Fondamentaux de l'apprentissage et science des données

Une brève introduction aux modèles et outils

E. Viennet

L2TI
Université Paris 13

Plan du cours

Plan

Intelligence Artificielle

L'*intelligence artificielle* (IA) est l'ensemble des sciences et techniques visant à rendre les machines intelligentes.

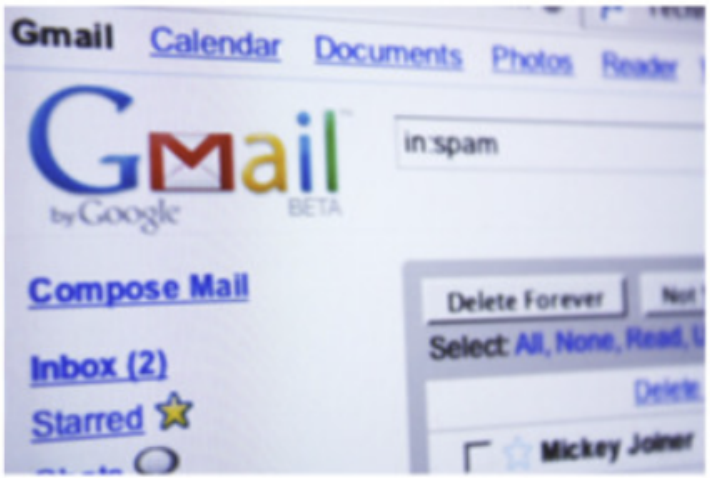
« intelligente » = indistinguable de l'humain ? (test de Turing)

Actuellement, on sait construire des système réalisant presque aussi bien, voire mieux, que l'humain des tâches variées :

- reconnaitre des formes (objets)
- analyser, prévoir des signaux
- reconnaître la voix
- traduire d'une langue à l'autre
- rechercher ou recommander de l'information

Voir `https://experiments.withgoogle.com/collection/ai`

Exemples d'applications du Machine Learning



https://flic.kr/p/5BLW6G [CC BY 2.0]

http://commons.wikimedia.org/wiki/File:American_book_company_1916._letter_envelope-2.JPG#filelinks [public domain]

http://commons.wikimedia.org/wiki/File:Netflix_logo.svg [public domain]

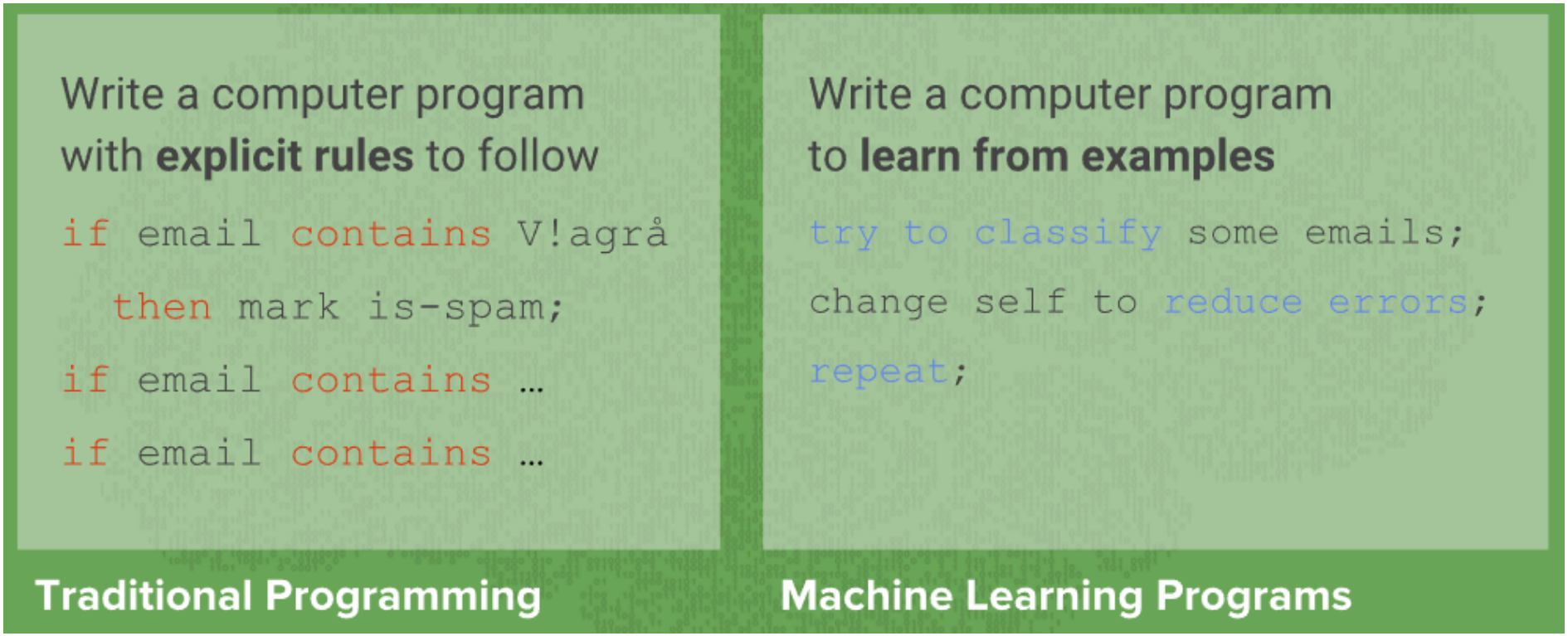
By Steve Jurvetson [CC BY 2.0]

And many, many more …

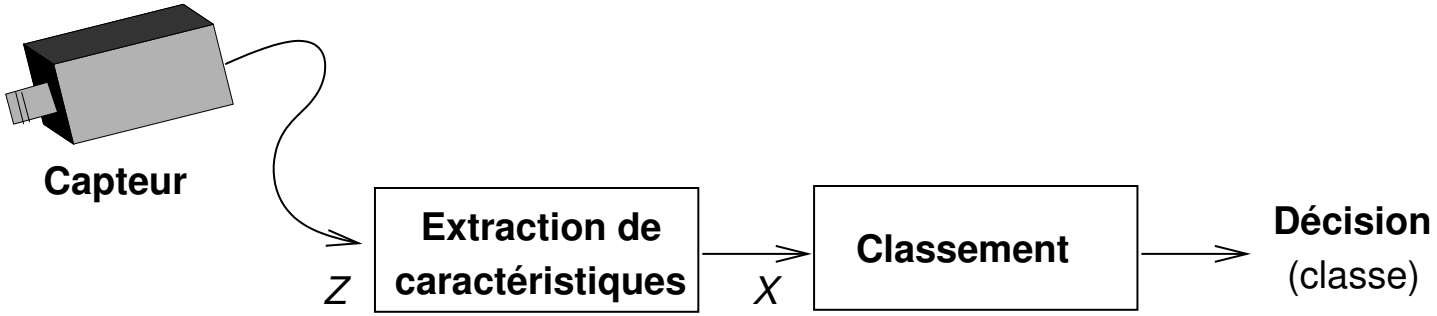
Source:
https://speakerdeck.com/rasbt/slides-from-machine-learning-with-scikit-learn-at-scipy-2016

Apprentissage vs Programmation

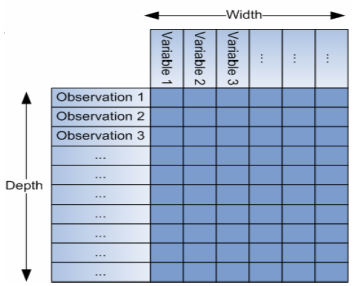
L'*apprentissage artificiel* (Machine Learning) est l'art de construire des systèmes capables d'apprendre à partir de données :

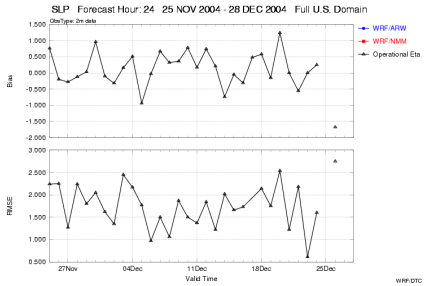


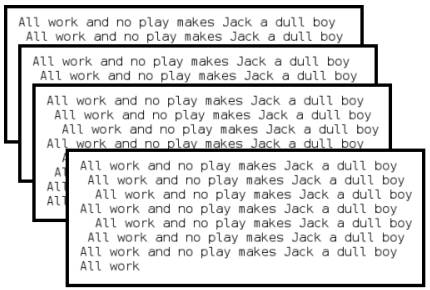
Apprentissage et reconnaissance des formes



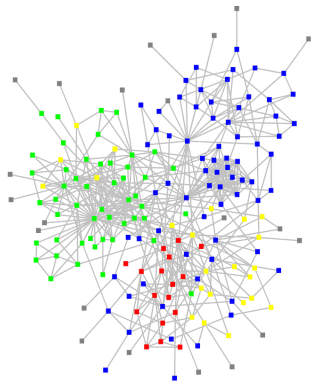
Données :



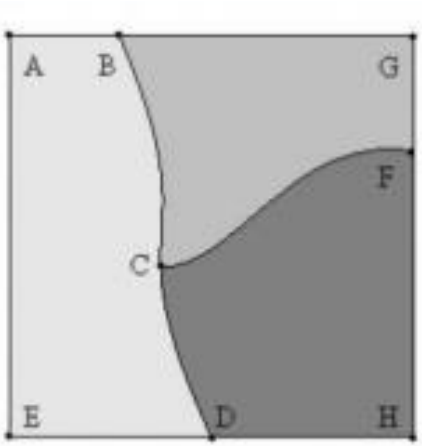




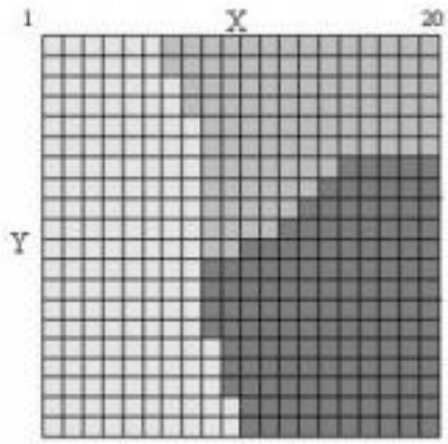

```
<texte id="exemple">
 <titre>Un document</titre>
 <partie>
   <par>
     Ceci est la première partie
   </par>
 </partie>
 <partie>
 ...
```



Représentation d'une image



Vector image



Raster image

How the pixels look:

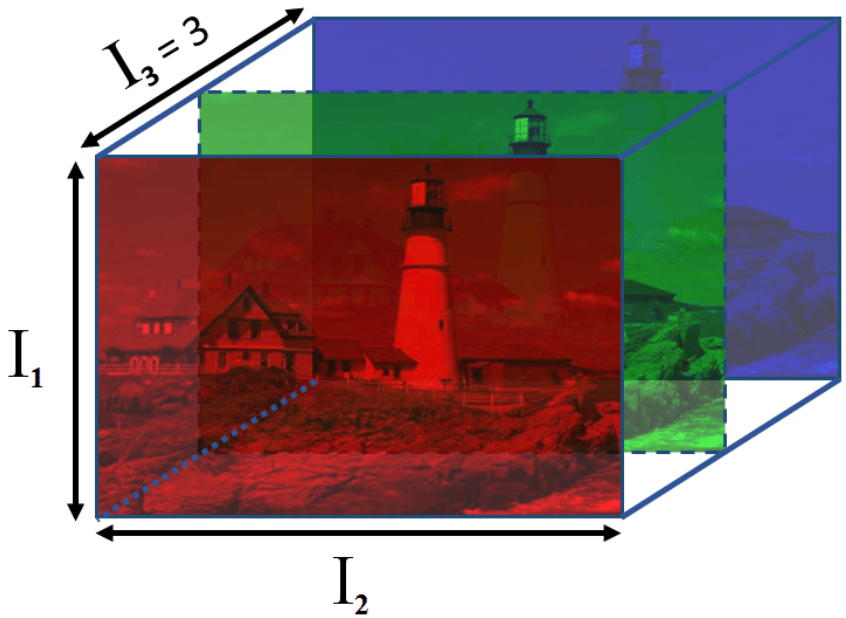
0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

How the pixels are stored:

0	1	2	3	4	5	6	7	8	9	·	·	·		

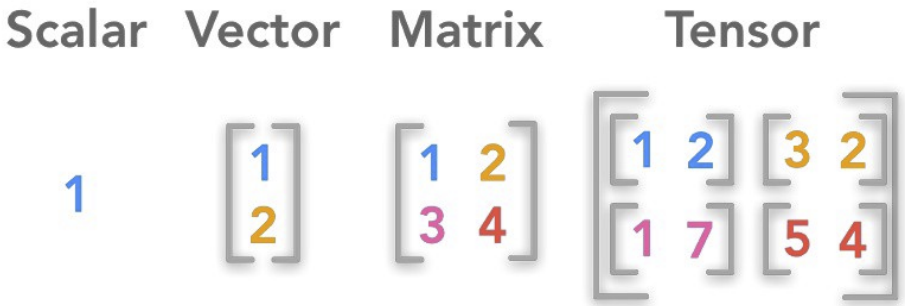
Niveaux de gris : $\{0, 1, \cdots, 255\}$ (sur 8 bits).
Parfois sur 10, 16 ou 32 bits. Souvent normalisés dans $[0, 1]$.

Représentation d'une image couleur RGB



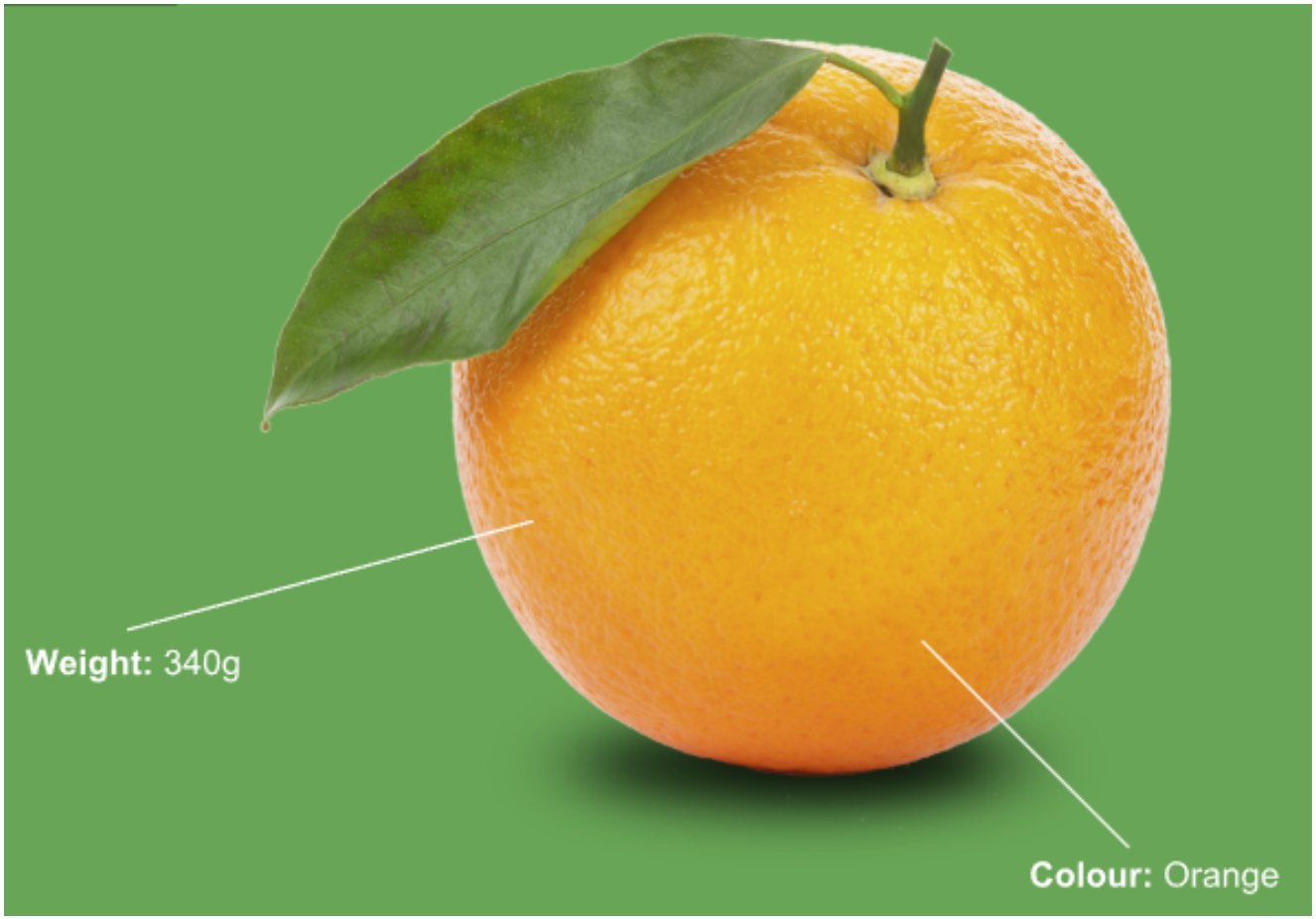
Chaque pixel est un triplet (r, g, b).

Une image de largeur W et de hauteur H pixels est représentée par un *tenseur* (matrice 3d), de dimension $(H, W, 3)$.



Attributs ou caractéristiques (*features*)

Les attributs sont les variables utilisées pour décrire les objets que l'on veut traiter

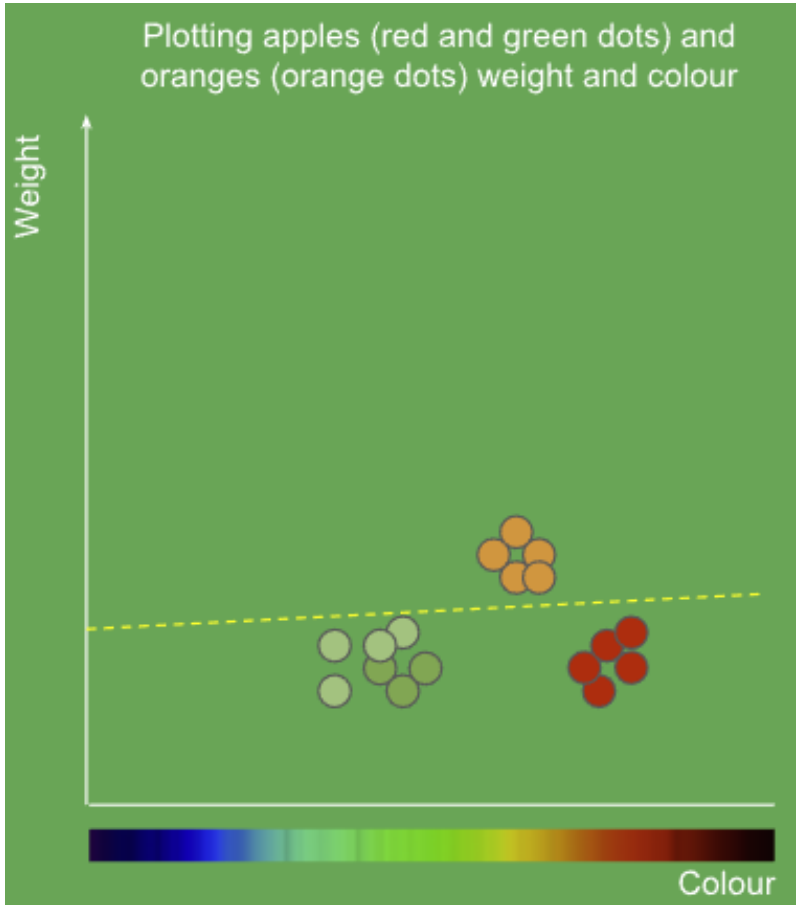


Attributs ou caractéristiques (*features*)

Exemple : distinguer des pommes et des oranges, à partir de leur masse et de leur couleur. Il y a des pommes rouges et des pommes vertes.

On peut calculer (apprendre) un modèle qui sépare ces fruits à partir de ces attributs.

Ce modèle pourra prévoir la nature d'un nouveau fruit.



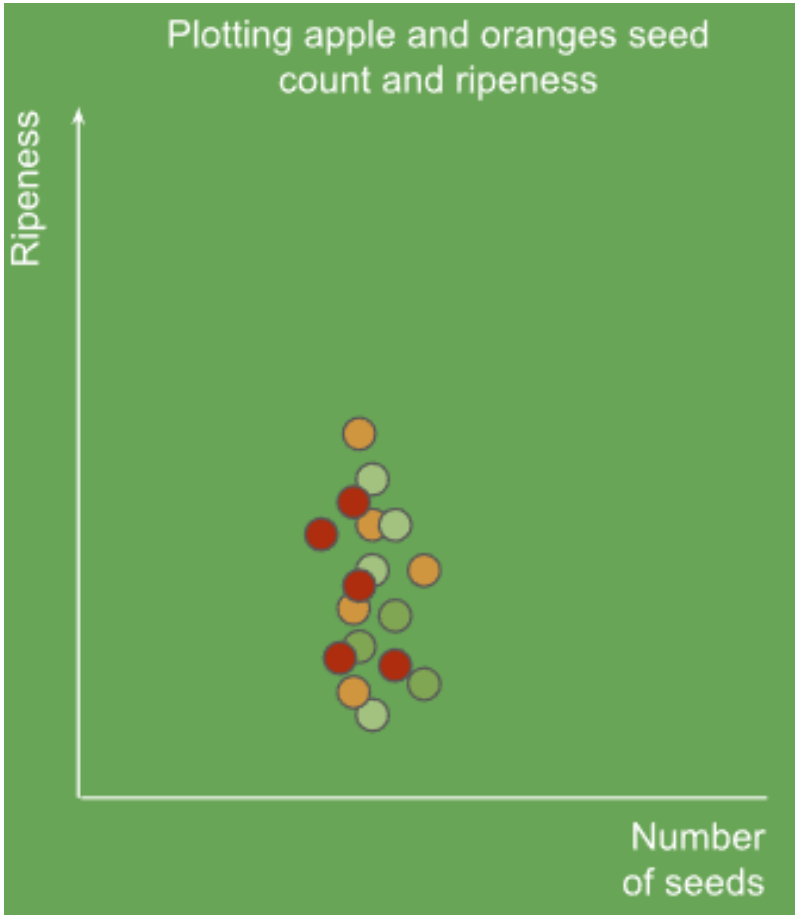
Source: Jason's Machine Learning 101, `https://docs.google.com`

Choix des attributs

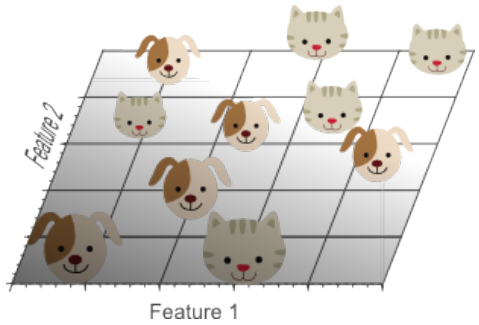
Le choix des caractéristiques est très important. Certaines n'apportent pas d'informations utiles pour le problème.

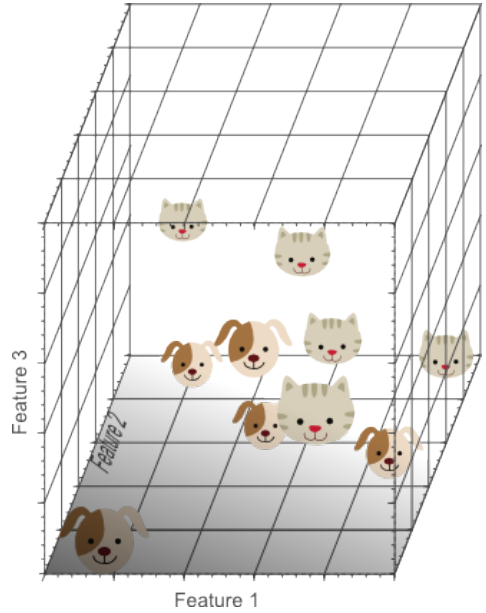
Par exemple, si on utilise le nombre de pépins et un indice de maturité des fruits, on ne peut pas séparer les pommes des oranges.

Le choix des variables est un sujet très important en apprentissage et data mining.



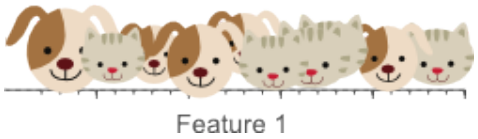
Dimensions

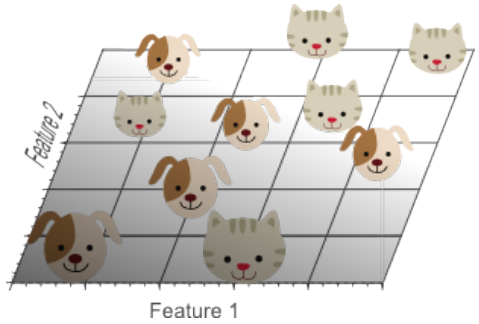


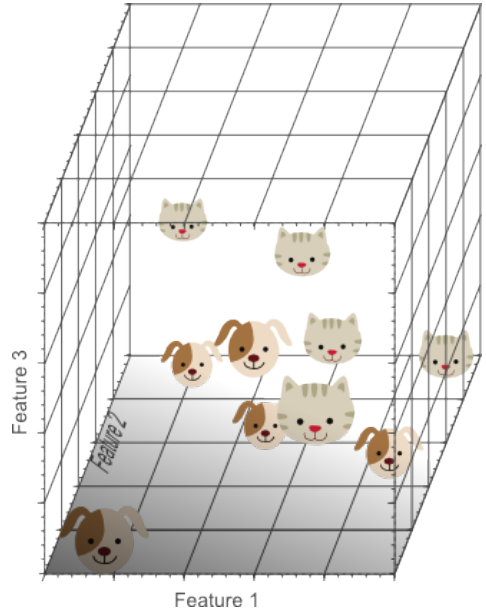


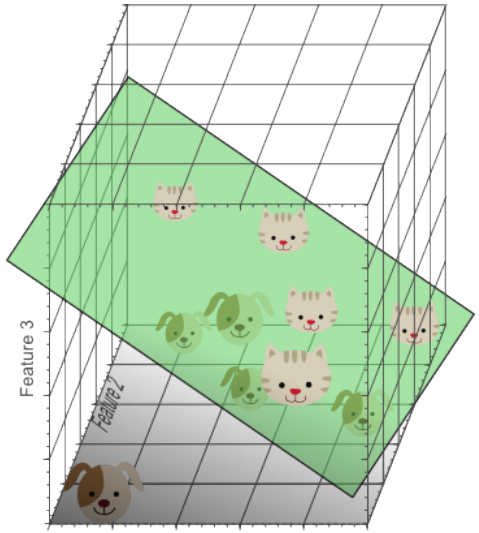
Source: http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

Dimensions









Source: http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

Dimensions

En pratique, on arrive à des modèles utilisants de nombreux attributs : des dizaines, jusqu'à des millions (signaux et images).

Mais l'augmentation du nombre d'attributs, donc de la dimension de l'espace à explorer, augmente la difficulté de l'apprentissage (on parle de « malédiction de la dimension »).

Plus on a de variables, plus on a de paramètres et donc plus il faut d'exemples pour apprendre. Les approches *deep learning* sont une bonne approche pour réduire la gravité de ce problème.

Données

- l'apprentissage demande des données
- si les données sont rares, méthodes « classiques » plus adaptées (extraction de caractéristiques à la main)
- Les résultats dépendent des données (attention aux biais)

Exemple

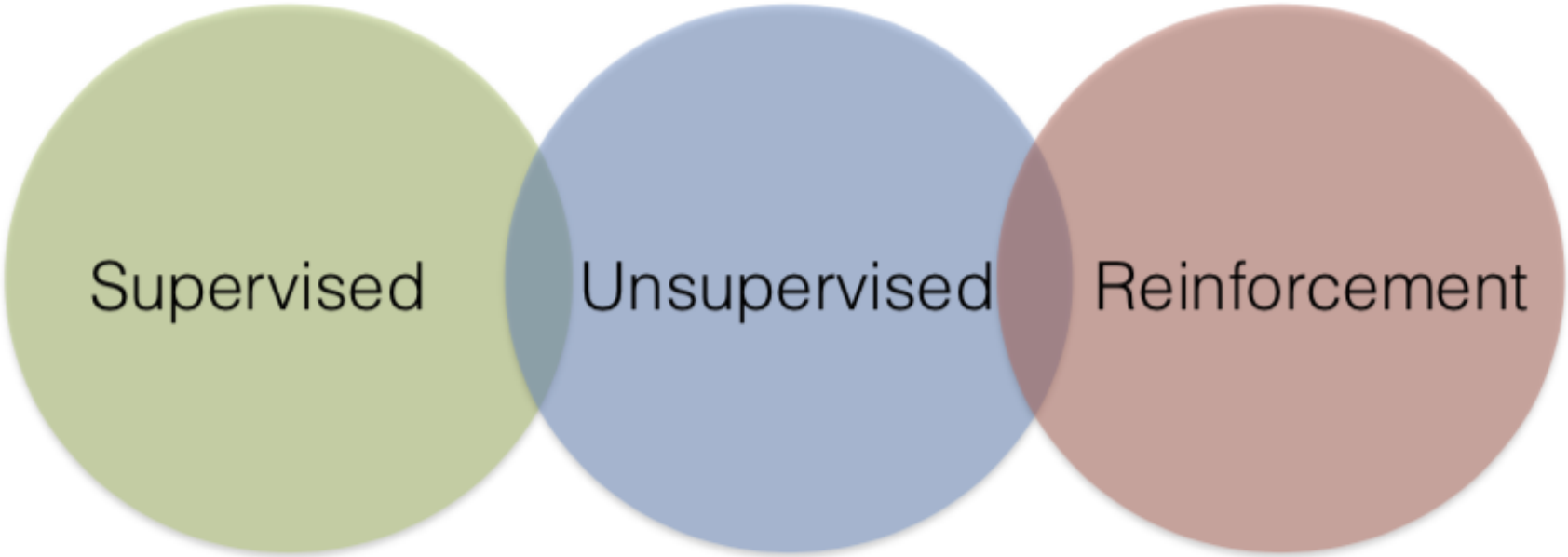
On apprend un modèle identifiant des animaux à partir des attributs :

Nb de pattes	Couleur	Poids	Animal
4	noir	10kg	**Chien**
2	orange	3kg	**Poule**
...	...	...	...

Si on lui présente une **vache** (4 pattes, noire, 200kg), elle sera reconnue comme un **chien**.

Plan

Trois types de problèmes d'apprentissage



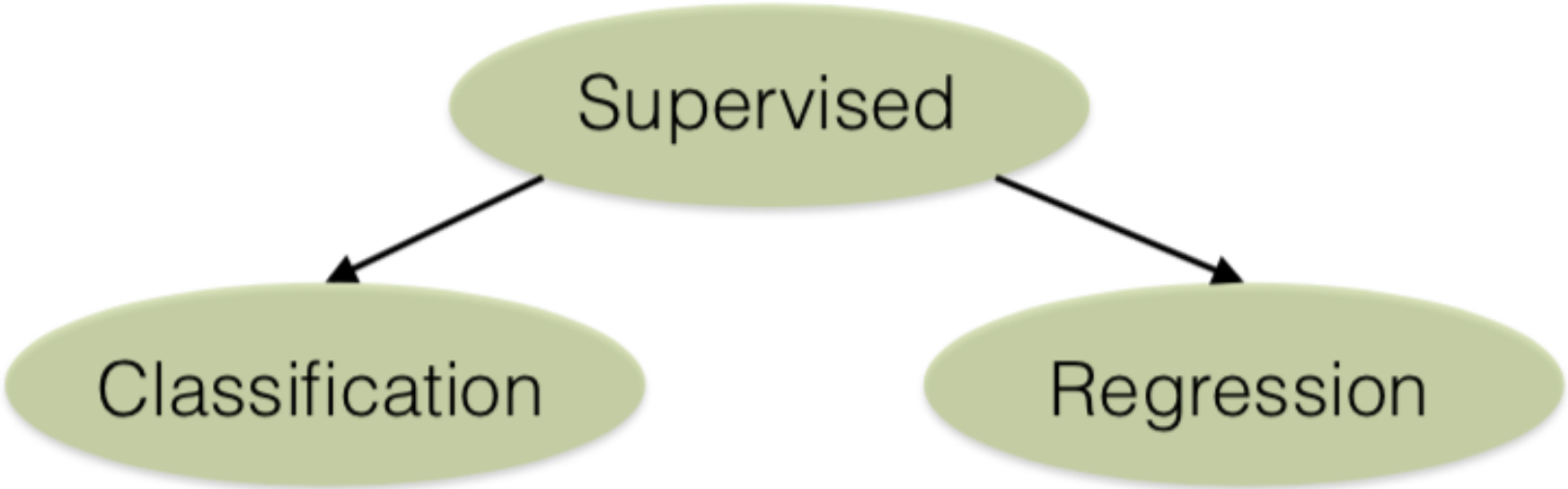
- Learning from labeled data
- E.g., Spam classification

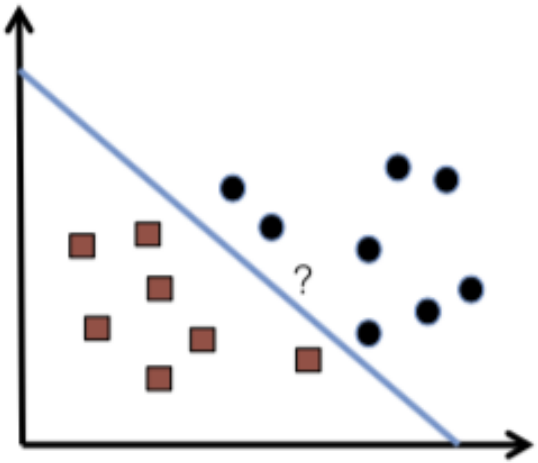
- Discover structure in unlabeled data
- E.g., Document clustering

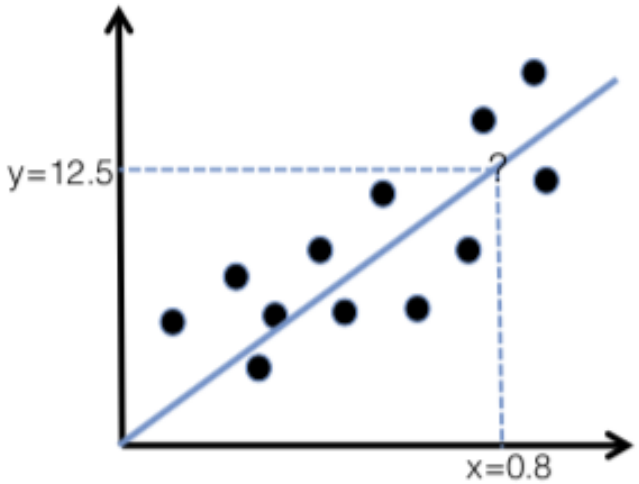
- Learning by "doing" with delayed reward
- E.g., Chess computer

Source:
https://speakerdeck.com/rasbt/slides-from-machine-learning-with-scikit-learn-at-scipy-2016

Apprentissage supervisé







Source:
https://speakerdeck.com/rasbt/slides-from-machine-learning-with-scikit-learn-at-scipy-2016

Apprentissage supervisé

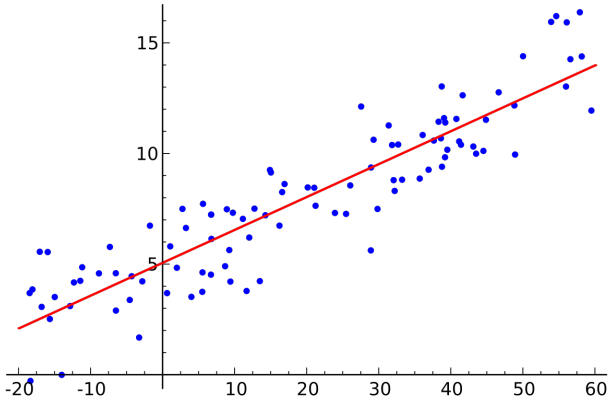
On a des *exemples*, et chacun a une étiquette (valeur cible).
Classement :

Nb de pattes	Couleur	Poids	**Animal**
4	noir	10kg	**Chien**
2	orange	3kg	**Poule**
...	...	...	...

Prévision de série (régression) :

Température	Férié	Nb Clients	Consommation
12	0	134	**1456**
14	0	124	**1234**
12	0	154	**1512**
...	...	...	...

Apprentissage supervisé : régression linéaire



- Données bivariées : $(x_0, y_0), (x_1, y_1), \ldots, (x_n, yn)$
- Modèle : $\hat{y} = f(x) + \epsilon$,
  où $f(x) = w.x + b$ et ϵ est un bruit
- Critère de performance : erreur quadratique $E = \sum_{i=0}^{n-1} (y_i - f(x_i))^2$

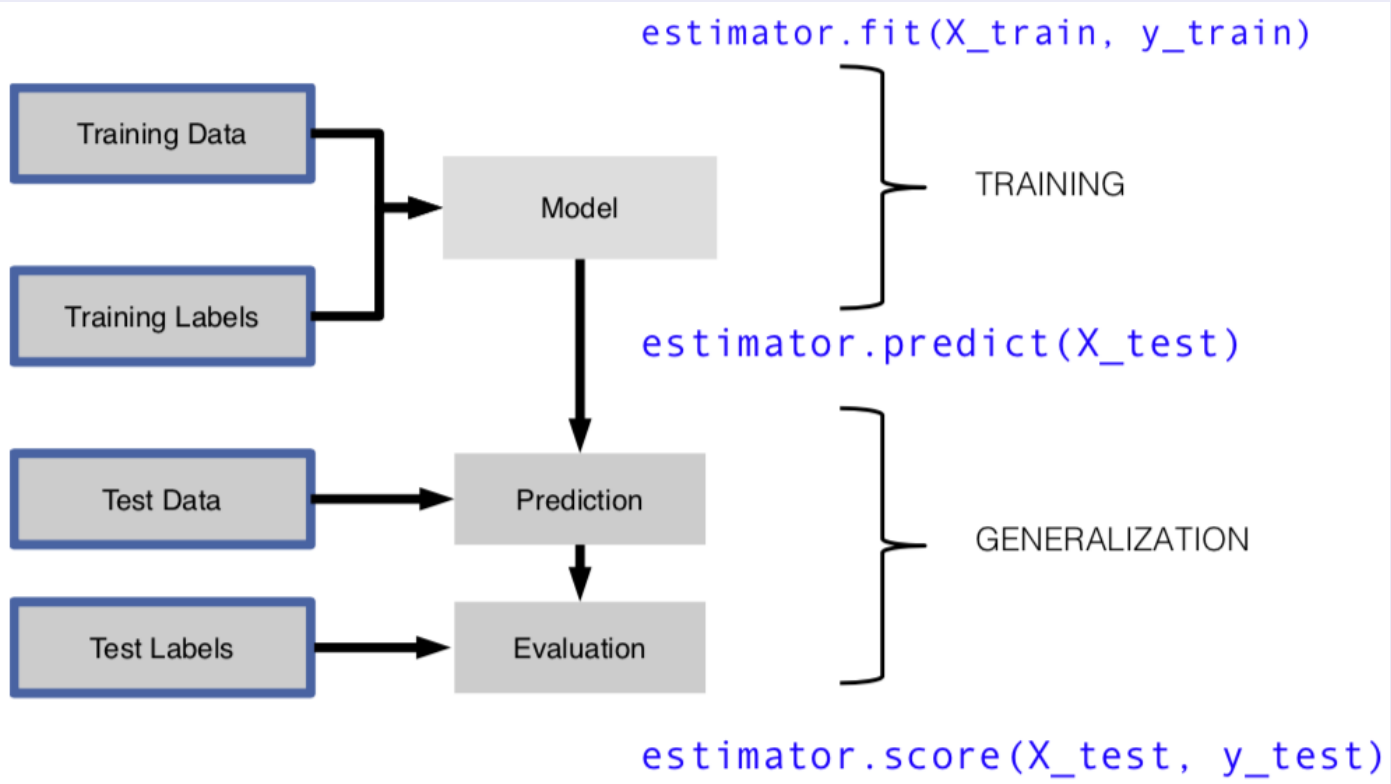
Apprentissage

Trouver w et b qui minimisent l'erreur E sur les données d'apprentissage.

Source: https://fr.wikipedia.org/wiki/Ajustement_affine

Apprentissage supervisé : régression linéaire

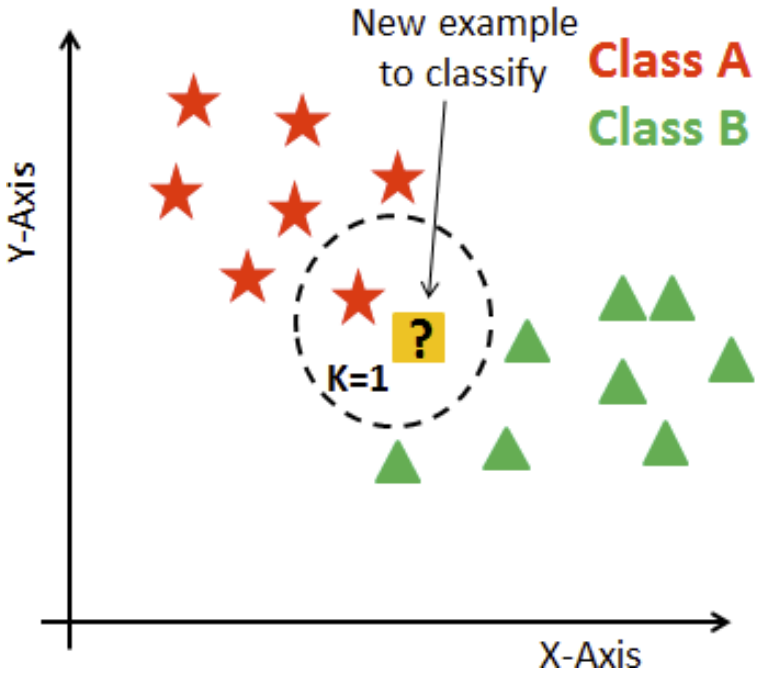
Workflow scikit-learn en apprentissage supervisé

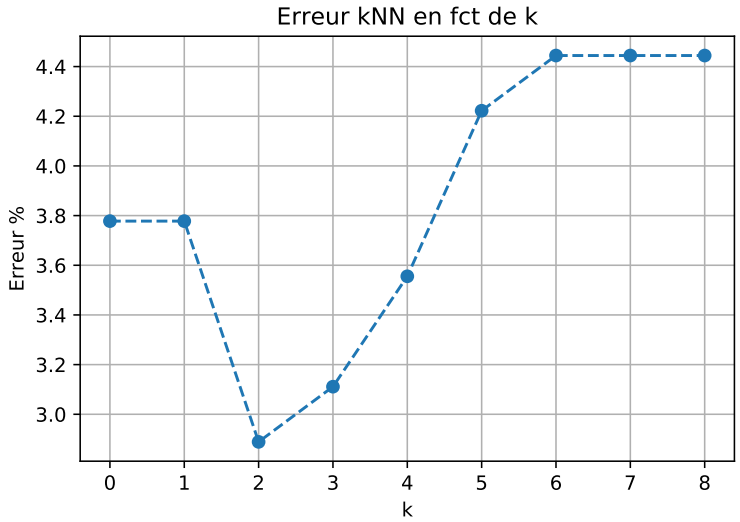


Notebook Jupyter

`01-RegressionLineaire.ipynb`

Apprentissage supervisé : plus proches voisins (*k*NN)





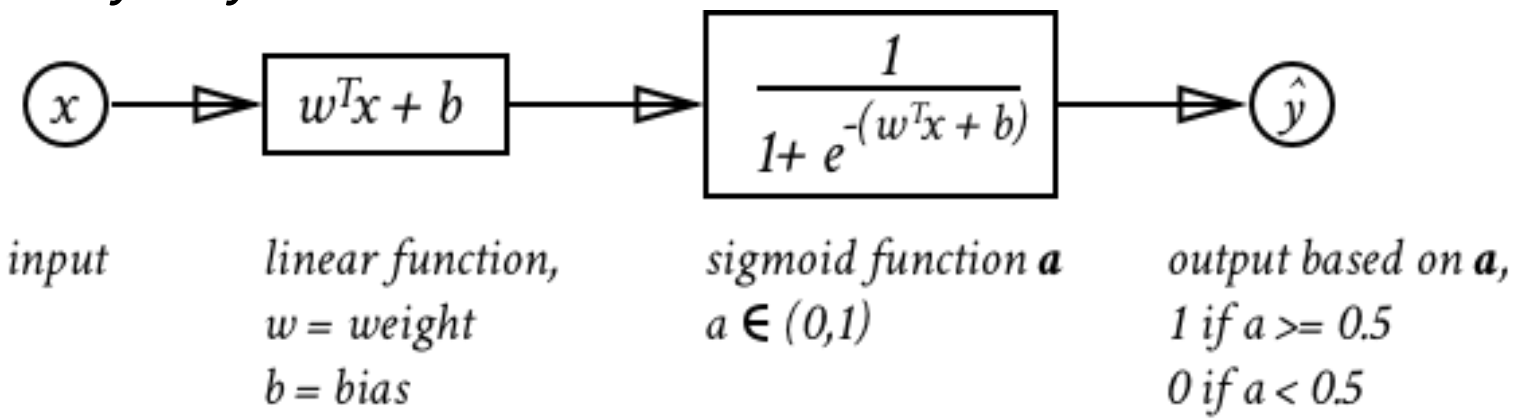
Notebook Jupyter

`02-kNN.ipynb`

Source:
https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

Apprentissage supervisé : régression logistique

Semblable à la régression (multivariée), utile pour les prévisions discrète (classes). On estime la probabilité que l'entrée x appartienne à la classe $\hat{y} = y$.

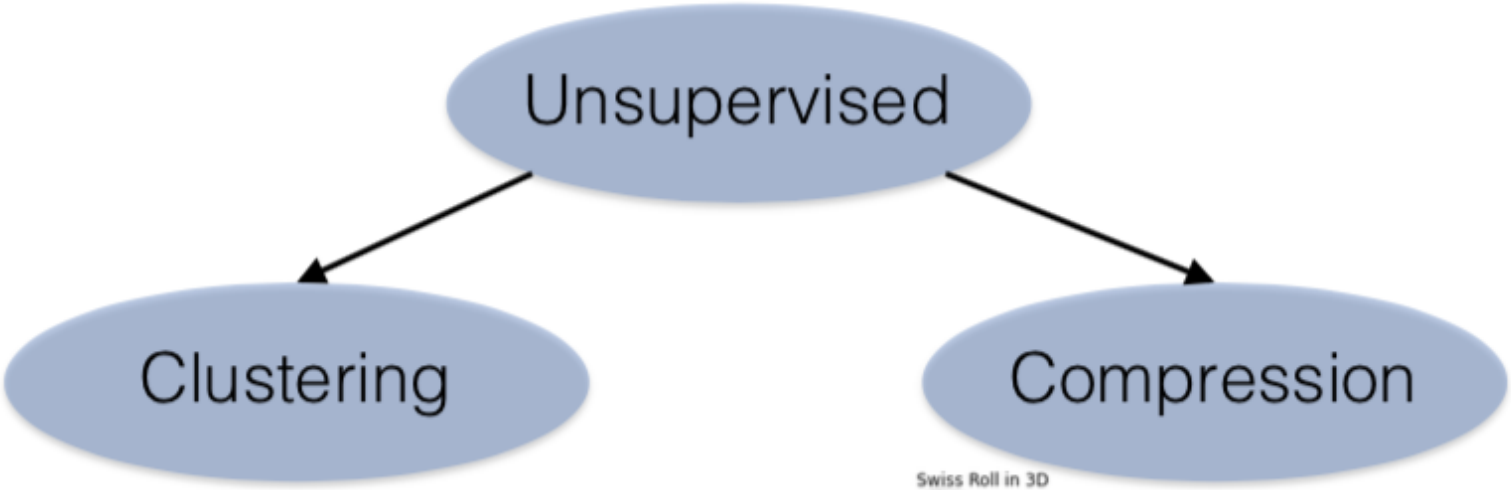


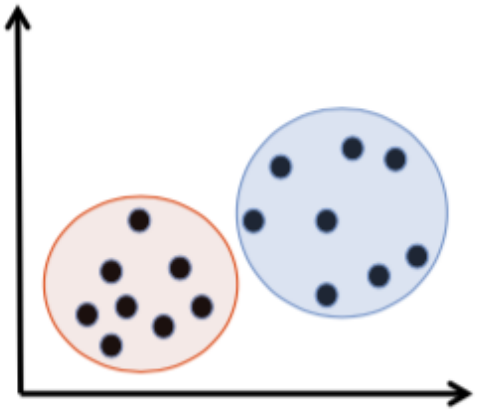
Autres modèles pour la classification supervisée

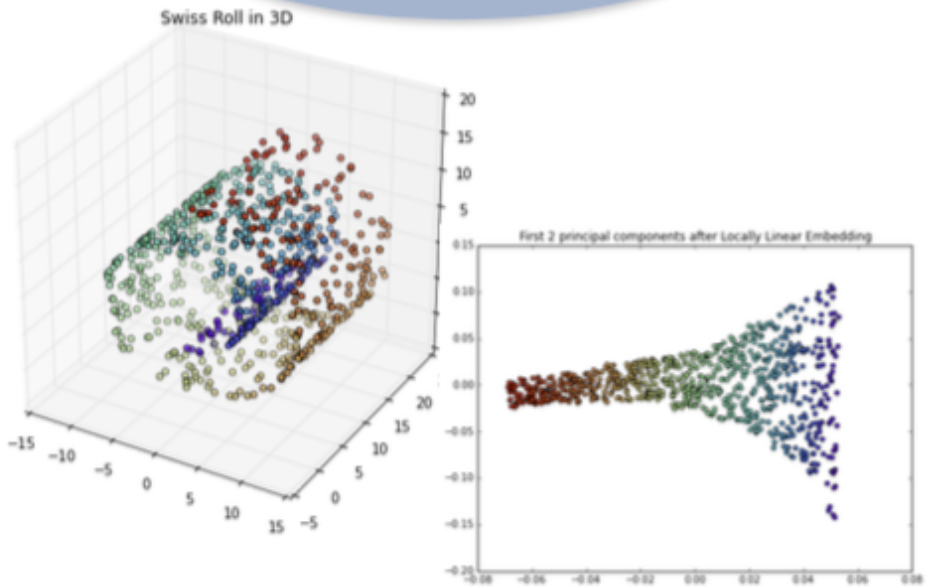
Il existe de nombreux modèles (ou algorithmes) pour la classification ou la régression supervisée. Citons :

- Classifieur de Bayes naïf
- Arbres de décision
- Séparateurs à vaste marge (*Support vector Machines*, SVM)

Apprentissage non supervisé





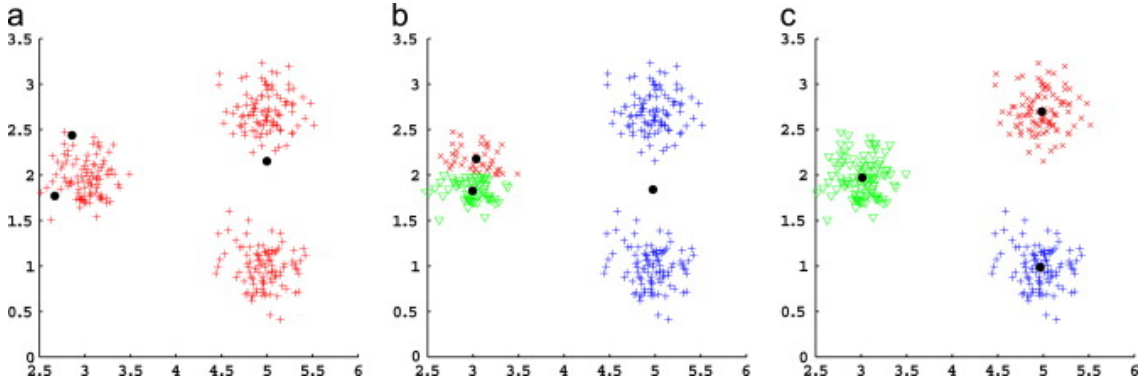


Source:
https://speakerdeck.com/rasbt/slides-from-machine-learning-with-scikit-learn-at-scipy-2016

Apprentissage non-supervisé : clustering (partitionnement)

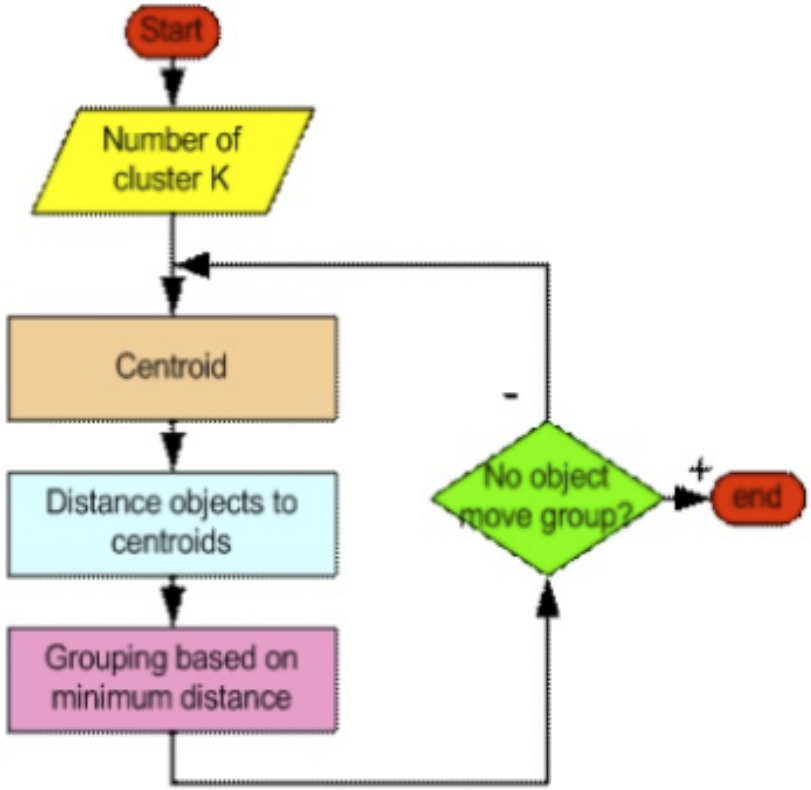
- grouper les points en paquets similaires
  - segmentation (exemple : groupes de clients semblables)
  - exploration des données
  - compression
- il faut une mesure de similarité
- pas de critère universel de performance : dépend de l'application

Apprentissage non-supervisé : *k*-moyennes (k-means)



k-means est un algorithme itératif :

- le nombre de paquets *k* est fixé a priori
- on minimise l'erreur de quantification



Apprentissage non-supervisé : *k*-moyennes (k-means)

Notebook Jupyter

- En deux dimensions : `03-kmeans.ipynb`
- Pour quantifier les couleurs d'une image `04-kmeans-couleurs.ipynb`

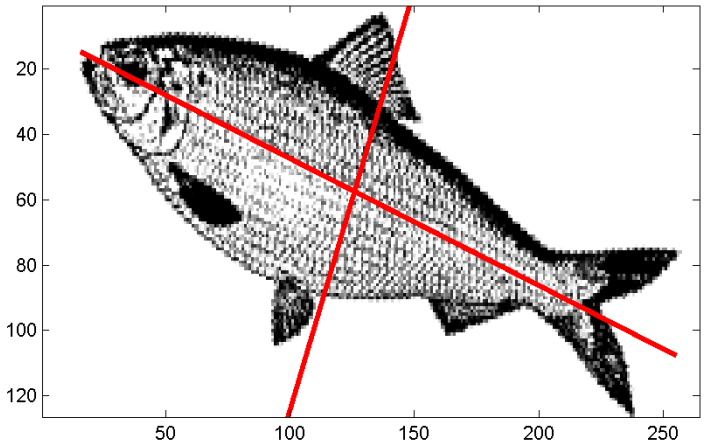
Original image (96615 colors)

Quantifiée sur 16 couleurs k-means

Quantifiée sur 16 couleurs aléatoires

Apprentissage non-supervisé : analyse en composante principale (ACP)

L'ACP construit le sous-espace linéaire qui permet de décorréler les variables.
Les axes, *composantes principales*, sont ordonnés selon leur importance pour expliquer les données.
Le calcul est une recherche des valeurs propres de la matrice de corrélation.



L'ACP est très utile pour

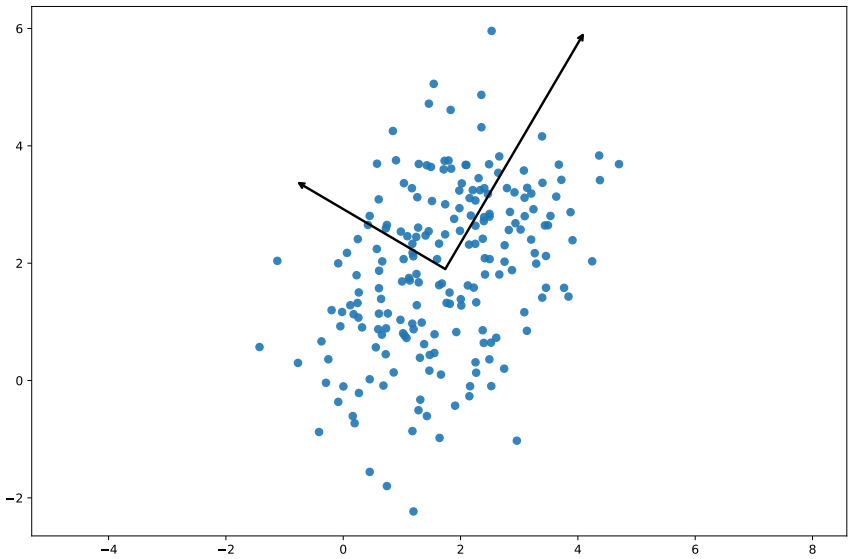
- réduire la dimension des données
- visualiser les données en 2 ou 3 dimensions
- comprimer des données ou images

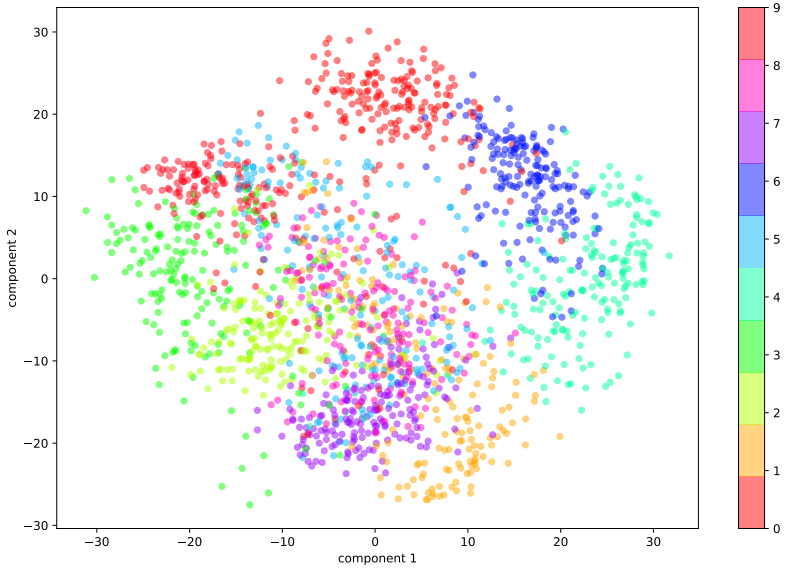
Source: Image Wikipedia

Apprentissage non-supervisé : analyse en composante principale (ACP)

Notebook Jupyter

- En deux dimensions : `05-PCA-2D.ipynb`
- Pour réduire la dimension d'images de chiffres `06-PCA-digits.ipynb`





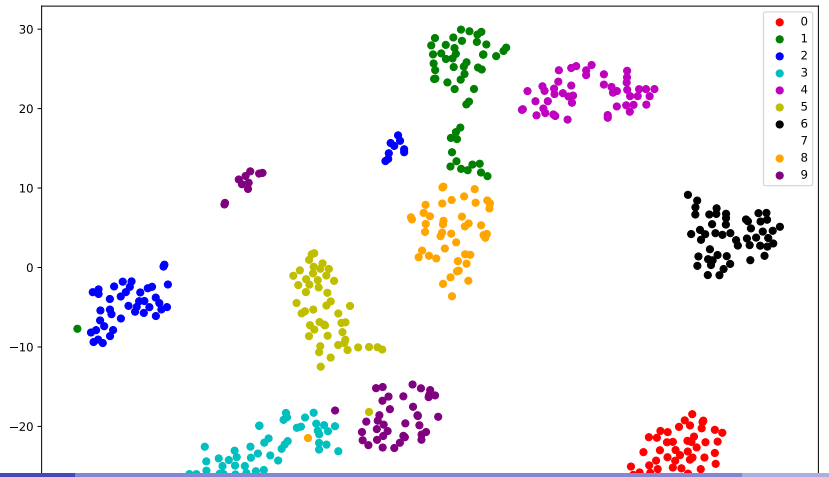
Apprentissage non-supervisé : t-SNE

t-Distributed Stochastic Neighbor Embedding
(van der Maaten et Hinton, 2008)

Méthode non-linéaire permettant de représenter un ensemble de points d'un espace à grande dimension dans un espace de deux ou trois dimensions. L'algorithme t-SNE tente de trouver une configuration optimale selon un critère de théorie de l'information pour respecter les proximités entre points : deux points qui sont proches dans l'espace d'origine devront être proches dans l'espace de faible dimension.

Notebook Jupyter

- Pour réduire la dimension d'images de chiffres
  `07-tSNE-digits.ipynb`

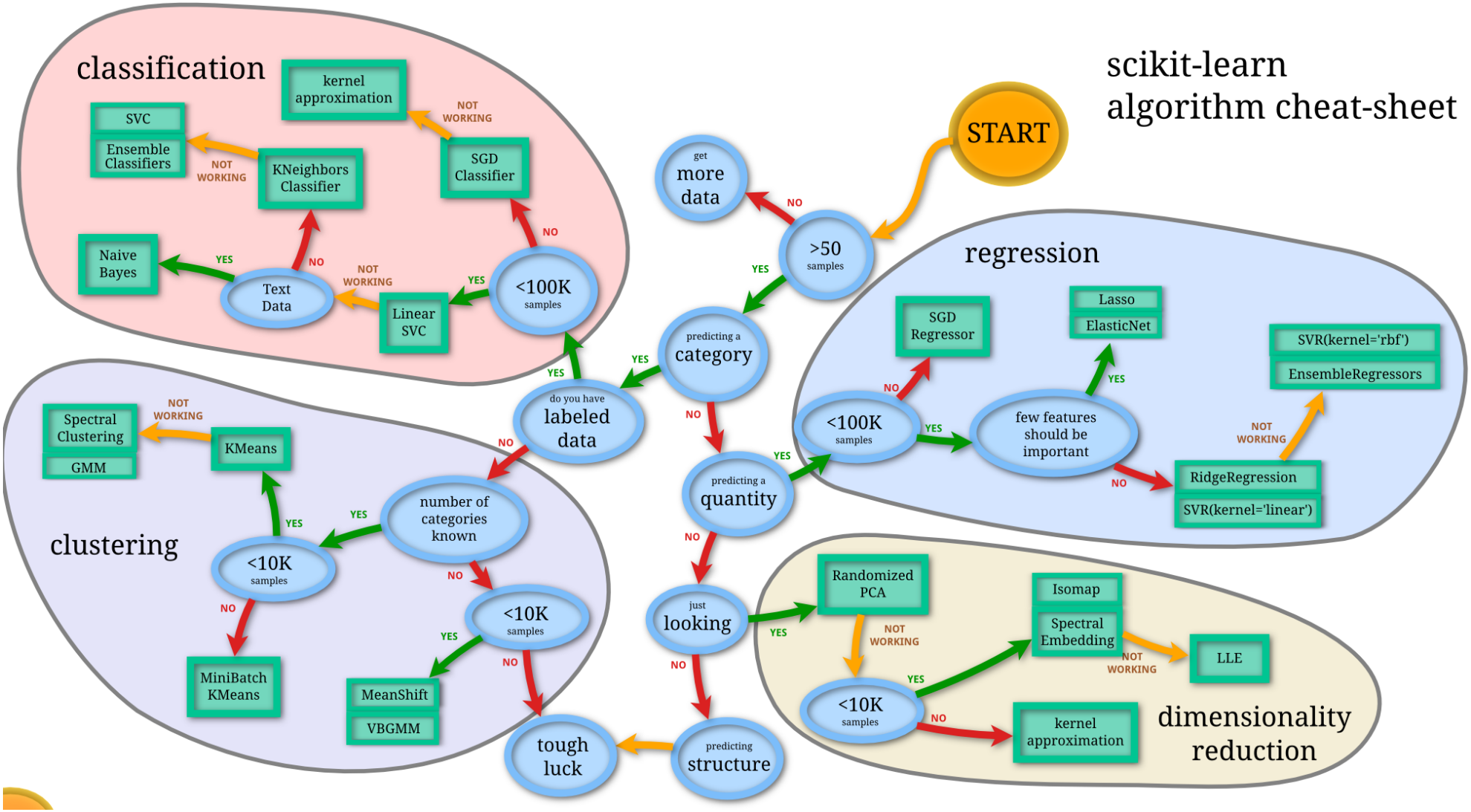


Apprentissage non-supervisé : UMAP

UMAP : très récent, serait plus précis et plus rapide que tSNE.

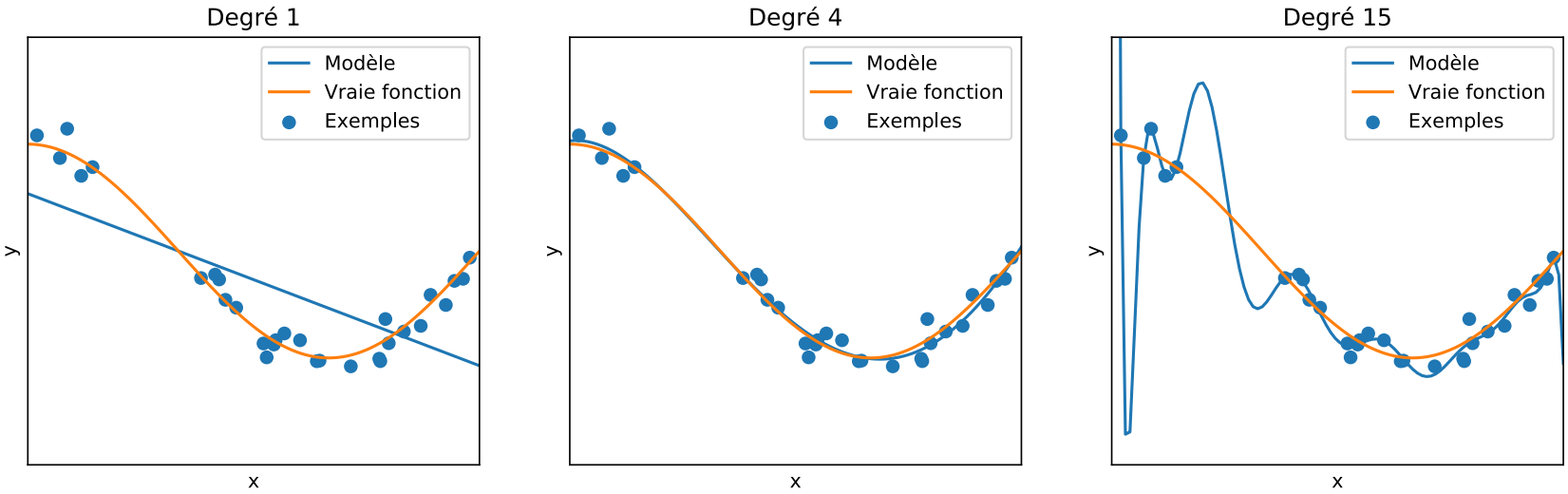
- Article : Leland McInnes, John Healy, James Melville « *UMAP : Uniform Manifold Approximation and Projection for Dimension Reduction* » (2018) `https://arxiv.org/abs/1802.03426`
- Exemple visualisation chiffres en 2D `https://www.kaggle.com/mrisdal/dimensionality-reduction-with-umap-on-mnist`

Guide pour choisir un modèle

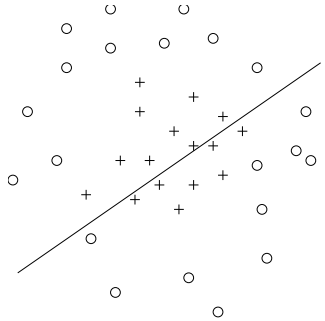


Source: https://scikit-learn.org/stable/tutorial/machine_learning_map

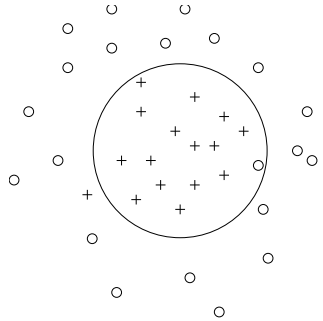
Compromis Apprentissage/Généralisation



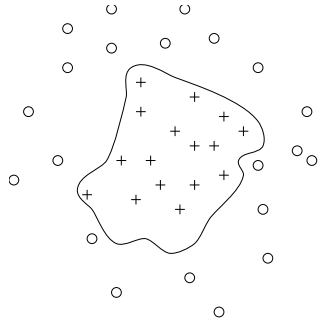
Compromis Apprentissage/Généralisation



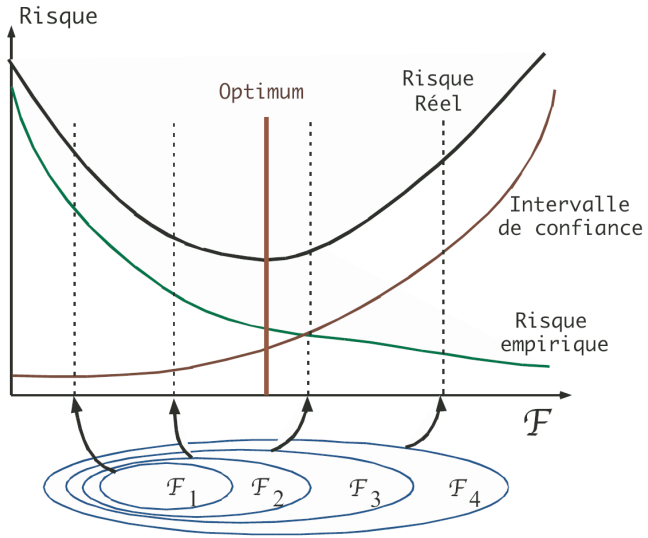
(a) Modèle trop contraint



(b) Optimal



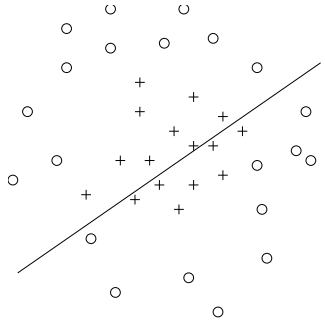
(c) Modèle trop libre



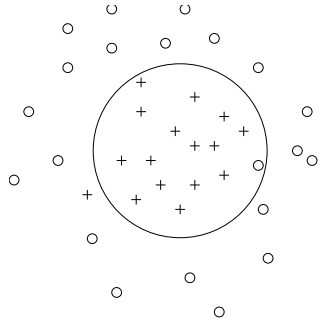
$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \sqrt{\frac{1}{l}\left(h(\log(2\frac{l}{h}) + 1) - \log(\eta/4)\right)}$$

= Minimisation Structurelle du Risque (SRM, Vapnik)

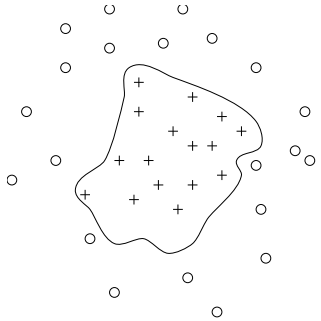
Compromis Apprentissage/Généralisation



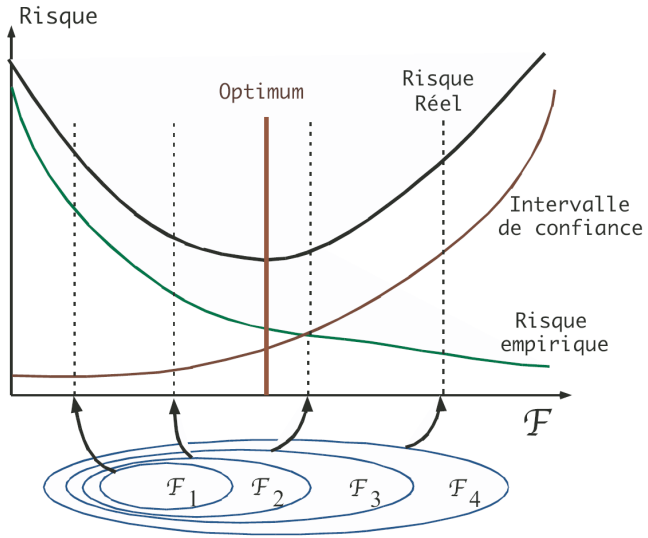
(a) Modèle trop contraint



(b) Optimal



(c) Modèle trop libre



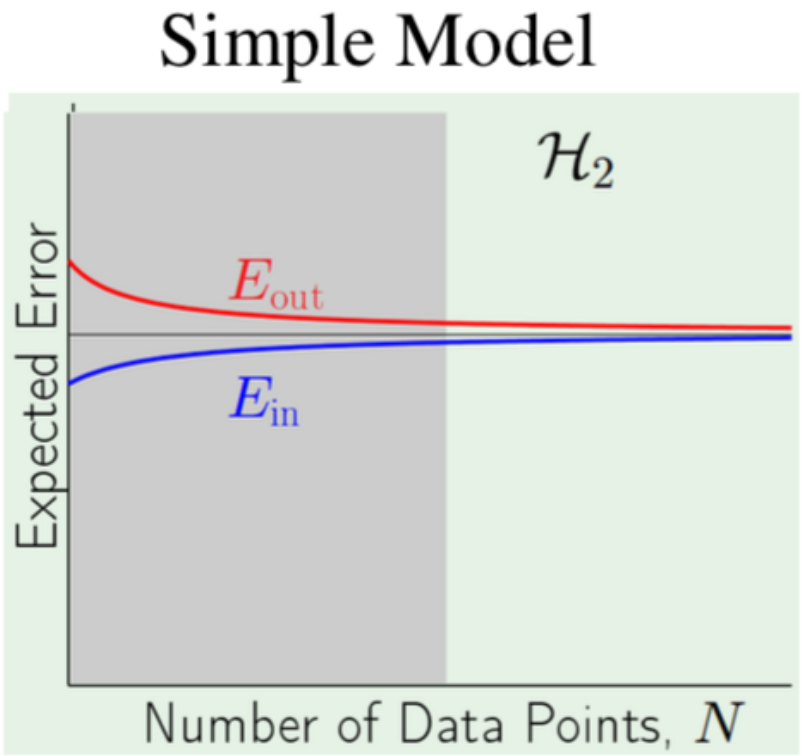
$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \sqrt{\frac{1}{l}\left(h(\log(2\frac{l}{h}) + 1) - \log(\eta/4)\right)}$$

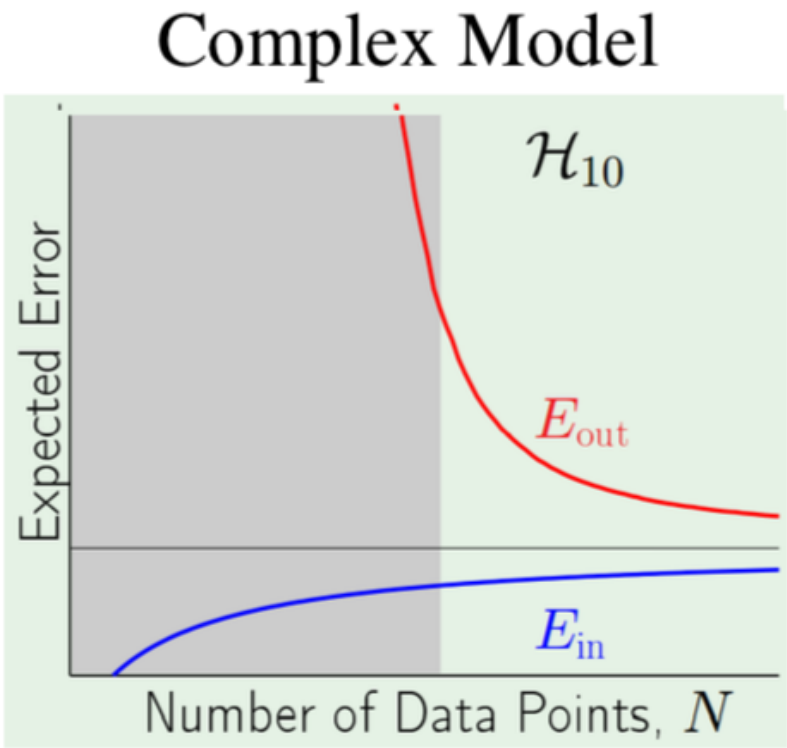
Méthodes réseaux connexionnistes :

- choix architecture
- régularisation, early stopping

= Minimisation Structurelle du Risque (SRM, Vapnik)

Complexité d'un modèle vs volume de données

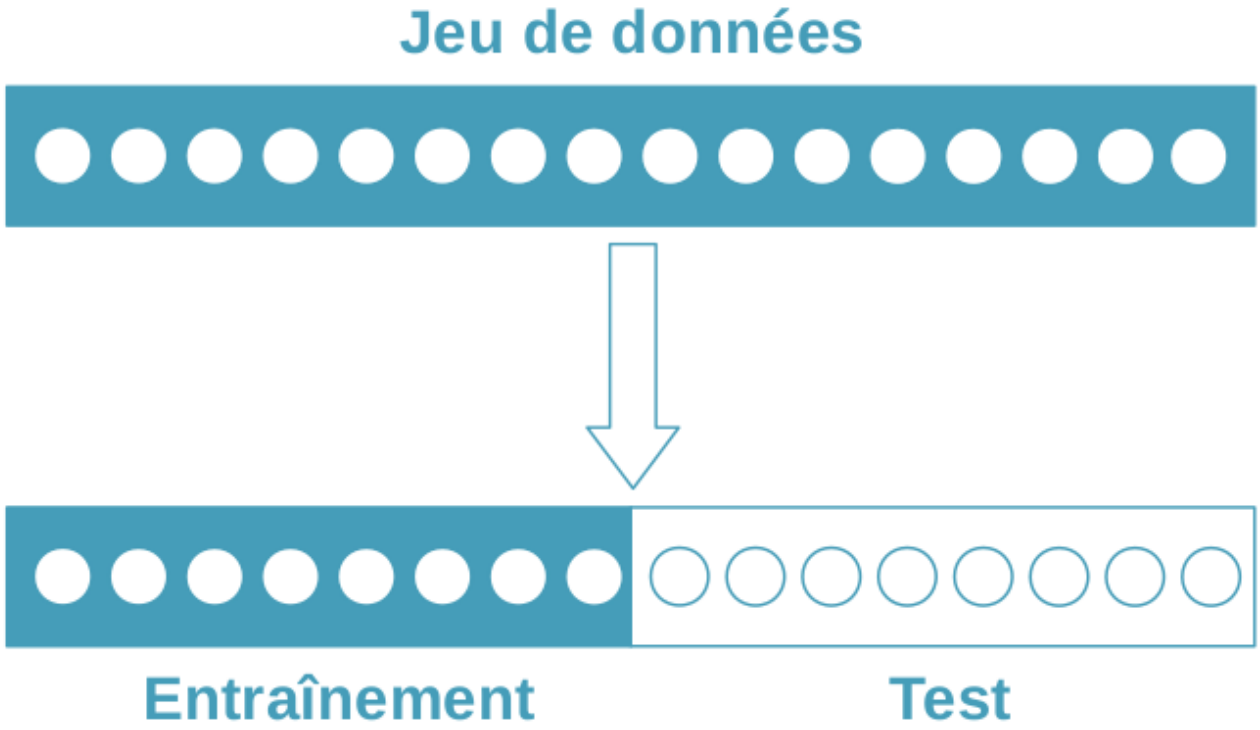




Overfitting only happens for small training sets

Comment choisir un modèle ?

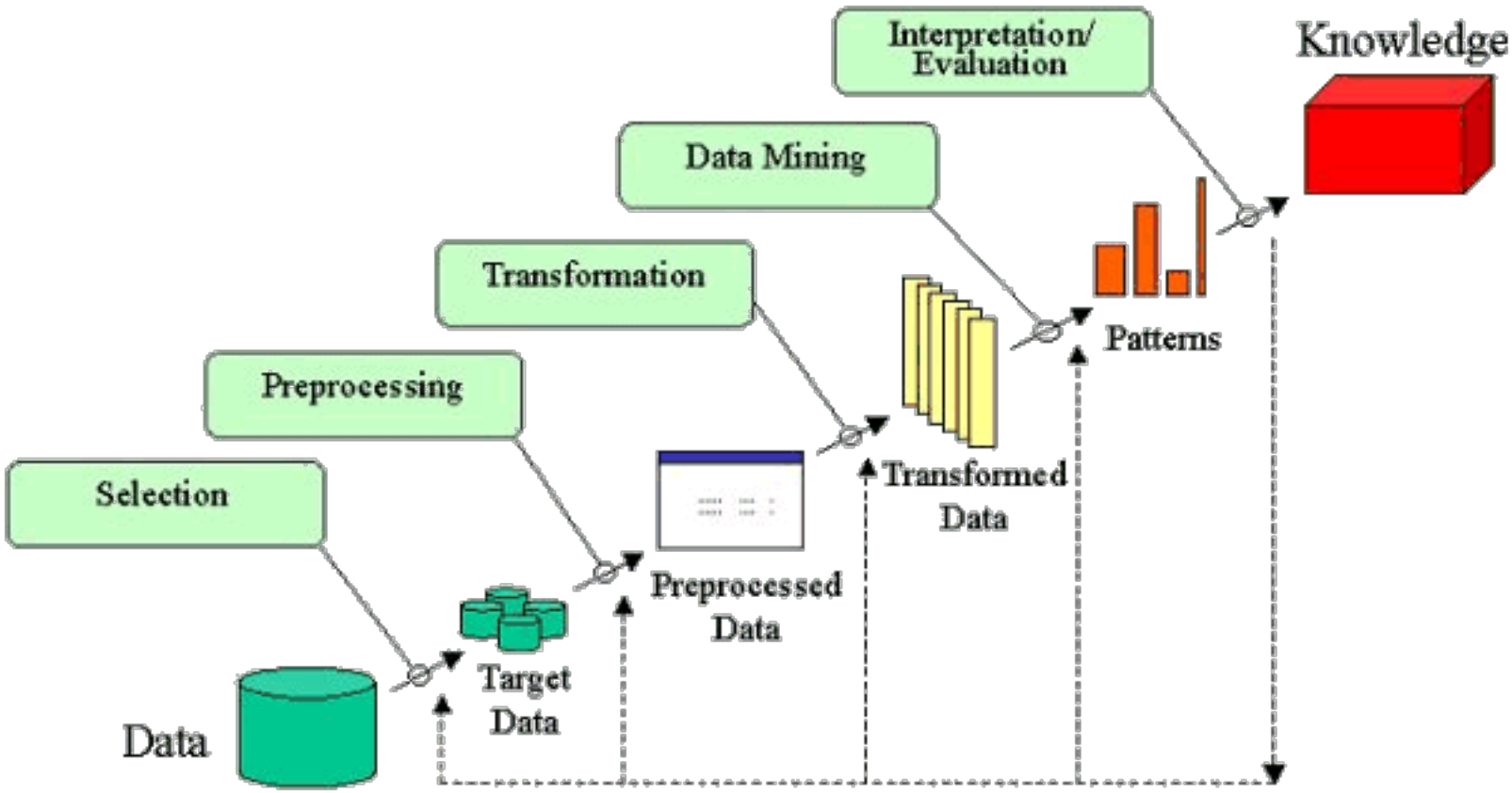
validation croisée



En pratique, on a souvent 3 ensembles : apprentissage, validation, test.

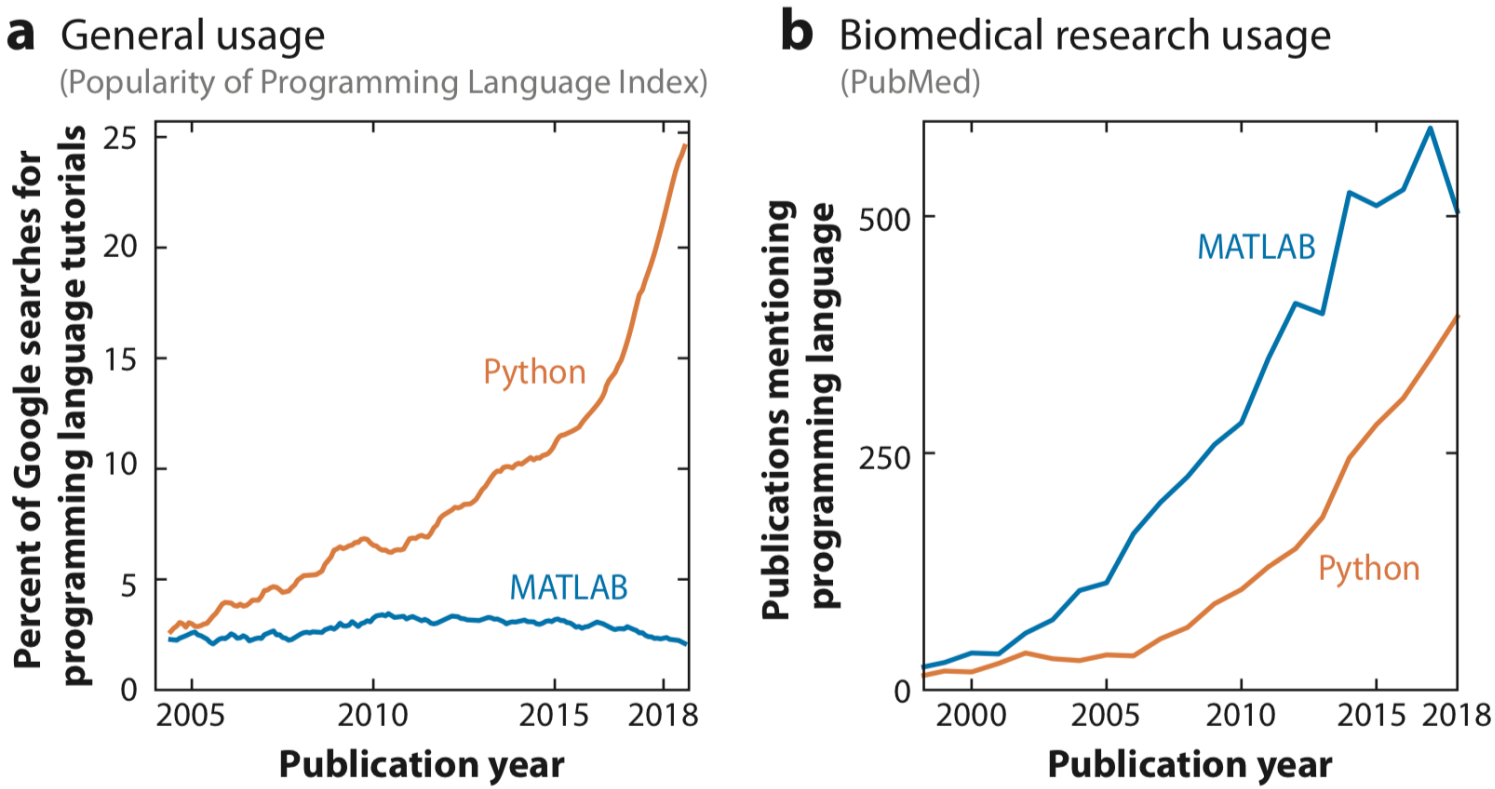
Plan

L'extraction de connaissances à partir de données (KDD)



CRISP : Cross-industry standard process for data mining

Au fait, pourquoi présenter Python et ses outils ?



Plodrack et al., Computational and Informatic Advances for Reproducible Data Analysis in Neuroimaging, *Annual Review of Biomedical Data Science*, March 2019

1- Outils (python) pour la préparation des données

- 10 Minutes to pandas `https://pandas.pydata.org/pandas-docs/stable/getting_started/10min.html`
- Matplotlib Beginner's Guide `https://matplotlib.org/users/beginner.html`
- Official seaborn tutorial `https://seaborn.pydata.org/tutorial.html`

Pour aller plus loin

- Intro to pandas data structures, by Greg Reda `http://www.gregreda.com/2013/10/26/intro-to-pandas-data-structures`
- Modern Pandas (in 7 parts), by Tom Augspurger `http://tomaugspurger.github.io/modern-1-intro.html`

2- Analyse exploratoire des données (EDA)

Comprendre les données : statistiques, visualisations
$\rightsquigarrow$ tendances, qualité, hypothèses.
Très important avant d'appliquer une modélisation.

Références

- 7 Steps to Mastering Data Preparation for Machine Learning with Python - 2019 Edition
  `https://www.kdnuggets.com/2019/06/7-steps-mastering-data-preparation-python.html`
- Prof. Patrick Meyer of the University of Virginia which provides an overview of EDA : `https://youtu.be/zHcQPKP6NpM`
- Exploratory data analysis (EDA)
  `https://datascienceguide.github.io/exploratory-data-analysis`
- EDA and Data Visualization with Python `https://kite.com/blog/python/data-analysis-visualization-python`

Panda Profiling, un outil pour l'EDA

Dataset info

Number of variables	16
Number of observations	45726
Missing cells	117837 (16.1%)
Duplicate rows	0 (0.0%)
Total size in memory	5.3 MiB
Average record size in memory	121.0 B

Variables types

Numeric	6
Categorical	5
Boolean	1
Date	1
URL	0
Text (Unique)	1
Rejected	2
Unsupported	0

Warnings

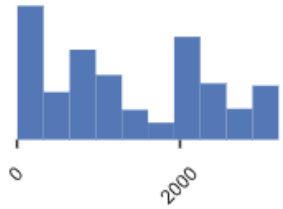
`Counties` has 44067 (> 99.9%) missing values	Missing
`GeoLocation` has a high cardinality: 17101 distinct values	Warning
`GeoLocation` has 7315 (16.0%) missing values	Missing
`mass_(g)` is highly skewed (γ1 = 76.918)	Skewed
`recclass` has a high cardinality: 466 distinct values	Warning
`reclat` has 6438 (14.1%) zeros	Zeros
`reclat` has 7315 (16.0%) missing values	Missing
`reclat_city` is highly correlated with `reclat` (ρ = 0.99424)	Rejected
`reclong` has 6214 (13.6%) zeros	Zeros
`reclong` has 7315 (16.0%) missing values	Missing
`source` has constant value "NASA"	Rejected
`States` has 44067 (> 99.9%) missing values	Missing

https://github.com/pandas-profiling/pandas-profiling

Panda Profiling, un outil pour l'EDA (suite)

Counties
Numeric

Distinct count	663	Mean	1353.3
Unique (%)	< 0.1%	Minimum	5
Missing (%)	> 99.9%	Maximum	3210
Missing (n)	44067	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		



Toggle details

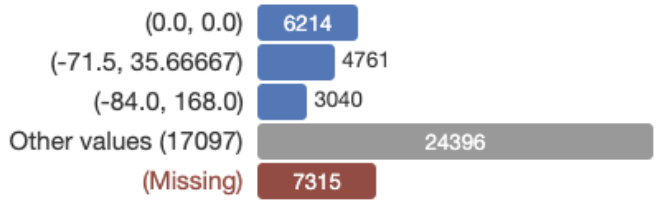
fall
Categorical

Distinct count	2
Unique (%)	< 0.1%
Missing (%)	0.0%
Missing (n)	0

Toggle details

GeoLocation
Categorical

Distinct count	17101
Unique (%)	37.4%
Missing (%)	16.0%
Missing (n)	7315



https://github.com/pandas-profiling/pandas-profiling

3- Valeurs manquantes

Méthodes usuelles simples :

- supprimer les exemples avec valeurs manquantes ;
- supprimer les attributs (colonnes) avec valeurs manquantes ;
- utiliser la moyenne, la médiane ou le mode pour toutes les valeurs manquantes ;
- utiliser une régression pour estimer chaque valeur manquante.

Le meilleur choix dépend aussi du modèle.

Références

- Panda : Working with missing data
  `http://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html`
- video from codebasics on handling missing values with Pandas
  `https://youtu.be/EaGbS7eWSs0`

4- *Outliers* (valeurs aberrantes)

Références

- http://www.theanalysisfactor.com/outliers-to-drop-or-not-to-drop
- Exemple simple : *Removing Outliers Using Standard Deviation with Python* https://www.kdnuggets.com/2017/02/removing-outliers-standard-deviation-python.html
- Discussion technique : *Remove Outliers in Pandas DataFrame using Percentiles.* https://stackoverflow.com/questions/35827863/remove-outliers-in-pandas-dataframe-using-percenti

5- Classes déséquilibrées

Arrive fréquemment dans les problèmes de détection, de diagnostic, etc. (évènements rares).

Références

- Learning from Imbalanced Classes `https://www.kdnuggets.com/2016/08/learning-from-imbalanced-classes.html`
- 7 Techniques to Handle Imbalanced Data by Ye Wu & Rick Radewagen, `https://www.kdnuggets.com/2017/06/7-techniques-handle-imbalanced-data.html`

6- Transformation des données

Échantillon x_i ou, en supervisé, (x_i, y_i).
La transformation $(x'_i, y'_i) = f(x_i, y_i)$ vise à améliorer les performances du modèle en :

- satisfaisant mieux les hypothèses (eg normalité) ;
- codant les variables pour rendre les données plus faciles à traiter.

Quelques références

- Preprocessing data `https://scikit-learn.org/stable/modules/preprocessing.html`
- Normalization vs Standardisation : quantitative analysis `https://towardsdatascience.com/normalization-vs-standardization-quantitative-anal`
- One-hot encoding : a method for transforming categorical features to a format which will better work for classification and regression `https://youtu.be/9yl6-HEY7_s`

Plan

Conclusion de la 1ère partie

Nous avons introduit :

- l'apprentissage à partir de données : concept et applications ;
- quelques outils (Python) pour les sciences des données et le machine learning ;
- les principaux modèles pour l'apprentissage supervisé et non-supervisé.
- la préparation des données avant leur modélisation.