

# Apprentissage et Fouille de Données Visuelles

E. Viennet

L2TI  
Université Paris 13

2019-2020

# Plan du cours

## 1 Introduction

## 2 Bases de l'apprentissage (machine learning)

- Apprentissage supervisé
- Apprentissage non-supervisé
- Apprentissage et généralisation
- Conclusion du cours 1

## 3 Introduction au Deep Learning

- Histoire du connexionnisme
- Perceptrons multi-couches
- Apprentissage par descente du gradient
- Réseaux convolutionnels pour les images
- Architectures profondes

# Plan

## 1 Introduction

## 2 Bases de l'apprentissage (machine learning)

- Apprentissage supervisé
- Apprentissage non-supervisé
- Apprentissage et généralisation
- Conclusion du cours 1

## 3 Introduction au Deep Learning

- Histoire du connexionnisme
- Perceptrons multi-couches
- Apprentissage par descente du gradient
- Réseaux convolutionnels pour les images
- Architectures profondes

# Intelligence Artificielle

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

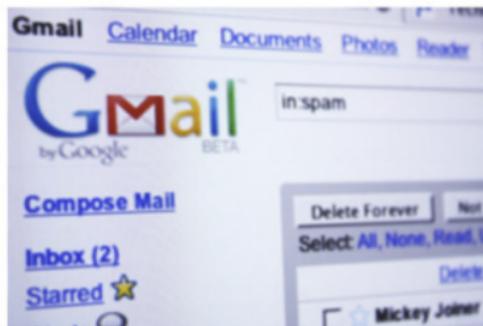
« intelligente » = indistinguishable de l'humain ? (test de Turing)

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

- reconnaître 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](http://commons.wikimedia.org/wiki/File:American_book_company_1916_letter_envelope-2.JPG#filelinks) [public domain]

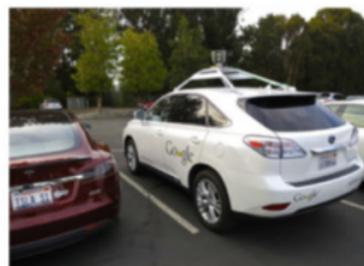


[http://commons.wikimedia.org/wiki/File:Netflix\\_logo.svg](http://commons.wikimedia.org/wiki/File:Netflix_logo.svg) [public domain]

And many, many more ...

Source:

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



By Steve Jurvetson [CC BY 2.0]

# Apprentissage vs Programmation

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

Write a computer program with **explicit rules** to follow

```
if email contains V!agrå
  then mark is-spam;
if email contains ...
if email contains ...
```

Traditional Programming

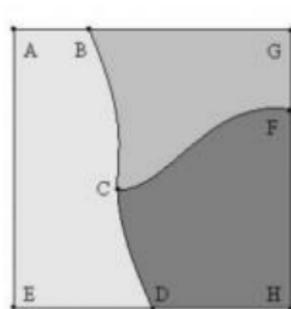
Write a computer program to **learn from examples**

```
try to classify some emails;
change self to reduce errors;
repeat;
```

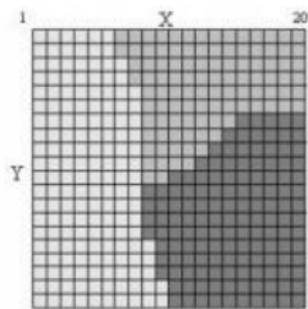
Machine Learning Programs



# 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

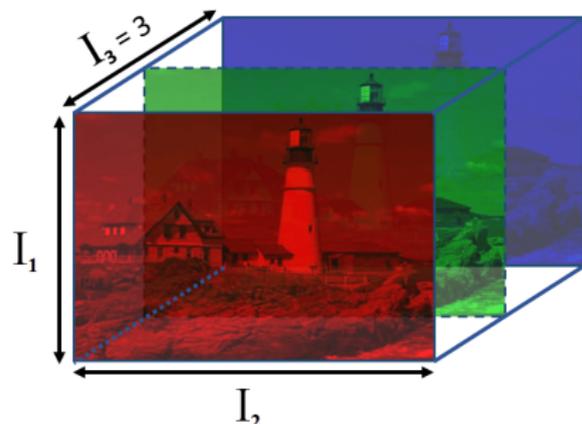
How the pixels are stored:

0	1	2	3	4	5	6	7	8	9	.	.	.		
---	---	---	---	---	---	---	---	---	---	---	---	---	--	--

Niveaux de gris :  $\{0, 1, \dots, 255\}$  (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  $(H, W, 3)$ .

Scalar   Vector   Matrix   Tensor

1

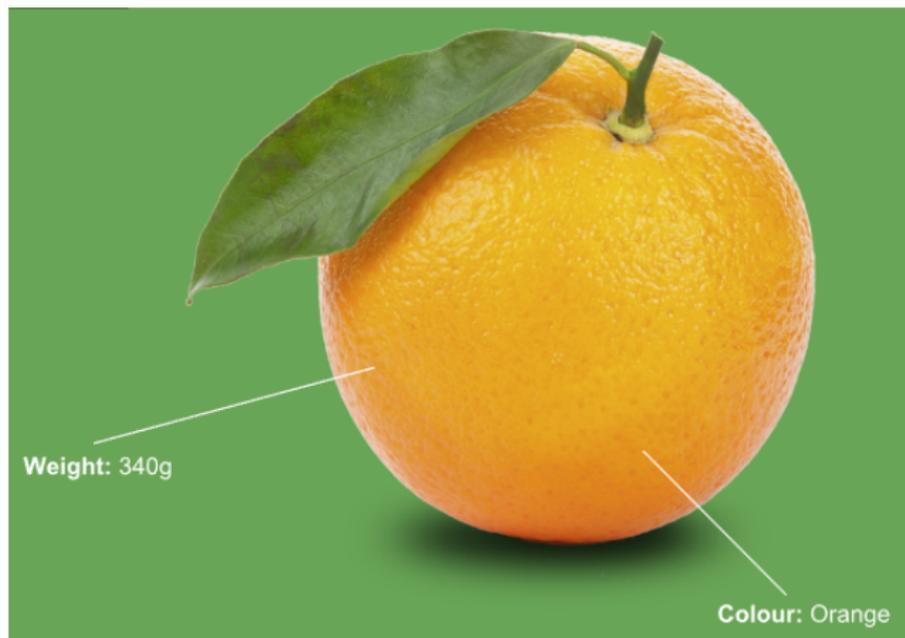
$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

$\begin{bmatrix} 1 & 2 & 3 & 2 \\ 1 & 7 & 5 & 4 \end{bmatrix}$

# Attributs ou caractéristiques (*features*)

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

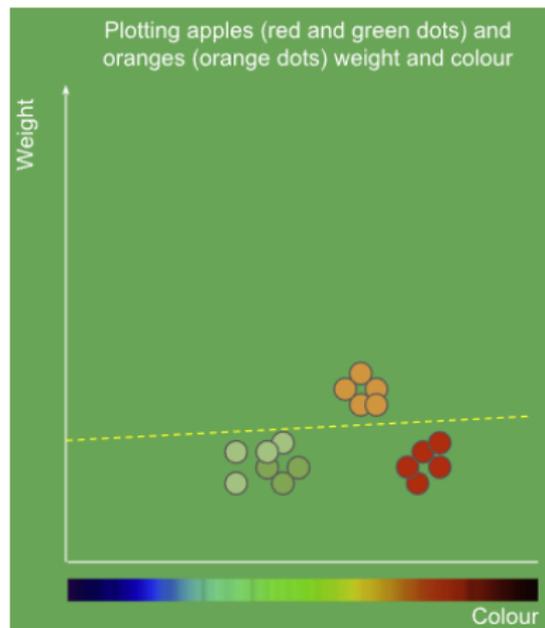


# Attributs ou caractéristiques (*features*)

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://www.welcome.ai/news_info/learn/jason-mayes-machine-learning-101),

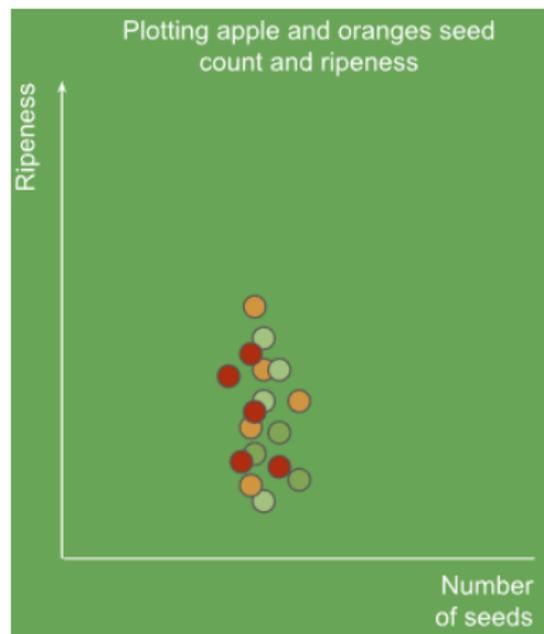
[https://www.welcome.ai/news\\_info/learn/jason-mayes-machine-learning-101](https://www.welcome.ai/news_info/learn/jason-mayes-machine-learning-101)

# Choix des attributs

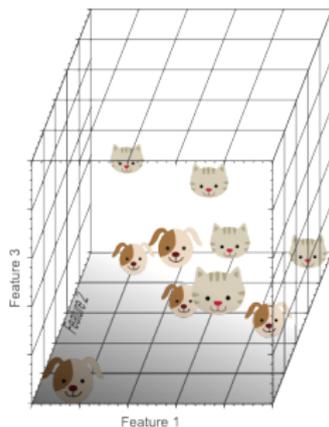
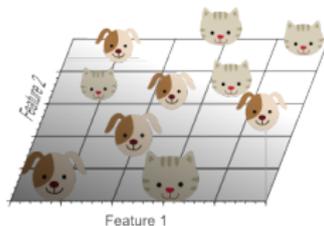
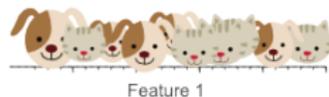
Le choix des caractéristiques est très important. Certaines n'apportent pas d'informations utiles pour le problème.

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.



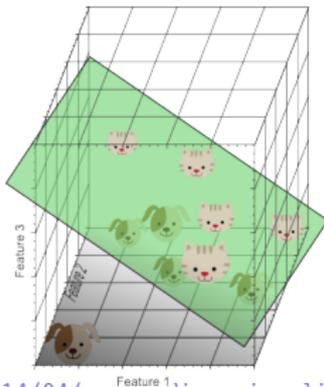
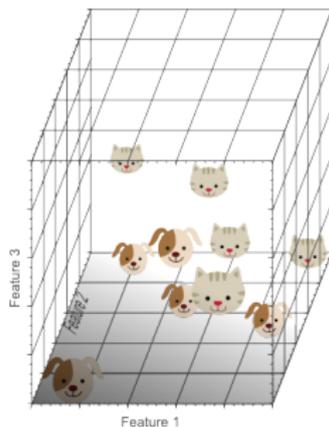
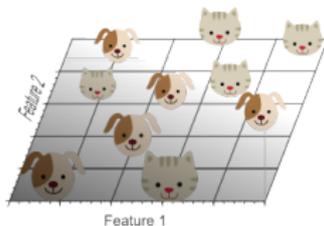
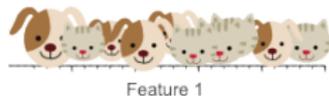
# Dimensions



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 utilisant 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	<b>Chien</b>
2	orange	3kg	<b>Poule</b>
...	...	...	...

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

# Plan

## 1 Introduction

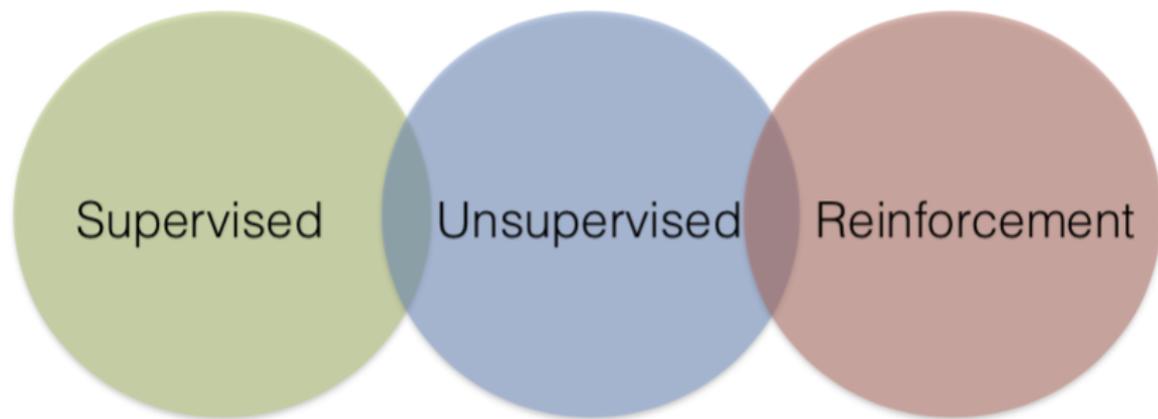
## 2 Bases de l'apprentissage (machine learning)

- Apprentissage supervisé
- Apprentissage non-supervisé
- Apprentissage et généralisation
- Conclusion du cours 1

## 3 Introduction au Deep Learning

- Histoire du connexionnisme
- Perceptrons multi-couches
- Apprentissage par descente du gradient
- Réseaux convolutionnels pour les images
- Architectures profondes

# Trois types de problèmes d'apprentissage



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

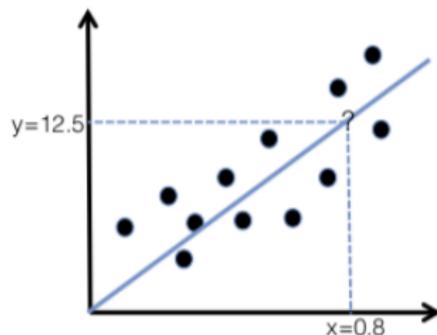
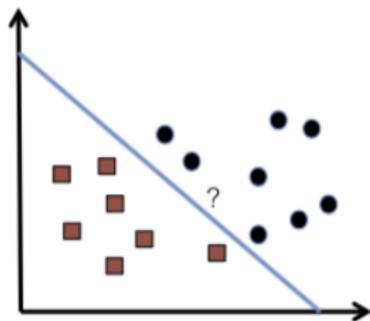
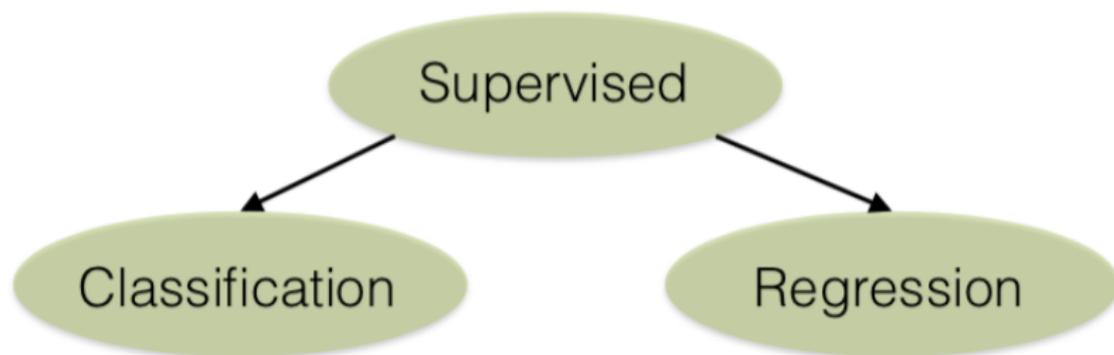
- Discover structure in unlabeled data
- E.g., Document clustering

- 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).

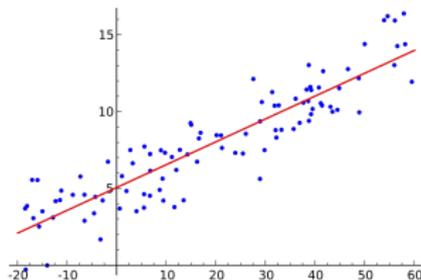
Classement :

Nb de pattes	Couleur	Poids	<b>Animal</b>
4	noir	10kg	<b>Chien</b>
2	orange	3kg	<b>Poule</b>
...	...	...	...

Prévision de série (régression) :

Température	Férié	Nb Clients	Consommation
12	0	134	<b>1456</b>
14	0	124	<b>1234</b>
12	0	154	<b>1512</b>
...	...	...	...

# Apprentissage supervisé : régression linéaire



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

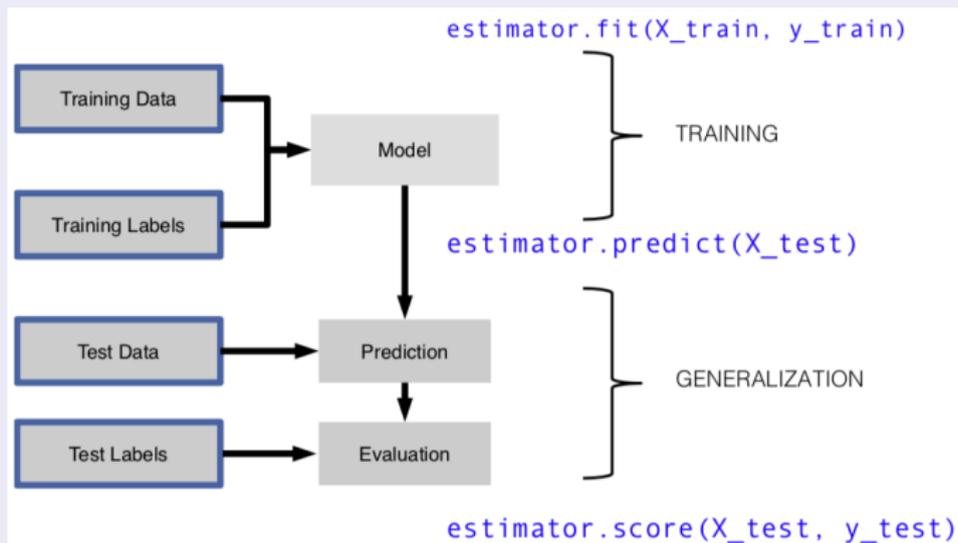
## Apprentissage

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

Source: [https://fr.wikipedia.org/wiki/Ajustement\\_affine](https://fr.wikipedia.org/wiki/Ajustement_affine)

# Apprentissage supervisé : régression linéaire

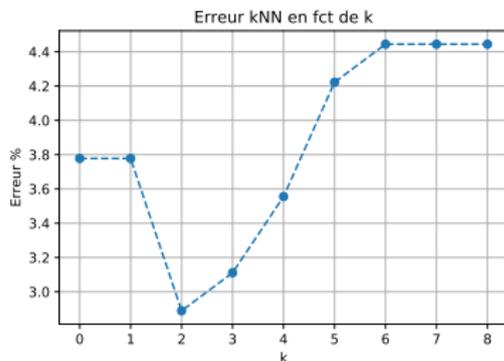
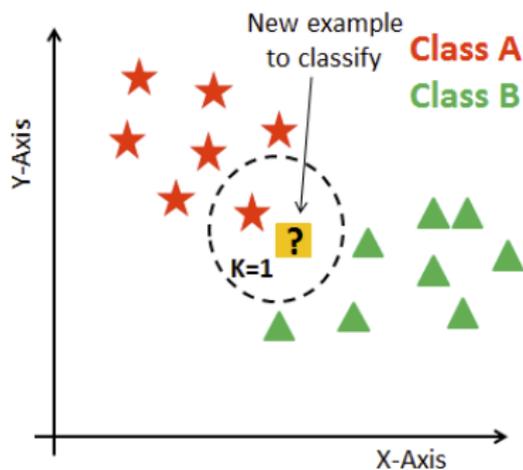
## Workflow scikit-learn en apprentissage supervisé



## Notebook Jupyter

01-RegressionLineaire.ipynb

# Apprentissage supervisé : plus proches voisins (kNN)



Notebook Jupyter

02-kNN.ipynb

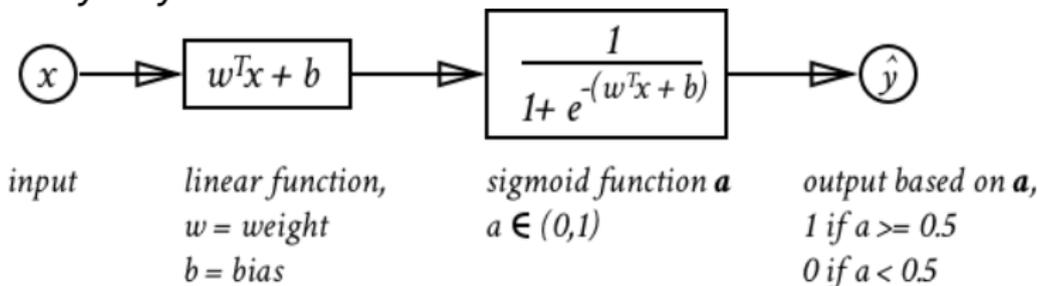
Source:

<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  $\hat{y} = y$ .

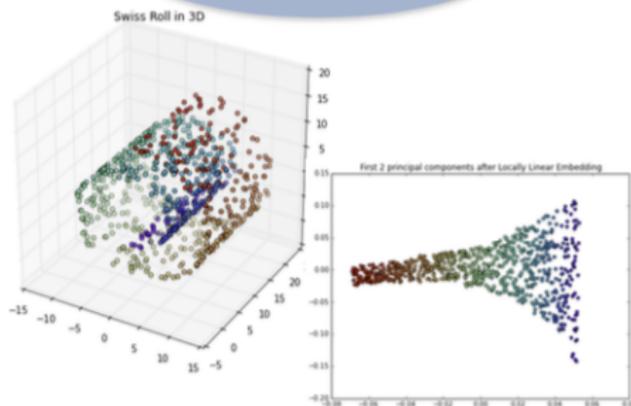
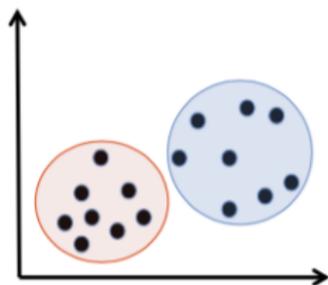
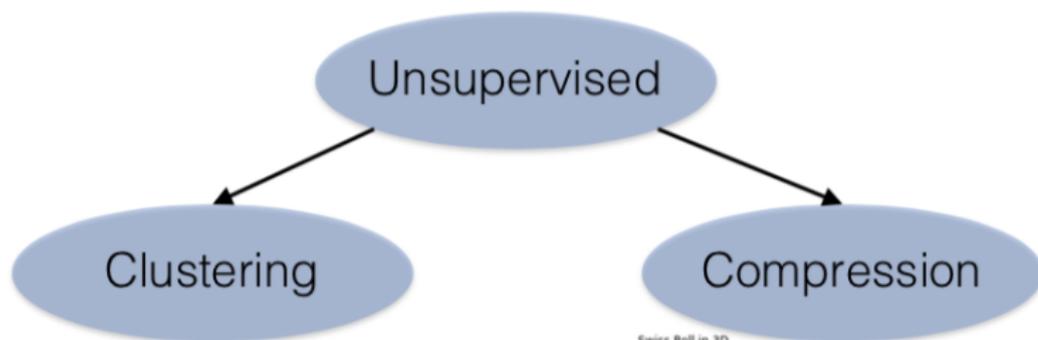


# Autres modèles pour la classification supervisée

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

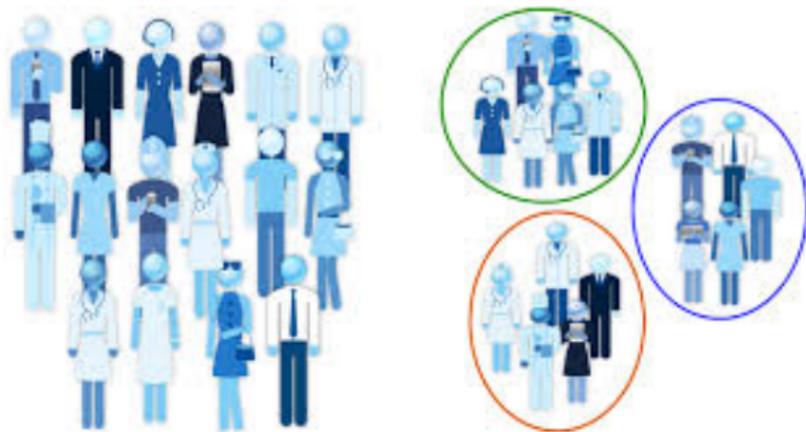


Source:

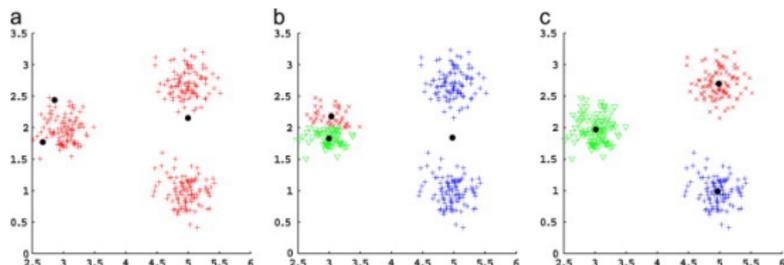
<https://speakerdeck.com/rasbt/slides-from-machine-learning-with-scikit-learn-at-scipy-2016>

# 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

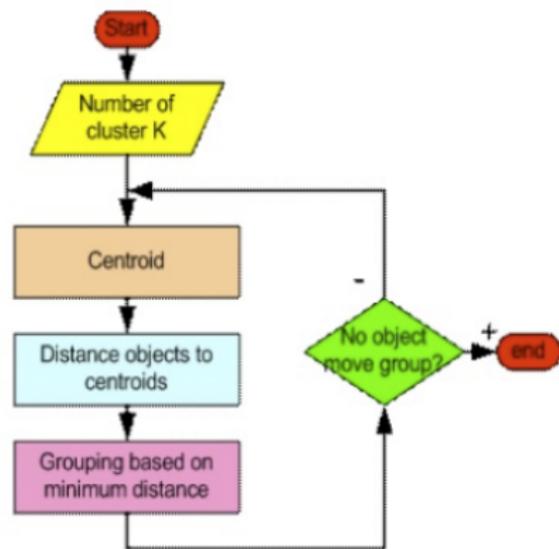


# Apprentissage non-supervisé : $k$ -moyennes ( $k$ -means)



$k$ -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

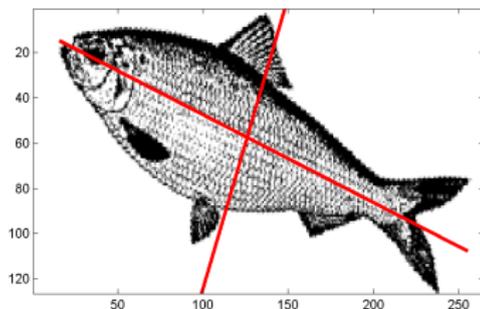


Quantifiée sur 16 couleurs aléatoires



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

L'ACP construit le sous-espace linéaire qui permet de décorrélérer 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

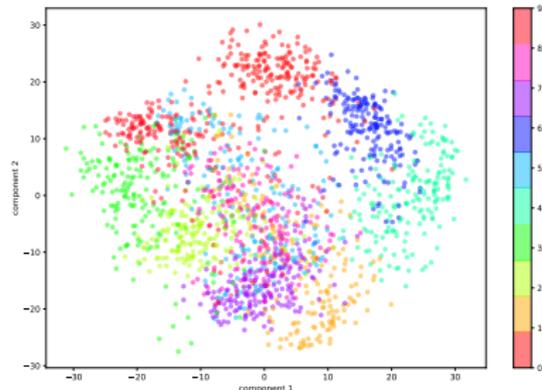
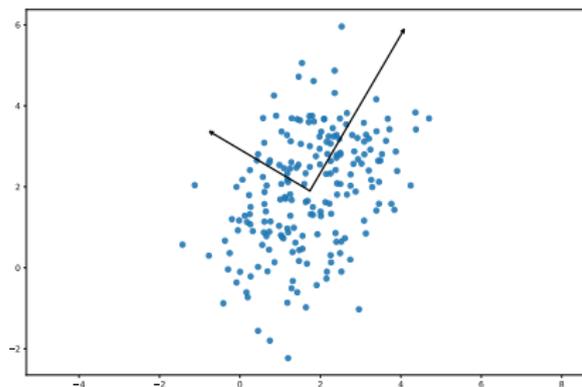
- réduire la dimension des données
- visualiser les données en 2 ou 3 dimensions
- compresser des données ou images

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*

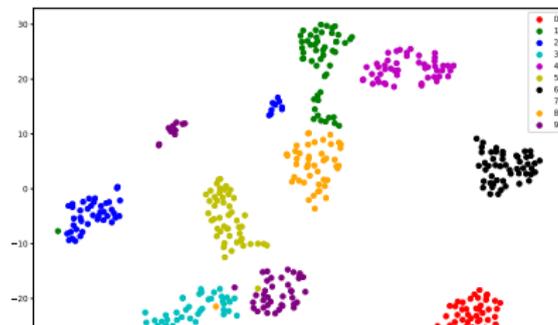
(van der Maaten et Hinton, 2008)

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`



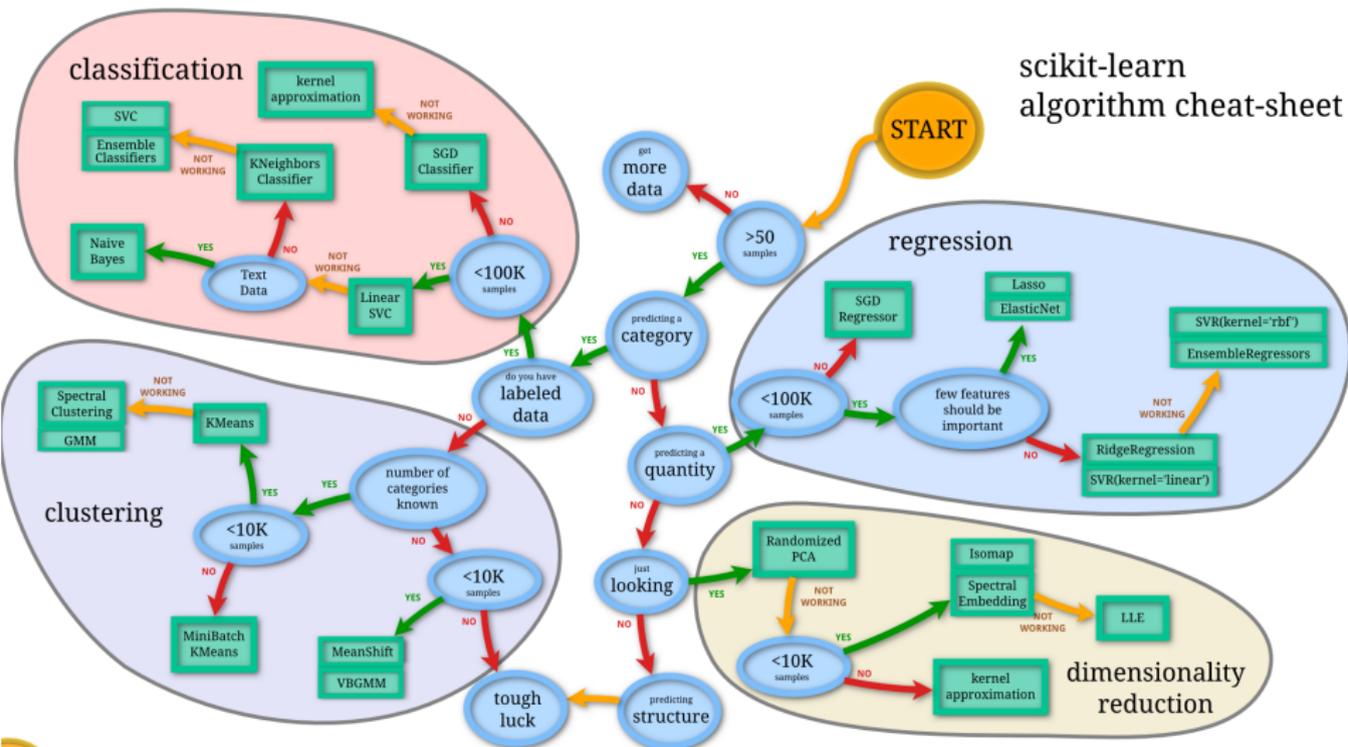
# Apprentissage non-supervisé : UMAP

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>

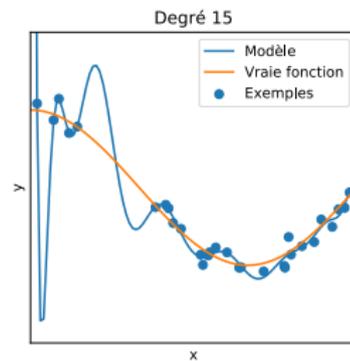
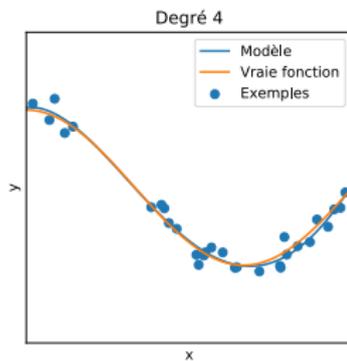
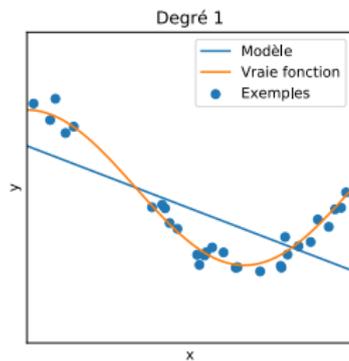
# Guide pour choisir un modèle

scikit-learn  
algorithm cheat-sheet

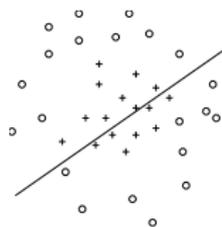


Source: [https://scikit-learn.org/stable/tutorial/machine\\_learning\\_map](https://scikit-learn.org/stable/tutorial/machine_learning_map)

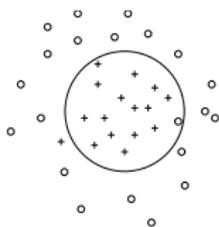
# Compromis Apprentissage/Généralisation



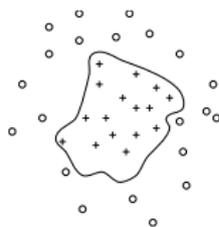
# Compromis Apprentissage/Généralisation



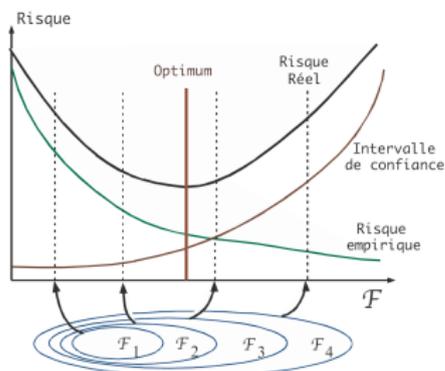
(a) Modèle trop contraint



(b) Optimal



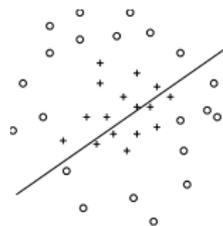
(c) Modèle trop libre



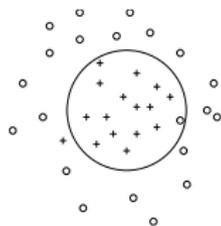
$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \sqrt{\frac{1}{l} (h(\log(2\frac{l}{h}) + 1) - \log(\eta/4))}$$

= **Minimisation Structurale du Risque** (SRM, Vapnik)

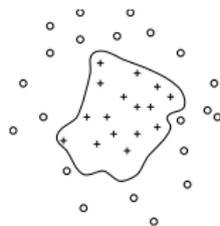
# Compromis Apprentissage/Généralisation



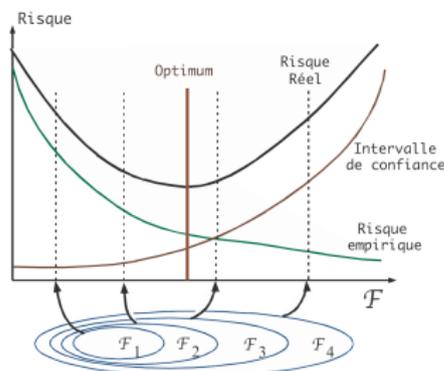
(a) Modèle trop contraint



(b) Optimal



(c) Modèle trop libre



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

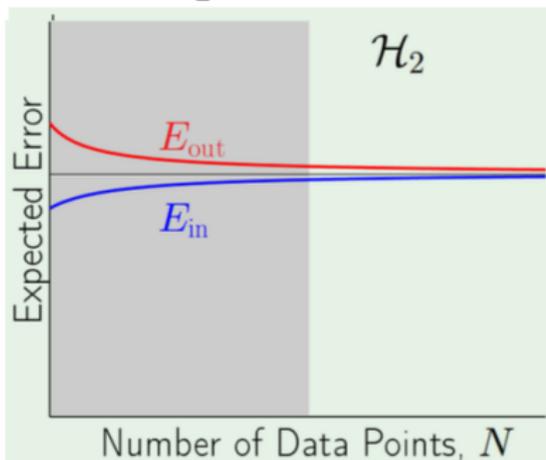
## Méthodes réseaux connexionnistes :

- choix architecture
- régularisation, early stopping

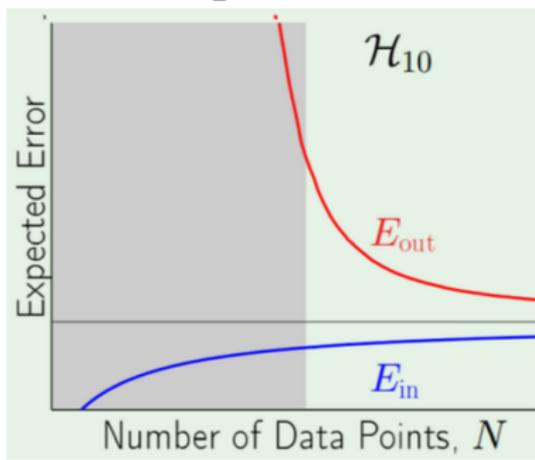
= **Minimisation Structurale du Risque** (SRM, Vapnik)

# Complexité d'un modèle vs volume de données

## Simple Model



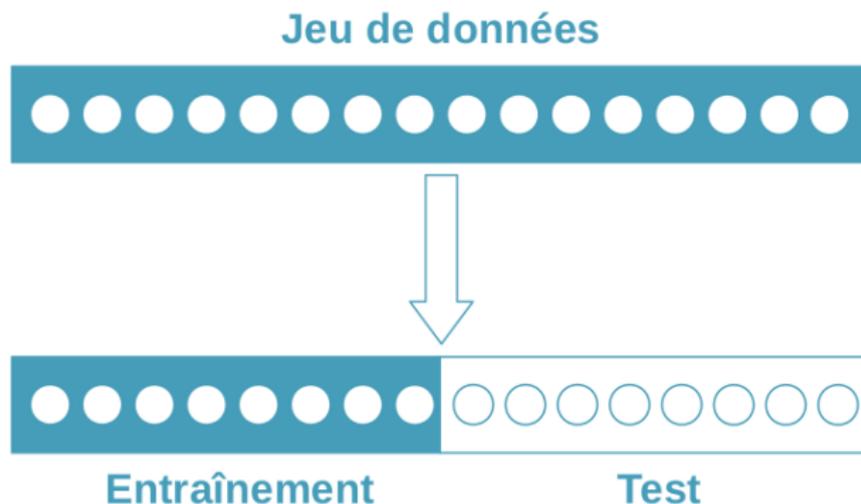
## Complex Model



Overfitting only happens  
for small training sets

# Comment choisir un modèle ?

validation croisée



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

# Conclusion du cours 1

Nous avons introduit :

- apprentissage à partir de données : concept et applications ;
- outils (Python) pour les sciences des données et le machine learning ;
- Principaux modèles pour l'apprentissage supervisé et non-supervisé.

A suivre : méthodes connexionnistes (*deep learning*).

# Plan

## 1 Introduction

## 2 Bases de l'apprentissage (machine learning)

- Apprentissage supervisé
- Apprentissage non-supervisé
- Apprentissage et généralisation
- Conclusion du cours 1

## 3 Introduction au Deep Learning

- Histoire du connexionnisme
- Perceptrons multi-couches
- Apprentissage par descente du gradient
- Réseaux convolutionnels pour les images
- Architectures profondes

