

# Introduction au Machine Learning

Yann Chevaleyre, Paul Caillon

# Intervenants



Yann Chevaleyre

- Professeur des Universités @ Dauphine-PSL;
- `yann.chevaleyre@lamsade.dauphine.fr`



Paul Caillon

- Chercheur post-doctorant @ Dauphine-PSL;
- `paul.caillon@dauphine.psl.eu`

# Planning du cours

## Séance 1 : Jeudi 15 mai 18-21h (3 h) – Introduction et Définitions

- Définition du Machine Learning
- Types de données (numériques, catégorielles, textuelles, images)
- Jeux d'entraînement et de test : découpage, overfitting et underfitting
- définition, sur- et sous-apprentissage
- Approches non-supervisée vs supervisée : quelques exemple d'applications sur des données
- k-plus proches voisins: utilisation en classification, régression, estimation de densité
  - distance, choix de k, sensibilité aux outliers
  - Malédiction de la dimension : concentration des distances
  - sur- et sous-apprentissage en fonction de k

## Séance 2 : Jeudi 22 mai matin (3,5 h) – Apprentissage supervisé Linéaire

- Risque statistique et fonctions de perte : MSE, log-loss
- Modèles linéaires en régression+classification
  - Régression Linéaire
  - Classifieur Bayes naïf : hypothèse d'indépendance, calcul des probabilités
  - Régression logistique : modèle linéaire, sigmoïde, multiclasse (softmax), interprétation des coefficients
- Descente de Gradient

## Séance 3 : Jeudi 22 mai après-midi (3,5 h) – Apprentissage supervisé : Arbres de Décision + méthodes ensemblistes

- Decision Trees, interprétabilité des Decision Trees
- Bagging et Random Forest : bootstrap, agrégation d'arbres, importance des variables
- Boosting (AdaBoost, Gradient Boosting) : pondération séquentielle
- Comparaison des approches ensemblistes vs modèles simples
- Hyperparamètres méthodes ensemblistes

# Planning du cours

## Séance 4 : Mardi 27 mai 18-21h (3 h) – Réduction de Dimension

- Pourquoi réduire la dimension ?
  - pour visualiser
  - pour combattre le surapprentissage
- Réduction de Dimension
  - sélection de variables
  - PCA et SVD
  - LSA (Latent Semantic Analysis)
  - UMAP

## Séance 5 : Mardi 3 juin 18-21h (3 h) - Clustering

- Clustering
  - Clustering hiérarchique : distances, liens et dendrogrammes
  - K-means : principe, initialisation, choix de k, convergence
  - DBSCAN : notion de densité, paramètres  $\epsilon$  & min\_samples, gestion du bruit
  - Soft-Clustering (EM)

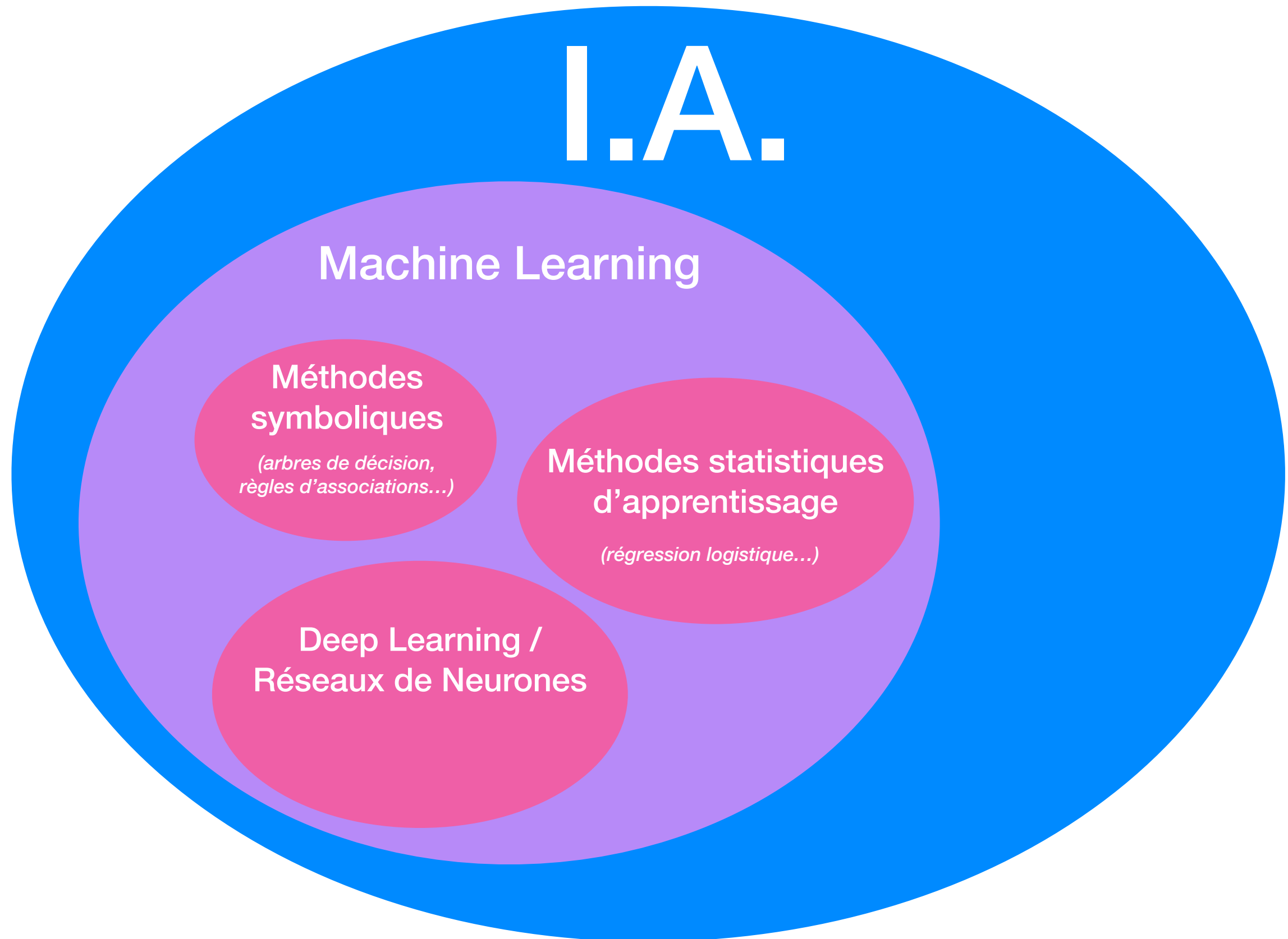
## Séance 6 : Jeudi 19 juin matin (3,5 h) – Introduction au Deep Learning (MLP)

- Perceptron multicouche (MLP) : architecture, couches denses, fonctions d'activation
- Rétropropagation : calcul du gradient, descente de gradient, régularisation (weight decay)
- Entraînement d'un MLP simple sur MNIST avec Pytorch

## Séance 7 : Jeudi 19 juin après-midi (3,5 h) – CNN + Grands Modèles de langue

- Réseaux convolutionnels (CNN) : convolution, pooling, architectures classiques (LeNet, VGG)
- Expliquer les LLMs en simplifiant (embeddings, attention, etc...)
- expliquer le pre-training, fine-tuning, etc.
- Leur faire jouer avec des LLMs en inférence LLMs en TP (soit la librairie huggingface, soit avec une librairie en ligne, comme openai ou openrouter)
- Jouer en TP avec des réseaux pre-trained, non post-trained

# Les Méthodes de ML en IA



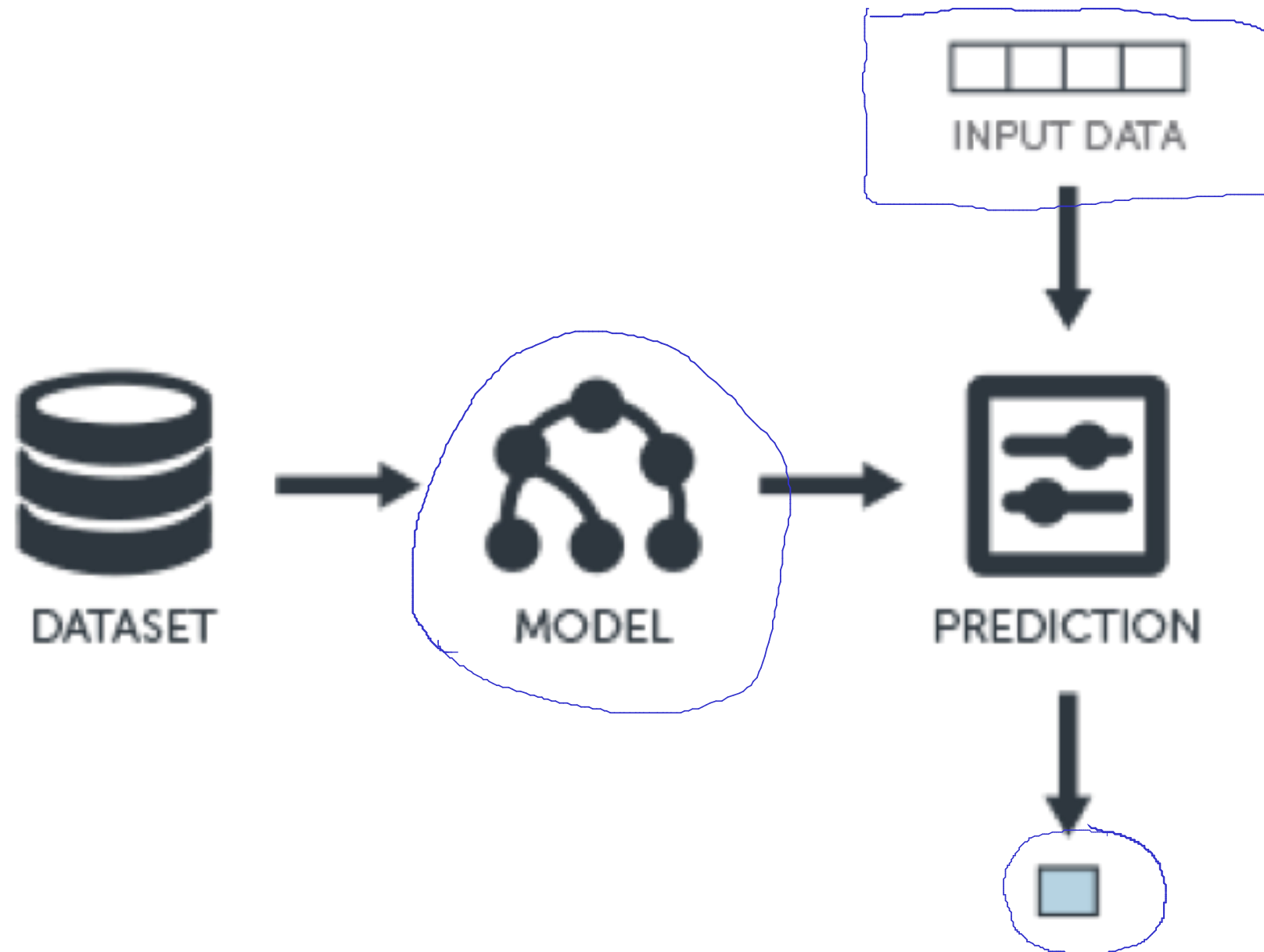
# Quelques repères historiques sur les méthodes d'Apprentissage Supervisé

- XIX siècle - Régression linéaire [Legendre, Gauss]
- 1936 - Linear Discriminant Analysis [Fisher]
- 1943 - Modèle mathématique du neurone. Pas d'apprentissage [McCulloch & Pitts]
- 1949 - Algorithme d'apprentissage non supervisé pour neurone [Hebb]
- 1958 - Algorithme du Perceptron [Rosenblatt]
- 1951 - Algorithme de la descente de gradient stochastique [Robbins, Monro]
- 1963 - Arbres de décision [Morgan, Sonquist]  
Classifieurs à vaste marge [Vapnik]
- 1971 - Réseaux de neurones à 8 couches [Ivakhnenko & Lapa]
- 1974 - Algorithme de rétro-propagation du gradient [Werbos]
- 1979 - Convolutional neural networks (*Neocognitron*, pas de back-prop. [Fukushima])
- 1988 - Naïve Bayes [Ohmann]
- 1989 - Réseau de neurone LeNet [Y. Lecun]
- 1990 - Méthodes de Boosting
- 1992 - Méthodes à Noyaux [V. Vapnik]
- 2012 - AlexNet [A. Krizhevsky]
- 2015 - ResNets
- 2017 - Transformers

# Typologie des problèmes d'apprentissage

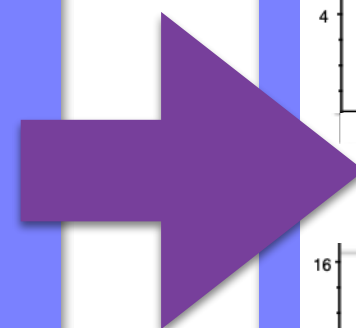
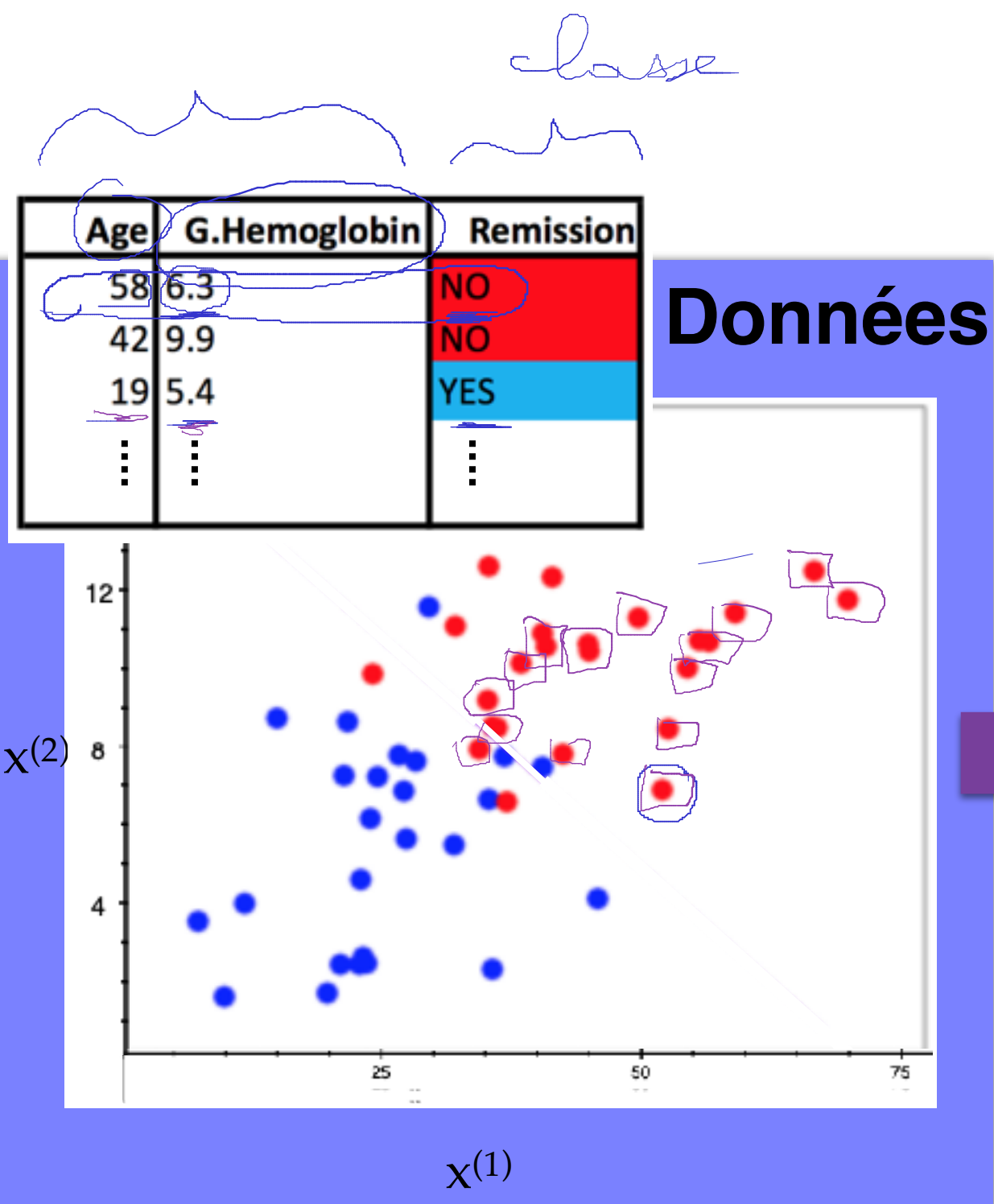
- **Apprentissage Supervisé**
  - Régression, classification, analyse de série temporelle  
filtrage collaboratif (recommandation)
- **Apprentissage non-supervisé**
  - Clustering, réduction de dimension, estimation de densité
- **Apprentissage par renforcement**

# Supervised ML: Learning a Model from Data

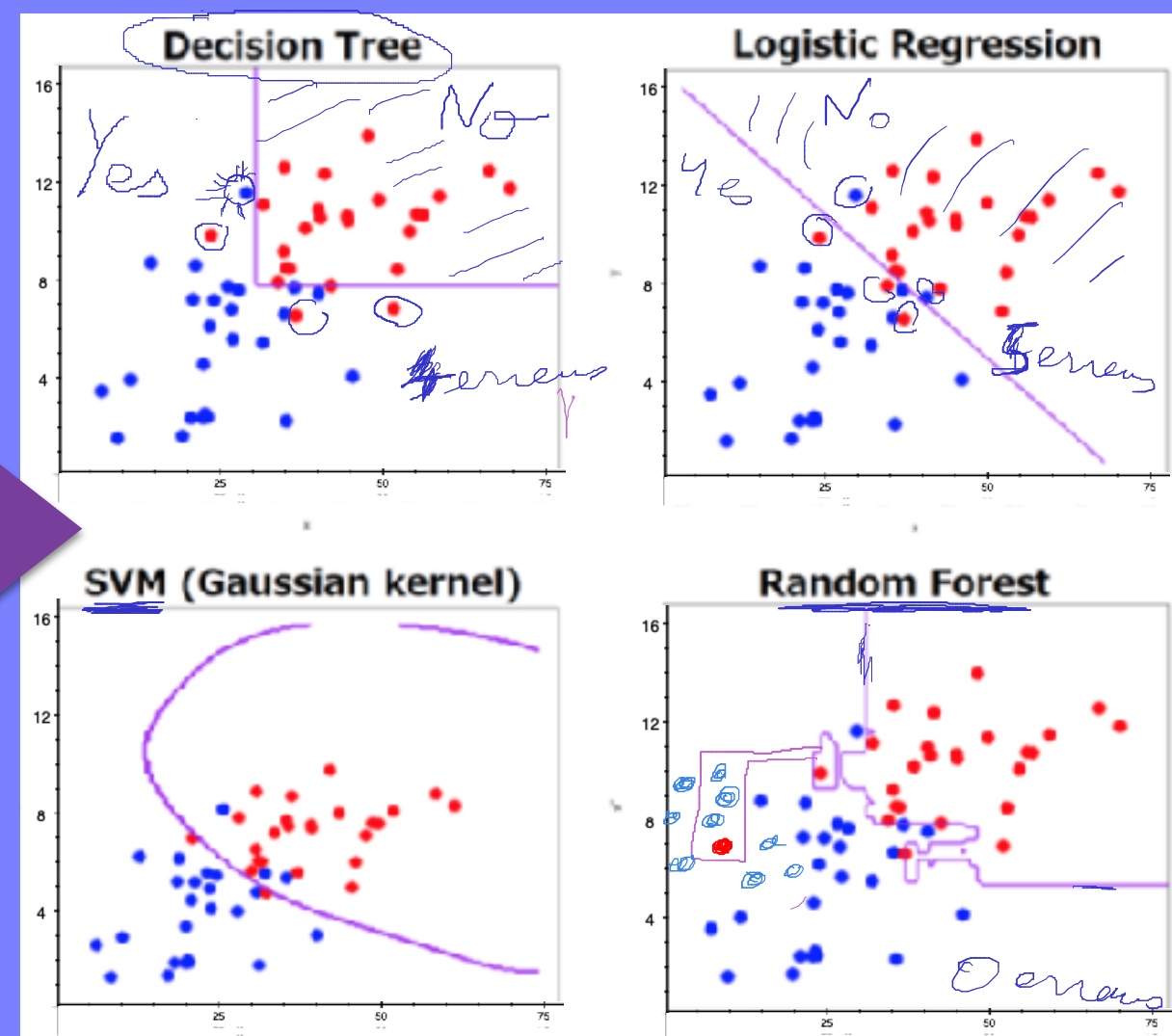




# Supervised ML: Learning a Model from Data



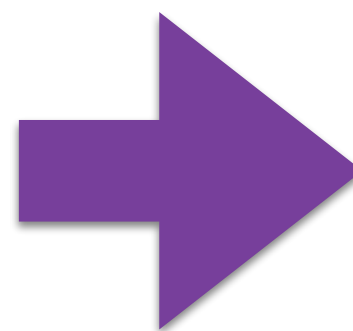
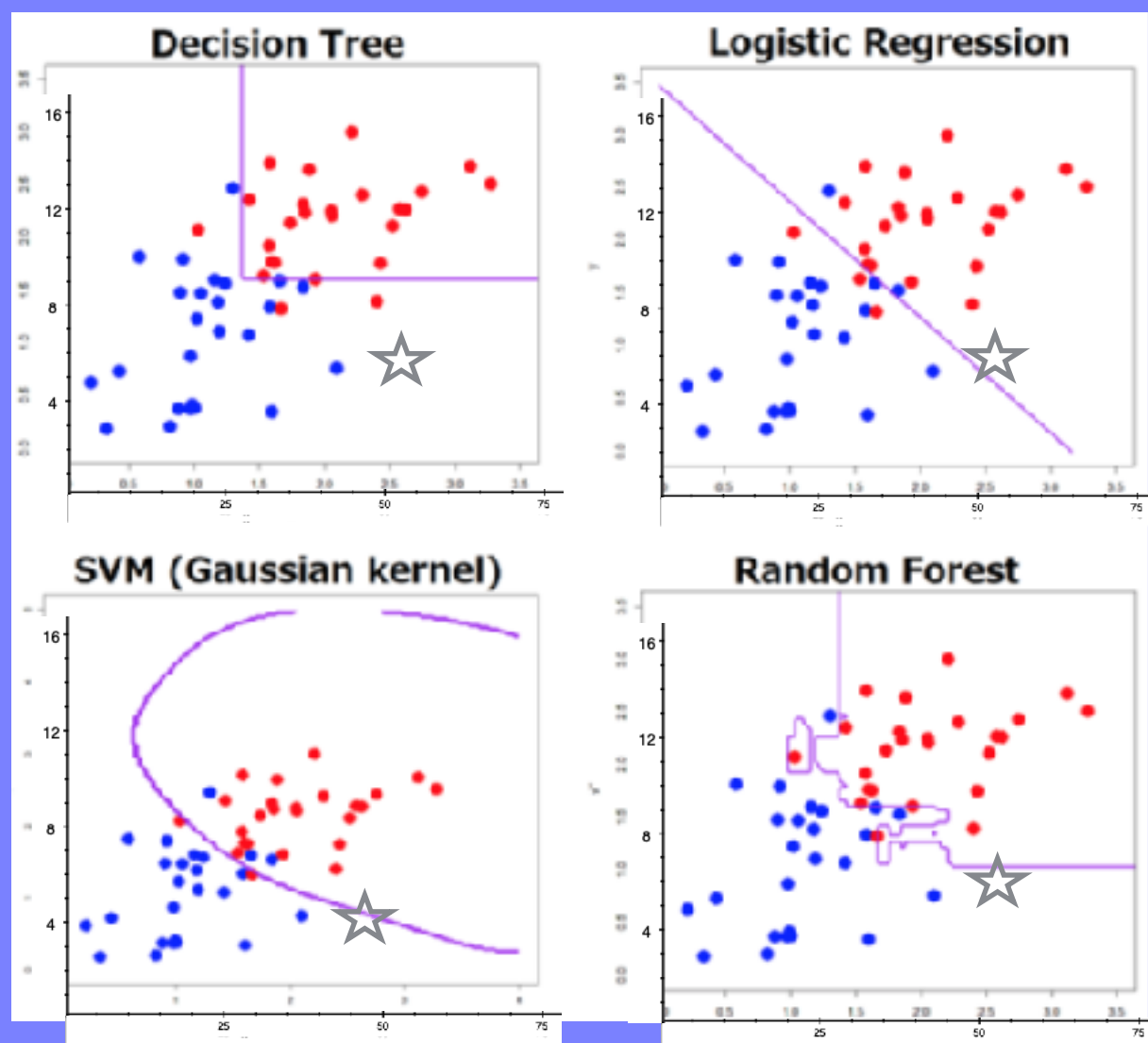
## Modèles



# Supervised ML: Learning a Model from Data

## Using this Model for Prediction

### Modèles



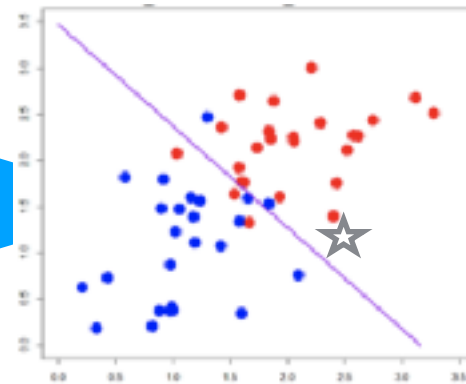
Age	G.Hemoglobin	Remission
55	6.1	?

# Apprentissage Supervisé: Classification vs Régression

- Si les prédictions sont des catégories (Yes/No, Colors), c'est de la classification

Age	G.Hemoglobin	Remission
58	6.3	NO
42	9.9	NO
19	5.4	YES
⋮	⋮	⋮

Apprentissage



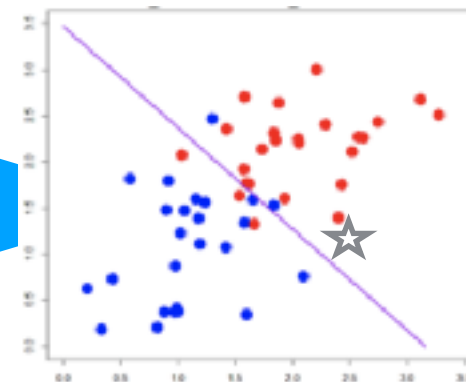
Prédiction

$h(x)=\text{NO}$

- Si les prédictions sont des nombre, c'est de la régression

Age	G.Hemoglobin	Remission
58	6.3	NO
42	9.9	NO
19	5.4	YES
⋮	⋮	⋮

Apprentissage

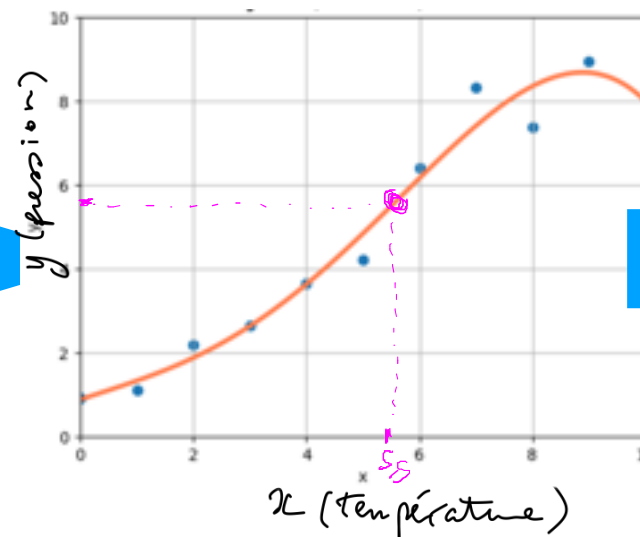


Prédiction

$h(x)=\text{YES w.p. } 30\%$   
 $\text{NO w.p. } 70\%$

Température	Pression
2	2.1
1.1	0.9
6	6.2
...	...

Apprentissage



Prédiction

$h(x)=1.9$

# Qu'est ce que l'apprentissage Supervisé ?

## Jeu de données d'apprentissage

Handwritten annotations:

- categorical* (circled in pink)
- categorical* (circled in pink)
- continuous* (circled in pink)
- class* (circled in pink)
- (numérique)* (circled in pink)
- exemple n°1* (circled in pink)
- étiquette n°1* (circled in pink)
- Variable, attributs explicatifs* (circled in pink)
- étiquette, classe, Variable à prédire* (circled in pink)

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

# Qu'est ce que l'apprentissage Supervisé ?

Jeu de données d'apprentissage

					categorical		categorical		continuous	class
Tid	Refund	Marital Status	Taxable Income	Cheat						
1	Yes	Single	125K	No	$x_1$		$x_2$		$y_1$	$y_2$
2	No	Married	100K	No						
3	No	Single	70K	No						
4	Yes	Married	120K	No						
5	No	Divorced	95K	Yes						
6	No	Married	60K	No						
7	Yes	Divorced	220K	No						
8	No	Single	85K	Yes						
9	No	Married	75K	No						
10	No	Single	90K	Yes						

$x^{(1)}$   $x^{(2)}$   $x^{(3)}$

Exemple:

$$x_{2}^{(3)} = 100$$

# Qu'est ce que l'apprentissage Supervisé ?

Jeu de données d'apprentissage

		categorical		categorical	continuous	class
Tid	Refund	Marital Status	Taxable Income	Cheat		
1	Yes	Single	125K	No	$x_1$	$y_1$
2	No	Married	100K	No	$x_2$	$y_2$
3	No	Single	70K	No		
4	Yes	Married	120K	No		
5	No	Divorced	95K	Yes		
6	No	Married	60K	No		
7	Yes	Divorced	220K	No		
8	No	Single	85K	Yes		
9	No	Married	75K	No		
10	No	Single	90K	Yes		

$x^{(1)}$   $x^{(2)}$   $x^{(3)}$

But: Trouver une fonction (un modèle)  $f: X \rightarrow Y$  tel que pour tout couple  $(x_i, y_i)$ ,  $f(x_i) \approx y_i$



# Qu'est ce que l'apprentissage Supervisé ?

Jeu de données d'apprentissage

<i>Tid</i>	<i>Refund</i>	<i>Marital Status</i>	<i>Taxable Income</i>	<i>Cheat</i>
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

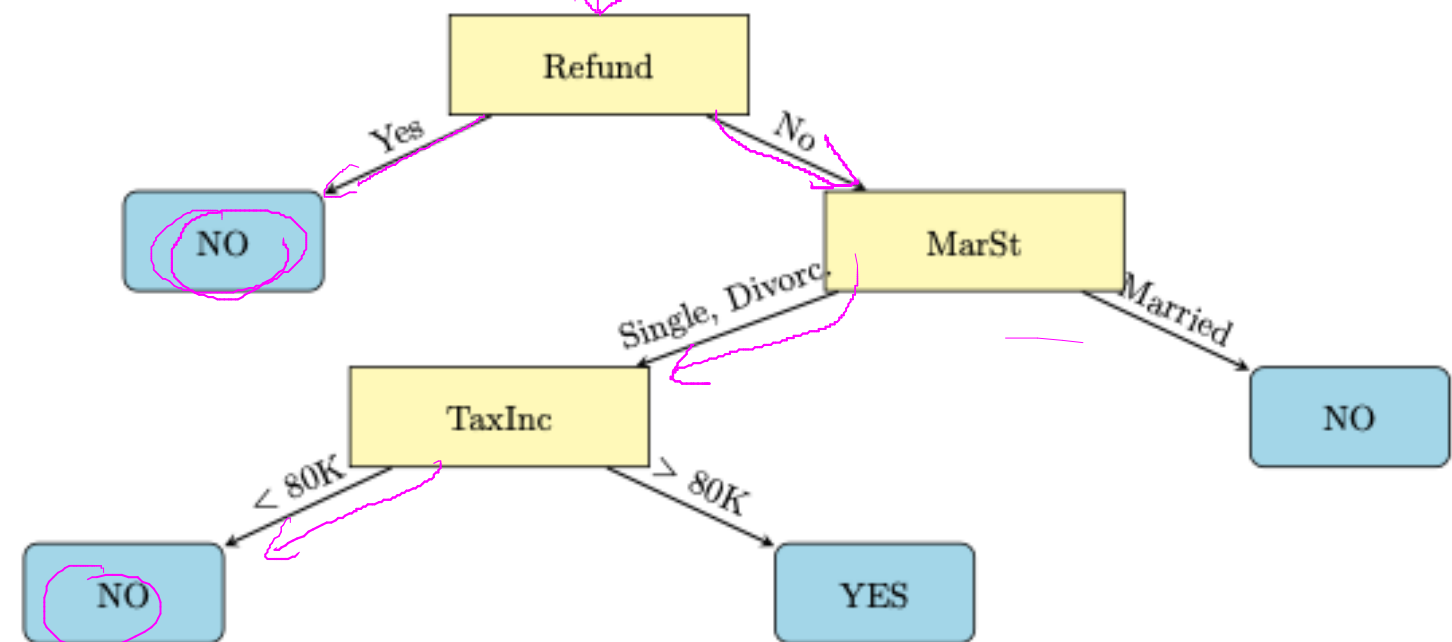
*categorical* (Refund, Marital Status)  
*categorical* (Marital Status)  
*continuous* (Taxable Income)  
*class* (Cheat)

$x_1$  (Refund)  
 $x_2$  (Marital Status)  
 $x^{(2)}$  (Marital Status)  
 $x^{(3)}$  (Taxable Income)

$x = (\text{Yes}, \text{Divorce}, 60K)$   
 $h(x) = \text{No}$

$x' = (\text{No}, \text{Single}, 30K)$   
 $h(x') = \text{No}$

Decision tree representing  $f(x)$



for example

$f(x) = \begin{cases} YES & \text{if Refun}=\text{No and Mar} \in \{\text{Sing, Div}\} \text{ and Tax} > 80 \\ NO & \text{otherwise} \end{cases}$

# Qu'est ce que l'apprentissage Supervisé ?

Jeu de données d'apprentissage

Jeu de test

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?

~~Apprentissage~~

~~Prediction~~

for example

$$f(x) = \begin{cases} YES & \text{if Refun=No and Mar} \in \{\text{Sing, Div}\} \text{ and Tax} > 80 \\ NO & \text{otherwise} \end{cases}$$






# Application de l'apprentissage supervisé

- **Banque/Assurance/Commerce:**

- Prédire la solvabilité d'un individu
- Détection de fraude
- Identifier des profils type de client
- Prédire combien un client potentiel peut rapporter
- « Analyse de sentiment »

- **Vision**

- Reconnaissance d'images, d'objet dans les images

x	y
	Chien
	Pandas
	Bus

- **Texte**

- Traduction de texte d'une langue vers une autre
- Texte->parole
- parole->texte
- Chatbots

x	y
My tailor is rich	Mon tailleur est riche
The cat eats the mouse	Le chat mange la souris
...	...

- **Robotique:**

- Voiture autonome
- Robotique humanoïde

x	y
My tailor is	Rich
...	...

# Apprentissage *non* supervisé

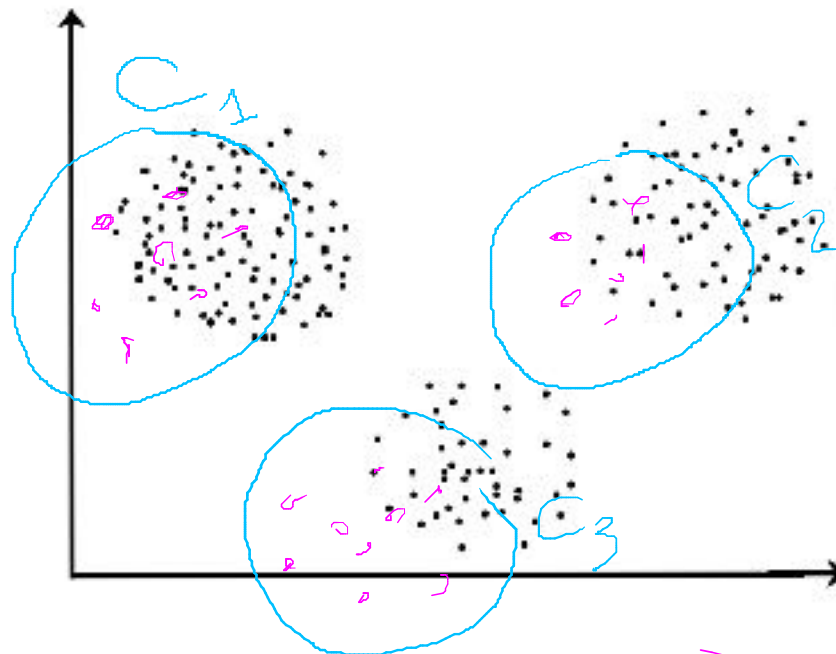
- **Clustering**

Pas de supervision (juste  $x_1 \dots x_N$ , pas d'étiquette  $y$ )

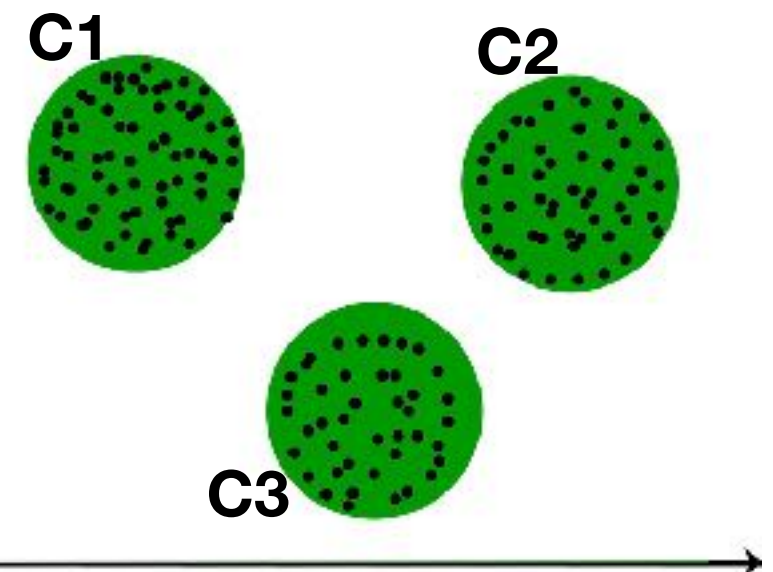
Jeu de données

$x^{(1)}$	$x^{(2)}$
3	1.1
10.12	4.9
4.2	2.2
...	...

Représentation graphique  
des données



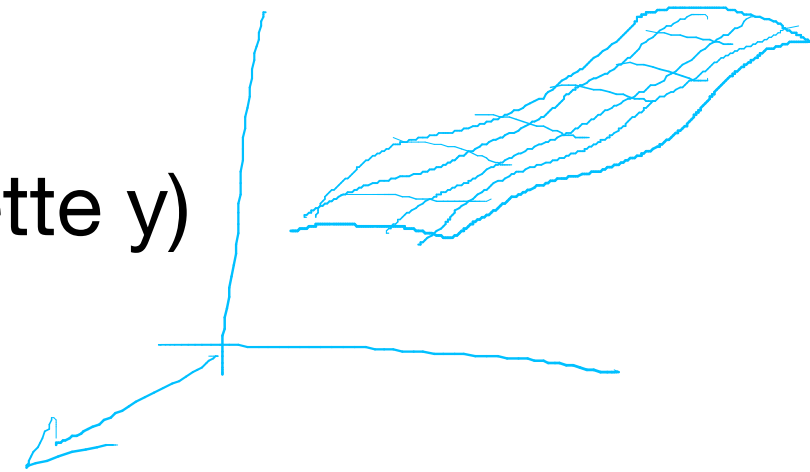
Clustering



# Apprentissage *non* supervisé

- **Réduction de dimension**

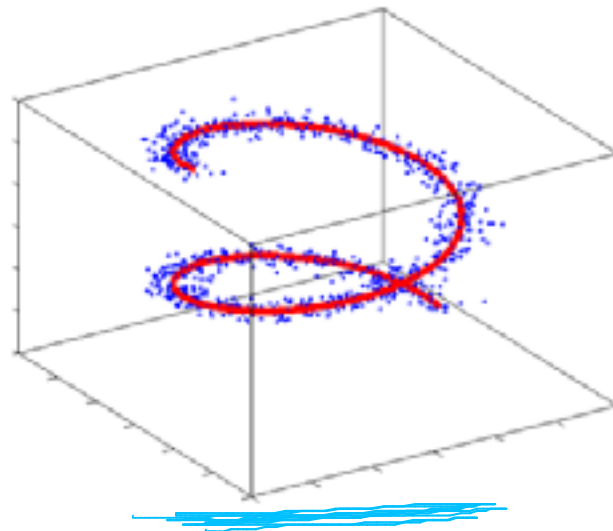
Pas de supervision (juste  $x_1 \dots x_N$ , pas d'étiquette  $y$ )



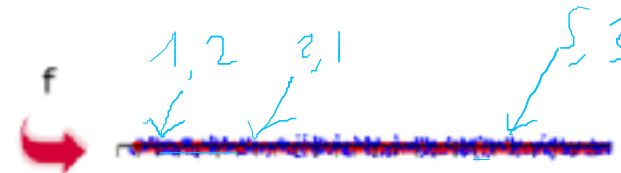
**Jeu de données**

$x^{(1)}$	$x^{(2)}$	$x^{(3)}$
3	1.1	2.1
10.1	4.9	5.2
4.2	2.2	23
...	...	

**Représentation graphique des données**



**Représentation graphique des données en basse dimension**



**Jeu de données en basse dimension**

$z^{(1)}$
21
4.3
4.9
...

# Apprentissage *non* supervisé

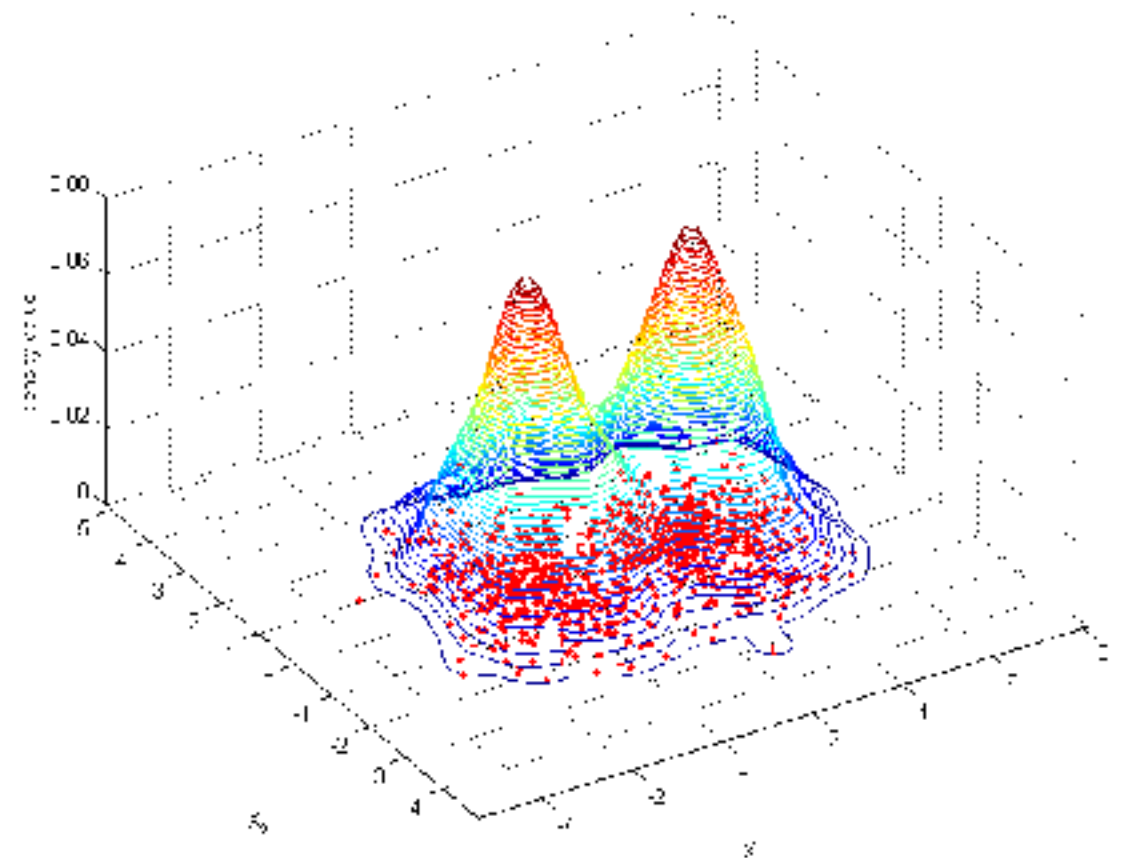
- **Estimation de density**

Pas de supervision (juste  $x_1 \dots x_N$ , pas d'étiquette  $y$ )

Jeu de données

$x^{(1)}$	$x^{(2)}$
3	1.1
10.12	4.9
4.2	2.2
...	...

Apprentissage



$$P(x^{(1)}, x^{(2)})$$

[https://commons.wikimedia.org/wiki/File:Bivariate\\_example.png](https://commons.wikimedia.org/wiki/File:Bivariate_example.png)

# Apprentissage par Renforcement

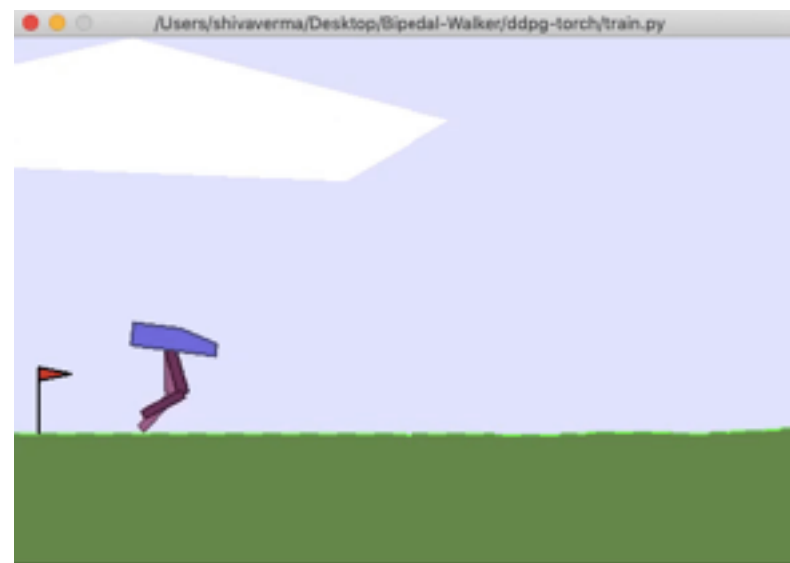
- Apprentissage par essais/erreurs



- Ex: apprendre à marcher, à gagner à des jeux vidéos, à exécuter une tâche dans une usine, à conduire une voiture automatiquement



1ière itération



500 itérations



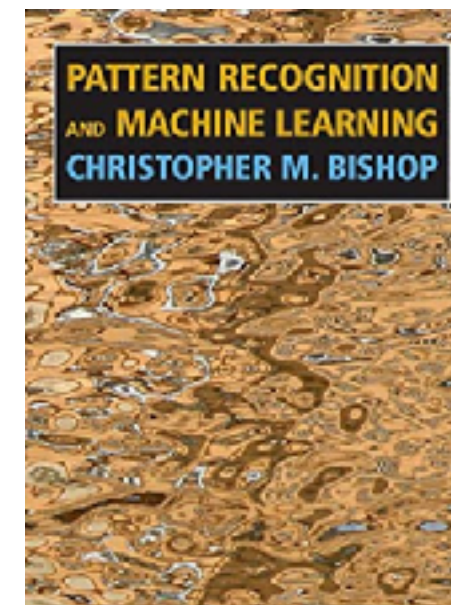
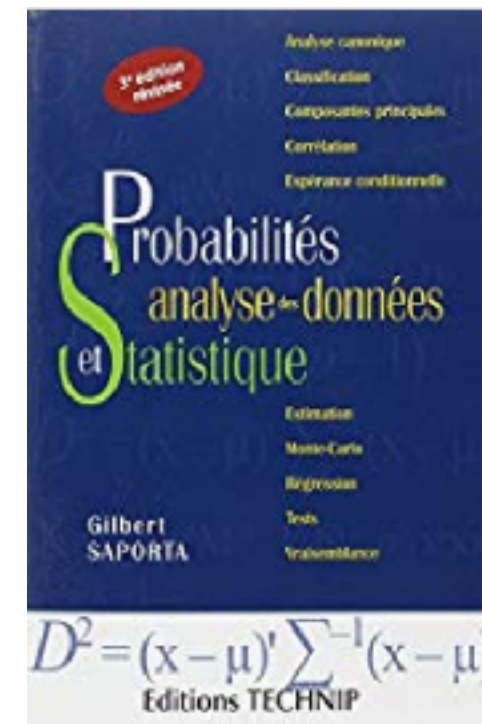
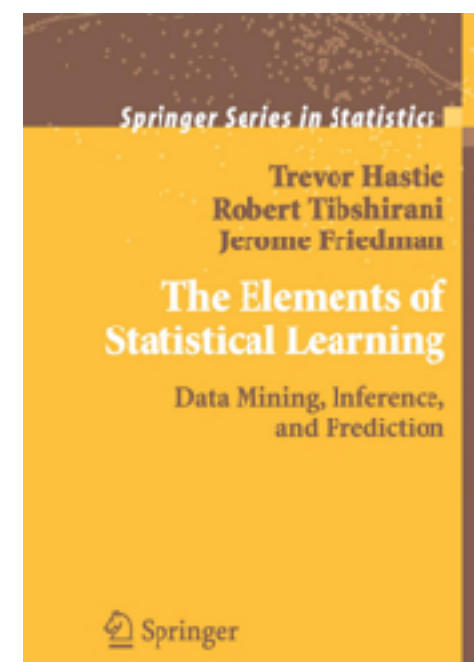
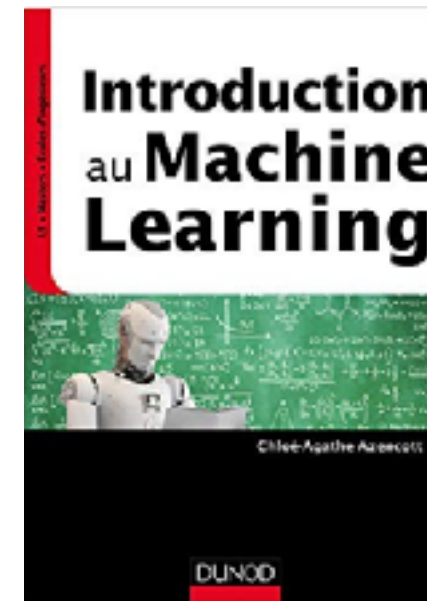
+1000 itérations

<https://www.freecodecamp.org/news/a-brief-introduction-to-reinforcement-learning-7799af5840db/>  
<https://towardsdatascience.com/teach-your-ai-how-to-walk-5ad55fce8bca>

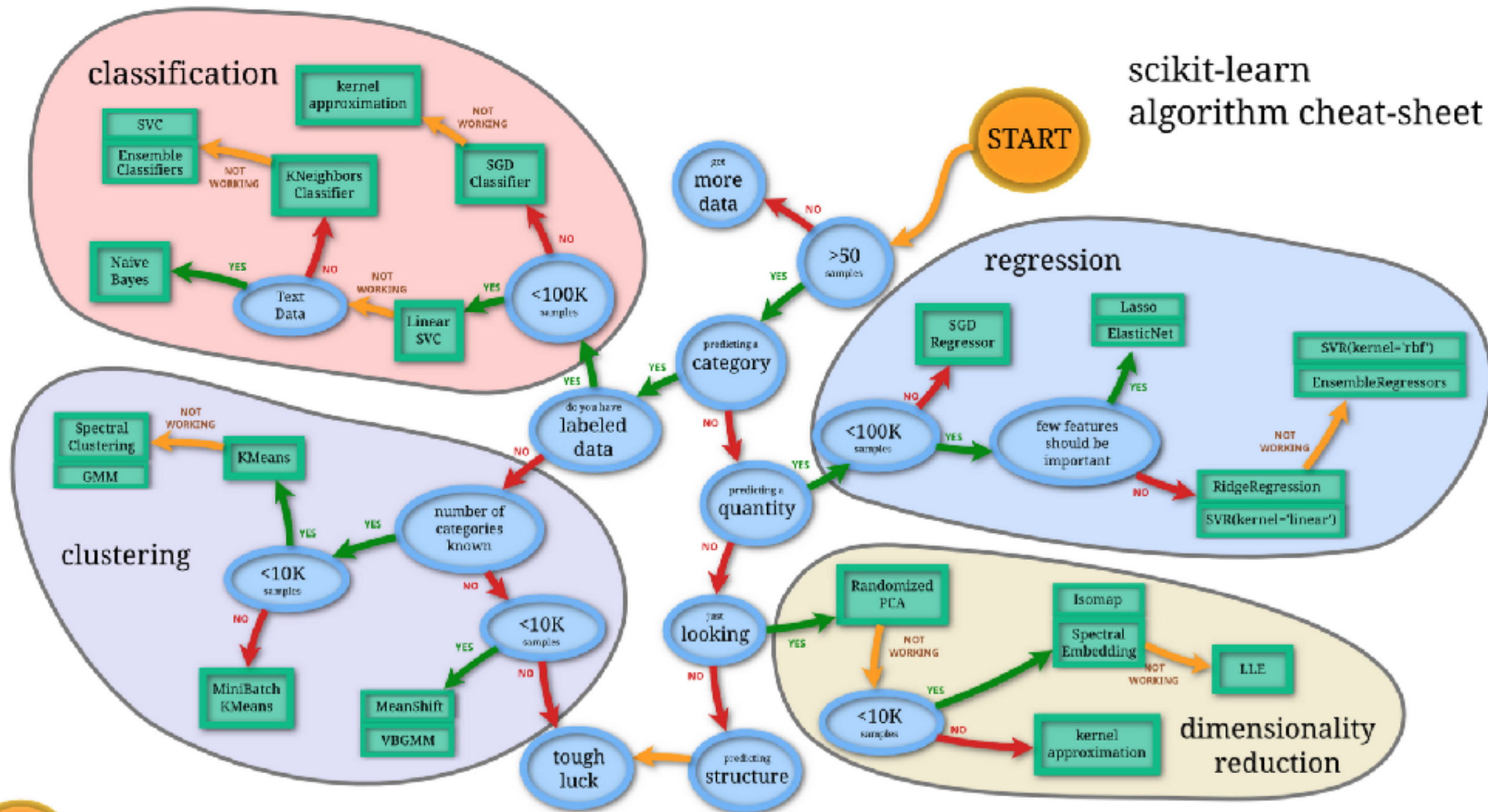


# Références

- T. Hastie,  
*The Elements of Statistical Learning*
- G. Saporta  
*Probabilités, Analyse de données  
et Statistiques*
- A. Cornuéjols  
*Apprentissage Artificiel,  
concepts et algorithmes*
- C.A. Azencott  
*Introduction au Machine Learning*



# ***Scikit Learn***: Un package Python pour le Machine Learning

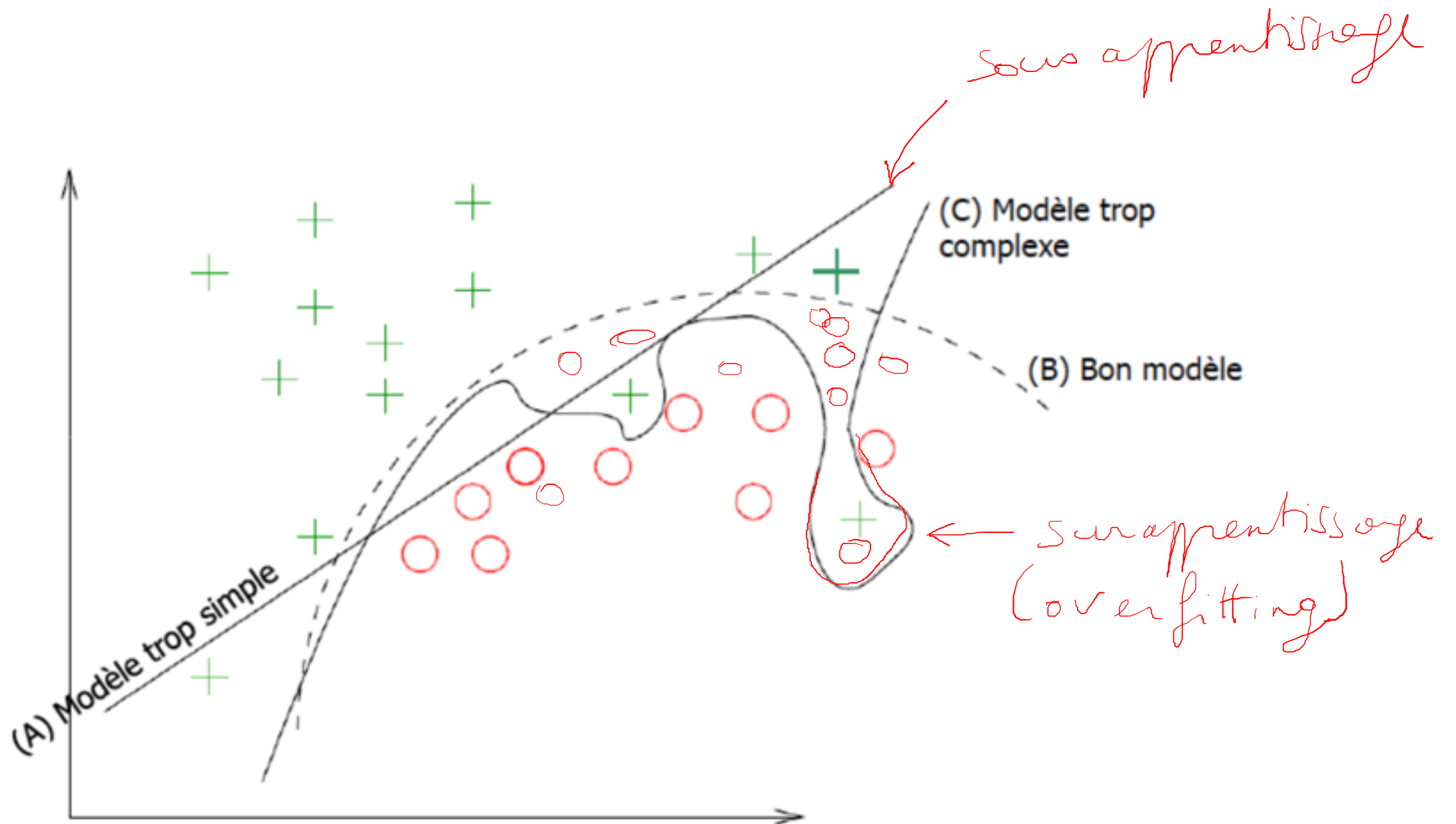


# Apprentissage Supervisé

## - Evaluation des Modèles



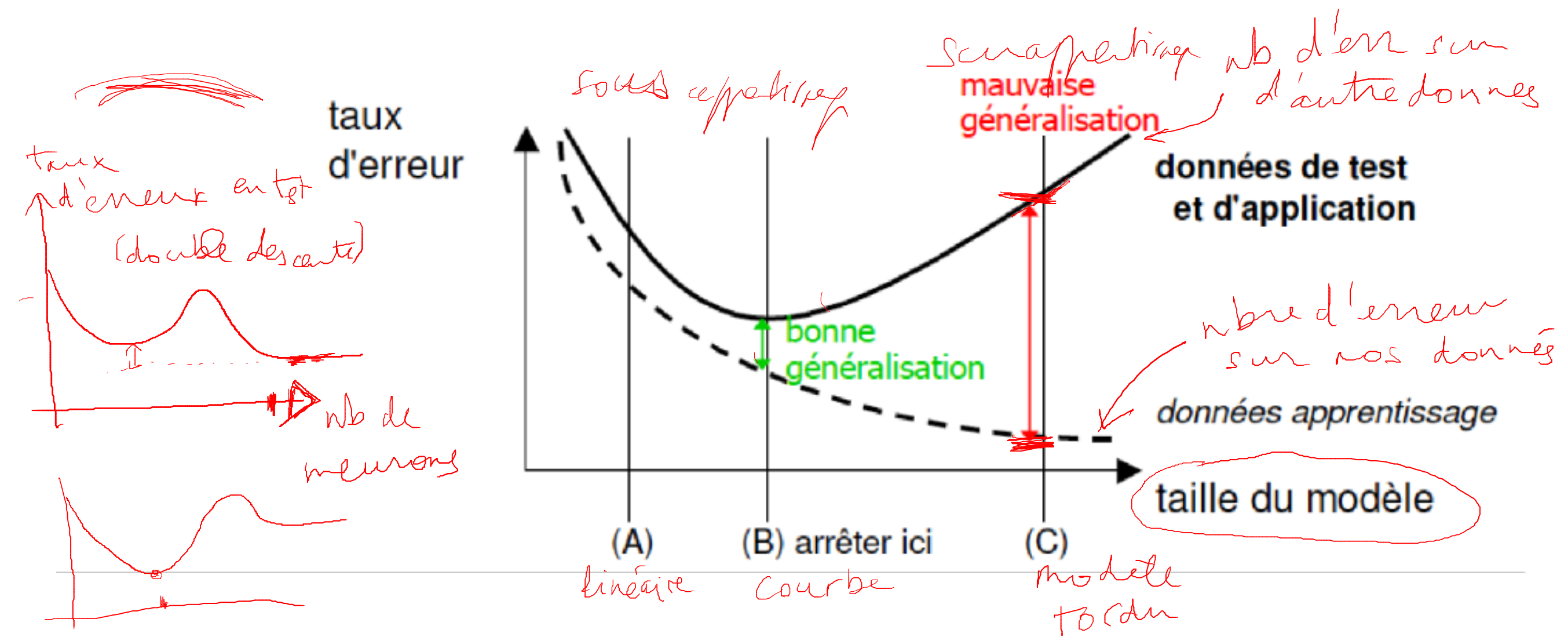
# Evaluation des modèles



Source : Olivier Bousquet

# Evaluation des modèles

taux d'erreur en fonction de la complexité du modèle.



# Evaluation des modèles

## Matrice de confusion

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)  
b: FN (false negative)  
c: FP (false positive)  
d: TN (true negative)

- True positive = correctly identified
- False positive = incorrectly identified
- True negative = correctly rejected
- False negative = incorrectly rejected

# Evaluation des modèles

Taux d'erreur : accuracy

ACTUAL CLASS	PREDICTED CLASS	
	Class=Yes	Class=No
	Class=Yes	Class=No
	a (TP)	b (FN)
	c (FP)	d (TN)

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

# Evaluation des modèles

Taux d'erreur : accuracy

Quelques limitations

- On considère un problème à 2 classes avec : 9990 instances de classe 0 et 10 instances de classe 1.
- Si le modèle prédit que tout instance est de classe 0, on a

$$\text{Accuracy} = \frac{9990}{10000} = 99,9$$

# Evaluation des modèles

## Recall vs precision

ACTUAL CLASS	PREDICTED CLASS	
	Class=Yes	Class=No
	Class=Yes	Class=No
	a (TP)	b (FN)
	c (FP)	d (TN)

## Recall (True positive rate sensitivity)

De ceux qui existent, combien l'algorithme a pu trouver  $TPR = \frac{TP}{TP+FN}$

## Precision

De ceux que l'algorithme a pu classer, combien sont corrects.  $PPV = \frac{TP}{TP+FP}$

# Evaluation des modèles

## F-mesure

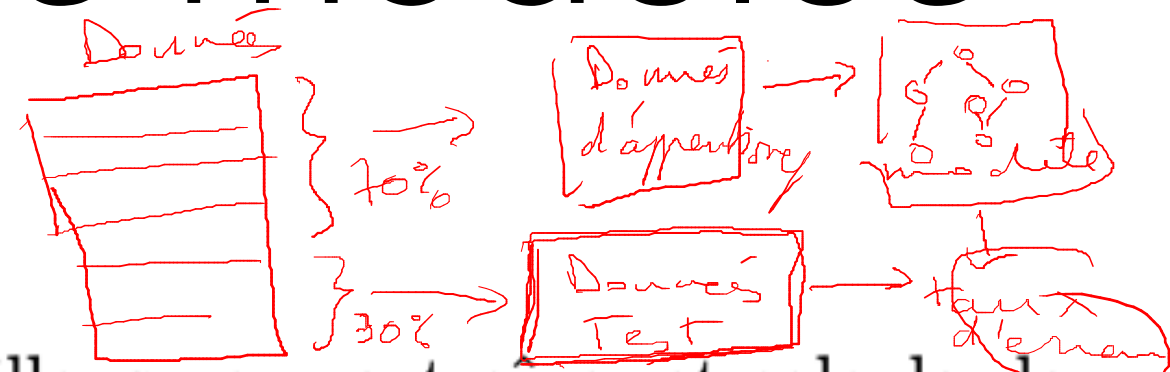
ACTUAL CLASS	PREDICTED CLASS	
	Class=Yes	Class=No
	Class=Yes	Class=No
Class=Yes	a (TP)	b (FN)
Class=No	c (FP)	d (TN)

Moyenne harmonique entre la precision et le rappel :

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{2TP}{2TP + FP + FN}$$



# Evaluation des modèles



- Comment estimer le taux d'erreur ?
- Méthode naïve : utiliser tous les échantillons pour entraîner et calculer le taux d'erreur sur l'ensemble d'apprentissage.
- Un classifieur tend à s'ajuster aux données d'apprentissage
- Un taux d'erreur généralement trop optimiste : pas rare d'avoir un taux de 0 à l'entraînement.
- Nécessité d'un ensemble de test indépendant de l'ensemble d'entraînement.
- Typiquement 10% de l'ensemble  $\mathcal{D}$  pour tester.



# Evaluation des modèles

- Qu'arrive-t-il si on dispose de très peu d'échantillons ?
- Comment savoir si le taux d'erreur est précis ou si on est pas tombé par hasard sur une situation particulière en coupant l'ensemble  $\mathcal{D}$  ?
- Si pour un ensemble de données  $\mathcal{D}$ , 2 classifieurs  $C_1$  et  $C_2$  ont 80% et 85% de précision, est-ce que  $C_2 > C_1$  ?
- Solution : **Validation croisée** :
  - aléatoire
  - k- blocs
  - n-blocs (leave one out)

downes



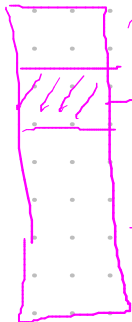
test

→

evaluate →  $e_1$

→ app

→  $e_2$



test

→

evaluate →  $e_2$

→ app

→  $e_1$

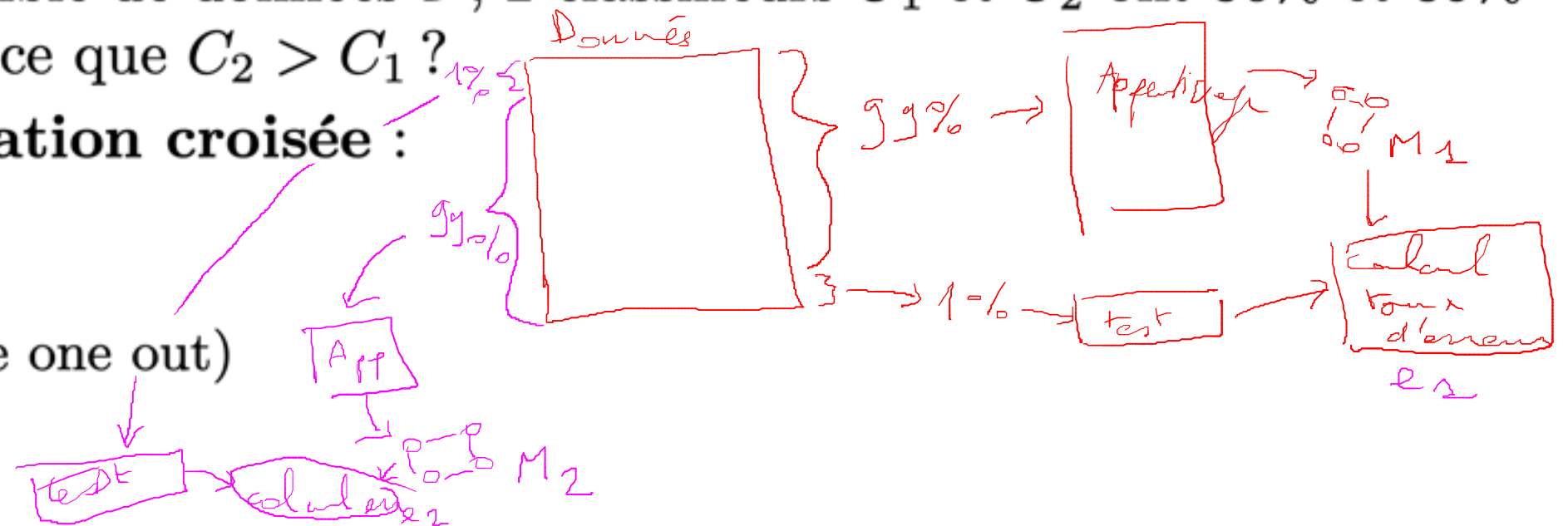


$e_{\text{final}} = \text{moyenne}(e_1, e_2, \dots)$

# Evaluation des modèles

- Qu'arrive-t-il si on dispose de très peu d'échantillons ?
- Comment savoir si le taux d'erreur est précis ou si on est pas tombé par hasard sur une situation particulière en coupant l'ensemble  $\mathcal{D}$  ?
- Si pour un ensemble de données  $\mathcal{D}$ , 2 classifieurs  $C_1$  et  $C_2$  ont 80% et 85% de précision, est-ce que  $C_2 > C_1$  ?
- Solution : **Validation croisée** :

- aléatoire
- k- blocs
- n-blocs (leave one out)



# Evaluation des modèles

- Qu'arrive-t-il si on dispose de très peu d'échantillons ?
- Comment savoir si le taux d'erreur est précis ou si on est pas tombé par hasard sur une situation particulière en coupant l'ensemble  $\mathcal{D}$  ?
- Si pour un ensemble de données  $\mathcal{D}$ , 2 classifieurs  $C_1$  et  $C_2$  ont 80% et 85% de précision, est-ce que  $C_2 > C_1$  ?
- Solution : **Validation croisée** :
  - aléatoire
  - k- blocs
  - n-blocs (leave one out)

# Evaluation des modèles

## Validation croisée $K$ -blocs

- On prend  $K$  ensemble disjoints de  $\frac{n}{K}$  échantillons chacun
- On teste avec l'un d'entre eux.
- Taux d'erreur = moyenne des  $K$  expériences.
- La variance peut être calculée
- Avantage : tous les échantillons de  $\mathcal{D}$  seront utilisés.