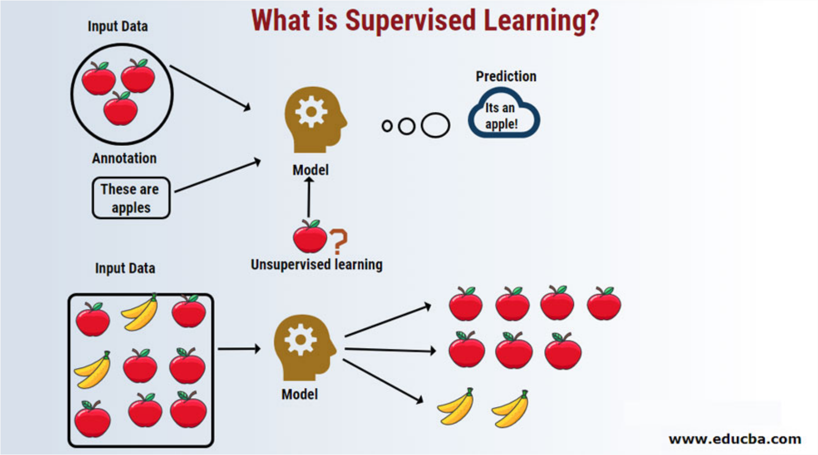
`Iris_dendrogram.png`

# t-SNE, and the others...




Src: `bigsnarf.wordpress.com`

# Supervised learning

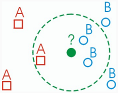


# Qualities of a supervised learning algorithm

- Efficiency (consistency / minimax)
- Tuning: parameter-free
- Result-critic
- Interpretability / Explicability
- Complexity
- Sequentiality
- Privacy

# La méthodes des $k$ plus proches voisins

Règle des $k$ plus proches voisins : pour tout $x \in \mathcal{X}$, trouver ses plus proches voisins $x_{(1)}, \ldots, x_{(k)}$



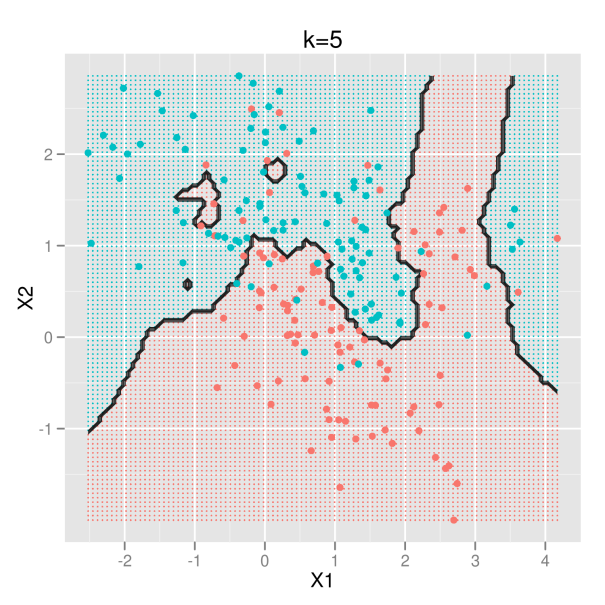

- classification:

$$f_n(x) = \mathrm{argmax}_{y \in \mathcal{Y}} \sum_{j=1}^{k} \mathbb{1}\{y_{(j)} = y\}$$

- régression:

$$f_n(x) = \frac{1}{k} \sum_{j=1}^{k} y_{(j)}$$

# Visualisation d'une règle k-NN



# Propriétés de k-NN

Qualités :

- simplicité
- interprétabilité (?)
- pas de consistance (quel que soit $k$)
- MAIS asymptotiquement erreur au plus 2x supérieure à la règle de Bayes.
- possibilité de faire croître $k$ avec $n$, consistance (théorique) par exemple pour $k = \log(n)$

Paramétrage:

- quelle métrique sur $\mathcal{X}$ ?

$\Longrightarrow$ au minimum, *normaliser* les variables (pb pour les qualitatives)

- Quelle valeur de $k$ choisir ?

# Qualités de kNN

**Interprétabilité:** OUI et NON

**Critique:** OUI mais pas très fiable

**Consistance:** NON mais possible si $k = \log(n)$ (par exemple)

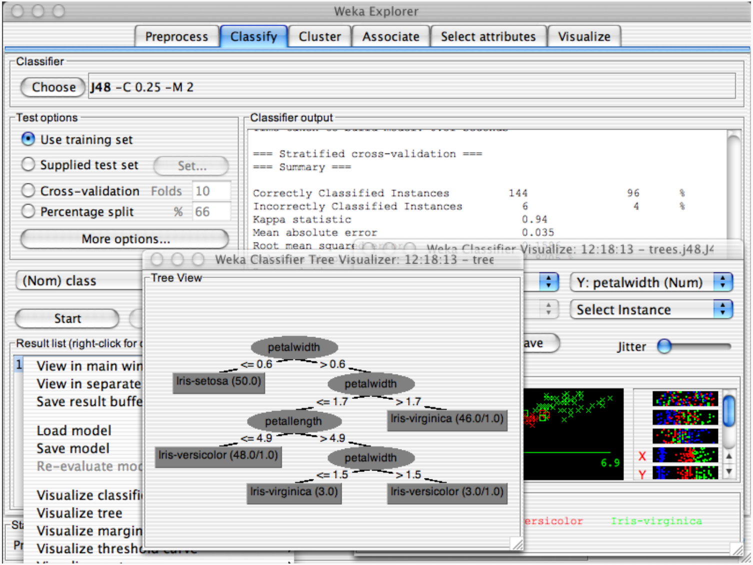
**Minimax:** NON

**Parameter-free:** NON

**Vitesse:** OUI et NON, implémentation possible en $O(n \log n)$

**Online:** OUI

# Arbres de décision



La segmentation par arbre est une approche non-paramétrique de l'analyse discriminante.

But : expliquer une variable réponse (qualitative ou quantitative) à l'aide d'autres variables.

Principe : construire un arbre à l'aide de divisions successives des individus d'un ensemble $E$ en deux segments (appelés aussi noeuds) homogènes par rapport à une variable $Y$ (binaire, nominale, ordinale ou quantitative) en utilisant l'information de $p$ variables $X^1, \ldots, X^p$ (binaires, nominales, ordinales ou quantitatives).

L'arbre obtenu est sous forme d'un arbre inversé comportant à la racine l'échantillon total $E$ à segmenter et les autres segments sont

- soit des segments intermédiaires (encore divisibles),
- soit des segments terminaux.

L'ensemble des segments terminaux constitue une partition de l'ensemble $E$ en classes homogènes et distinctes, relativement à la variable $Y$.

Il s'agit d'arbre de classement si $Y$ est qualitative et d'arbre de régression si $Y$ est quantitative.

## Avantages / Inconvénients

- La méthode CART (Classification And Regression Tree) fournit des solutions sous formes graphiques simples à interpréter.
- Elle est complémentaire des méthodes statistiques classiques, très calculatoire et efficace à condition d'avoir de grandes tailles d'échantillon.
- Elle est capable de gérer à la fois les variables quantitatives ET qualitatives simultanément.
- Peu d'hypothèses requises !
- Algorithme étant basé sur une stratégie pas à pas hiérarchisée, il peut passer à côté d'un optimum global.

Soient $p$ variables quantitatives ou qualitatives explicatives $X^j$ et une variable à expliquer $Y$ qualitative à $m$ modalités $\{\tau_l, l = 1, \dots, m\}$ ou quantitative réelle, observée sur un échantillon de $n$ individus.

La construction d'un arbre de discrimination binaire consiste à déterminer une séquence de noeuds.

- Un noeud est défini par le choix conjoint d'une variable parmi les explicatives et d'une division qui induit une partition en deux classes.
- Une division est elle-même définie par une valeur seuil de la variable quantitative sélectionnée ou un partage en deux groupes des modalités si la variable est qualitative.
- À la racine ou au noeud initial correspond l'ensemble de l'échantillon. La procédure est ensuite itérée sur chacun des sous-ensembles.

L'algorithme considéré nécessite

1. la définition d'un critère permettant de sélectionner la "meilleure" division parmi toutes celles admissibles pour les différentes variables ;
2. une règle permettant de décider qu'un noeud est terminal : il devient alors feuille ;
3. l'affectation de chaque feuille à l'une des classes ou à une valeur de la variable à expliquer.

Le point 2. correspond encore à la recherche d'un modèle parcimonieux. Un arbre trop détaillé, associé à une sur-paramétrisation, est instable et donc probablement plus défaillant pour la prévision d'autres observations.

Breiman et al. ont mise en place une stratégie de recherche de l'arbre optimal.

1. Construire l'arbre maximal $A_{max}$.
2. Ordonner les sous-arbres selon une séquence emboîtée suivant la décroissance d'un critère pénalisé de déviance ou de taux de mal-classés.
3. Sélectionner le sous-arbre optimal : c'est la procédure d'élagage.

**Objectif** : Rechercher le meilleur compromis entre

- un arbre très détaillé, fortement dépendant des observations qui ont permis son estimation, qui fournira un modèle de prévision très instable
- un arbre trop robuste mais grossier qui donne des prédictions trop approximatives.

**Principe**

- Construire une suite emboîtée de sous-arbres de l'arbre maximum par élagage successif.
- Choisir, parmi cette suite, l'arbre optimal au sens d'un critère.

La solution obtenue par algorithme pas à pas n'est pas nécessairement, globalement optimale mais l'efficacité et la fiabilité sont préférées à l'optimalité.

# Qualités du classifieur CART

**Interprétabilité:** OUI !

**Critique:** OUI mais pas très précis

**Consistance:** OUI (sous certaines réserves) MAIS instable!

**Minimax:** NON !

**Parameter-free:** NON

**Vitesse:** OUI

**Online:** NON