

Apprentissage Automatique

Introduction

Stéphane Herbin

stephane.herbin@onera.fr

« Machine Learning »

- Un domaine scientifique hybride:
 - Statistique
 - Intelligence artificielle
 - « Computer science »
- Utilisant des techniques généralistes:
 - Optimisation numérique
 - Hardware
 - Gestion de base de données

Pourquoi le « Machine Learning »?

- Thème à la mode: Intelligence Artificielle, « deep learning », « big data »...
- Raison épistémologique
 - On ne sait pas modéliser les problèmes complexes
... mais on dispose d'exemples en grand nombre représentant la variété des situations
 - « Data driven » vs. « Model Based »
- Raison scientifique
 - L'apprentissage est une faculté essentielle du vivant
- Raison économique
 - La récolte de données est plus facile que le développement d'expertise

Domaines techniques utilisant du ML

- ML comme outil de conception pour certains fonctions
 - Vision & Reconnaissance des formes
 - Traitement du langage
 - Traitement de la parole
 - Robotique
 - « Data Mining »
 - Recherche dans BDD
 - Recommandations
 - Marketing...
- ML comme outil explicatif
 - Neuroscience
 - Psychologie
 - Sciences cognitives

Artificial Intelligence: A Modern Approach, 4th Global ed.

by Stuart Russell and Peter Norvig

The authoritative, most-used AI textbook, adopted by over 1500 schools.

Table of Contents for the Global Edition (or see the US Edition)

[Preface \(pdf\)](#); [Contents with subsections \(pdf\)](#)

I Artificial Intelligence

- 1 Introduction
- 2 Intelligent Agents

II Problem-solving

- 3 Solving Problems by Searching
- 4 Search in Complex Environments
- 5 Constraint Satisfaction Problems
- 6 Adversarial Search and Games

III Knowledge, reasoning, and planning

- 7 Logical Agents
- 8 First-Order Logic
- 9 Inference in First-Order Logic
- 10 Knowledge Representation
- 11 Automated Planning

IV Uncertain knowledge and reasoning

- 12 Quantifying Uncertainty
- 13 Probabilistic Reasoning
- 14 Probabilistic Reasoning over Time
- 15 Making Simple Decisions
- 16 Making Complex Decisions
- 17 Multiagent Decision Making
- 18 Probabilistic Programming

V Machine Learning

- 19 Learning from Examples
- 20 Knowledge in Learning
- 21 Learning Probabilistic Models
- 22 Deep Learning
- 23 Reinforcement Learning

VI Communicating, perceiving, and acting

- 24 Natural Language Processing
- 25 Deep Learning for Natural Language Processing
- 26 Robotics
- 27 Computer Vision

VII Conclusions

- 28 Philosophy, Ethics, and Safety of AI
- 29 The Future of AI
- Appendix A: Mathematical Background
- Appendix B: Notes on Languages and Algorithms
- Bibliography ([pdf](#) and [LaTeX .bib file](#) and [bib data](#))
- Index ([pdf](#))

[Exercises \(website\)](#)

[Figures \(pdf\)](#)

[Code \(website\)](#); [Pseudocode \(pdf\)](#)

Covers: [US](#), [Global](#)

Apprentissage automatique : applications

Anti-Spam (*Classifieur Bayésien*)



1997 : DeepBlue bat Kasparov (pas de ML)

2017: Alpha GO bat Ke Jie

2019: AlphaStar champion de StarCraft

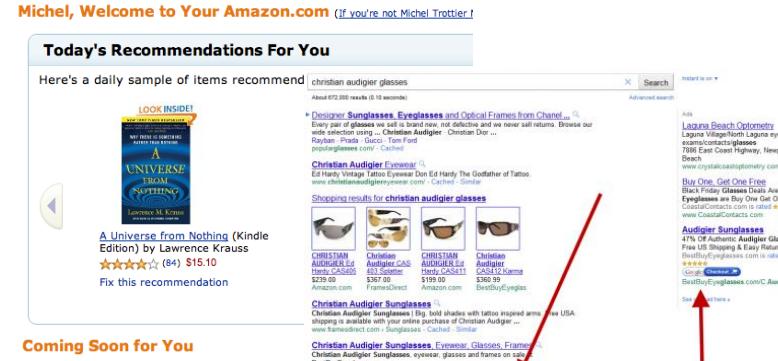


Tri postal automatique (déttection de chiffres manuscrits par réseaux de neurones)



Apprentissage automatique : applications

Recommandation ciblée
(régression logistique)



Appareil photo avec détection de visages (boosting)

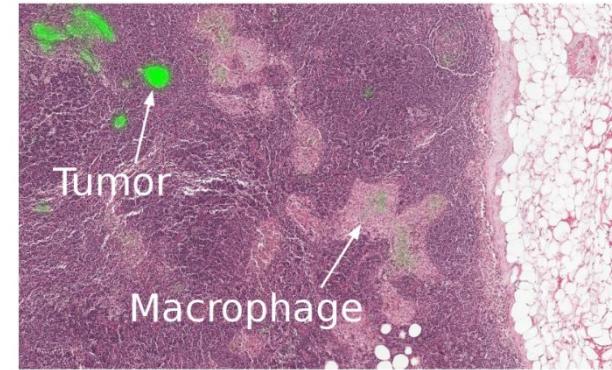


Chat Bots
(Réseaux de neurones)



Apprentissage automatique : applications

Diagnostic médical
(Réseaux de neurones)



Traduction multi-lingue
(Réseaux de neurones)

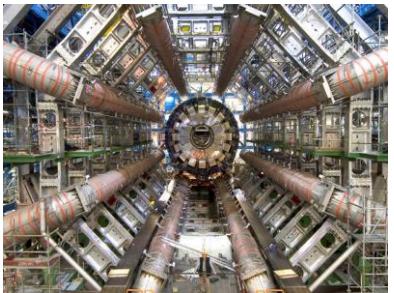


Biochimie (prédiction de la structure 3D des protéines)
(Réseaux de neurones + modèles biologiques)

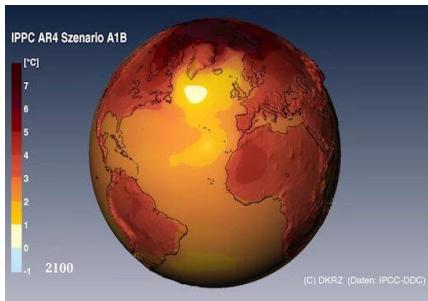


Données = carburant du ML

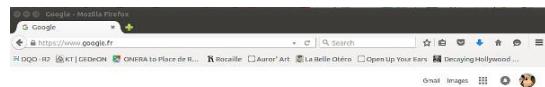
CERN /
Large Hadron Collider
~70 Po/an



DKRZ (Climat)
500 Po



Google :
24 PetaOctets/jour



Copernicus :
> 1Po/an

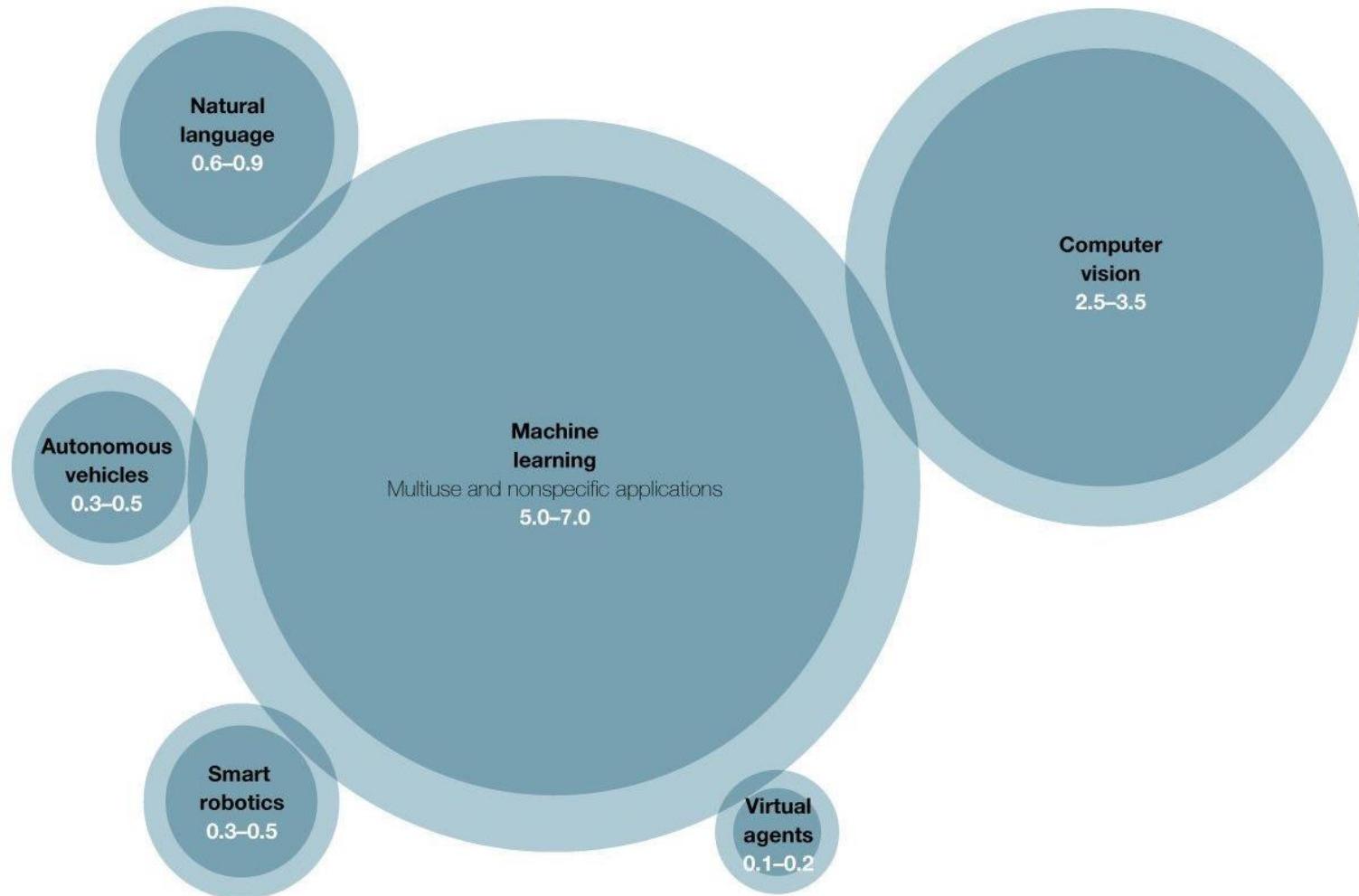


Square Kilometer Array
1376 Po/an (en 2024)



BIG DATA

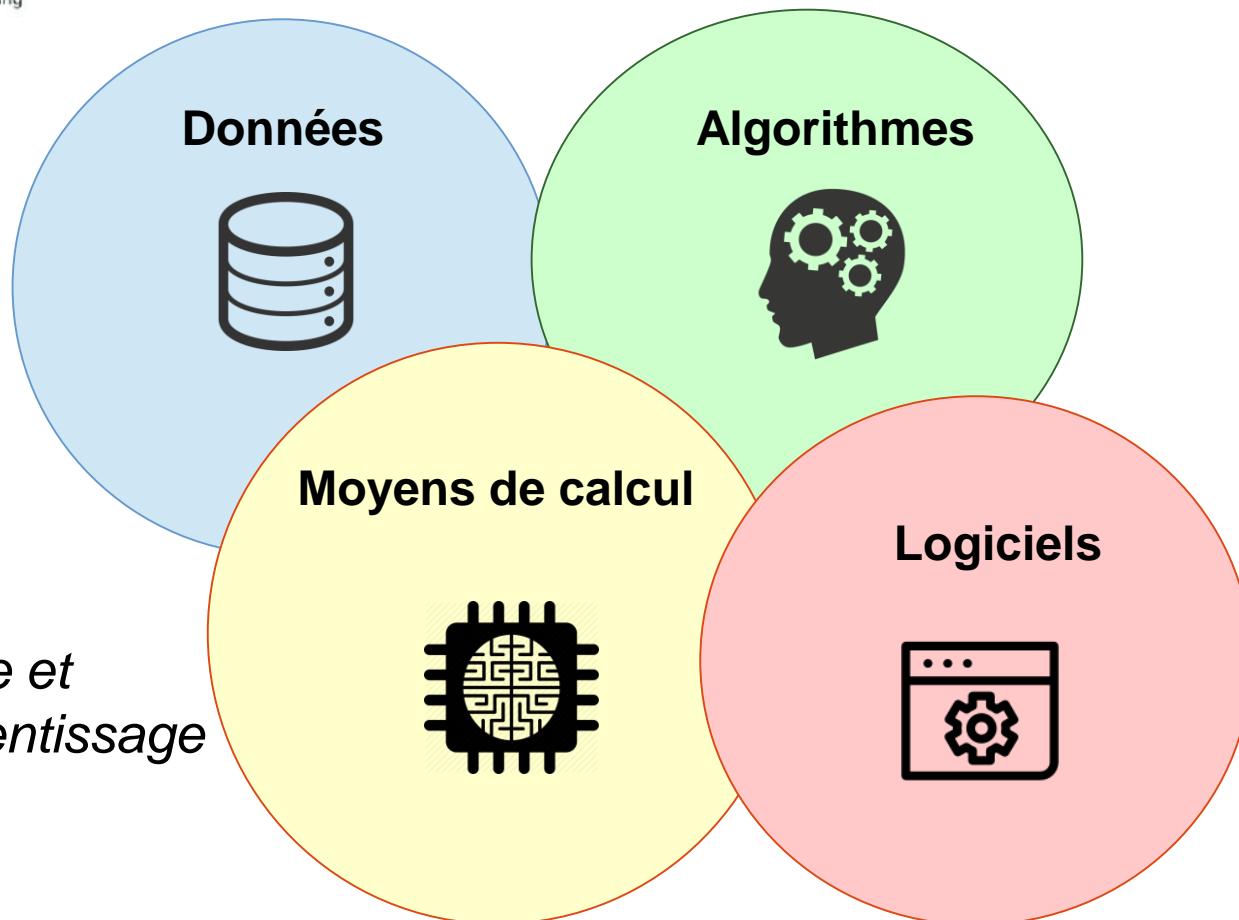
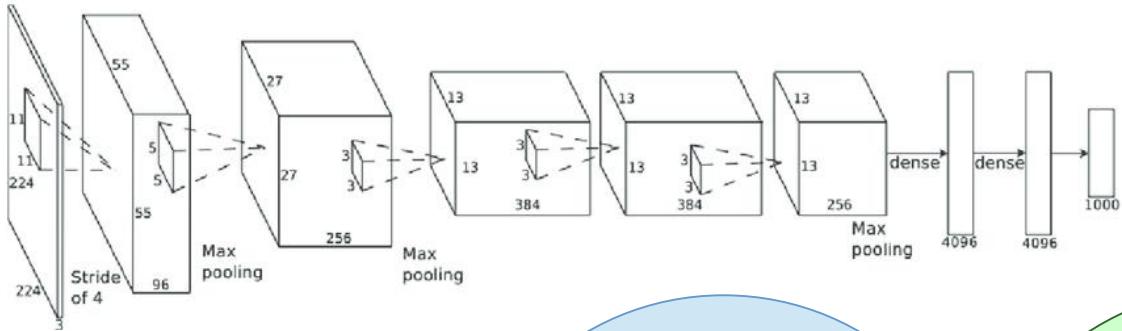
External investment in AI-focused companies by technology category, 2016¹
\$ billion



¹ Estimates consist of annual VC investment in AI-focused companies, PE investment in AI-related companies, and M&A by corporations. Includes only disclosed data available in databases, and assumes that all registered deals were completed within the year of transaction.

McKinsey&Company | Source: Capital IQ; Pitchbook; Dealogic; McKinsey Global Institute analysis

« Deep Learning » : le mot clé inévitable



Une rupture scientifique et technologique en apprentissage

Deux pratiques du ML

- ML fournit la solution
 - Prédicteur à partir des données brutes
 - Démarche « end-to-end »
 - Modèles de fondation
- ML est une aide
 - Modèles de substitution (« surrogate »)
 - IA hybride
 - « Physically Informed Neural Networks » (PINNs)

MACHINE LEARNING

Problématique générale

Dans ce cours

L'apprentissage automatique est:

- une démarche de **conception** d'un **prédicteur**
- par une modélisation ou programmation **non explicite** à partir **d'exemples** (signaux, images, texte, mesures...)

Formalisation élémentaire

- Donnée à interpréter (x)
 - Mesures, texte, image, enregistrement, vidéo ou caractéristiques extraites de ...
- Prédiction (y)
 - Décision, choix, action, réponse, préférence, groupe, commande, valeur...
- Echantillons ($\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N\}$)
 - Exemples de données et de (bonnes) prédictions
 - « Base d'apprentissage »: \mathcal{D}
- Hypothèses fortes:
 - Les échantillons contiennent toute l'information exploitable et utile
 - Le futur = le passé: les données à prédire suivent la distribution d'apprentissage

Formalisation élémentaire

Prédicteur = Fonction paramétrique de paramètres \mathbf{W}

$$y = F(\mathbf{x}; \mathbf{W})$$

Apprentissage = trouver le \mathbf{W} qui optimise un critère C

$$\mathbf{W} = \arg \min_{\mathbf{W}'} C(\mathcal{D}, \mathbf{W}')$$

A partir de la base d'apprentissage $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$

Le critère C est « empirique » = il dépend de données, c'est une **statistique**

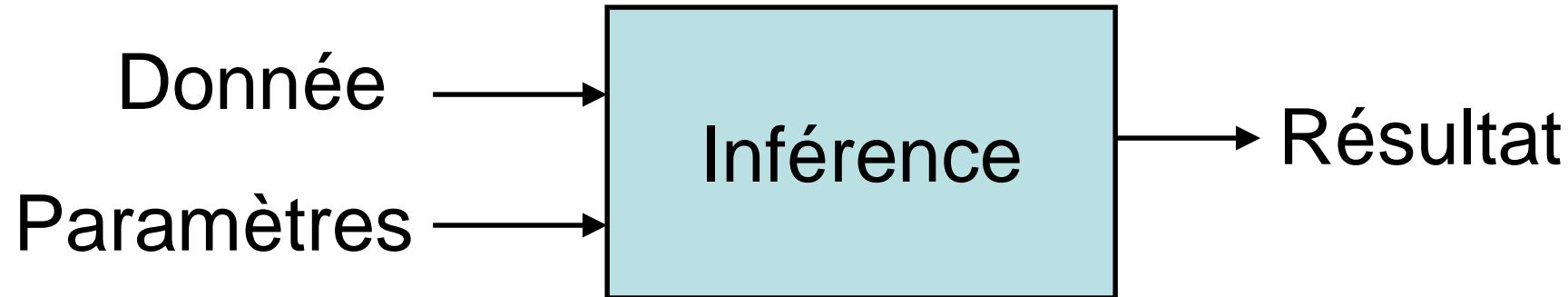
$$\text{ex. } C(\mathcal{D}, \mathbf{W}) = \sum_i \|y_i - F(\mathbf{x}_i; \mathbf{W})\|^2$$

Deux phases

Apprentissage (train)



Prédiction (inférence et test)



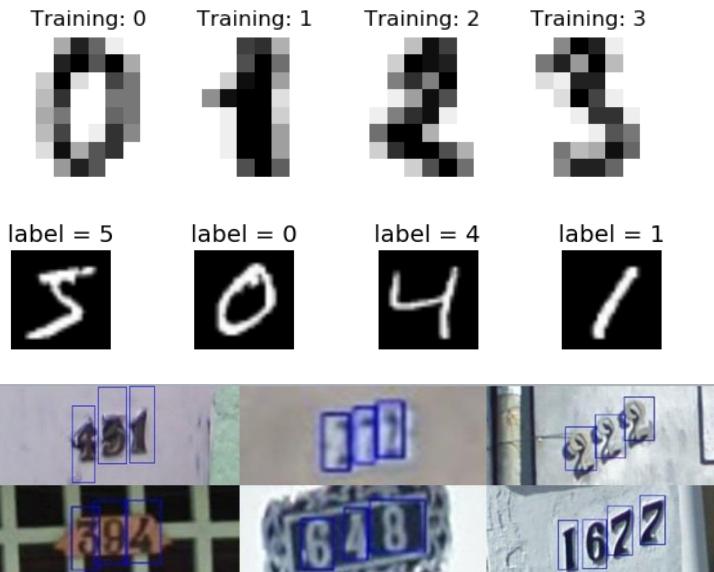
Exemple: Reconnaissance de chiffres manuscrits



- Comment définir les éléments ?
 F, W, x, y
- Les fonctions d'apprentissage et de prédiction?
 $\mathcal{D} \mapsto W$
 $W, x \mapsto y$

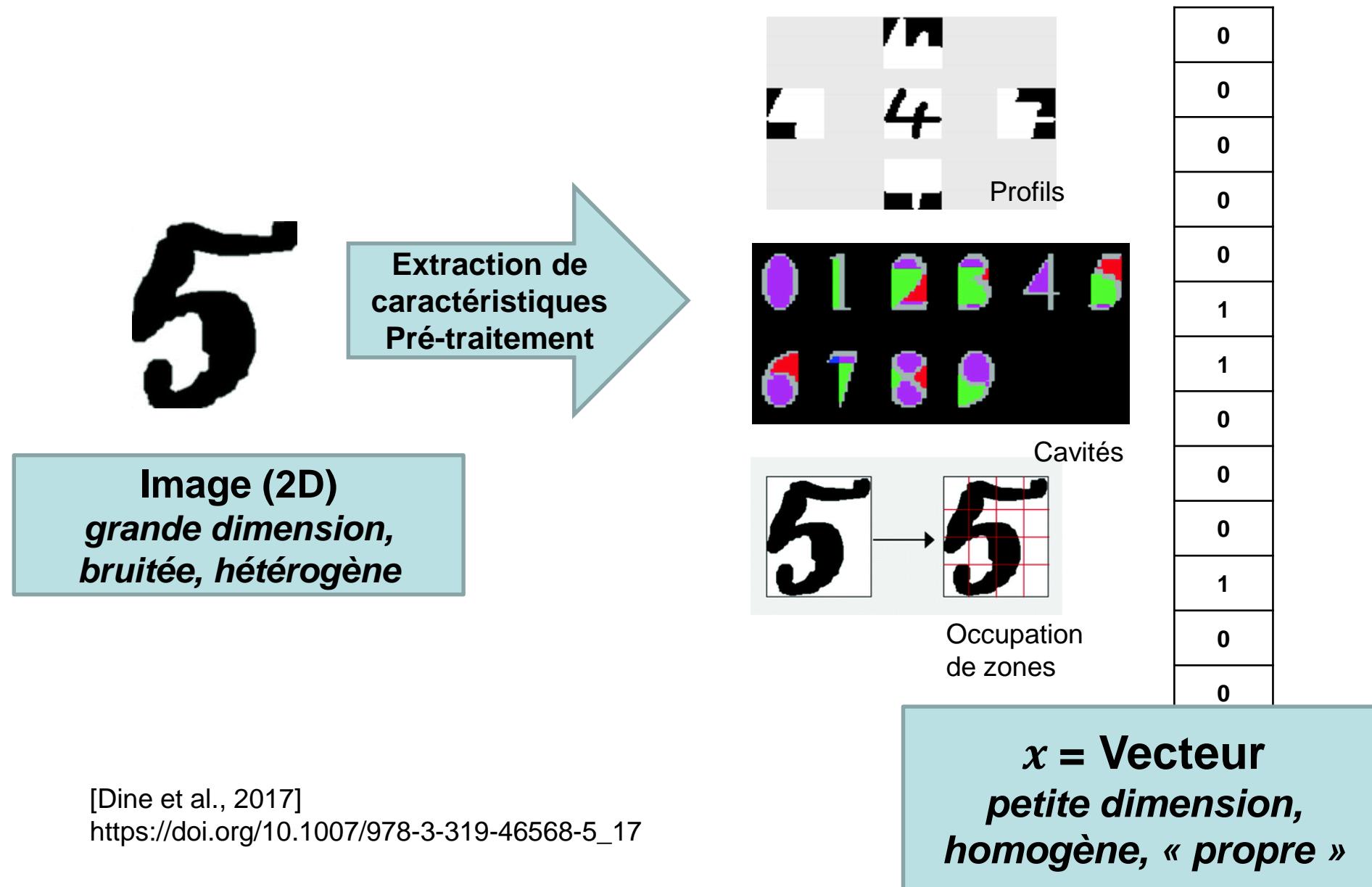
Etape 1: choix de la base de données

- Elle existe:
 - Scikit-learn:
 - MNIST:
 - SVHN:



- Il faut la construire:
 - Recueil de données existantes
 - Expérimentations (photos, mesures...)

Etape 2: mise en forme des données



[Dine et al., 2017]

https://doi.org/10.1007/978-3-319-46568-5_17

Etape 3: choix de l'approche

- Quelle fonction?
 - Classification
- Quel type d'apprentissage?
 - Apprentissage supervisé (On connaît les classes cibles)
- Nature des données?
 - Vecteurs de taille fixe mais de grandes dimensions (>10)
- Taille de la base de données?
 - Moyenne/Grande (> 10000 exemples)
- Modèle de prédicteur?
 - Arbres de décision, SVM, Réseaux de neurones...

Types de prédiction

- **Classification**
 - Binaire: spam / non spam
 - Identification: « tata Monique »
- **Régression**
 - Prédiction de température, de cours de bourse
 - Localisation d'objet dans image
 - Commande
- **Structure**
 - Graphe des articulations d'une personne
- **Regroupement**
 - Photos dans base de données personnelle
- **Texte**
 - « C'est un chat qui saute sur une table. »

Types d'apprentissage

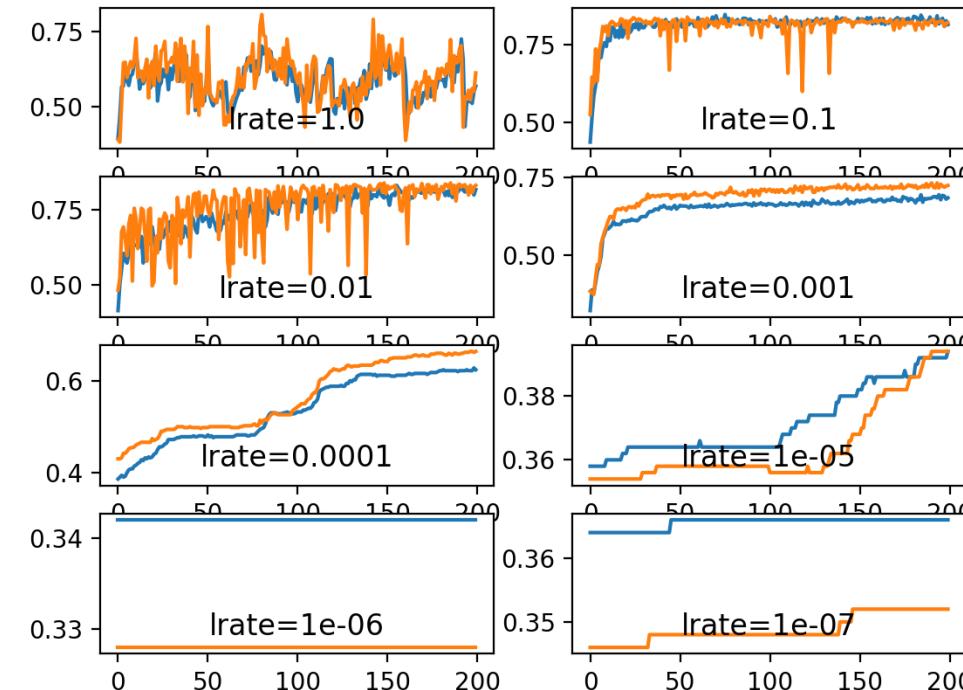
- **Apprentissage supervisé**
 - Les données d'apprentissage contiennent les objectifs de prédiction (annotations)
- **Apprentissage non supervisé**
 - Les données d'apprentissage sont brutes
- **Apprentissage par renforcement**
 - Les prédictions sont issues d'une séquence d'actions et sont caractérisées par une mesure de qualité (« reward »)
- **Apprentissage semi-supervisé**
 - Les données d'apprentissage sont partiellement annotées
- **Apprentissage par transfert**
 - Les données d'apprentissage sont proches du problème visé

Modèle de prédicteur

- Dépend de la forme des données (vecteurs, listes, réels/discret) et du type de prédiction
- Exemples
 - Plus proches voisins
 - Machines à vecteurs de supports (SVM)
 - Arbre de décision
 - Ensembles de classifieurs (forêts aléatoires, « boosting »...)
 - Réseaux de neurones
 - Modèles probabilistes (Réseaux bayésiens, Chaînes ou champs de Markov...)
 - Règles/Programmation logique
 - Etc.

Etape 4: apprentissage

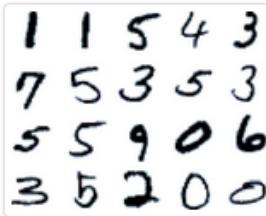
- Définir un espace fonctionnel et un critère paramétrique (coût, énergie...)
- Appliquer un **optimiseur** et régler ses paramètres
- Vérifier que l'apprentissage se passe bien
 - Valeur du critère
 - Convergence
 - Paramètres du prédicteur
 - ...



Optimisation

- **Optimisation convexe**
 - Ex. Minimisation séquentielle de problème quadratique
- **Optimisation stochastique**
 - Ex. Descente de gradient stochastique, Algorithmes génétiques
- **Optimisation sous contraintes**
 - Ex. Programmation linéaire
- **Optimisation combinatoire**
 - Ex. Algorithmes gloutons

Etape 5: évaluation



MNIST 50 results collected

Units: error %

Classify handwritten digits. Some additional results are available on the [original dataset page](#).

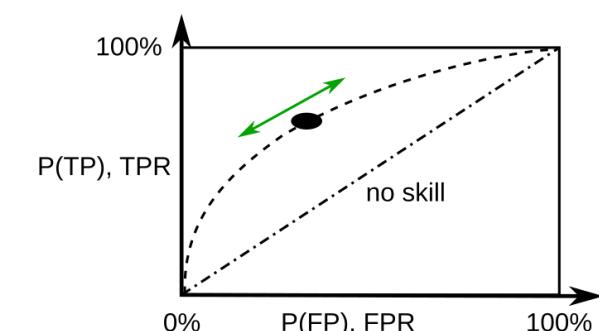
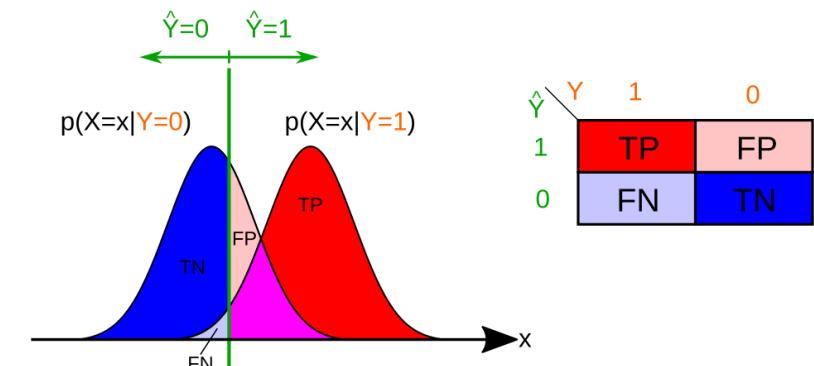
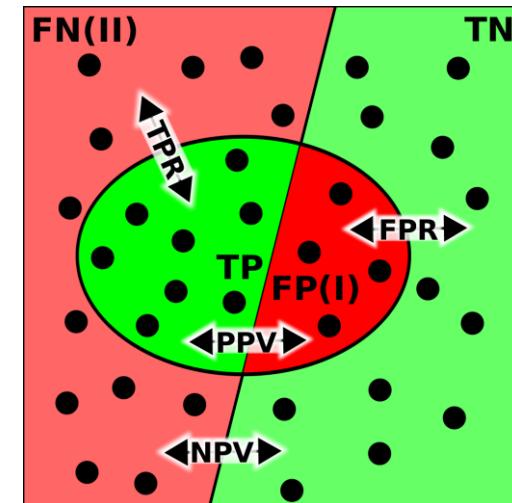
Result	Method	Venue	Details
0.21%	Regularization of Neural Networks using DropConnect	ICML 2013	
0.23%	Multi-column Deep Neural Networks for Image Classification	CVPR 2012	
0.23%	APAC: Augmented PAttern Classification with Neural Networks	arXiv 2015	
0.24%	Batch-normalized Maxout Network in Network	arXiv 2015	Details
0.29%	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree	AISTATS 2016	Details
0.31%	Recurrent Convolutional Neural Network for Object Recognition	CVPR 2015	
0.31%	On the Importance of Normalisation Layers in Deep Learning with Piecewise Linear Activation Units	arXiv 2015	

Métriques d'évaluation

- Dépend du type de prédiction
- Classification
 - Taux d'erreur moyen
 - Matrice de confusion
 - Précision/rappel
 - Courbe ROC
- Régression
 - Erreur quadratique
- Détection
 - Taux de recouvrement moyen

https://scikit-learn.org/stable/modules/model_evaluation.html

https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers



Train et Test

- Apprentissage (« train »)
 - Exploité pour calculer le prédicteur
 - C'est un moyen de modélisation
- Evaluation (« test »)
 - Utilisé pour estimer l'erreur de prédiction une fois l'apprentissage achevé
 - C'est la situation réelle (ou censée l'être), l'inférence
 - Données pour lesquelles on veut une bonne prédiction
 - NE PAS UTILISER POUR L'APPRENTISSAGE
- Validation
 - Utilisé pour simuler/estimer l'erreur de test pendant l'apprentissage

Résumé des étapes de conception

1. Constituer des bases de données
2. Préparer les données: Analyser, visualiser, prétraiter, transformer, extraire, constituer les ensembles train/test
3. Concevoir le modèle (type de prédicteur, principe d'apprentissage)
4. Définir un critère et Optimiser (l'apprentissage proprement dit)
5. Evaluer

ML = Travailler avec des données

Différentes activités/métiers

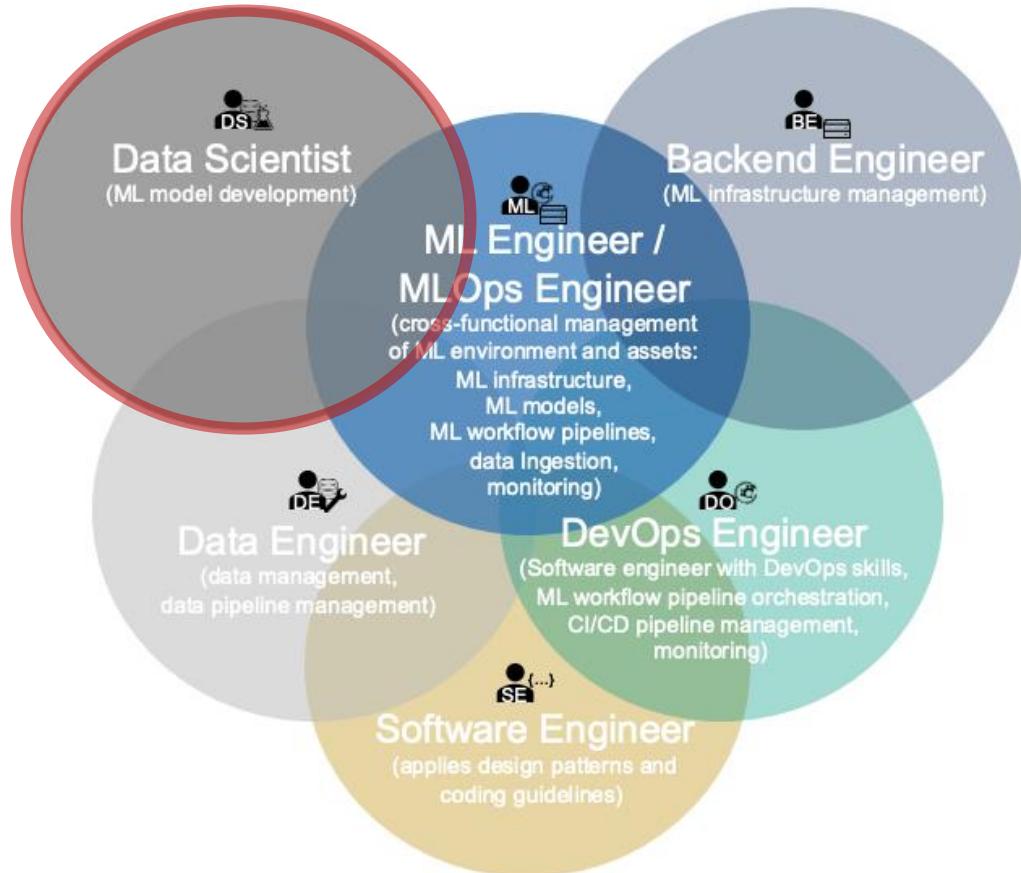
Préparer les données

- Etape coûteuse mais indispensable
- Objectif: rendre possible l'apprentissage avec des données
 - Propres, homogènes, recalées, calibrées, organisées, facilement accessibles, renseignées...
- « Data engineering » (un nouveau métier!)

Transformer les données

- Objectif: Extraire l'information des données, leurs caractéristiques (« features »), calculer leur « forme »
- « Data scientist »

Les métiers du ML et des données



Kreuzberger, D., Kühl, N., & Hirschl, S. (2022). Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *arXiv preprint arXiv:2205.02302*.

Les “process” du ML

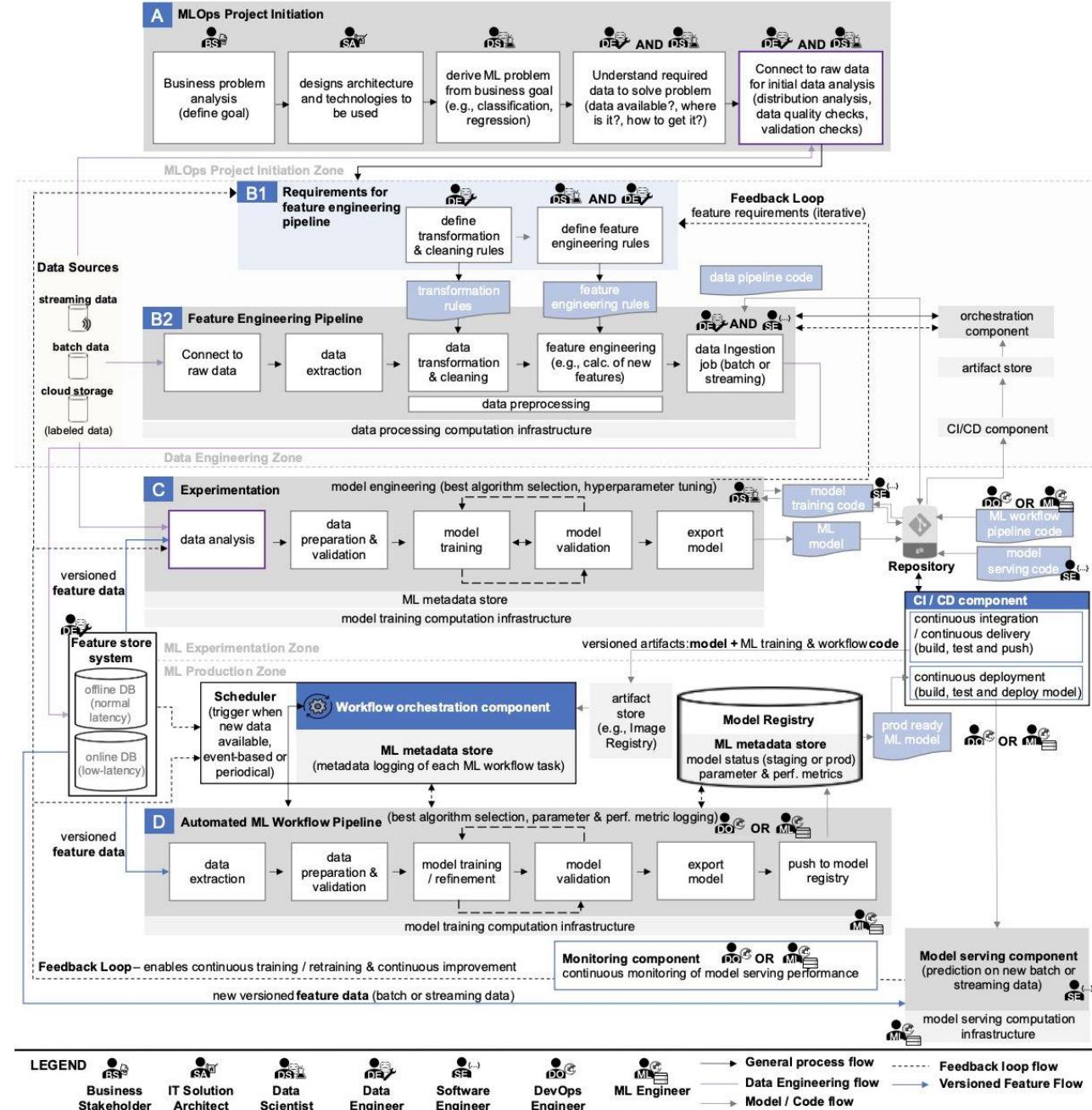


Figure 4. End-to-end MLOps architecture and workflow with functional components and roles

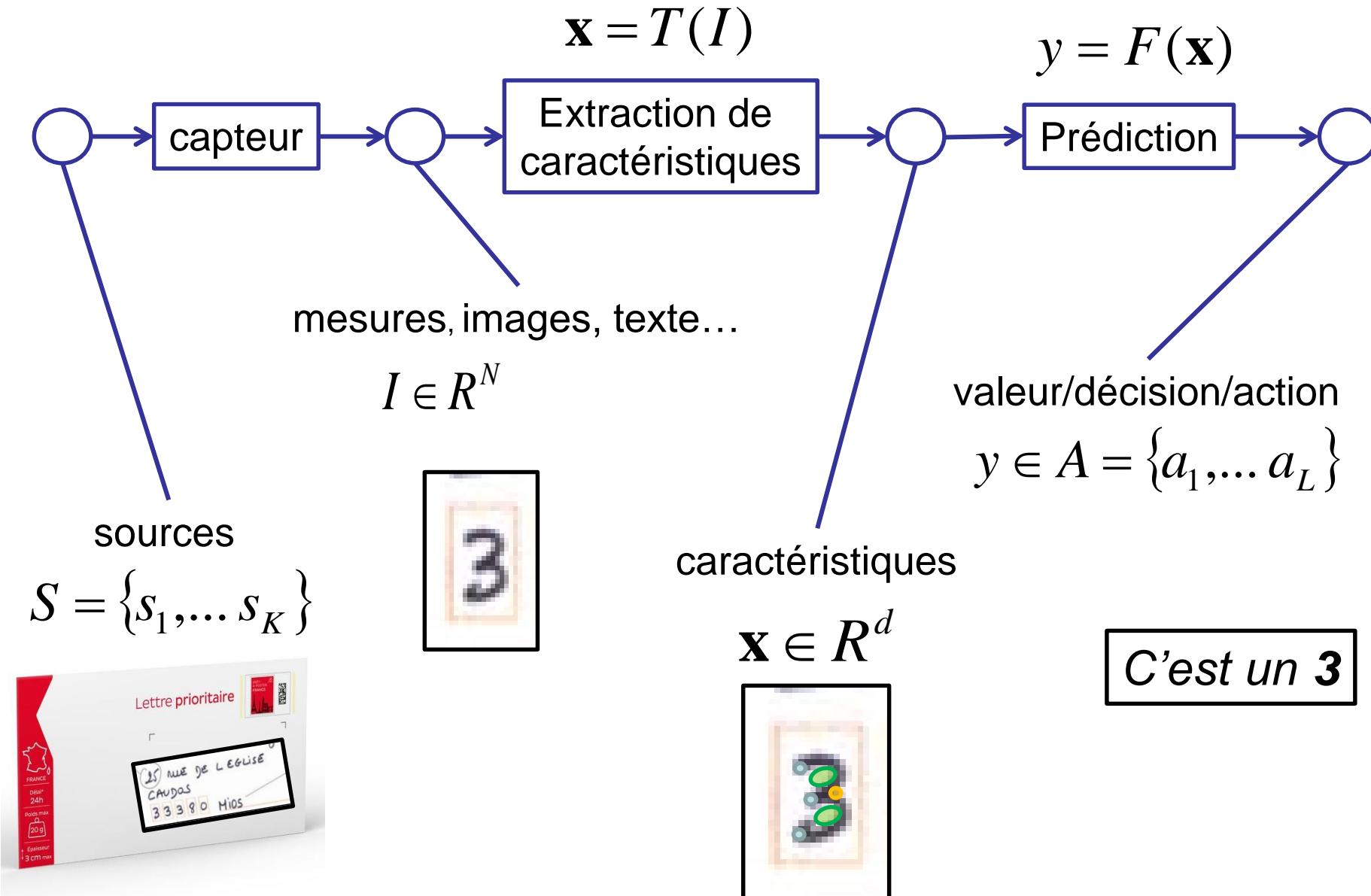
Des données vers l'information

- Données brutes pas exploitables directement:
 - Bruitées
 - Grandes dimensions (image, audio)
 - Information utile noyée

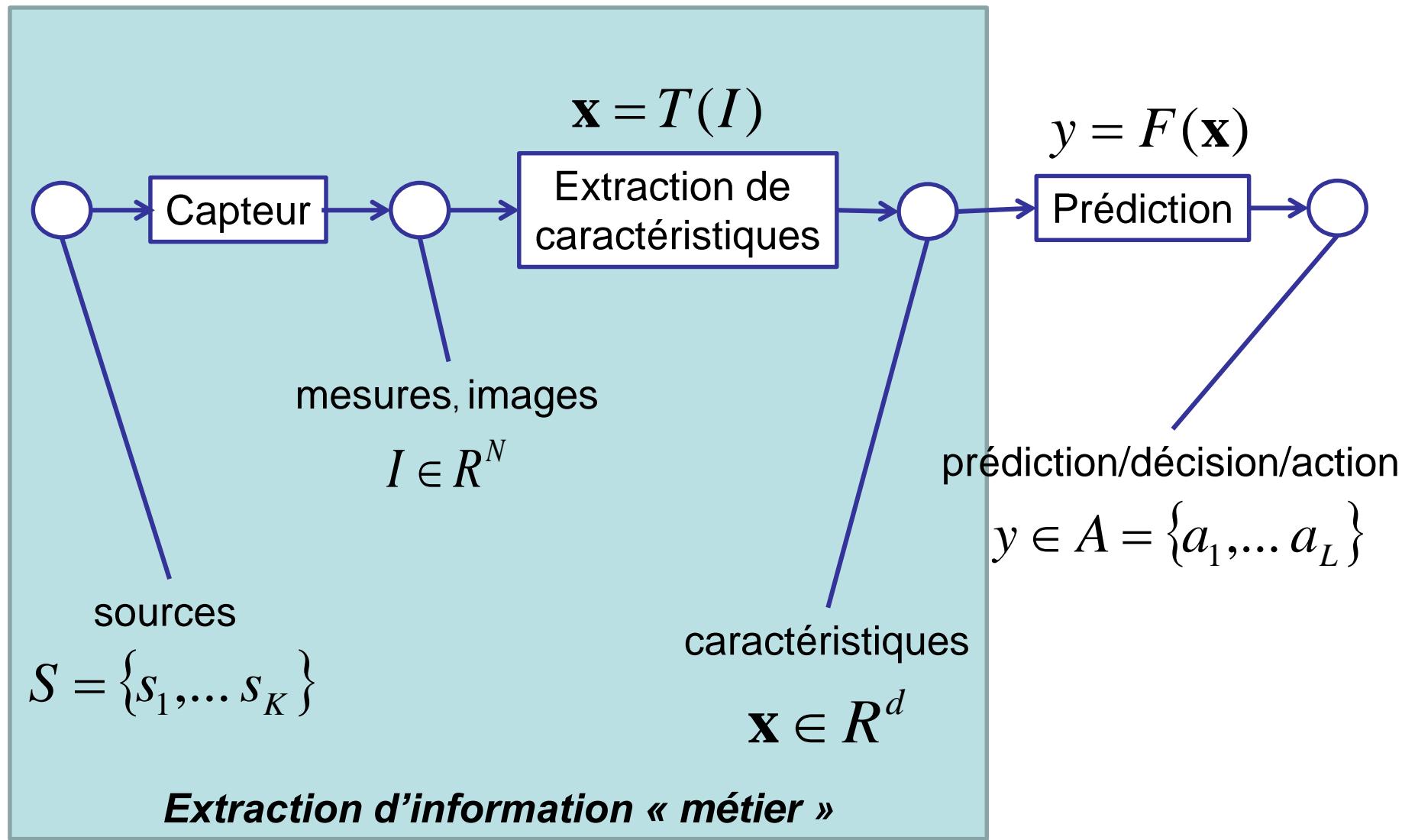
→ **Extraction de caractéristiques** (« Feature extraction » en anglais)

- Etape critique de la « reconnaissance des formes »
 - Caractéristiques trop simples: pas assez d'information, confusion
 - Caractéristiques trop riches: complexité, bruit, grande variabilité
- Compromis difficile à régler entre expressivité, invariance, robustesse, taille, coût de calcul...
- Deux cas de figure:
 - On sait ce qui est important et pourquoi (expertise « métier »)
→ modélisation
 - On ne sait pas décrire ce qui est important
→ on l'apprend!

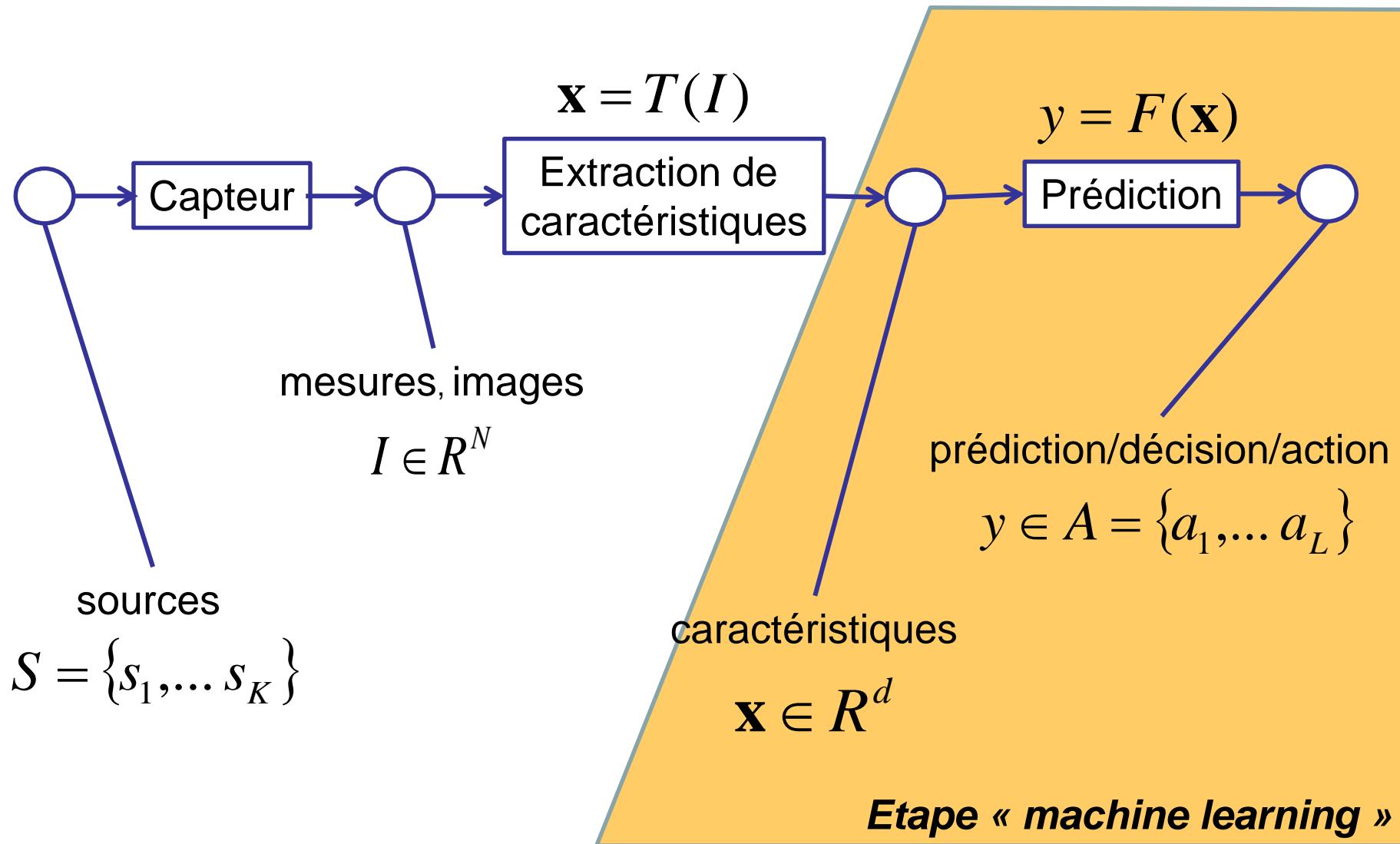
Chaîne de traitement « générique »



1 – Construire une représentation (forme)

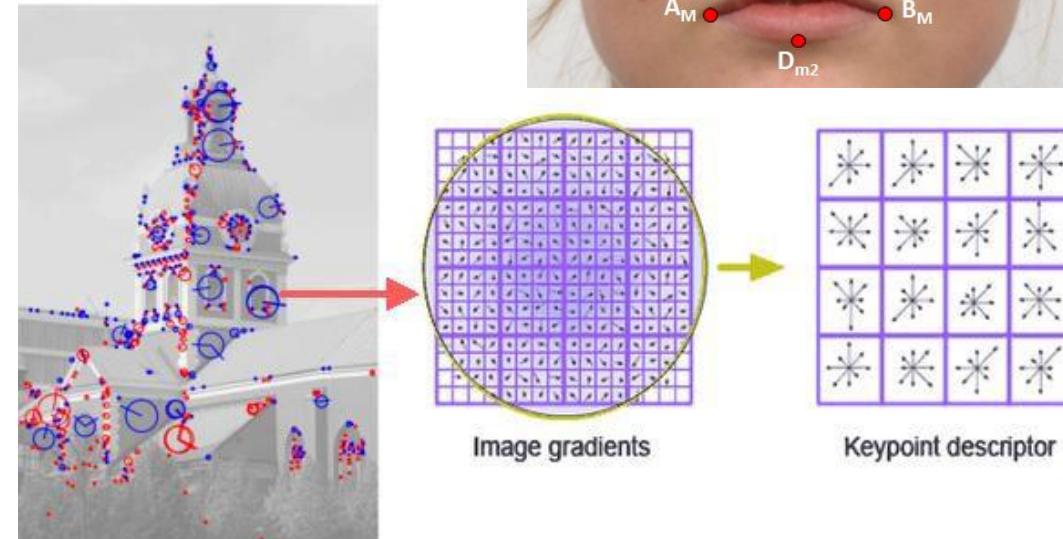
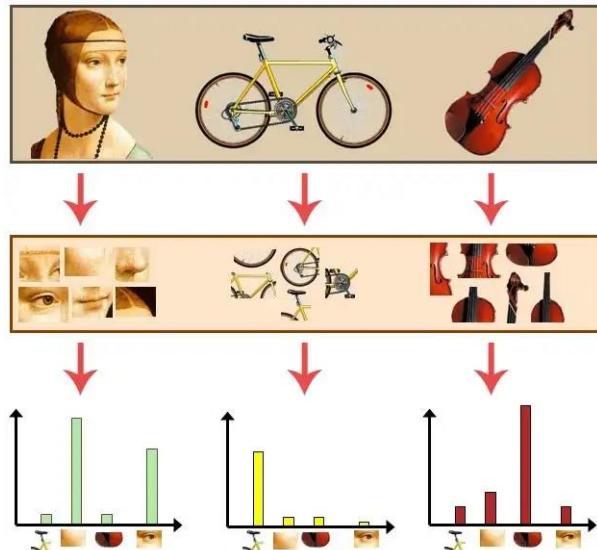


2 - Prédire

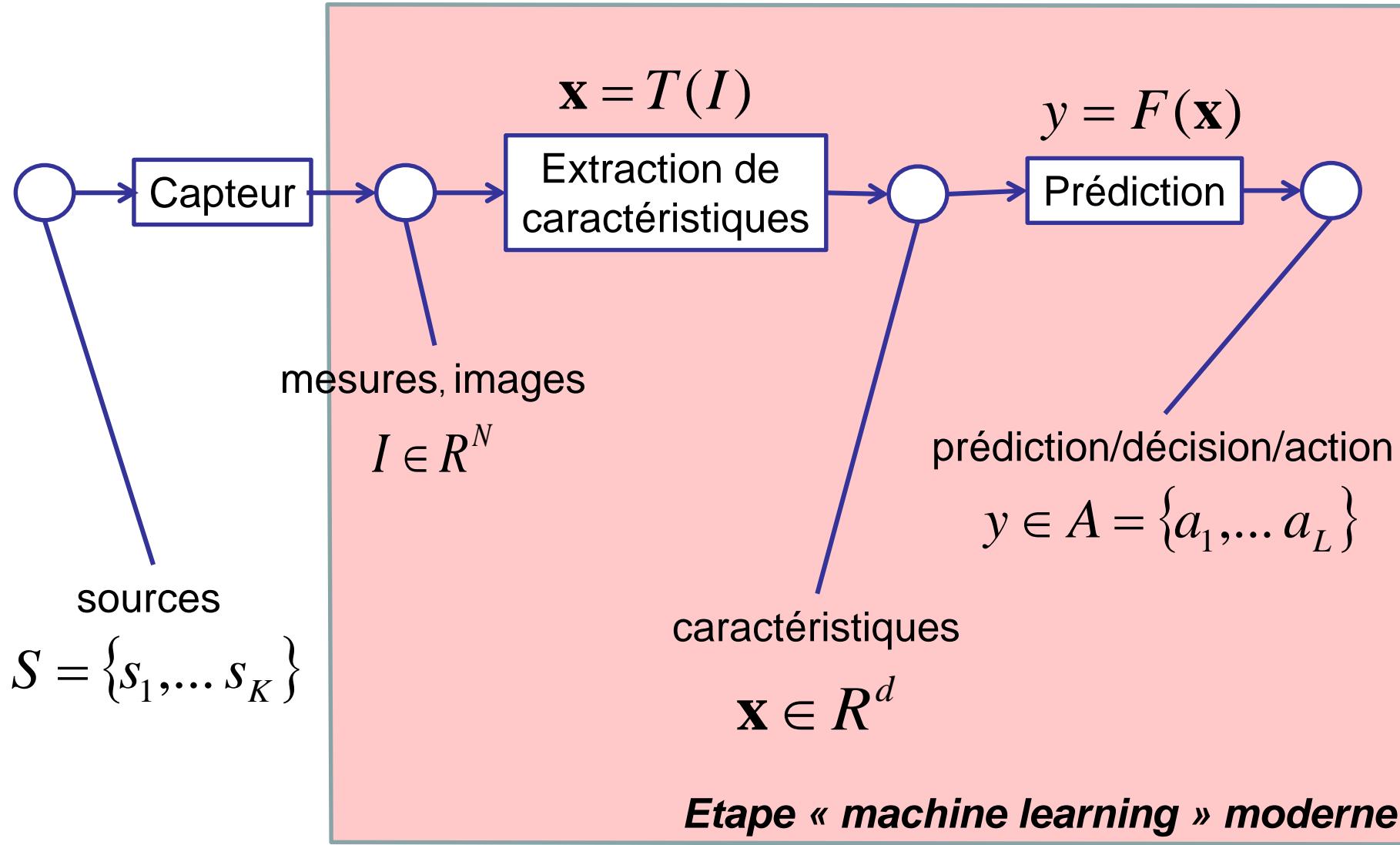


Exemples de caractéristiques en image

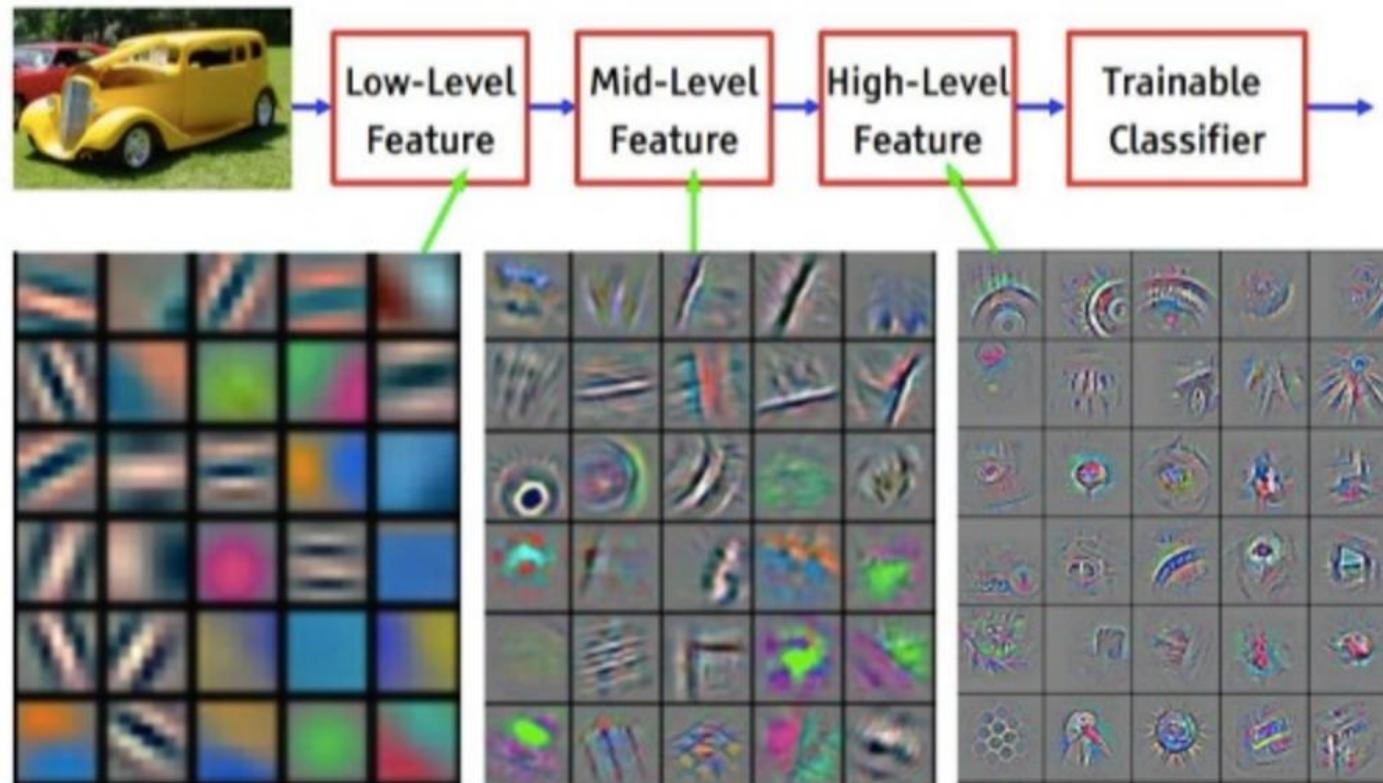
- Deux grandes classes: forme ou texture
- Forme
 - Dépend d'une étape de séparation du fond (segmentation, saillance)
 - Caractéristiques structurelles/géométriques
- Texture
 - Globales et/ou locales
 - Plus difficiles à associer à un objet précis



Prédire & extraire en même temps



« Deep features »



On peut apprendre les caractéristiques image
Réseaux convolutifs (cours DL)

ML POUR CLASSIFICATION

Modélisation probabiliste

Classification

- Classification = choisir **une** hypothèse parmi un ensemble donné
- Aujourd'hui: classification **binaire**:
 - Deux hypothèses à séparer: vrai/faux, malade/sain...
 - Rend compte de multiples situations pratiques de décision
 - Théorie plus facile à développer
- Deux stratégies de conception:
 - Modéliser les données pour décider: approche **générationnelle**
 - Décider directement à partir des données: approche **discriminative**
- Base statistique et probabiliste

Formalisme (suite)

- Fonction de décision (ou de prédiction):

$$y = F(\mathbf{x}; \mathbf{W})$$

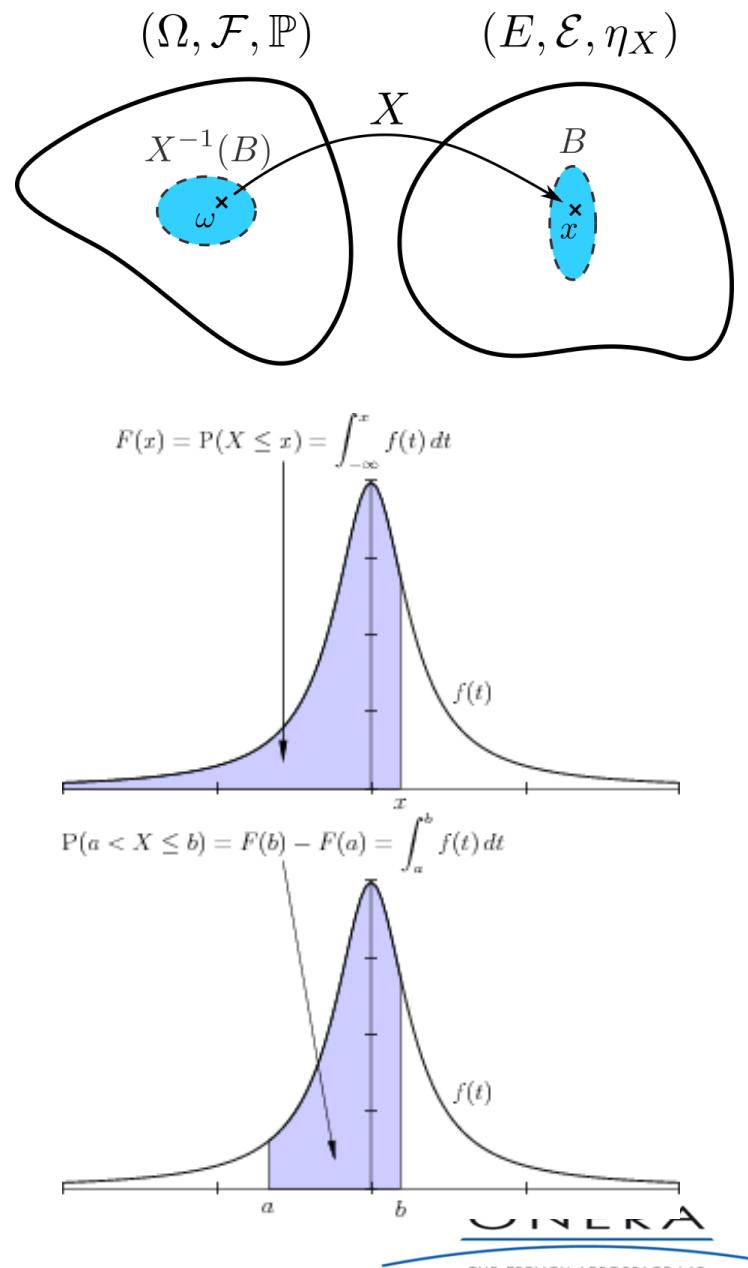
$$\mathbf{x} \in \mathbb{R}^d, y \in \{0,1\} \text{ ou } \{1,2 \dots C\}$$

Deux visions:
fonctions vs. probabilités

- On considère les données \mathbf{x}, y comme des **variables aléatoires**
- Base d'apprentissage: échantillons $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ de la loi $P(\mathbf{x}, y)$
- Les échantillons sont Indépendants Identiquement Distribués (i.i.d.)

Probabilités / statistiques (rappels)

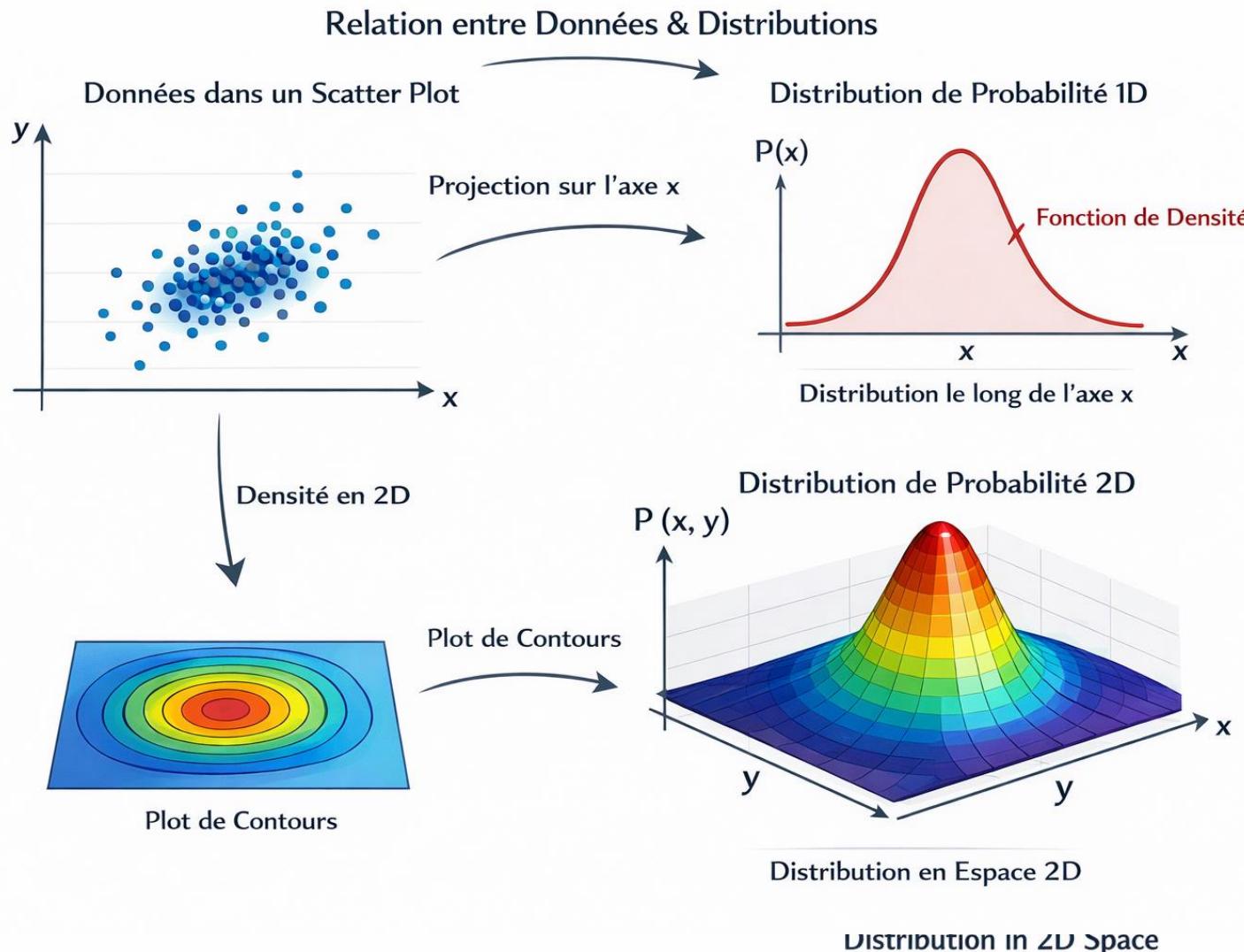
- Variable aléatoire
 - Fonction X d'un espace probabilisé vers un espace mesurable
 - Mesure de probabilité: $\mathbb{P}_X(B) = \mathbb{P}[X^{-1}(B)] = \mathbb{P}(X \in B)$
 - En pratique, $E = \mathbb{R}^D$ (continu) ou $\{1, \dots, C\} \in \mathbb{N}$ (discret)
- Distribution de probabilité (cas continu)
 - Fonction de répartition: $F_X(x) = \mathbb{P}(X \leq x)$
 - Densité de probabilité: $f_X(x) = \frac{dF_X(x)}{dx}$
- Mesures
 - Espérance mathématique: $E_X[g(X)] = \int g(x)f_X(x)$
 - Moyenner des phénomènes ou des quantités
 - Moments: Moyenne, variance... $E_X[X^p] = \int_{-\infty}^{+\infty} x^p f_X(x)$
 - Résumer/analyser la distribution



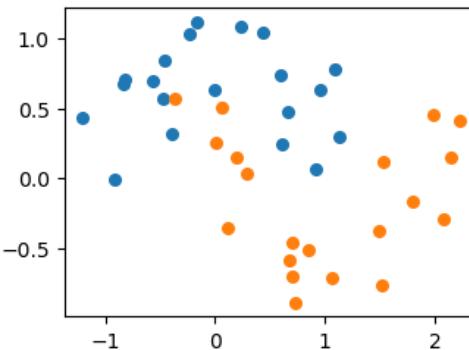
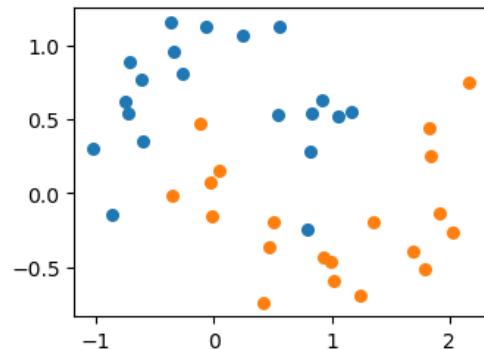
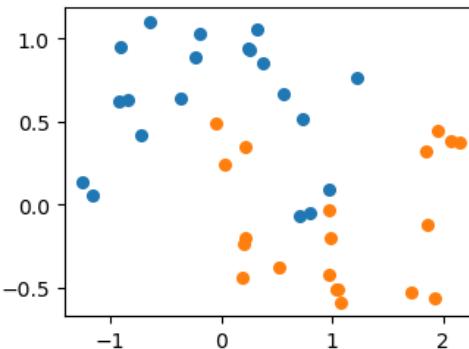
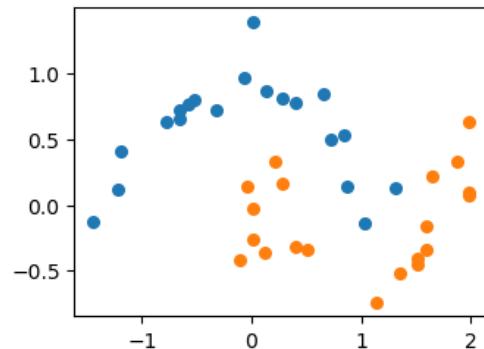
Combiner/associer des variables aléatoires

- Plusieurs types de lois de probabilités:
 - $P(x, y)$: loi jointe
 - $P(x)$: loi a priori sur les données (« comment est le monde »)
 - $P(y)$: loi a priori sur les prédictions (« quel sens donner au monde »)
 - $P(x | y)$: vraisemblance conditionnelle
 - $P(y | x)$: loi a posteriori
- Conditionnement
 - $P(x, y) = P(x|y)P(y) = P(y|x)P(x)$
 - Permet de simplifier la modélisation en décomposant la loi jointe
- Indépendance
 - Générale: $P(x, x') = P(x)P(x')$
 - Conditionnelle: $P(x, x'|y) = P(x|y)P(x'|y)$
 - Hypothèses de modélisation, mais potentiellement vérifiable empiriquement (en fait rarement)

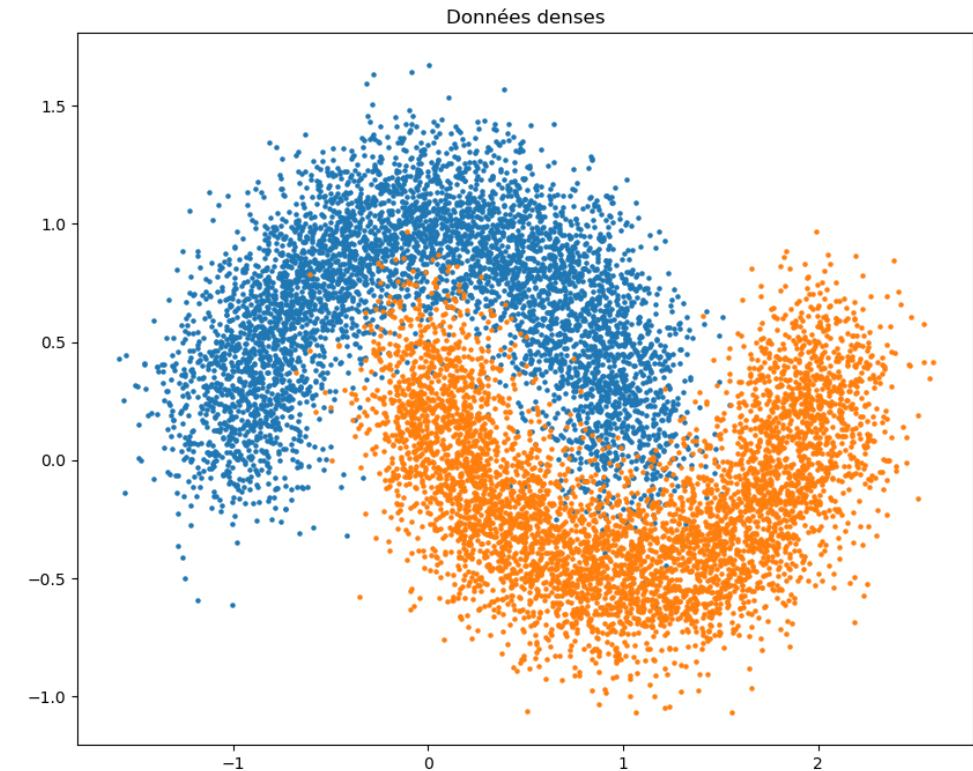
Données et distribution de probabilité



Echantillons et variance (d'apprentissage)



Quatre tirages différents (50 échantillons)



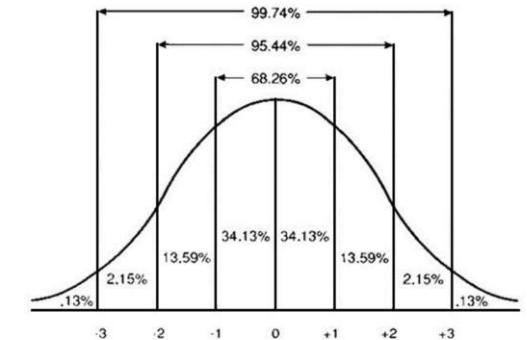
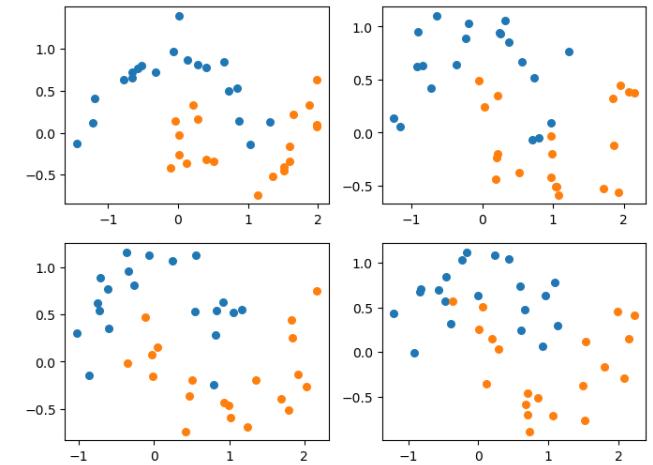
Un tirage (10000 échantillons)

Probabilités / statistiques (rappels)

- Données
 - La « matière » originelle du ML: pas de structure ou d'organisation
 - Hypothèse = échantillons d'une distribution (que l'on imagine exister...)
 - Train et test: deux ensembles d'échantillons de la même distribution

A partir de la modélisation probabiliste...

- Estimation de paramètres
 - Maximum de vraisemblance: meilleure explication
 - Loi des grands nombres: convergence vers la valeur moyenne
- Incertitude: relier quantités estimées, modèles probabilistes et données
 - Intervalles de confiance: compromis précision et confiance
 - Inégalités de concentration (Markov, Chebyshev, Chernoff, Hoeffding...): contrôler les écarts aux estimations
 - Tests statistiques: mesurer la validité d'une hypothèse



$$\mathbb{P}(X \geq t) \leq E_X[X]/t$$

$$\mathbb{P}\left(\left|\frac{1}{n} \sum_i X_i - E_X[X]\right| \geq t\right) \leq 2 \exp(-2nt^2)$$

Stratégie 1: Modéliser les données pour décider

1. Trouver une représentation des données (modèle paramétrique)
2. Construire une « bonne » fonction de prédiction à partir de ce modèle

Apprentissage = estimation des paramètres du modèle à partir de données

- Une « bonne » fonction de prédiction dépend de la nature du modèle et de l'impact potentiel des erreurs
- Modèle probabiliste: on peut générer (échantillonner) des données
→ On parle d'approche « générative » (attention: différent de l'IA générative des médias!)
- On peut considérer le problème comme une estimation de la loi jointe $P(x, y)$ ou des vraisemblances $P(x|y)$ et loi a priori $P(y)$.

Théorie Bayésienne de la décision

- Classification: $y \in \{1, 2, \dots, C\}$ est une étiquette (classe)
- On cherche à prédire une unique hypothèse y^* à partir de x
$$F: x \mapsto y^*$$
- On définit une fonction qui caractérise le coût (ou risque) de mauvaise décision: $l(y, y^*)$
- On cherche à **minimiser** son espérance, i.e. sa valeur moyenne:

$$E_{x,y}[l(y, F(x))] = \int_{x,y} l(y, F(x))P(x, y)$$

- Un coût usuel est $l(y, y') = 1 - 1_{y=y'}$ (risque 0/1 = comptage des erreurs)
- On peut montrer que la fonction de **décision optimale** est le « **classifieur bayésien** »:

$$F(x) = \arg \max_y P(y | x)$$

Remarque: la démonstration repose sur la factorisation:

$$E_{x,y}[l(y, F(x))] = \int_x P(x) \int_{y|x} l(y, F(x))P(y|x)$$

Théorie Bayésienne de la décision

- Deux questions:
 - Comment calculer $P(y | x)$ = apprentissage
 - Comment trouver le max = prédiction
- « Astuce »: utiliser la loi de Bayes

$$P(y | x) = \frac{P(x | y) P(y)}{P(x)}$$

- On connaît en général la fréquence d'occurrence des classes y
- On sait plus facilement calculer la **vraisemblance**: $P(x | y)$
 - « Si je sais dans quelle classe je suis, je sais décrire le comportement/distribution de mes données »
- Le max sur y ne dépend que de $P(x | y)$ et $P(y)$

$$F(x) = \arg \max_y P(x | y)P(y)$$

Exemple : Modèle gaussien multivarié

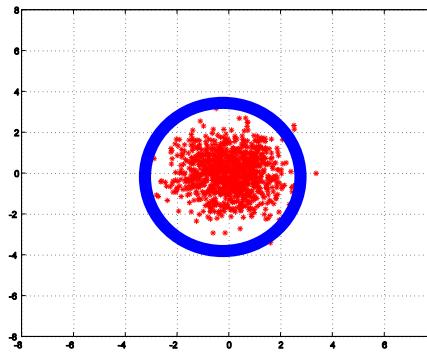
- Un exemple élémentaire (mais utile) de vraisemblance:

$$P(x; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} \sqrt{|\Sigma|}} \exp \left[-\frac{1}{2} (x - \mu)^t \Sigma^{-1} (x - \mu) \right]$$

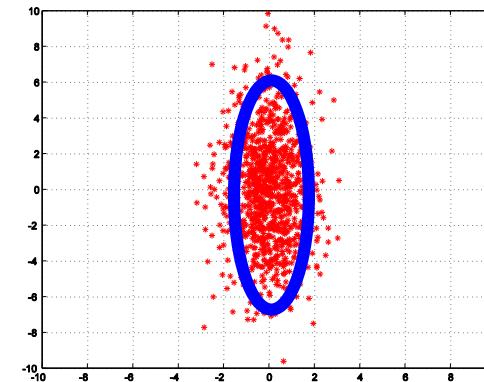
où $x = [x_1, x_2 \dots x_d] \in \mathbb{R}^d$

- Permet de décrire les corrélations entre dimensions (moments d'ordre 2).

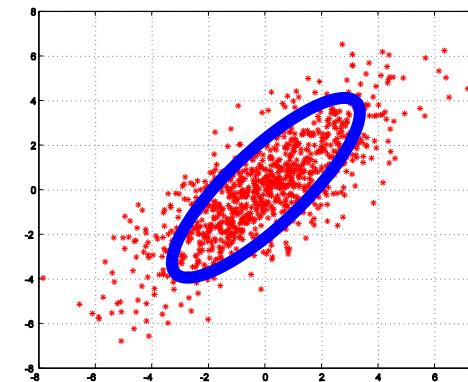
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 9 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix} = R \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix} R^{-1}$$



« Apprentissage » du modèle gaussien

- Log-vraisemblance d'un échantillon $X = \{x_1, x_2 \dots x_N\}$

$$\log P(X; \mu, \Sigma) = cste - \frac{N}{2} \log |\Sigma| - \sum_{i=1}^N (x_i - \mu)^t \Sigma^{-1} (x_i - \mu)$$

- L'estimateur du maximum de vraisemblance donne:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N x_i$$

et

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{\mu})(x_i - \hat{\mu})^t$$

- Remarques:

1. L'estimateur de la covariance est biaisé. On normalise en général par $\frac{1}{N-1}$.
2. On peut définir des estimateurs plus robustes qui gèrent les données aberrantes

Construction de la décision

- Problème à deux classes: décider consiste à calculer le signe du log-ratio!

$$\log \frac{P(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) P(y_1)}{P(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) P(y_2)} \geq 0$$

- En développant, on obtient une fonction de décision de la forme:

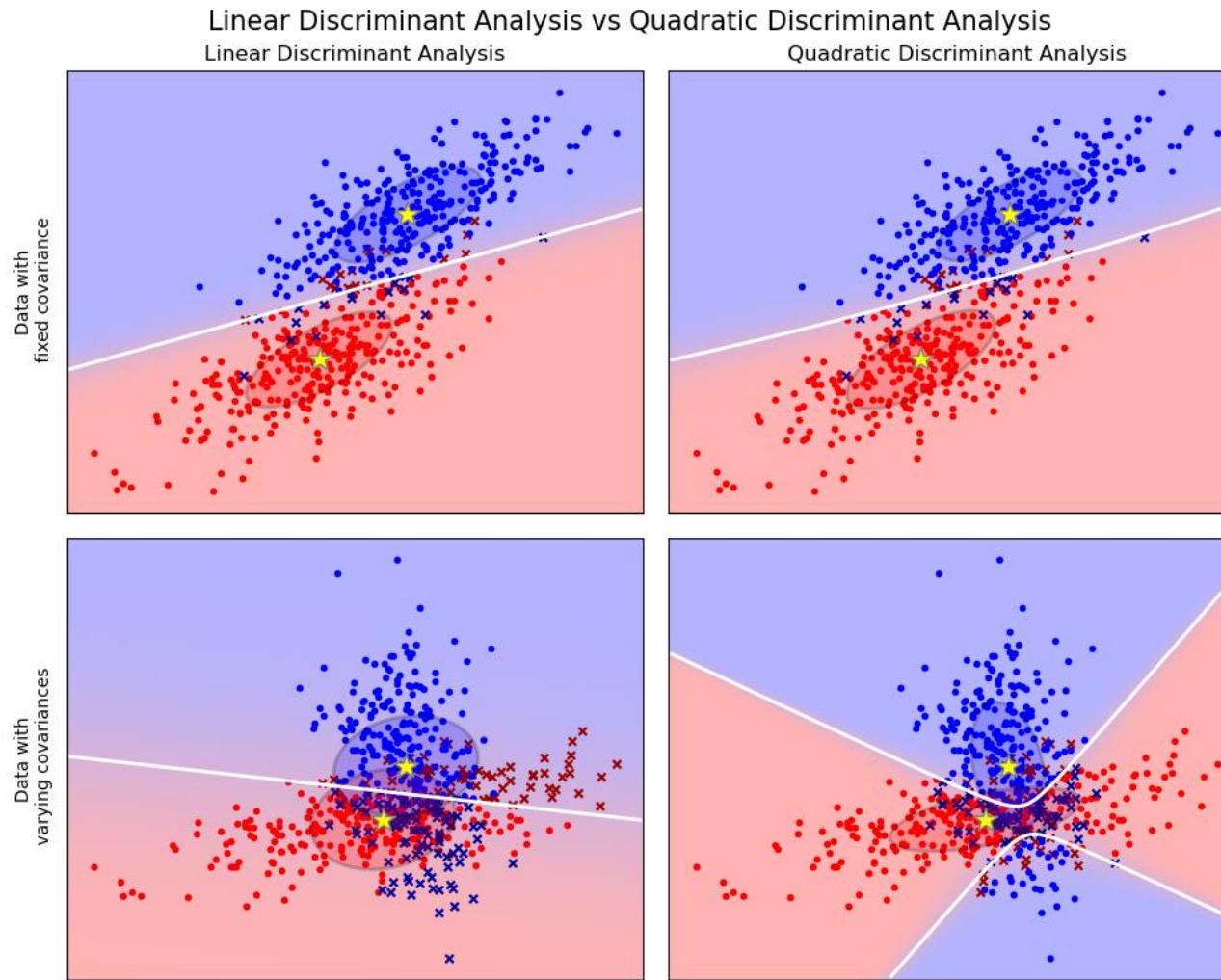
$$(\mathbf{x} - \boldsymbol{\mu}_1)' \boldsymbol{\Sigma}_1^{-1} (\mathbf{x} - \boldsymbol{\mu}_1) - (\mathbf{x} - \boldsymbol{\mu}_2)' \boldsymbol{\Sigma}_2^{-1} (\mathbf{x} - \boldsymbol{\mu}_2) + cste \geq 0$$

- C'est une forme quadratique.
- Remarque: si l'on constraint les deux populations à avoir la même covariance $\boldsymbol{\Sigma}$, on obtient une forme linéaire:

$$\mathbf{x}' \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + cste \geq 0$$

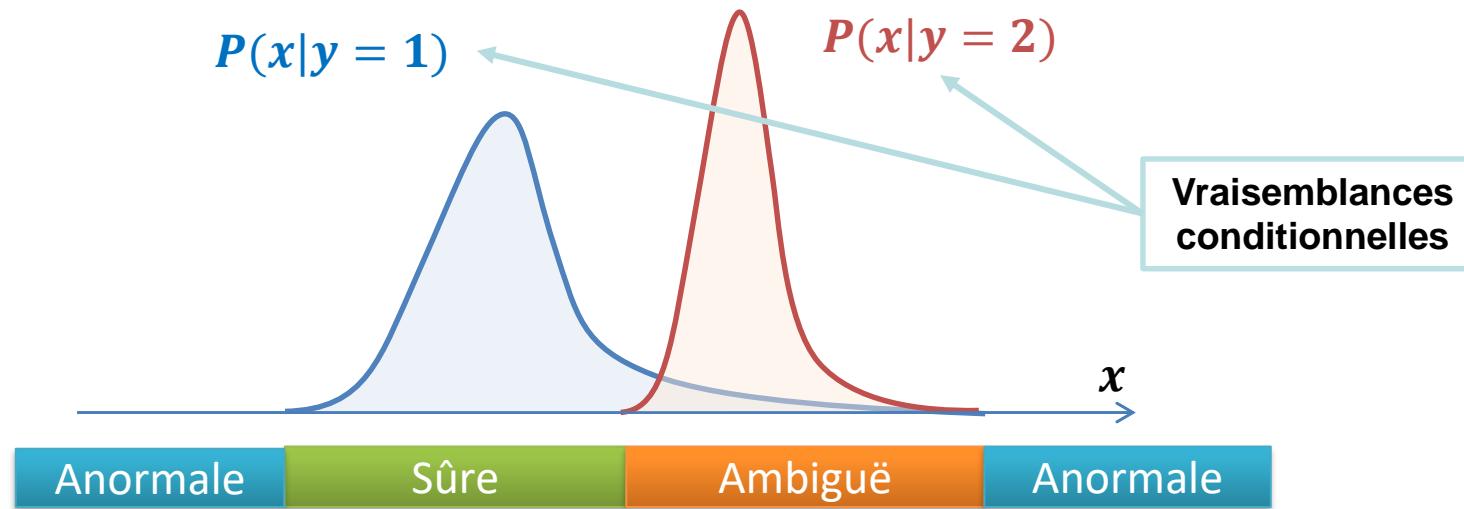
- On parle de « séparatrice »: la forme (quadratique ou linéaire) sépare l'espace en deux zones.

Exemple en 2D



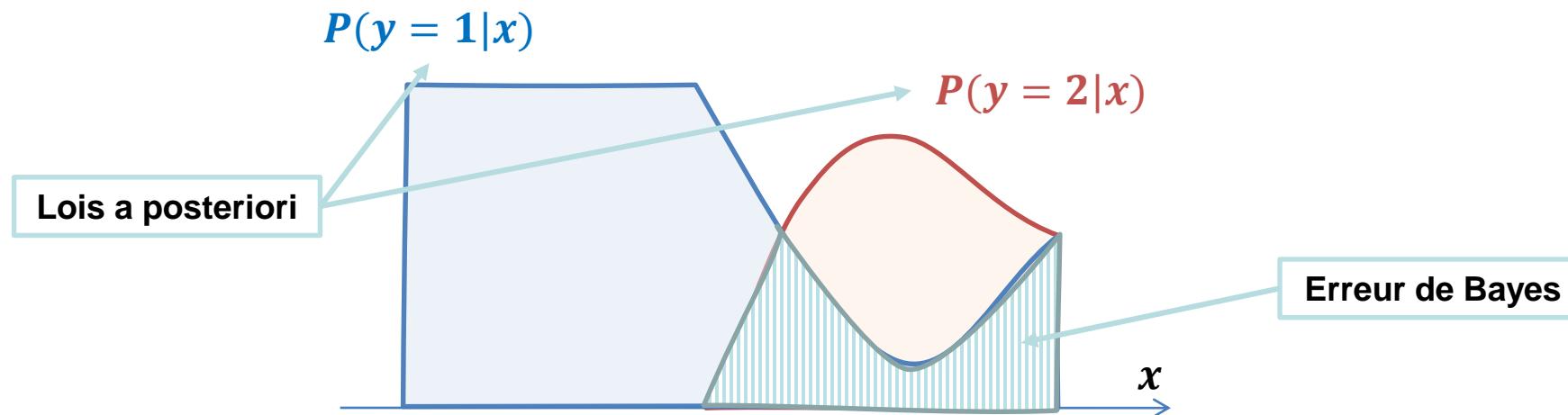
https://scikit-learn.org/stable/modules/lda_qda.html

Différentes sources d'erreur



- Certaines données sont ambiguës: on ne peut prendre de décision sûre sur la classe.
- Certaines données sont « impossibles » et sont considérées comme anormales si elles surviennent: elles sont « hors support ».

Une erreur incompressible

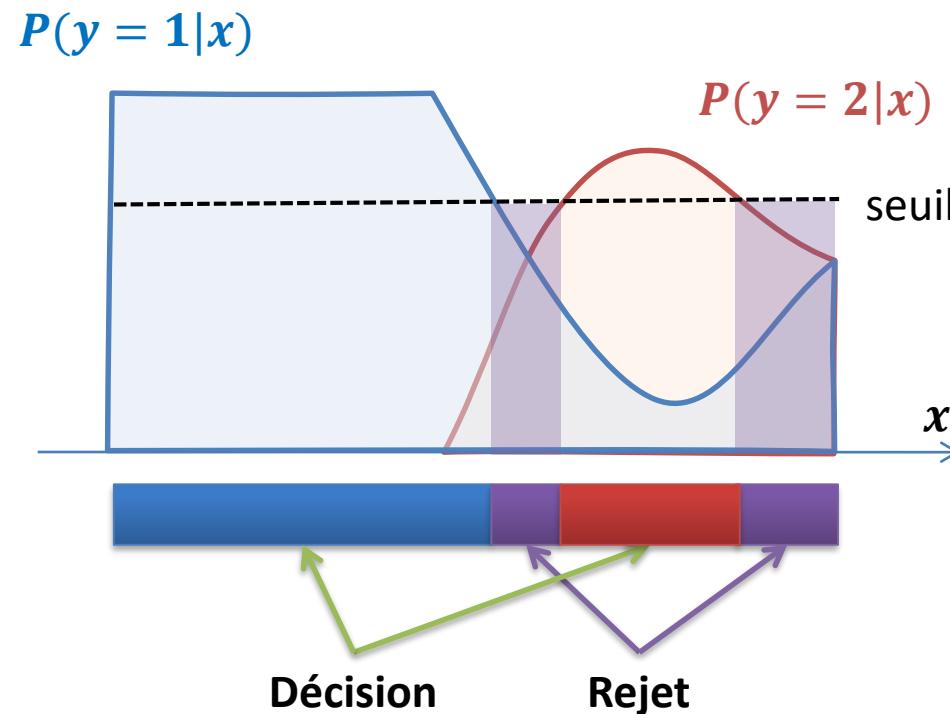


- Pour une décision bayésienne (argmax sur la loi a posteriori), l'erreur est alors

$$\text{error}_{\text{Bayes}} = E_x[1 - \max_k P(y = k|x)]$$

- C'est l'erreur minimale que peut réaliser un prédicteur.
- Rem: c'est un concept théorique, difficile à calculer en pratique

Rejet sur loi a posteriori



- On peut contrôler le processus de décision en seuillant sur la loi a posteriori, et en décidant de ne pas décider si en dessous du seuil (« rejet »).
- Intérêt: on diminue le taux d'erreur mais on décide moins souvent.

Remarques sur la modélisation bayésienne gaussienne

- Dans des espaces de grandes dimensions, il n'est pas possible d'estimer correctement la covariance → modèle gaussien imprécis.
- *Peut-on simplifier la modélisation des données?*
- *Approche bayésienne naïve*
- On peut souvent faire comme si la distribution était gaussienne même si ce n'est pas vrai: ce qui importe est de trouver une bonne fonction de décision (dans TD)
- *Peut-on trouver la fonction de décision sans modélisation des données?*
- *Approche discriminative*

Approche Bayésienne Naïve

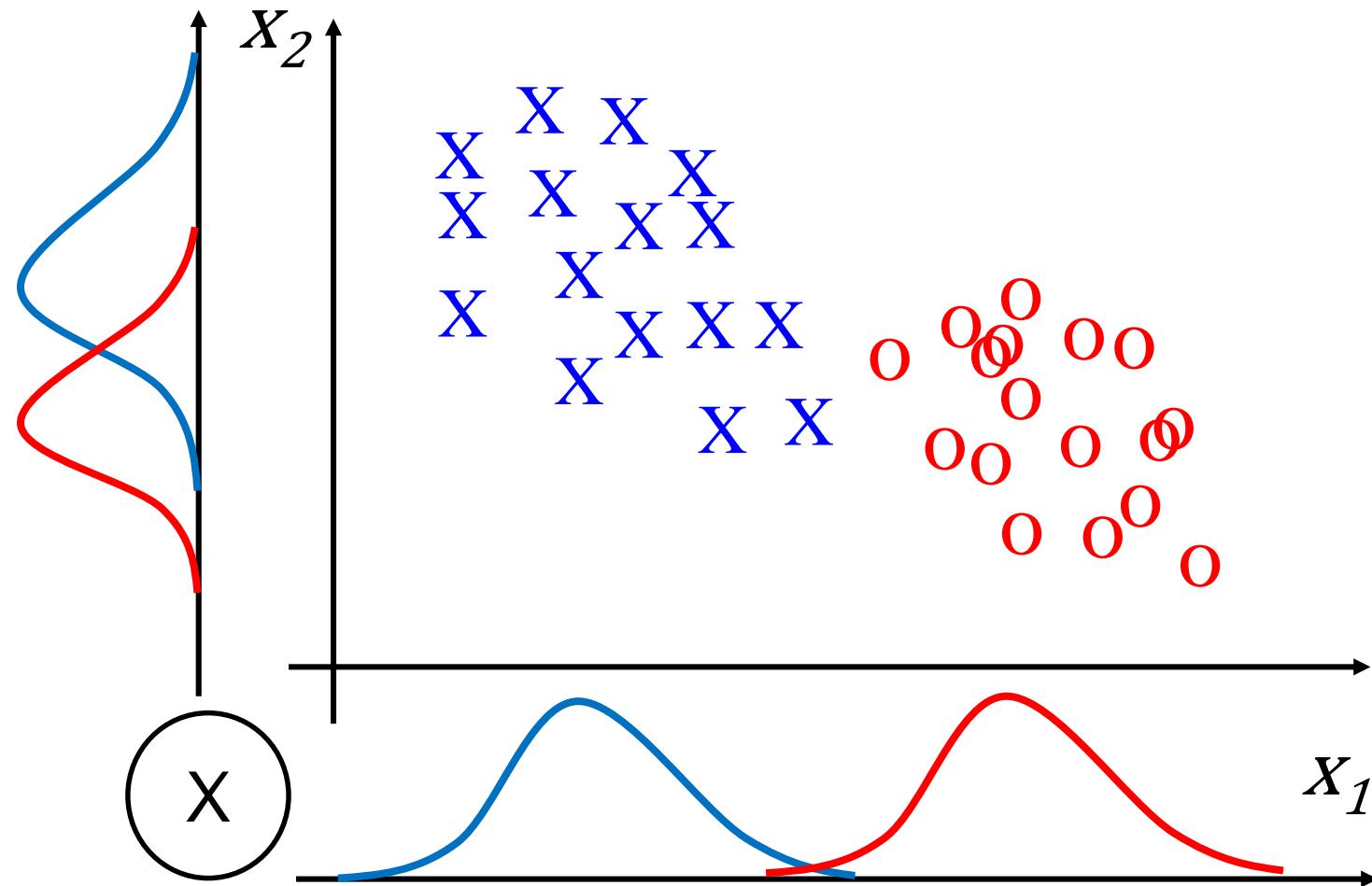
- Que faire pour des espaces à grande dimension?
- Calcul de la loi conditionnelle: hypothèse d'indépendance conditionnelle.

$$\begin{aligned} P(x_1, x_2 \dots x_d | y) &= P(x_1 | x_2 \dots x_d, y)P(x_2 \dots x_d | y) \\ &= P(x_1 | y)P(x_2 \dots x_d | y) \\ &= P(x_1 | y)P(x_2 | y) \dots P(x_d | y) \end{aligned}$$

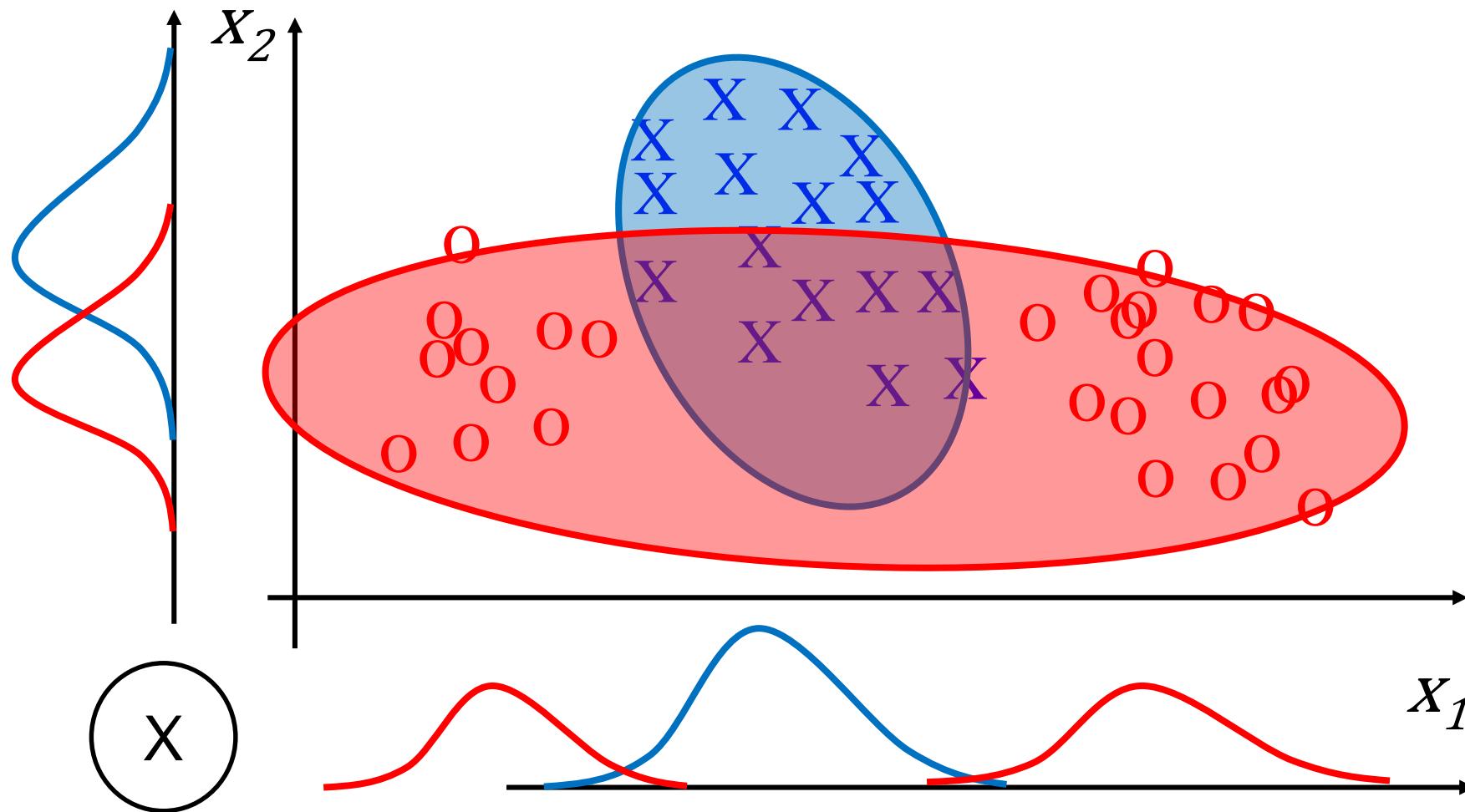
- On calcule la vraisemblance globale dimension par dimension
- ➔ Problème 1D, modèles plus faciles à estimer (gaussien, binomial, histogrammes, mélange de gaussiennes...)
- En pratique, on calcule plutôt la log-vraisemblance pour des questions de stabilité numérique

$$\begin{aligned} \log P(\mathbf{x}|y) &= \sum_i \log P(x_i|y) \\ y^* &= \arg \max_y \log P(\mathbf{x} | y) + \log P(y) \end{aligned}$$

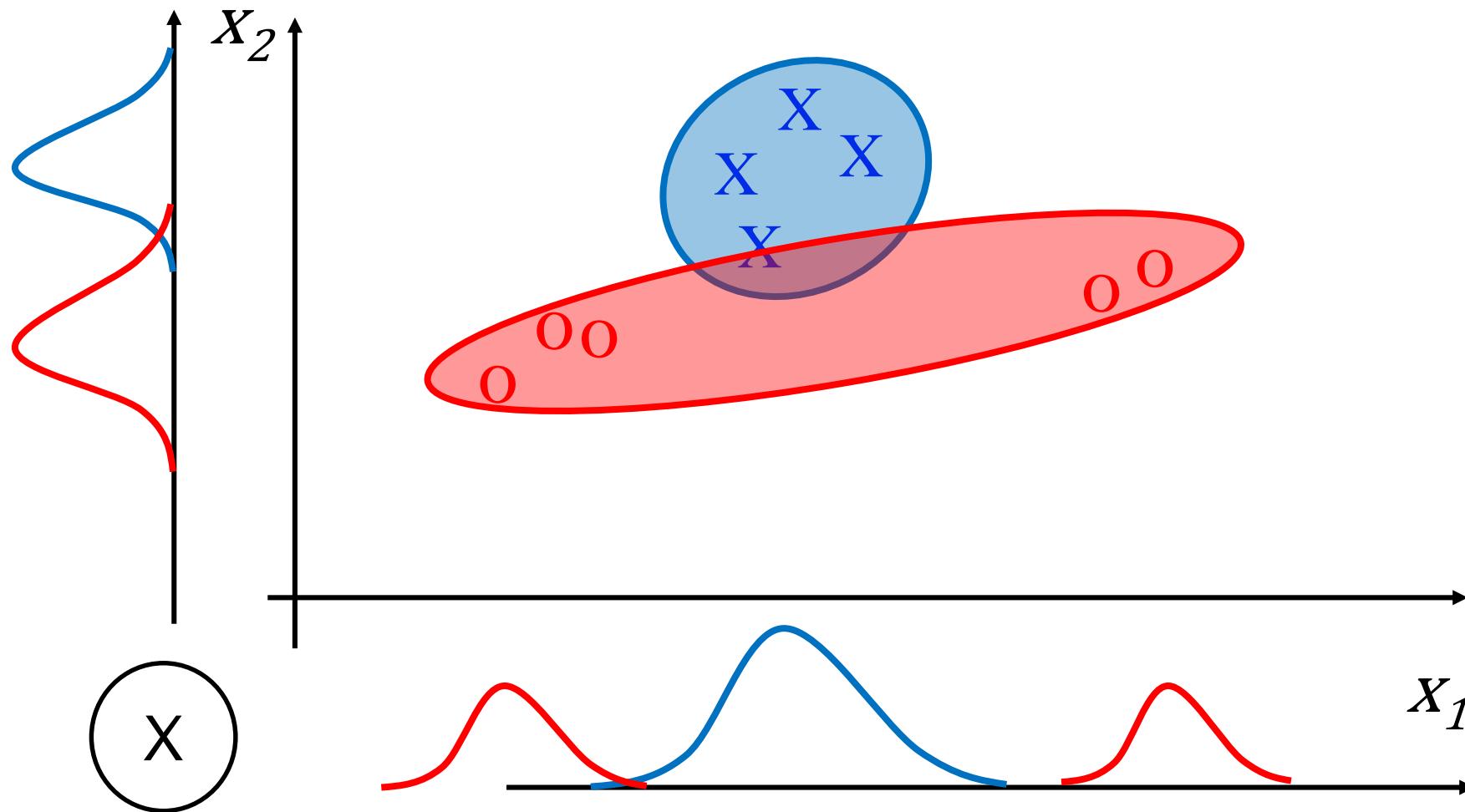
Approche bayésienne naïve



Approche bayésienne naïve vs. multivariée



Approche bayésienne naïve vs. multivariée



Approche bayésienne: résumé

- Théorie probabiliste de la décision → calcul de la loi a posteriori
- Expression de la loi a posteriori:
 - Modèle gaussien multivarié
 - Hypothèse d'indépendance conditionnelle.
- Apprentissage
 - Estimation de lois paramétriques simples
- Prédiction
 - Calcul de log-vraisemblance et max sur hypothèses
- Quand l'utiliser? (limitations)
 - Petits problèmes bien modélisés (gaussien multivarié)
 - Caractéristiques non corrélées (bayésien naïf, mais ça peut aussi marcher si c'est corrélé)

Stratégie 2: Décider/prédire directement

1. Construire une « bonne » fonction de prédiction à partir des données

Apprentissage = estimation des paramètres de la fonction de décision

- Une « bonne » fonction de prédiction dépend de l'impact potentiel des erreurs
- Modèle statistique: on peut produire des incertitudes de décision, mais pas échantillonner des données
→ On parle d'approche « discriminante »
- On peut considérer le problème comme une estimation de la loi a posteriori $P(y|x)$.

Discrimination linéaire binaire

Un exemple simple d'approches discriminante

Principe

- On cherche à projeter linéairement sur une droite les données de telle sorte qu'elles soient séparées au mieux
- Formellement, on définit la décision par un vecteur w et un biais b :

$$F(x) = (\mathbf{w}^t \cdot \mathbf{x} + b) \begin{cases} 1 & \text{if } F(x) \geq 0 \\ 2 & \text{if } F(x) < 0 \end{cases}$$

- Le signe de la fonction donne la classe
- Plusieurs stratégies pour définir une « bonne » projection linéaire:
 - moindre carrés
 - régression logistique
 - discrimination de Fisher

Moindres carrés

- Première idée: on cherche à prédire directement la classe et on minimise l'erreur quadratique (régression):

$$J(\mathbf{w}, b) = \frac{1}{N} \sum_i (\mathbf{w}^t \cdot \mathbf{x}_i + b - y_i)^2$$

- Qui peut se récrire:

$$J(\mathbf{W}) = \|\mathbf{W} \cdot \mathbf{X} - \mathbf{Y}\|^2$$

avec $\mathbf{W} = [\mathbf{w}, b]$ et $\mathbf{X}_i = [\mathbf{x}_i, 1]$

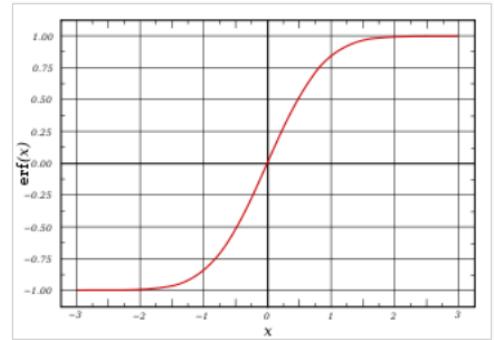
- La solution analytique est celle des moindres carrés:

$$\mathbf{W}^* = (\mathbf{X}^t \cdot \mathbf{X})^{-1} \mathbf{X}^t \mathbf{Y}$$

- Conceptuellement simple mais:

- Demande d'inverser une matrice de corrélation: donc très coûteux (en pratique on ne calcule pas les poids comme ça)
- Très sensible aux données aberrantes (« outliers »)

Régression logistique



- Principe: on modélise directement la **loi a posteriori** selon:

$$P(y | \mathbf{x}) = \sigma(\mathbf{w}^t \mathbf{x} + b) \quad \text{où} \quad \sigma(a) = \frac{1}{1 + \exp(-a)}$$

- Approximation « lisse » de la fonction créneau
- On cherche les paramètres \mathbf{w} et b qui maximisent la log-vraisemblance:

$$\begin{aligned} J(\mathbf{w}, b) &= -\sum_{n=1}^N \ln p(y_n | z_n) \\ &= -\sum_{n=1}^N y_n \ln z_n + (1 - y_n) \ln (1 - z_n) \end{aligned}$$

↑ ↑
si $y = 1$ si $y = 0$

avec

$$z_n = \sigma(\mathbf{w}^t \mathbf{x}_n + b)$$

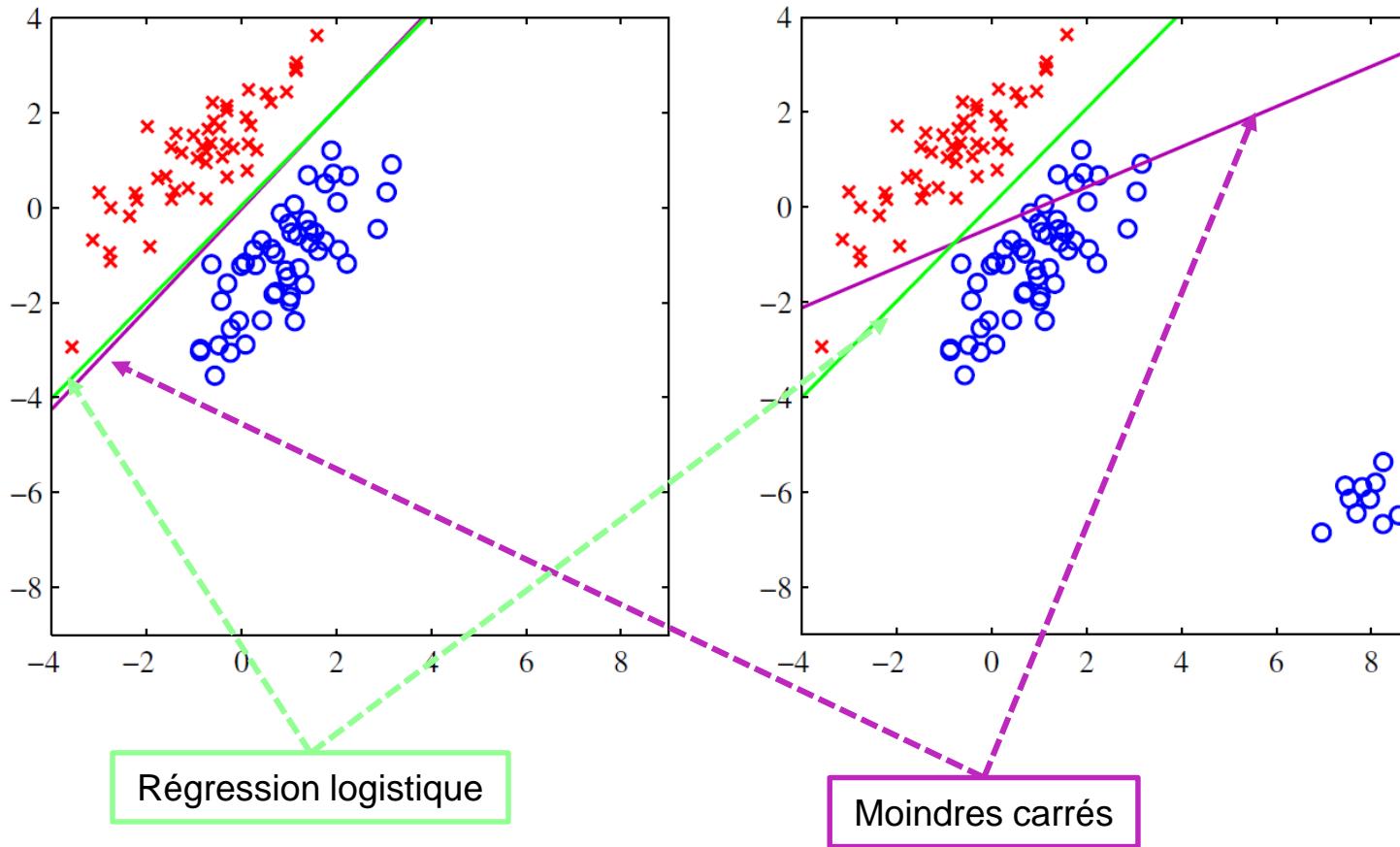
Optimisation de la régression logistique

- La minimisation du critère n'admet pas formulation analytique
- Astuce: Le gradient du critère se calcule facilement

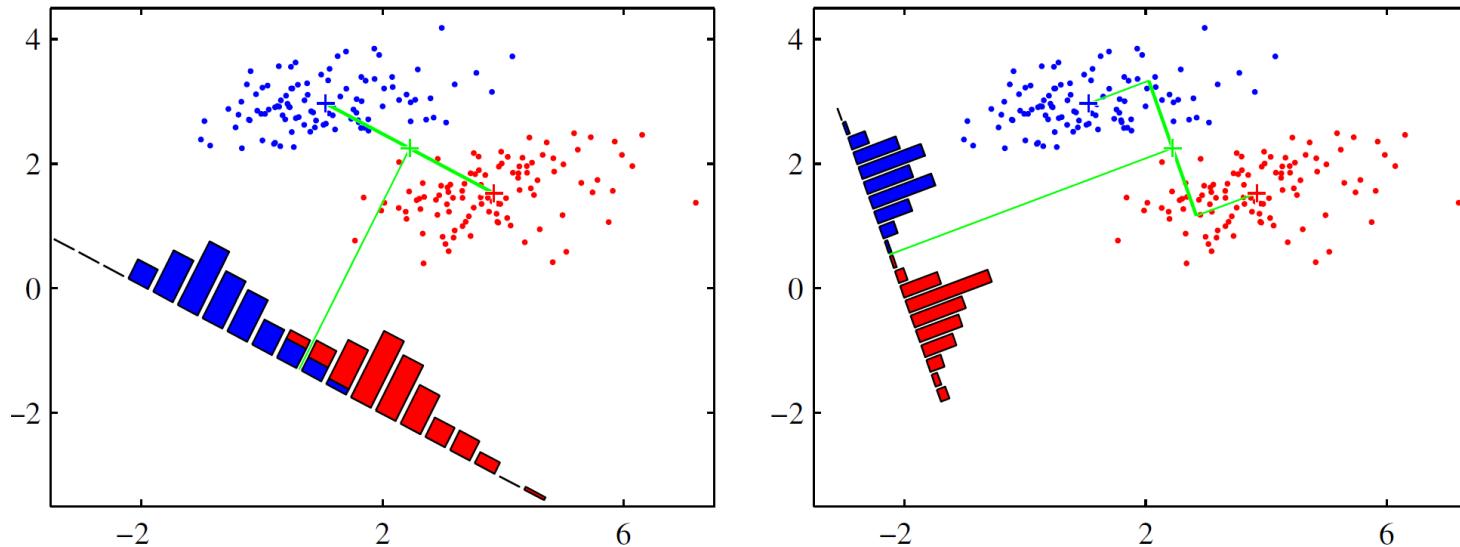
$$\frac{\partial J_n}{\partial \mathbf{w}} = (z_n - y_n) \mathbf{x}_n$$

- On utilise une descente de gradient pour minimiser.
- Il existe une version au second ordre (Newton) conduisant à un algorithme des moindres carrés pondérés.
- La direction de prédiction est moins sensible aux données aberrantes.

Sensibilité aux « outliers »



Analyse discriminante linéaire (Fisher)



Principe: maximiser le contraste entre deux populations projetées selon le critère de Rayleigh

$$J = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

Où m_k et s_k sont les moyennes et variances 1-D des deux populations à contraster.

Analyse discriminante linéaire (Fisher)

- Lorsque l'on projette selon $\mathbf{w}^t \cdot \mathbf{x} + b$, le critère s'exprime selon:

$$J(\mathbf{w}) = \frac{\mathbf{w}^t \mathbf{S}_B \mathbf{w}}{\mathbf{w}^t \mathbf{S}_I \mathbf{w}}$$

où

$$\mathbf{S}_B = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^t \cdot (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$$

est la matrice de covariance inter-classe et

$$\mathbf{S}_I = \sum_{i \in C_1} (\mathbf{x}_i - \boldsymbol{\mu}_1)(\mathbf{x}_i - \boldsymbol{\mu}_1)^t + \sum_{i \in C_2} (\mathbf{x}_i - \boldsymbol{\mu}_2)(\mathbf{x}_i - \boldsymbol{\mu}_2)^t$$

est la matrice de covariance intra-classe.

- On peut montrer que la direction qui maximise le contraste est colinéaire à:

$$\mathbf{w}^* = \mathbf{S}_I^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$$

C'est aussi le vecteur propre de la plus grande valeur propre de $\mathbf{S}_I^{-1} \cdot \mathbf{S}_B$.

Discrimination linéaire: résumé

- Décision réduite à rechercher une direction de projection
- Plusieurs algorithmes pour apprendre directement la direction:
 - Moindres carrés
 - Régression logistique
 - Discrimination de Fisher
- Quand l'utiliser? (limitations)
 - Petites dimensions
 - Distributions monomodales
- Remarques:
 - On peut retrouver une surface quadratique en augmentant l'espace de représentation par des combinaisons polynomiales (cf. TD)
 - On peut définir des versions « multi-classe »
 - On verra comment mieux contrôler le calcul et la forme d'une surface séparatrice (SVM)

A retenir

- « Programmer à partir des données »
 - Deux phases: apprentissage et prédition
 - Plusieurs variétés de prédicteurs et d'apprentissage
- Démarche générique:
 - Constitution d'une base d'apprentissage
 - Analyse préliminaire des données + préparation
 - Conception du modèle
 - Optimisation
 - Evaluation
- Deux approches élémentaires:
 - Modélisation bayésienne: approche « générative »
 - Analyse discriminante linéaire: approche « discriminante »

Références

- R.O. Duda and P.E. Hart, Pattern classification and scene analysis, John Wiley & Sons, New York, 1973.
- P.A. Devijver and J. Kittler, Pattern Recognition, a Statistical Approach, Prentice Hall, Englewood Cliffs, 1982)
- K. Fukunaga, Introduction to Statistical Pattern Recognition (Second Edition), Academic Press, New York, 1990.
- L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, Classification and regression trees, Wadsworth, 1984.
- S. Haykin, Neural Networks, a Comprehensive Foundation. (Macmillan, New York, NY., 1994)
- L. Devroye, L. Györfi and G. Lugosi, A Probabilistic Theory of Pattern Recognition, (Springer-Verlag 1996)
- V. N. Vapnik, The nature of statistical learning theory (Springer-Verlag, 1995)
- **C. Bishop, Pattern Recognition and Machine Learning, (<https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>).**
- Jerome H. Friedman, Robert Tibshirani et Trevor Hastie, The Elements of Statistical Learning: Data Mining, Inference, and Prediction (<https://web.stanford.edu/~hastie/ElemStatLearn/>).
- G. James, D. Witten, T. Hastie & R.Tibshirani, An Introduction to statistical learning, Springer Texts in Statistics (<https://www.statlearning.com/>)
- Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, An MIT Press book (<http://www.deeplearningbook.org>)
- B. Scholkopf, A. Smola, Learning with kernels, (MIT Press, 2001)
- Kevin Murphy, Machine Learning: a Probabilistic Perspective, (MIT Press, 2013)
- Hal Daumé III, A Course in Machine Learning (<http://ciml.info/>)
- Francis Bach, Learning Theory from First Principles (https://www.di.ens.fr/~fbach/ltp_book.pdf)

Bases de données

- UCI Repository: <http://www.ics.uci.edu/~mlearn/MLRepository.html>
- UCI KDD Archive: <http://kdd.ics.uci.edu/summary.data.application.html>
- Statlib: <http://lib.stat.cmu.edu/>
- Delve: <http://www.cs.utoronto.ca/~delve/>
- Kaggle: <https://www.kaggle.com/>
- Benchmarks (Vision):
 - ImageNet: <http://image-net.org/>
 - MS COCO: <http://cocodataset.org/>
 - MNIST et plus: http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html
 - CV on line: <https://computervisiononline.com/datasets>
 - Kitti: <http://www.cvlibs.net/datasets/kitti/>
 - Waymo: <https://waymo.com/open>

Journaux

- Journal of Machine Learning Research www.jmlr.org
- Machine Learning
- Neural Computation
- Neural Networks
- IEEE Transactions on Neural Networks
- IEEE Transactions on Pattern Analysis and Machine Intelligence
- Annals of Statistics
- Journal of the American Statistical Association
- ...

Conférences

- International Conference on Machine Learning (ICML)
- European Conference on Machine Learning (ECML)
- Neural Information Processing Systems (NIPS)
- International Conference on Learning Representations (ICLR)
- Uncertainty in Artificial Intelligence (UAI)
- International Joint Conference on Artificial Intelligence (IJCAI)
- International Conference on Neural Networks (ICNN)
- Conference of the American Association for Artificial Intelligence (AAAI)
- IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
- European Conference on Computer Vision (ECCV)
- International Conference on Computer Vision (ICCV)
- IEEE International Conference on Data Mining (ICDM)
- ...

Cours & tutoriaux

- Des MOOC (Français et Anglais)
- Des tutoriaux associés aux conférences (orientés recherche)
- Des cours en français:
 - <https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle>
 - https://www.college-de-france.fr/site/stephane-mallat/_course.htm
- Des « cheat sheets »
 - <https://stanford.edu/~shervine/teaching/>

Logiciels

- Environnement génériques: Matlab, ScikitLearn
- Environnements Deep Learning: Tensor Flow, Pytorch, mxnet...
- Beaucoup de codes sur GitHub



Le TD1

- Partie 1: Les approches élémentaires sur une première base
 - Programmation Python
 - Application de la démarche
- Partie 2: Utilisation de la bibliothèque scikit-learn
 - Les approches sur une autre base

Utilisation de Colab

- Environnement de développement Python (Notebook « .ipynb »)
- Ressources de calcul distantes (GPU)
- C'est proposé par Google
- <https://colab.research.google.com/>

Etapes

- Se créer un gmail (ou utiliser le votre)
- Se connecter à Colab
- Ouvrir le Notebook du TD (td_gaussien_bayesien.ipynb)
- Modifiez directement le notebook.

Utilisation des « Notebook Python »

The diagram shows a Jupyter Notebook interface in a Mozilla Firefox browser window. The notebook contains three main sections: Text, Code, and Exécution des cellules.

- Text:** A light blue box containing the text "après chaque fonction de visualisation."
- Code:** A light blue box containing the code for importing libraries and generating data.
- Exécution des cellules:** A light blue box containing the code for creating a classifier and processing data.

Annotations with arrows point from the labels to their corresponding parts in the notebook:

- An arrow points from the "Text" label to the "plt.show()" line in the first code cell.
- An arrow points from the "Code" label to the code in the second code cell.
- An arrow points from the "Exécution des cellules" label to the code in the third code cell.

```
[1]: # Librairies utiles standard
import numpy as np
import matplotlib.pyplot as plt

# Pour visualiser et récupérer les données
import iogs_td_util as td

# L'algorithme SVM dans la bibliothèque scikit-learn
from sklearn import svm

import random

In [2]: # Classifieur
svc = svm.SVC(kernel='linear', max_iter=-1)

# Premier jeu de données
trainX, trainY, testX, testY=td.generate_data(0)
```

Pour se mettre à jour sur Python & Numpy

- Intro à Numpy et Matplotlib
 - <https://sebastianraschka.com/blog/2020/numpy-intro.html>
 - <https://cs231n.github.io/python-numpy-tutorial/>
- « Cheat sheets »
 - <https://www.datacamp.com/community/data-science-cheatsheets>