

Approches statistiques du Machine Learning

Hugo Schmutz, Phd student

3IA

INRIA

Universite cote d'Azur

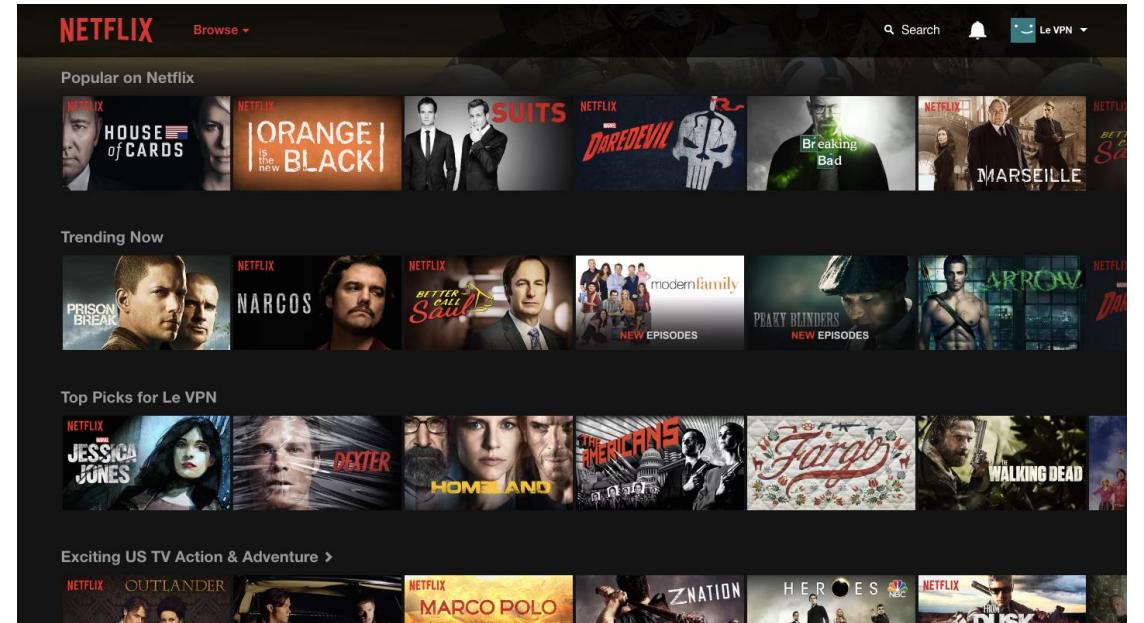
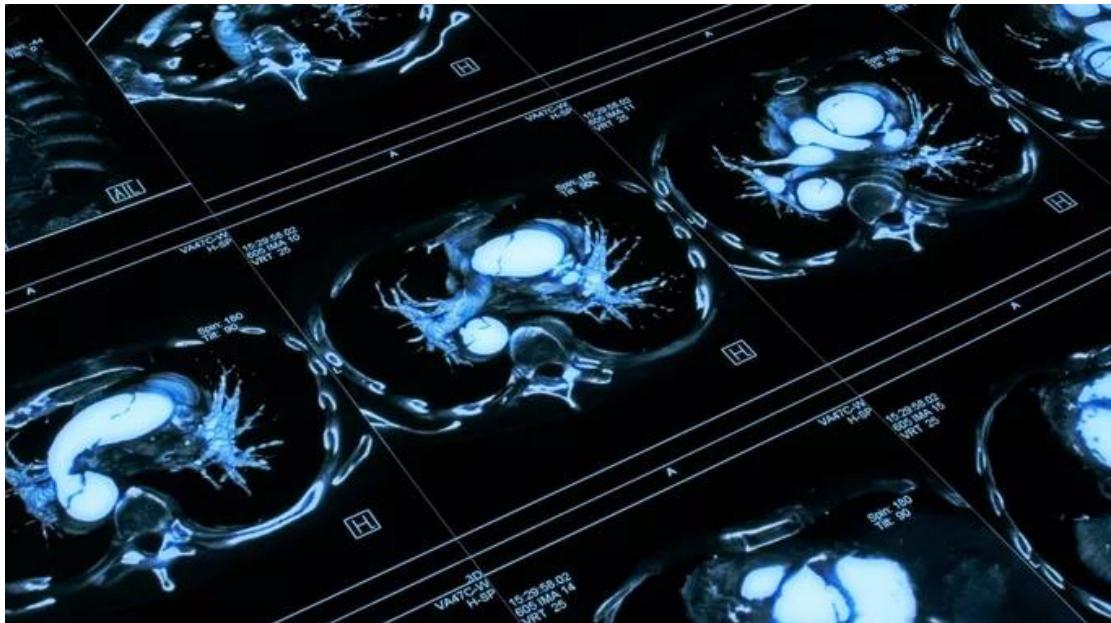


inria



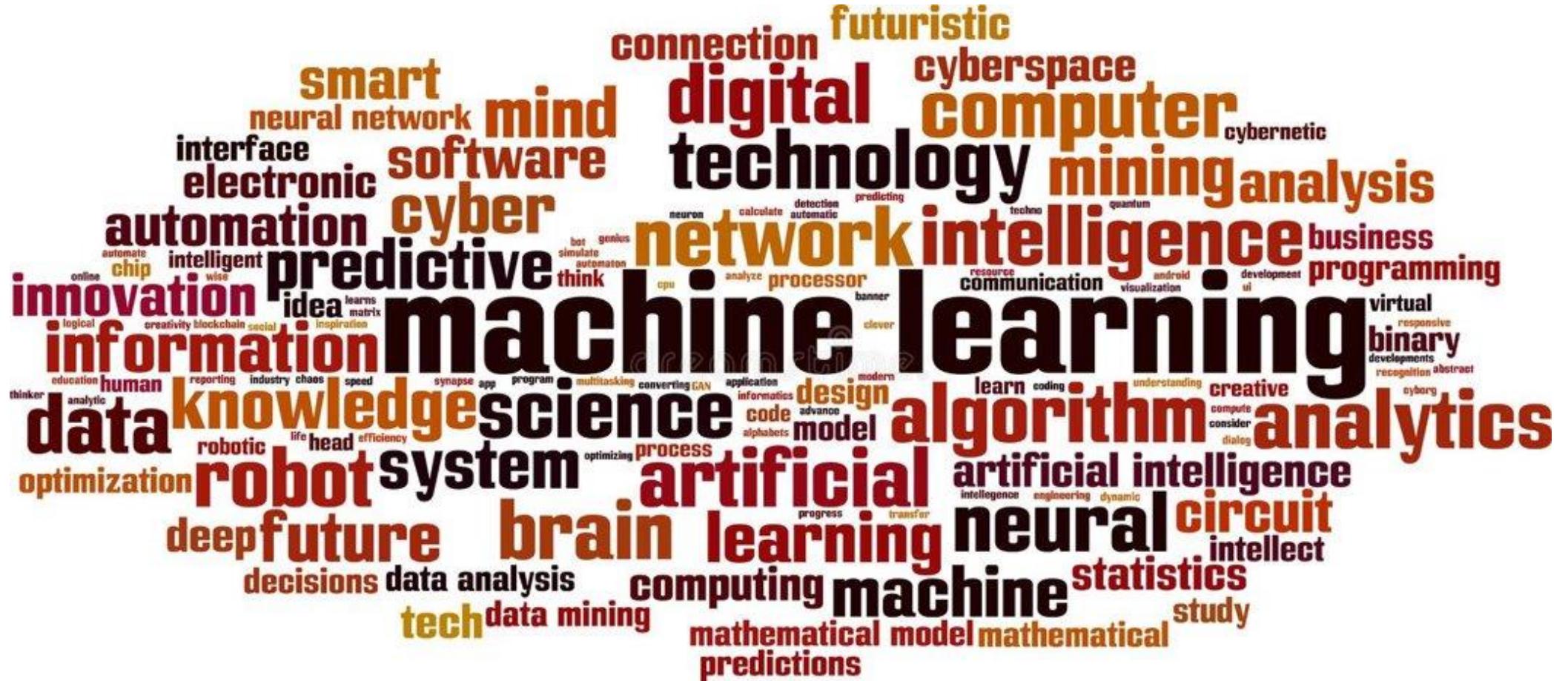
Introduction

"L'IA est déjà parmi nous de manière très visible et impressionnante, lorsque nous dictons un message à notre téléphone, quand notre appareil photo se charge de la mise au point sur les visages ou quand les meilleurs joueurs d'échecs et de go se font battre par les machines...", Francis Bach

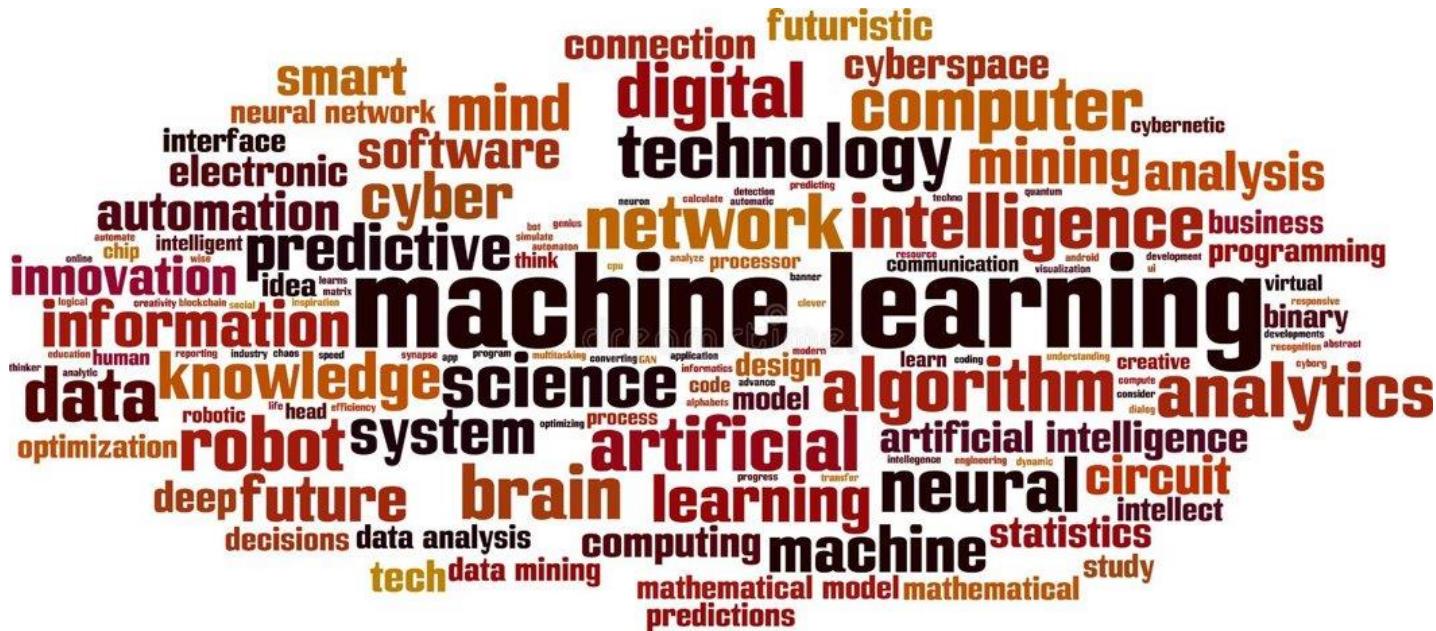


Qu'est-ce que le mot IA vous évoque ?

Qu'est-ce que le mot IA vous évoque ?



Qu'est-ce que le mot IA vous évoque ?



Tout cela repose sur 3 piliers:

- L'algèbre linéaire
- Les statistiques et probabilités
- L'optimisation

Plan du cours

Rappel de maths

1. Probabilité
2. Dérivée d'une fonction
3. Introduction au multidimensionnel
4. Introduction du gradient d'une fonction multidimensionnel

I. Algorithme d'apprentissage

1. Les Taches
2. La mesure de Performance
3. L'Experience
4. Comment ça marche ?
5. Exemple: Linear regression

II. Capacité, overfitting et underfitting

1. Training and test datasets
2. Overfitting et underfitting
3. Fonction de pertes et regularisation
4. Hyper-paramètres et datasets de validation

III. Supervised learning:

1. Definition
2. Risque et risque empirique: fonction de pertes
3. Optimisation et descente de gradient
4. Exemple: regression logistique

Rappel de maths:

1. Probabilité
2. Dérivée d'une fonction
3. Introduction au multidimensionnel
4. Introduction: gradient d'une fonction multidimensionnelle

1. Probabilité

- **Variable aléatoire?**

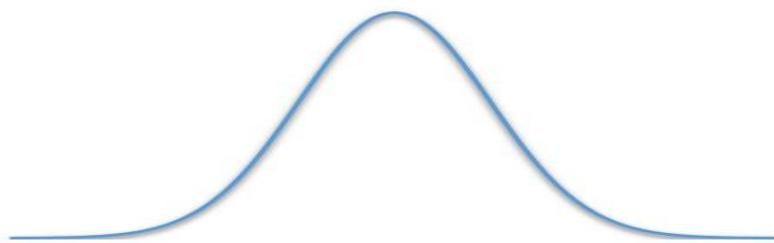
1. Probabilité

- **Variable aléatoire:** Une variable qui prend ses valeurs de manière aléatoire.

Exemple: On jette un dé, l'ensemble des issues possibles est $\{1, 2, 3, 4, 5, 6\}$. On peut considérer la variable aléatoire X qui, au jet du dé, associe le nombre obtenu.

1. Probabilité

- **Distribution?**



1. Probabilité

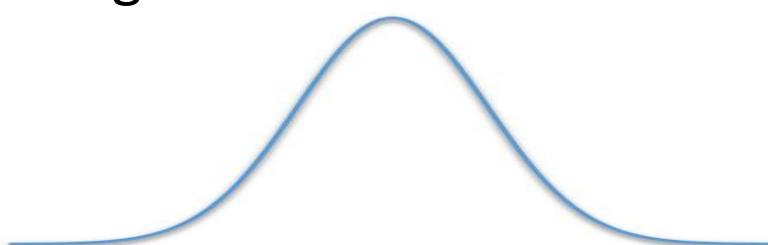
- **Distribution:** Une description de la chance qu'une variable X aléatoire prenne une de ses valeurs x . $p(X=x)$ ou simplement $p(x)$

- Peut être **discrète**, ex: distribution uniforme.

Exemple: On jette un dé, l'ensemble des issues possibles est $\{1, 2, 3, 4, 5, 6\}$. On peut considérer la variable aléatoire X qui, au jet du dé, associe le nombre obtenu.

$$p(X = 1) = p(X = 2) = \dots = p(X = 6) = \frac{1}{6}$$

- Ou **continue**, ex: distribution gaussienne



1. Probabilité

- **Probabilité marginale:** Soit x et y deux variables aléatoires.
 - Supposons que l'on cherche à calculer $p(x)$ mais que l'on connaît uniquement $p(x,y)$
- Exemple: X et Y variables aléatoires prenant leurs valeurs resp. dans $\{-2,0,2\}$ et dans $\{-1,1\}$

$$p(X = -2, Y = 1) = \frac{1}{4}$$

$$p(X = 0, Y = -1) = \frac{1}{2}$$

$P(X=x_i)$	$p(X = -2)$	$p(X = 0)$	$p(X = 2)$
$P(Y=y_j)$	$p(Y = 1)$?	$p(Y = -1)$
	1/4	0	1/2
-2			
0			
2	1/4		0

1. Probabilité

- **Probabilité marginale:** Soit x et y deux variables aléatoires.
 - Supposons que l'on cherche à calculer $p(x)$ mais que l'on connaît uniquement $p(x,y)$
 - Exemple: X et Y variables aléatoires prenant leurs valeurs resp. dans $\{-2,0,2\}$ et dans $\{-1,1\}$

$$p(X = -2) = p(X = -2, Y = 1) + p(X = -2, Y = -1)$$

$$p(X = 0) = p(X = 0, Y = 1) + p(X = 0, Y = -1)$$

$$p(X = 2) = p(X = 2, Y = 1) + p(X = 2, Y = -1)$$

$$p(Y = 1) = p(X = -2, Y = 1) + p(X = 0, Y = 1) + p(X = 2, Y = 1)$$

$$p(Y = -1) = p(X = -2, Y = -1) + p(X = 0, Y = -1) + p(X = 2, Y = -1)$$

$X \backslash Y$	1	-1	$P(X=x_i)$
-2	1/4	0	$p(X = -2)$
0	0	1/2	$p(X = 0)$
2	1/4	0	$p(X = 2)$
$P(Y=y_j)$	$p(Y = 1)$?	$p(Y = -1)$

1. Probabilité

- **Probabilité marginale:** Soit x et y deux variables aléatoires.
 - Supposons que l'on cherche à calculer $p(x)$ mais que l'on connaît uniquement $p(x,y)$
- Exemple: X et Y variables aléatoires prenant leurs valeurs resp. dans $\{-2,0,2\}$ et dans $\{-1,1\}$

$$p(X = -2) = p(X = -2, Y = 1) + p(X = -2, Y = -1)$$

$$p(X = 0) = p(X = 0, Y = 1) + p(X = 0, Y = -1)$$

$$p(X = 2) = p(X = 2, Y = 1) + p(X = 2, Y = -1)$$

$$p(Y = 1) = p(X = -2, Y = 1) + p(X = 0, Y = 1) + p(X = 2, Y = 1)$$

$$p(Y = -1) = p(X = -2, Y = -1) + p(X = 0, Y = -1) + p(X = 2, Y = -1)$$

$X \backslash Y$	1	-1	$P(X=x_i)$
-2	1/4	0	1/4
0	0	1/2	1/2
2	1/4	0	1/4
$P(Y=y_j)$	1/2	1/2	

1. Probabilité

- **Probabilité conditionnelle?**

1. Probabilité

- **Probabilité conditionnelle:** La probabilité d'un évènement x sachant qu'un autre évènement y est déjà arrivé.

$$P(X = x|Y = y) = \frac{p(X = x, Y = y)}{P(Y = y)}$$

Exemple:

Un laboratoire pharmaceutique a réalisé des tests sur 800 patients atteints d'une maladie. Certains sont traités avec le médicament A, d'autres avec le médicament B. Le tableau présente les résultats de l'étude :

	Médicament A	Médicament B	Total
Guéri	383	291	674
Non guéri	72	54	126
Total	455	345	800

On choisit un patient au hasard: La probabilité que le patient ait pris le médicament A **sachant qu'il est guéri** :

La probabilité que le patient soit guéri **sachant qu'il a pris le médicament B**

1. Probabilité

- **Probabilité conditionnelle:** La probabilité d'un évènement x sachant qu'un autre évènement y est déjà arrivé.

$$P(X = x|Y = y) = \frac{p(X = x, Y = y)}{P(Y = y)}$$

Exemple:

Un laboratoire pharmaceutique a réalisé des tests sur 800 patients atteints d'une maladie. Certains sont traités avec le médicament A, d'autres avec le médicament B. Le tableau présente les résultats de l'étude :

	Médicament A	Médicament B	Total
Guéri	383	291	674
Non guéri	72	54	126
Total	455	345	800

On choisit un patient au hasard: La probabilité que le patient ait pris le médicament A **sachant qu'il est guéri** :

La probabilité que le patient soit guéri **sachant qu'il a pris le médicament B**

1. Probabilité

- **Probabilité conditionnelle:** La probabilité d'un évènement x sachant qu'un autre évènement y est déjà arrivé.

$$P(X = x|Y = y) = \frac{p(X = x, Y = y)}{P(Y = y)}$$

Exemple:

Un laboratoire pharmaceutique a réalisé des tests sur 800 patients atteints d'une maladie. Certains sont traités avec le médicament A, d'autres avec le médicament B. Le tableau présente les résultats de l'étude :

	Médicament A	Médicament B	Total
Guéri	383	291	674
Non guéri	72	54	126
Total	455	345	800

On choisit un patient au hasard: La probabilité que le patient ait pris le médicament A **sachant qu'il est guéri** :

$$p(A|gueri) = \frac{p(A, gueri)}{p(gueri)} = \frac{383/800}{674/800} = 57\%$$

La probabilité que le patient soit guéri **sachant qu'il a pris le médicament B**

$$p(gueri|B) = \frac{p(B, gueri)}{p(B)} = \frac{291/800}{345/800} = 84\%$$

1. Probabilite

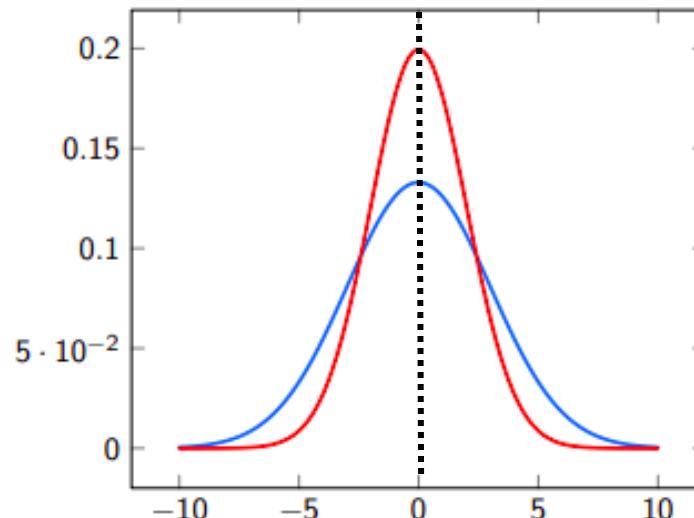
- **Espérance?**

1. Probabilité

- **Espérance:** Quelle est la valeur attendue de x selon une probabilité $p(x)$. (Quelle est la valeur la plus probable ?)

$$\mathbb{E}_{x \sim P}(x)$$

Que pouvez-vous dire des espérances relatives de ces densités ?



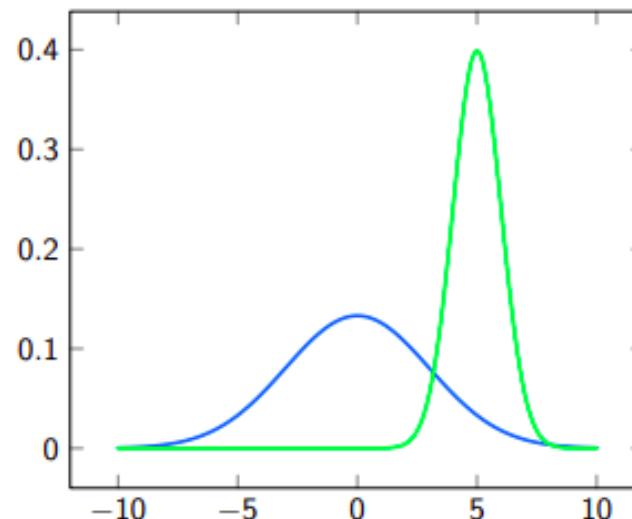
$$\mathbb{E}_{x \sim P_{rouge}}(x) = \mathbb{E}_{x \sim P_{bleu}}(x)$$

1. Probabilité

- **Espérance:** Quelle est la valeur attendue de x selon une probabilité $p(x)$. (Quelle est la valeur la plus probable ?)

$$\mathbb{E}_{x \sim P}(x)$$

Que pouvez-vous dire des espérances relatives de ces densités ?

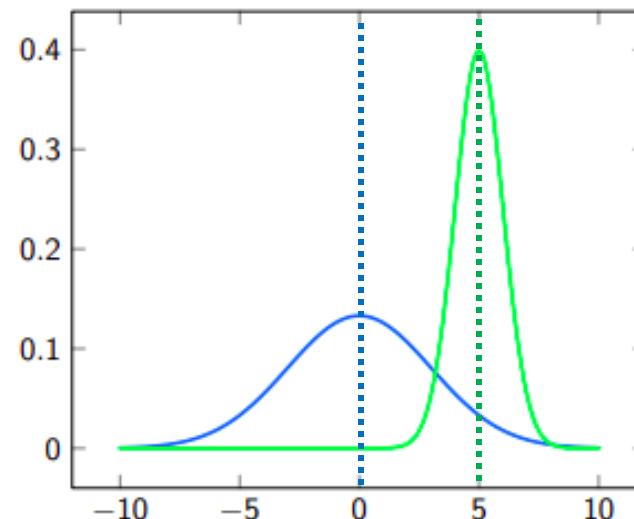


1. Probabilité

- **Espérance:** Quelle est la valeur attendue de x selon une probabilité $p(x)$. (Quelle est la valeur la plus probable ?)

$$\mathbb{E}_{x \sim P}(x)$$

Que pouvez-vous dire des espérances relatives de ces densités ?



$$\mathbb{E}_{x \sim P_{bleu}}(x) < \mathbb{E}_{x \sim P_{vert}}(x)$$

1. Probabilité

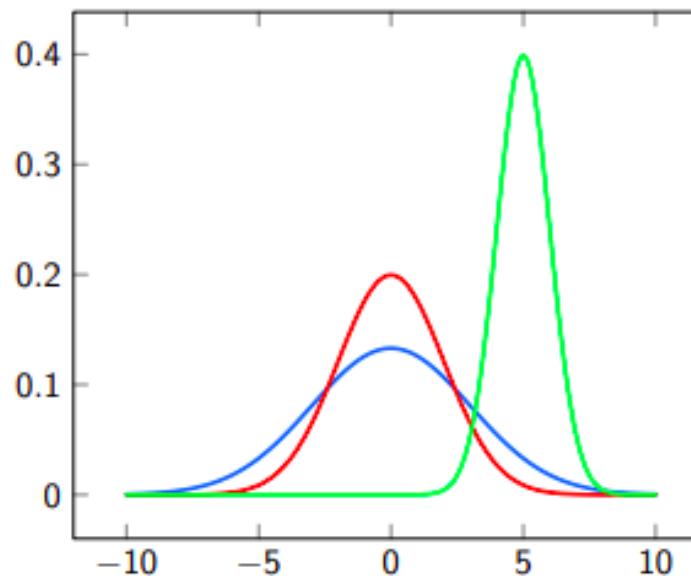
- Variance?

1. Probabilité

- **Variance:** Donne une idée de la variation des valeurs de $f(x)$

$$Var(x)$$

Que pouvez-vous dire des variances de ces densités ?

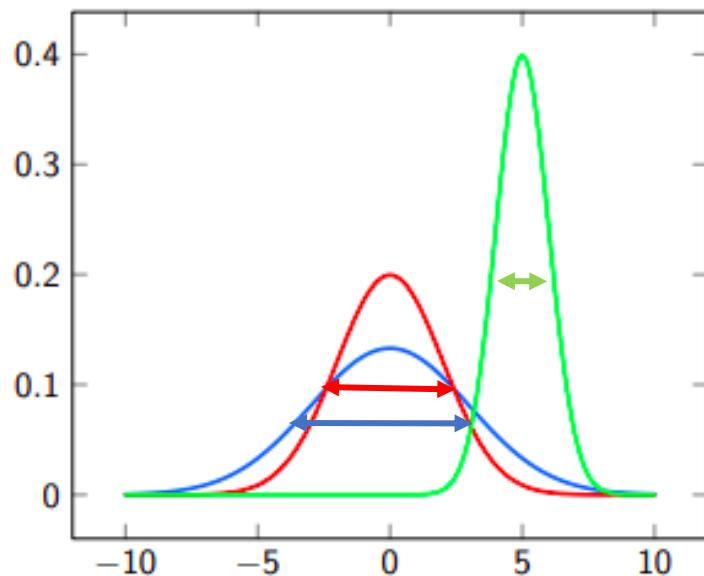


1. Probabilité

- **Variance:** Donne une idée de la variation des valeurs de x autour de son espérance.

$$Var(x)$$

Que pouvez-vous dire des variances de ces densités ?

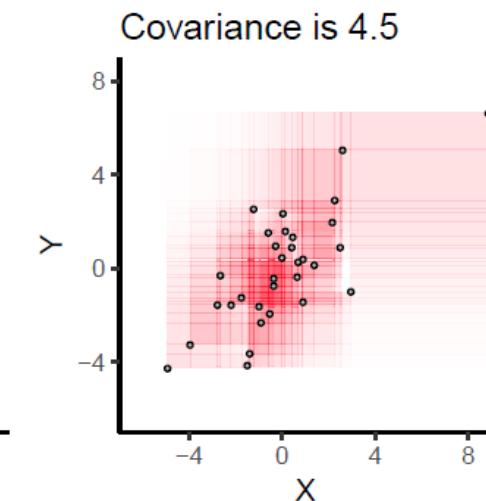
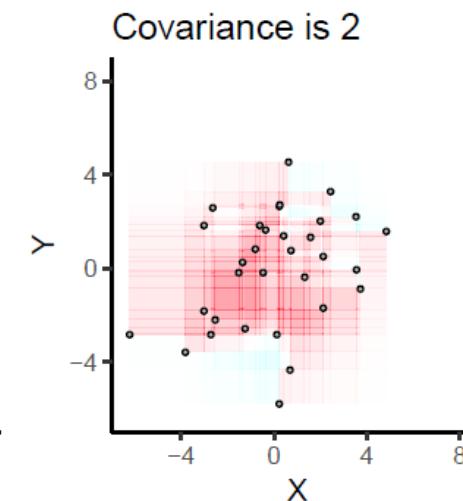
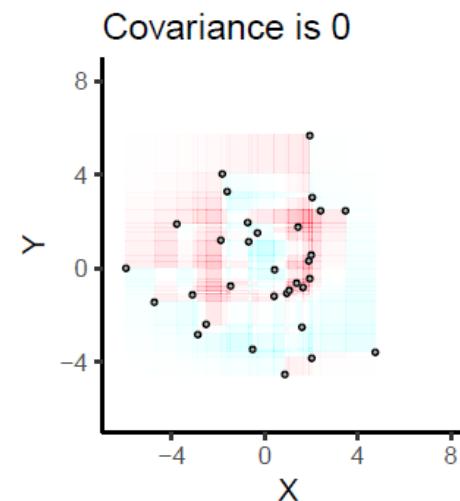
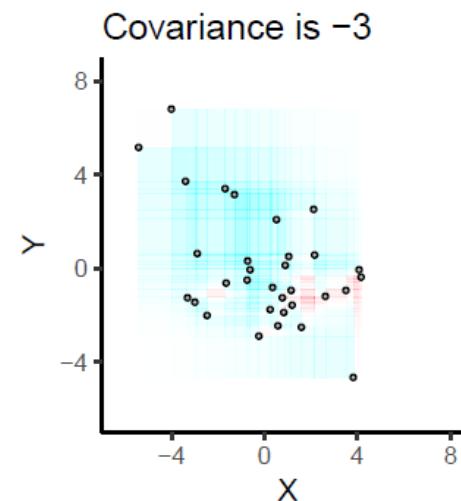
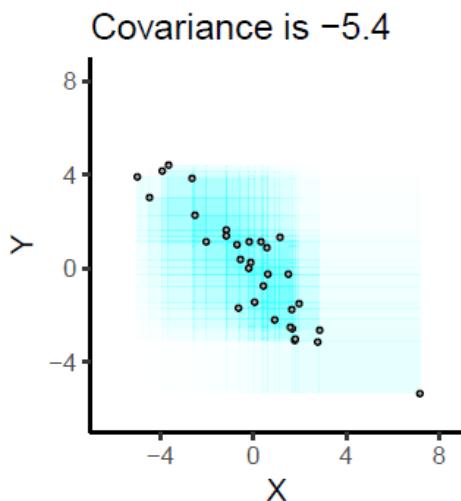


$$Var_{vert}(x) < Var_{rouge}(x) < Var_{bleu}(x)$$

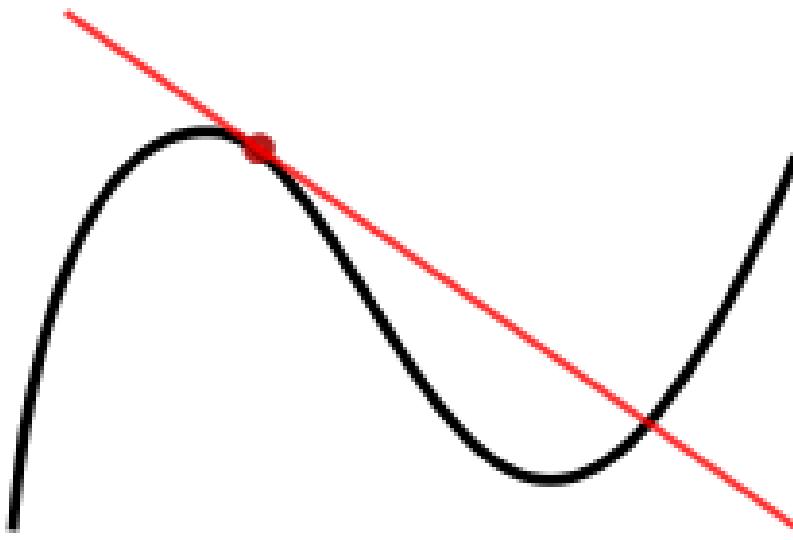
1. Probabilité

- **Covariance:** A quel point x et y sont-ils linéairement correlés ?

$$\text{Cov}(x, y)$$



2. Rappel dérivée

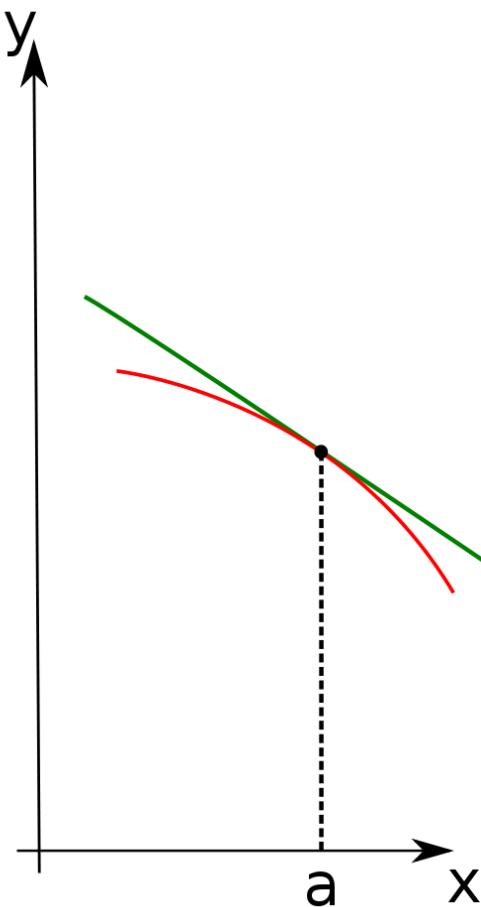
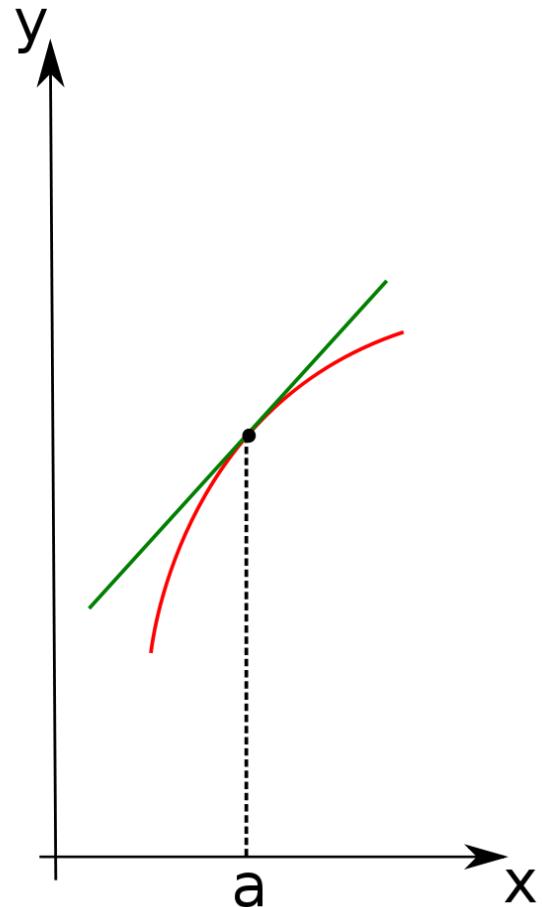


Dérivée = pente

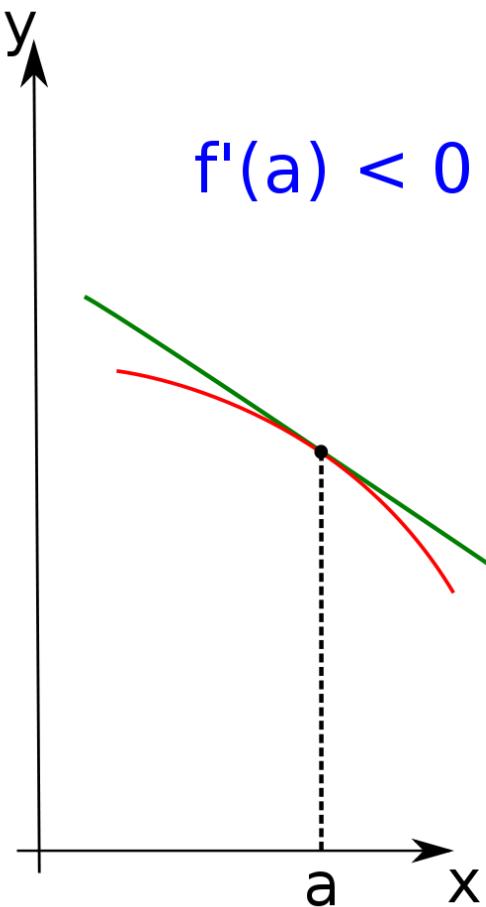
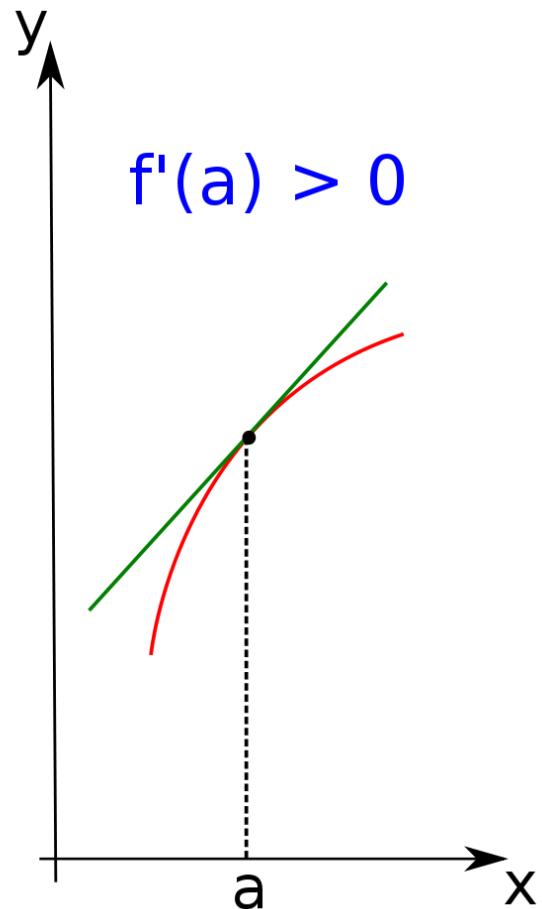
Dérivées des fonctions usuelles

Fonction	Dérivée
$x^n, n \in \mathbb{N}^*$	nx^{n-1}
$\frac{1}{x}$	$-\frac{1}{x^2}$
$\frac{1}{x^n}, n \in \mathbb{N}^*$	$-\frac{n}{x^{n+1}}$
$x^n, n \in \mathbb{Z}^*$	nx^{n-1}
\sqrt{x}	$\frac{1}{2\sqrt{x}}$
e^x	e^x
$\ln(x)$	$\frac{1}{x}$
$\sin(x)$	$\cos(x)$
$\cos(x)$	$-\sin(x)$

2. Rappel dérivée

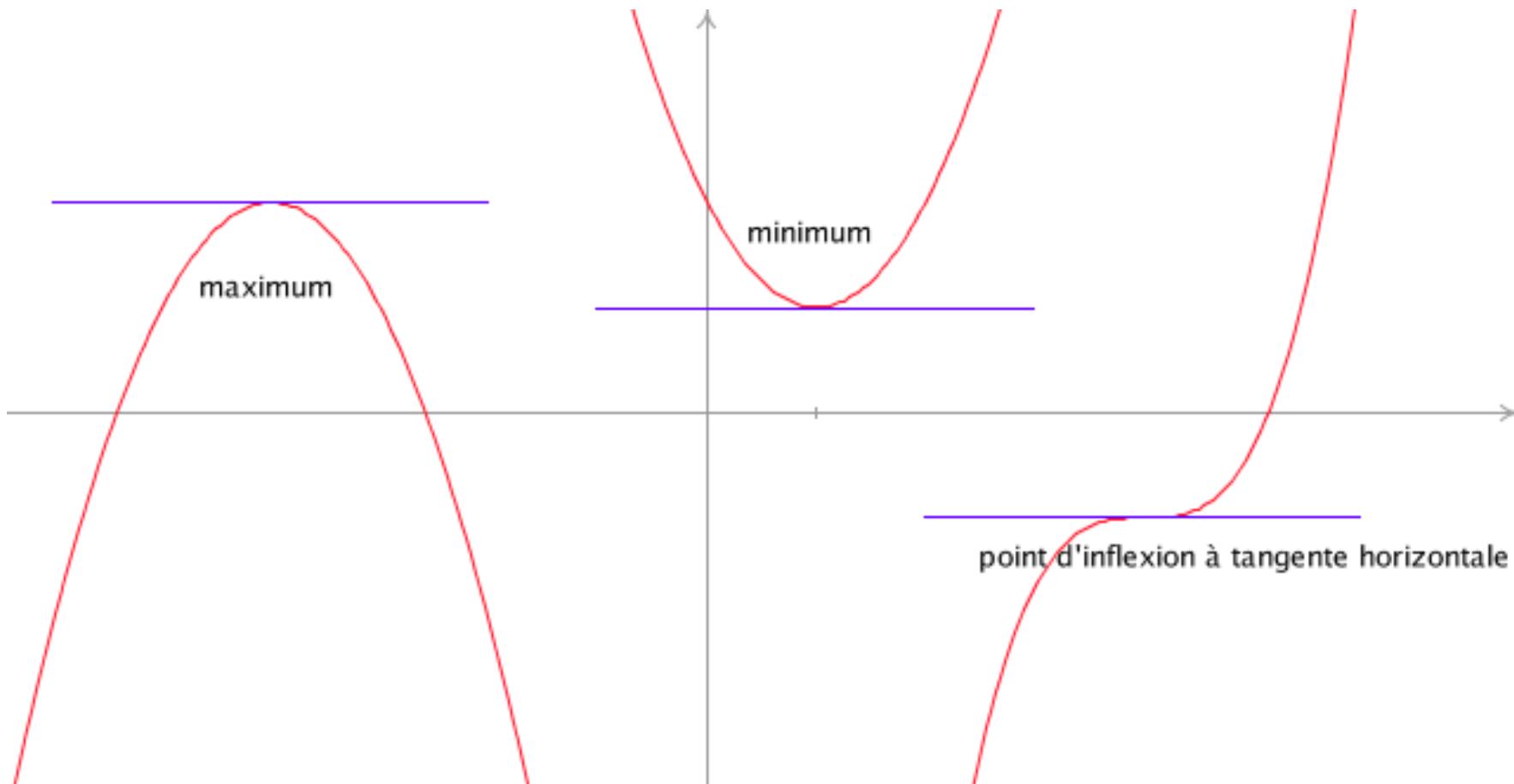


2. Rappel dérivée



2. Rappel dérivée

$$f'(a) = 0$$



3. Multidimensionnel

- Scalaire: un nombre réel, x

Scalar

3. Multidimensionnel

- Scalaire: un nombre réel, x
- Vecteurs: plusieurs nombre réels sous forme de liste

Scalar Vector

1

[
1
2]

3. Multidimensionnel

- Scalaire: un nombre réel, x
- Vecteurs: plusieurs nombre réels sous forme de liste
- Matrices: plusieurs nombre réels sous forme de tableau (dimension 2)

Scalar Vector Matrix

1

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

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

3. Multidimensionnel

- Scalaire: un nombre réel, x
- Vecteurs: plusieurs nombre réels sous forme de liste
- Matrices: plusieurs nombre réels sous forme de tableau (dimension 2)
- Tensors: plusieurs nombre réels sous forme de tableau de plus de deux dimensions

	Scalar	Vector	Matrix	Tensor
1		$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 2 \end{bmatrix} \\ \begin{bmatrix} 1 & 7 \end{bmatrix} & \begin{bmatrix} 5 & 4 \end{bmatrix} \end{bmatrix}$

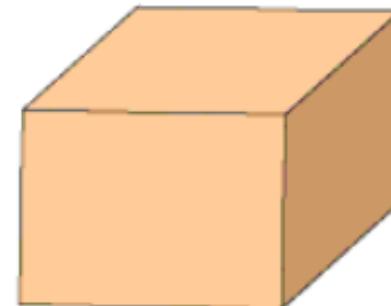
3. Multidimensionnel



1 d - Tensor



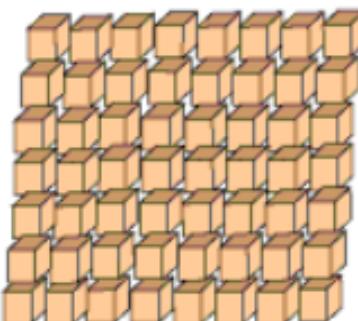
2 d - Tensor



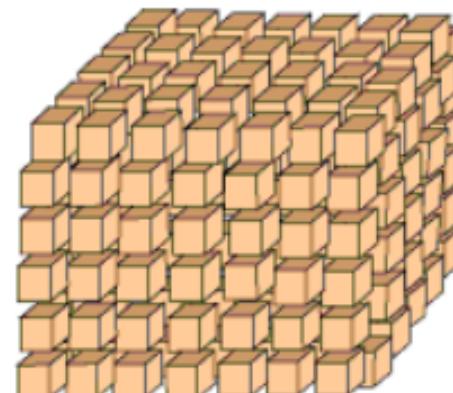
3 d - Tensor



4 d - Tensor



5 d - Tensor



6 d - Tensor

3. Multidimensionnel: fonctions

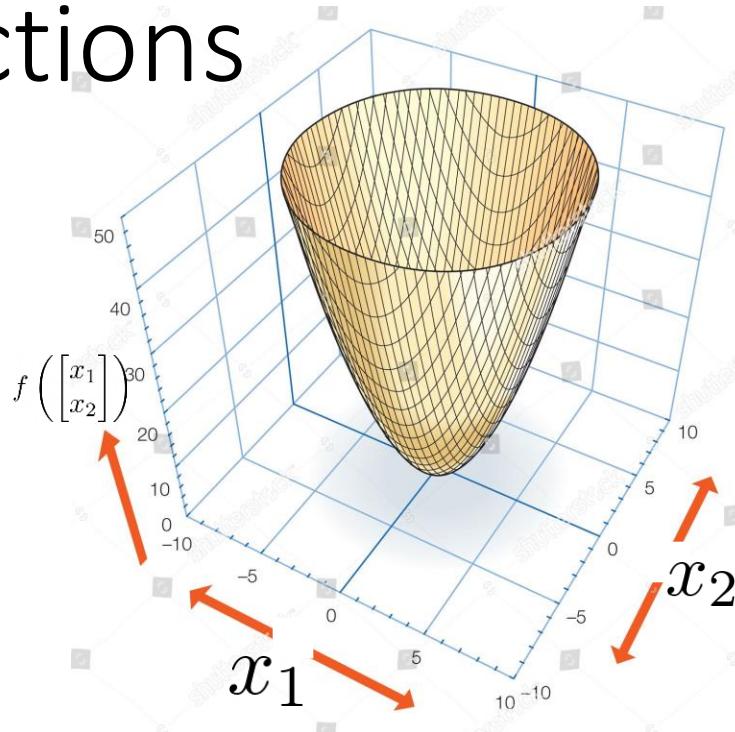
- Fonction de vecteurs:

$$f \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) = x_1^2 + x_2^2$$

- Fonctions de matrices:

$$f \left(\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} \right) = x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} \quad \text{"somme"}$$

$$f \left(\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \right) = x_{11} + x_{22} + x_{33} \quad \text{"trace"}$$



4. Introduction du gradient

Scalar-valued multivariable function

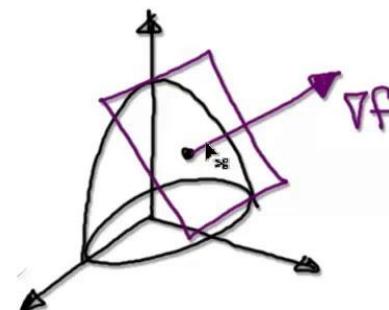
$$\nabla f(x_0, y_0, \dots) = \begin{bmatrix} \frac{\partial f}{\partial x}(x_0, y_0, \dots) \\ \frac{\partial f}{\partial y}(x_0, y_0, \dots) \\ \vdots \end{bmatrix}$$

Notation for gradient, called "nabla".

∇f takes the same type of inputs as f .

∇f outputs a vector with all possible partial derivatives of f .

$$f(x, y, z) = 3x^2 + 2y^2 + z - 20$$



$$\nabla f(x, y, z) = ?$$

4. Introduction du gradient

Scalar-valued multivariable function

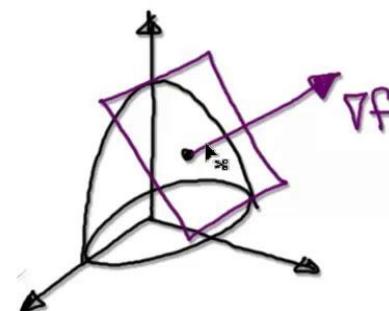
$$\nabla f(x_0, y_0, \dots) = \begin{bmatrix} \frac{\partial f}{\partial x}(x_0, y_0, \dots) \\ \frac{\partial f}{\partial y}(x_0, y_0, \dots) \\ \vdots \end{bmatrix}$$

Notation for gradient, called "nabla".

∇f takes the same type of inputs as f .

∇f outputs a vector with all possible partial derivatives of f .

$$f(x, y, z) = 3x^2 + 2y^2 + z - 20$$



$$\nabla f(x, y, z) = \begin{bmatrix} 6x \\ 4y \\ 1 \end{bmatrix}$$

4. Introduction du gradient

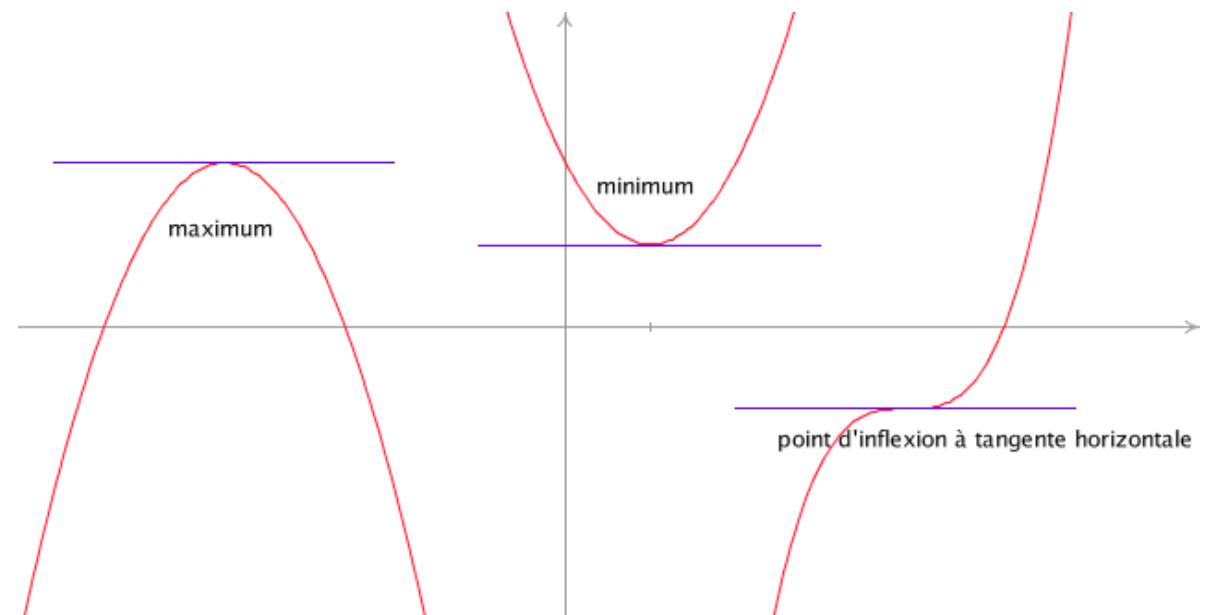
$$\nabla f(x, y, z) = 0$$

Scalar-valued multivariable function

$$\nabla f(x_0, y_0, \dots) = \begin{bmatrix} \frac{\partial f}{\partial x}(x_0, y_0, \dots) \\ \frac{\partial f}{\partial y}(x_0, y_0, \dots) \\ \vdots \\ \underbrace{\quad}_{\nabla f \text{ outputs a vector with all possible partial derivatives of } f.} \end{bmatrix}$$

∇f takes the same type of inputs as f .

Notation for gradient, called "nabla".



A savoir pour la suite: qu'est-ce qu'une ... ?

- Probabilité conditionnelle
- Espérance et variance
- Un gradient
- Comment caractériser le minimum ou le maximum d'une fonction avec son gradient?

Commencons !

L'IA n'existe pas, on parlera d'algorithme d'apprentissage.

I. Algorithme d'apprentissage

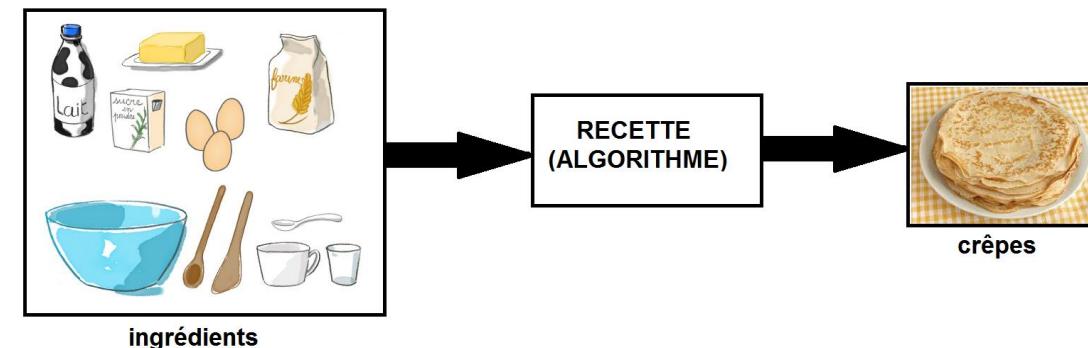
DEF: Un algorithme d'apprentissage est un algorithme capable d'apprendre a partir de données.

I. Algorithme d'apprentissage

DEF: Un algorithme d'apprentissage est un **algorithme** capable d'apprendre a partir de données.

DEF: Un **algorithme** est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre une classe de problèmes.
(Wikipedia)

Ex: recette de cuisine

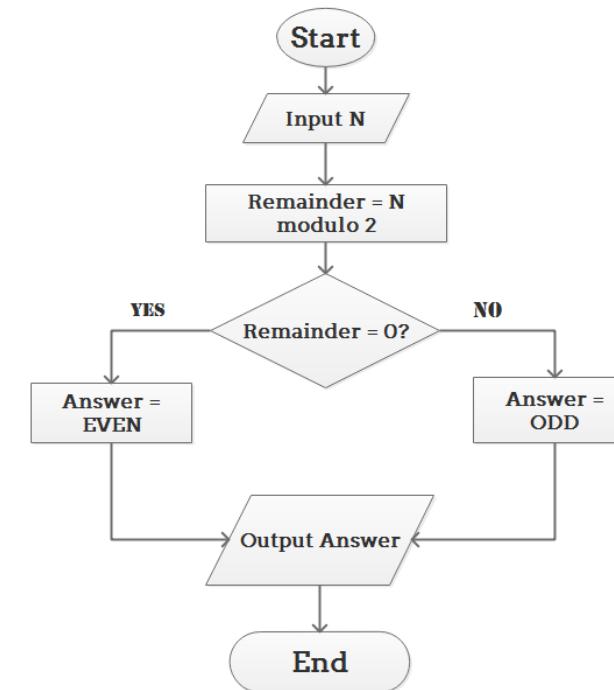


I. Algorithme d'apprentissage

DEF: Un algorithme d'apprentissage est un **algorithme** capable d'apprendre à partir de données.

DEF: Un **algorithme** est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre une classe de problèmes.
(Wikipedia)

Ex: recette de cuisine



I. Algorithme d'apprentissage

DEF: Un algorithme d'apprentissage est un **algorithme** capable d'apprendre à partir de données.

?????????????

DEF: Un **algorithme** est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre une classe de problèmes.
(Wikipedia)

Ex: recette de cuisine

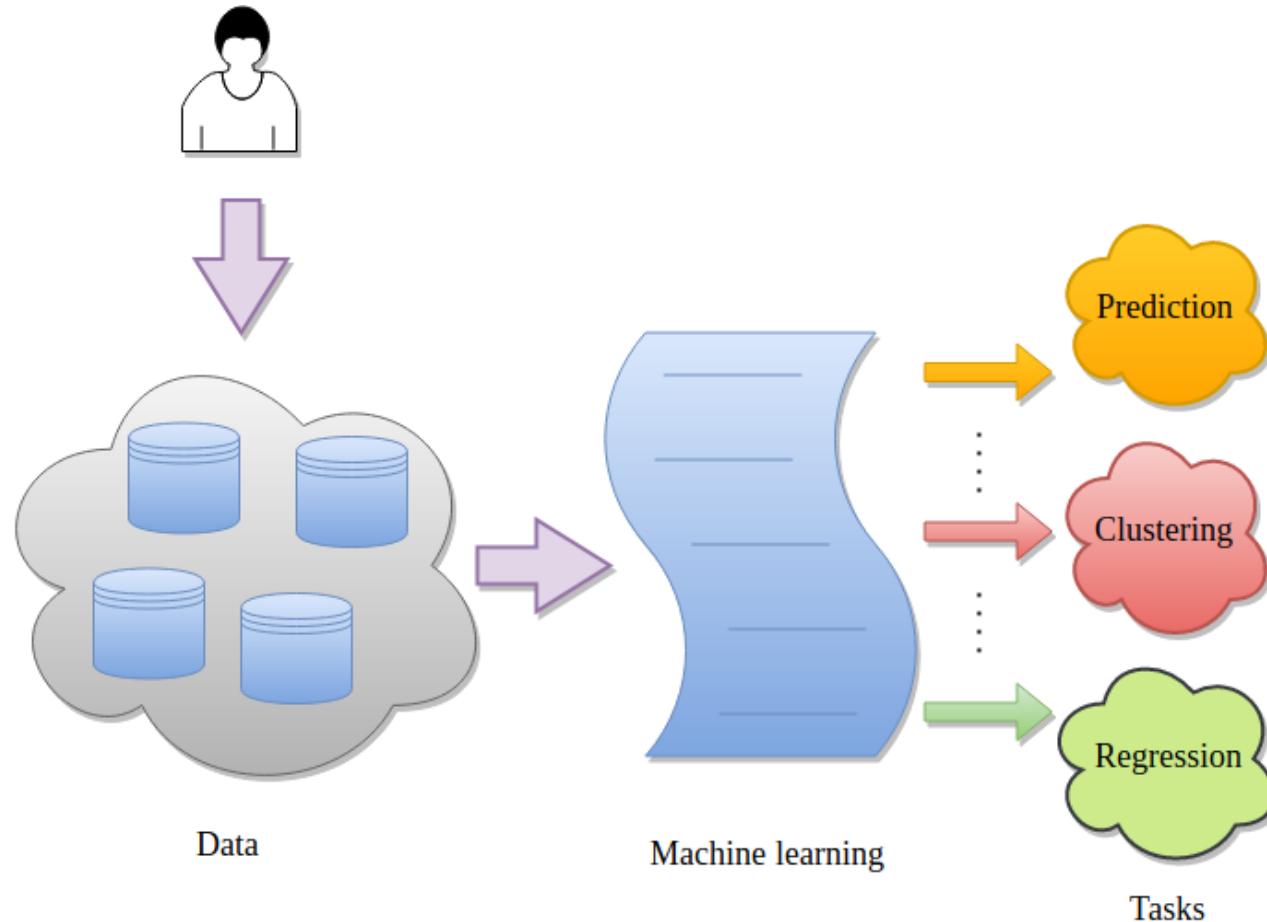
"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" (Mitchell, 1997)

MACHINE LEARNING

Par exemple:

Un élève améliore ses capacités en calcul mental (Tache T) en étudiant ses leçons et en faisant des exercices (expérience E). On observe l'amélioration de ses notes (performance P).

1. Les tâches: T



Quelle est la tâche à réaliser à partir des données collectées ?

Classification

Données:

Différents vins et bières mélangés.

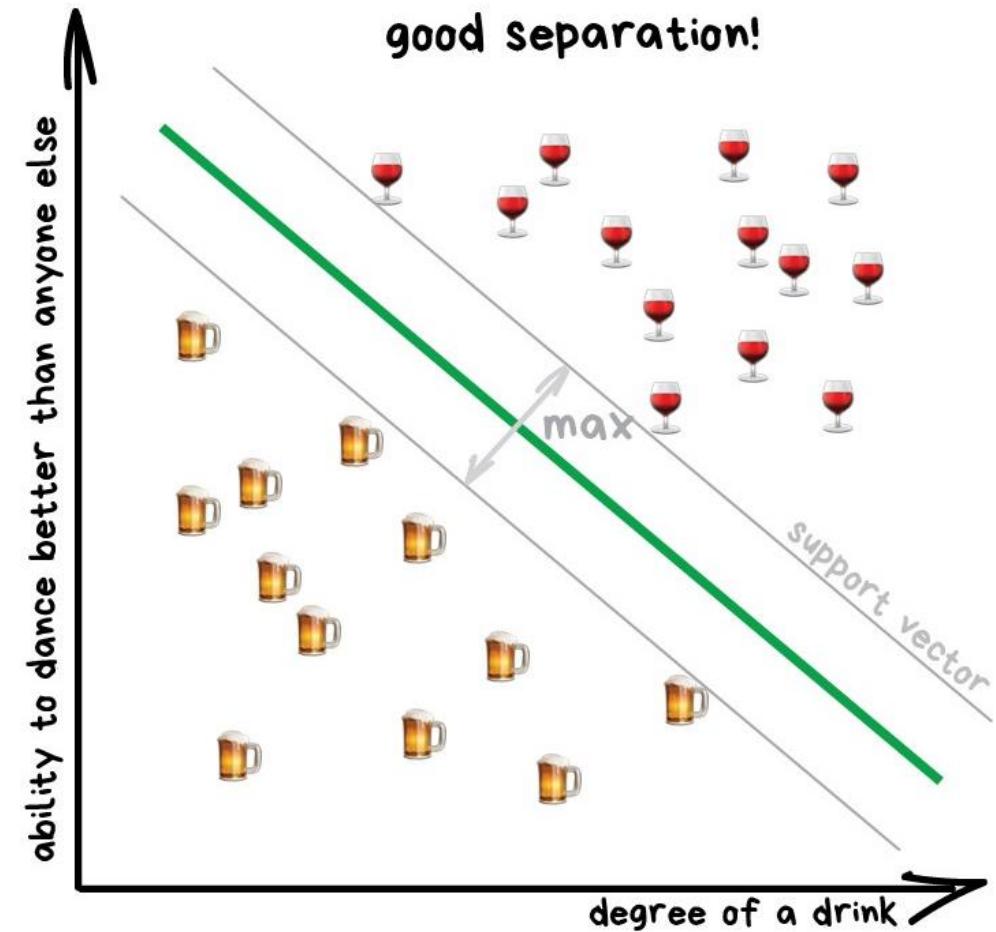
Les variables (features):

- Degré d'alcool
- Capacité à faire danser

Objectifs – la tâche:

Reconnaitre le type d'alcool à partir des features.

Exemple plus sérieux: diagnostiquer un cancer du poumon à partir d'un scanner. (Ardila et al., 2019)



Source :Machine Learning for Everyone

Classification avec données manquantes

Donnees: Patients

Variables: température, mal de tête, nausée

Tache: Le patient a-t'il la grippe (flu)?

Cases	Temperature	Headache	Nausea	Decision (flu)	
x_1	High	*	No	Yes	
x_2	Very high	Yes	Yes	Yes	
x_3	*	No	No	No	Très fréquent dans le domaine médical:
x_4	High	Yes	Yes	Yes	
x_5	High	*	Yes	No	Antécédents inconnus.
x_6	Normal	Yes	No	No	Examens couteux et invasifs.
x_7	Normal	No	Yes	No	Arrêt du suivi
x_8	*	Yes	*	Yes	...

Regression

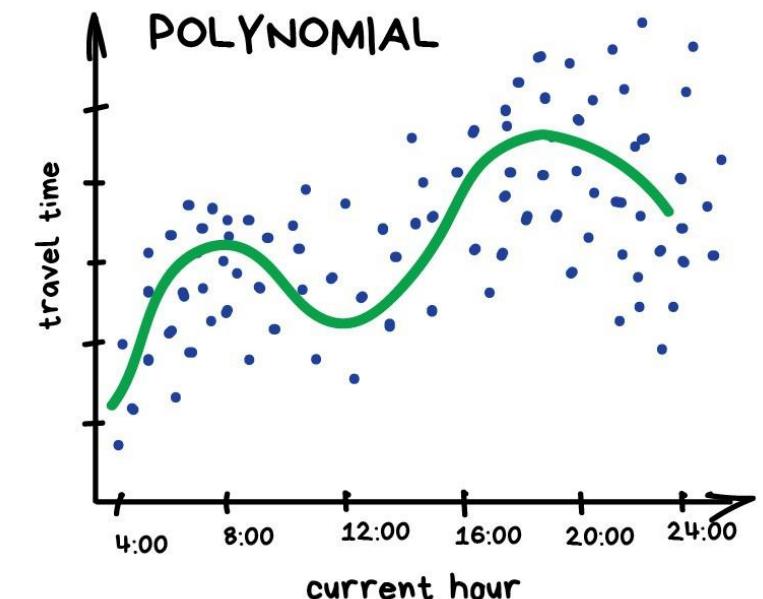
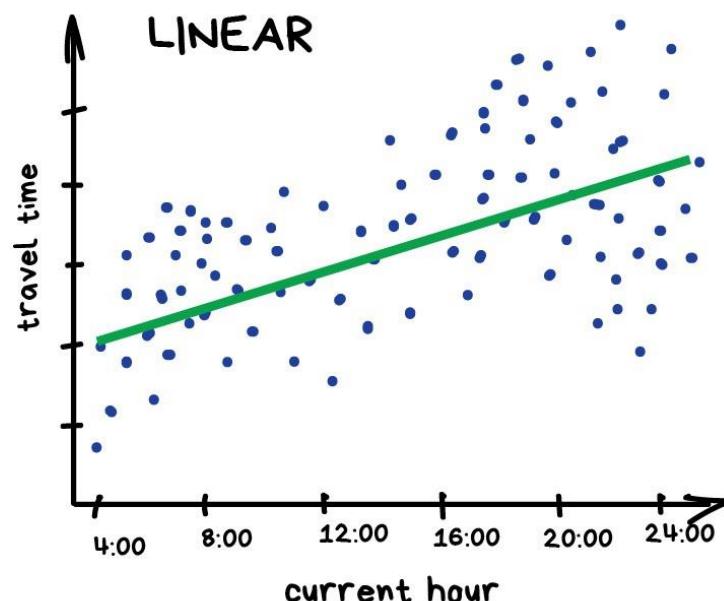
Données: moment de la journée

Variables: heure actuelle

Tache: Quel va etre le temps de trajet entre mon travail et chez moi ?

Ici, les valeurs du labels sont continues, nous ne sommes plus dans de la classification mais de la regression

PREDICT TRAFFIC JAMS

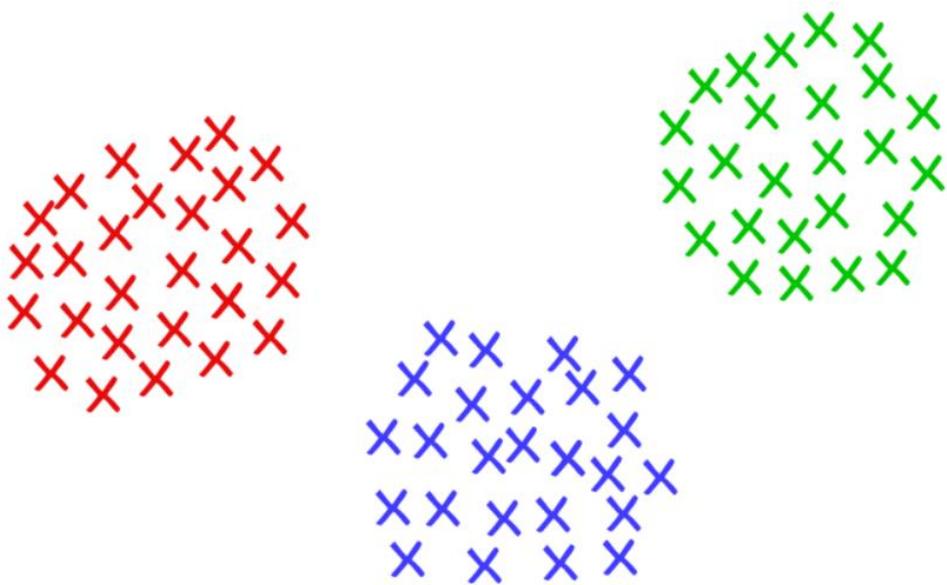


Source: Machine Learning for Everyone

REGRESSION

Exemple plus sérieux: Prédire le temps de survie d'un patient

Clustering



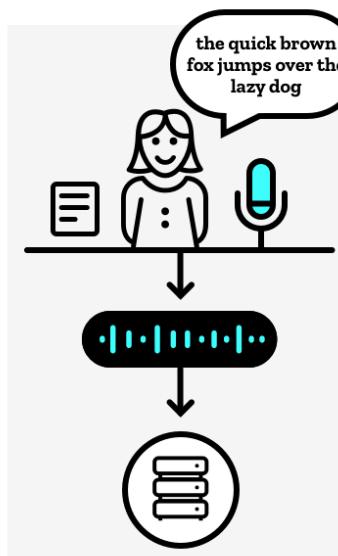
On a des données mais pas de label. On cherche à grouper des données ensemble.

Autres tâches

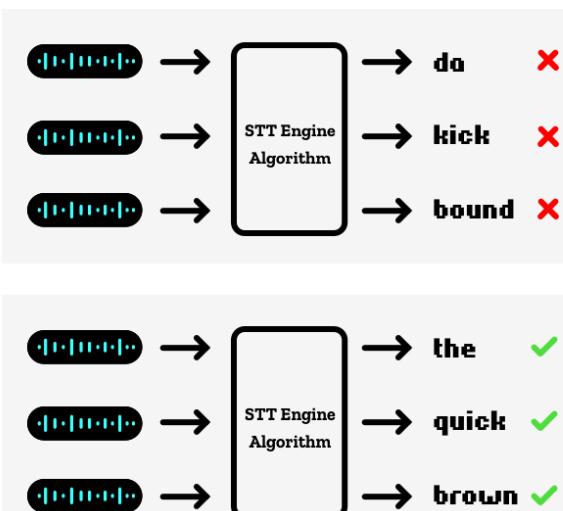
- Transcription exemple: Google Home, Alexia, ...

How a Speech Application Learns

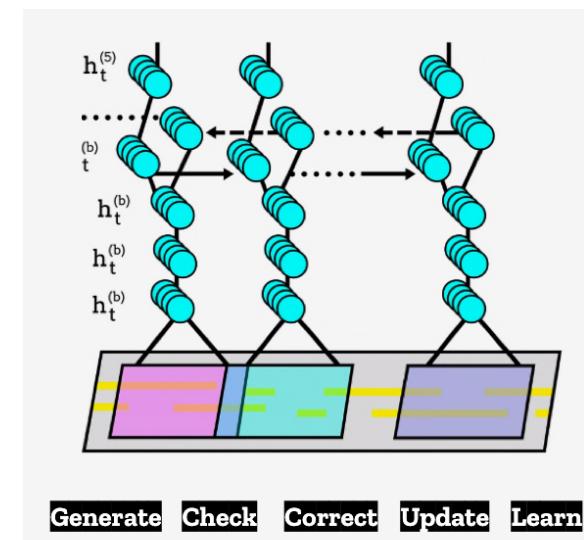
Step 1. Record voices



Step 2. Input voice data



Step 3. Train the speech algorithm



Common Voice Project

Open Source STT Engine

Deep Learning Architecture

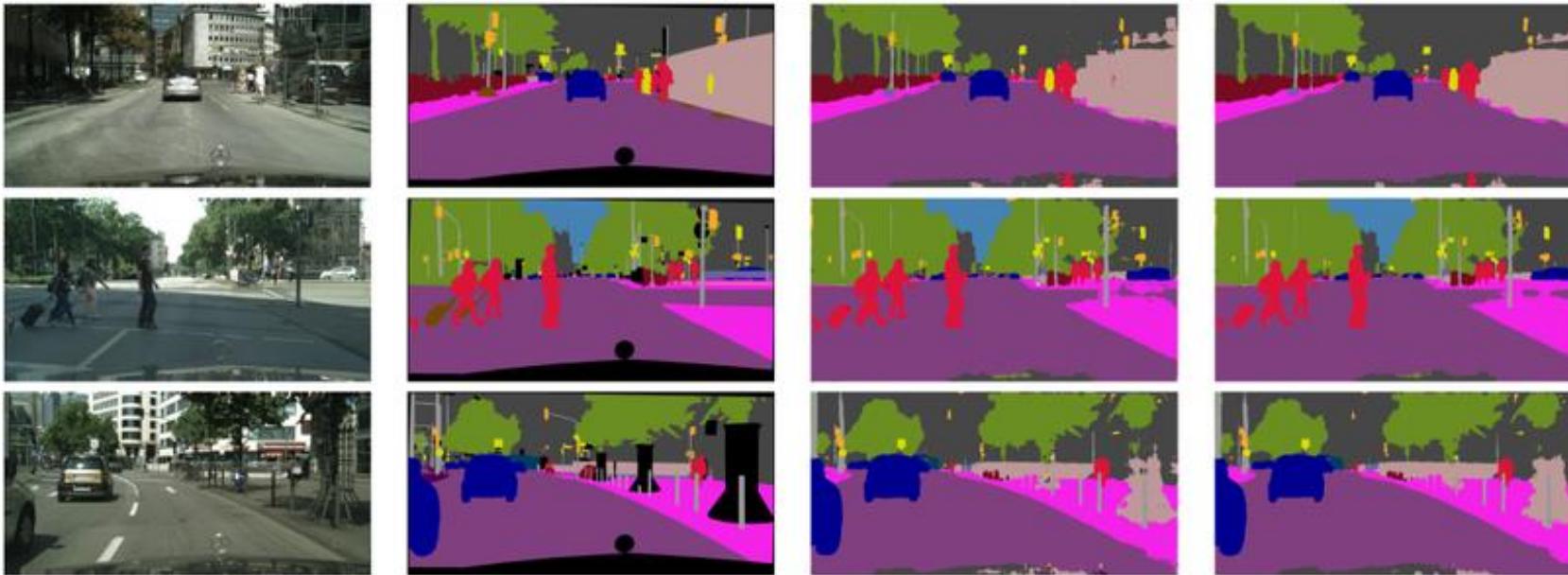
Autres tâches

- Machine translation ex: Google traduction, Deepl



Autres tâches

- Structured output ex: segmentation, description de situation



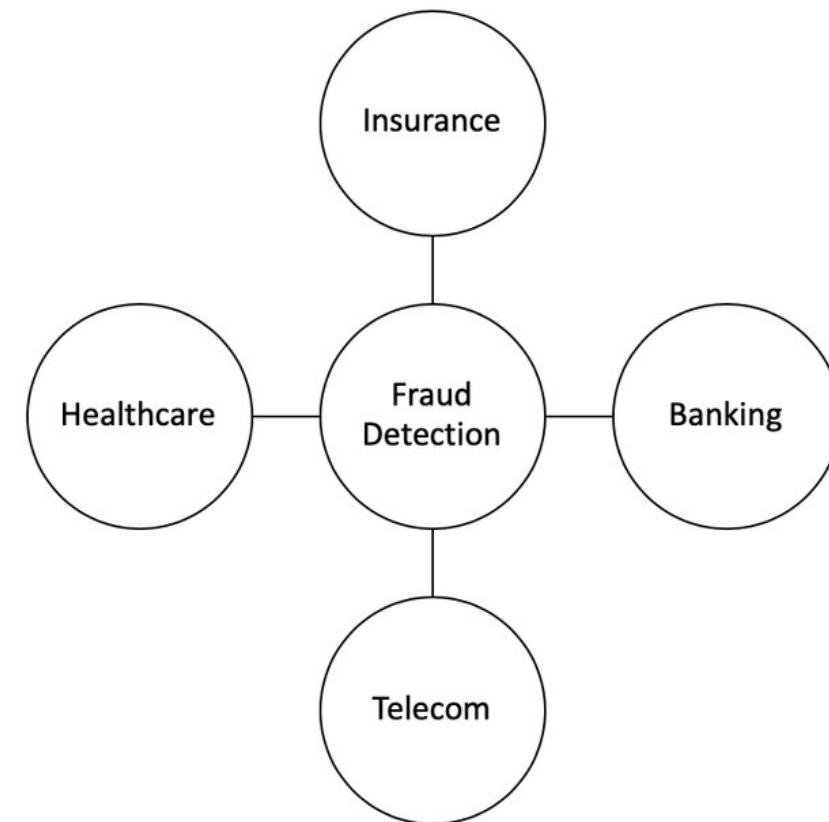
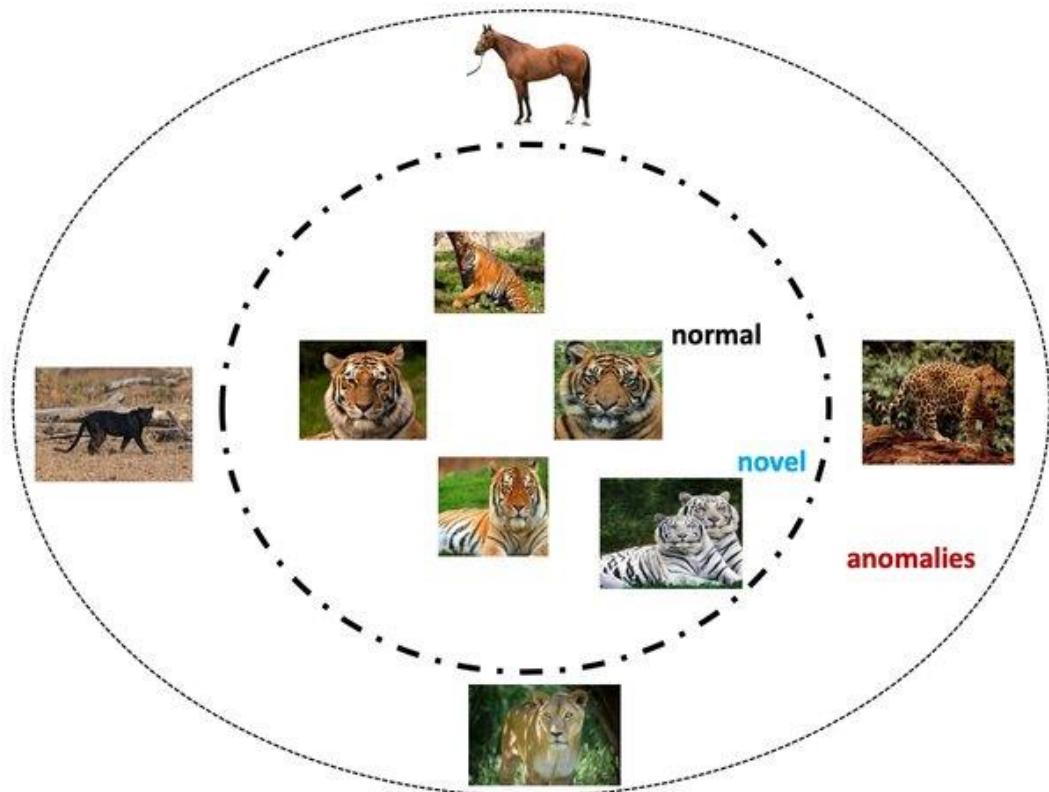
Autres tâches

- Structured output ex: segmentation, description de situation

Describes without errors	Describes with minor errors	Somewhat related to the image	Unrelated to the image
 A person riding a motorcycle on a dirt road.	 Two dogs play in the grass.	 A skateboarder does a trick on a ramp.	 A dog is jumping to catch a frisbee.
 A group of young people playing a game of frisbee.	 Two hockey players are fighting over the puck.	 A little girl in a pink hat is blowing bubbles.	 A refrigerator filled with lots of food and drinks.
 A herd of elephants walking across a dry grass field.	 A close up of a cat laying on a couch.	 A red motorcycle parked on the side of the road.	 A yellow school bus parked in a parking lot.

Autres tâches

- Anomaly detection ex: fraudes fiscales



Autres tâches

- Synthesis and sampling ex: création de faux visages (<https://thispersondoesnotexist.com/>)
- Ou de chats (<https://thiscatdoesnotexist.com/>)



Karras et al., 2018

Autres tâches

- Synthesis and sampling ex: Imagen



A strawberry mug filled with white sesame seeds.
The mug is floating in a dark chocolate sea.



Teddy bears swimming at the Olympics 400m
Butterfly event.



A photo of a Corgi dog riding a bike in Times Square.
It is wearing sunglasses and a beach hat.

Autres tâches

- Imputation of missing values

	col1	col2	col3	col4	col5	
0	2	5.0	3.0	6	NaN	
1	9	NaN	9.0	0	7.0	
2	19	17.0	NaN	9	NaN	

mean()



	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

	col1	col2	col3	col4	col5	
0	2	5.0	3.0	6.0	NaN	
1	9	NaN	9.0	0.0	7.0	
2	19	17.0	NaN	9.0	NaN	
3	7	10.0	3.0	6.0	4.0	
4	2	8.0	10.0	NaN	3.0	

mice()

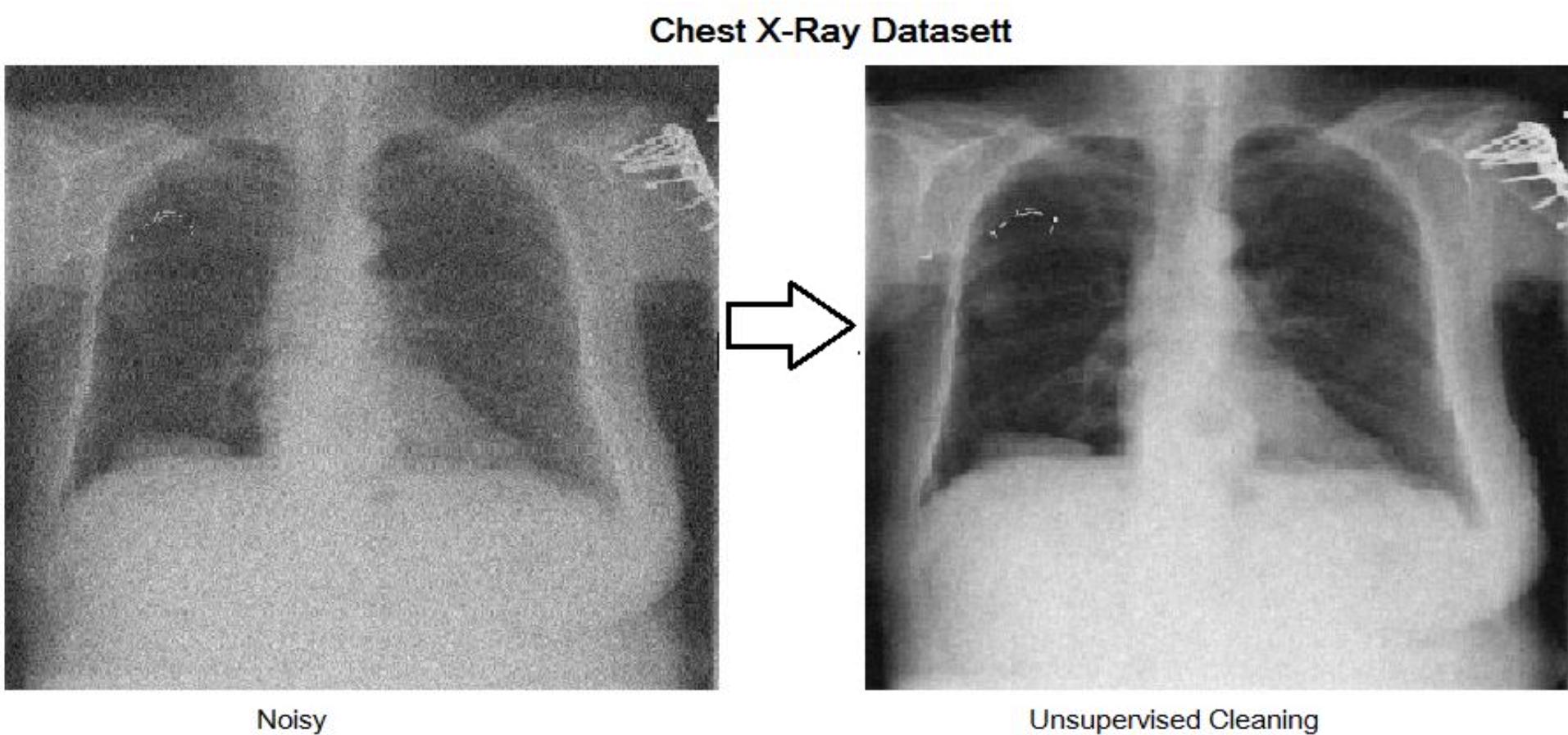


0	2.0	5.0	3.00	6.00	4.666667
1	9.0	10.0	9.00	0.00	7.000000
2	19.0	17.0	6.25	9.00	4.666667
3	7.0	10.0	3.00	6.00	4.000000
4	2.0	8.0	10.00	5.25	3.000000

MICE = Multivariate imputation by chained equations

Autres tâches

- Denoising

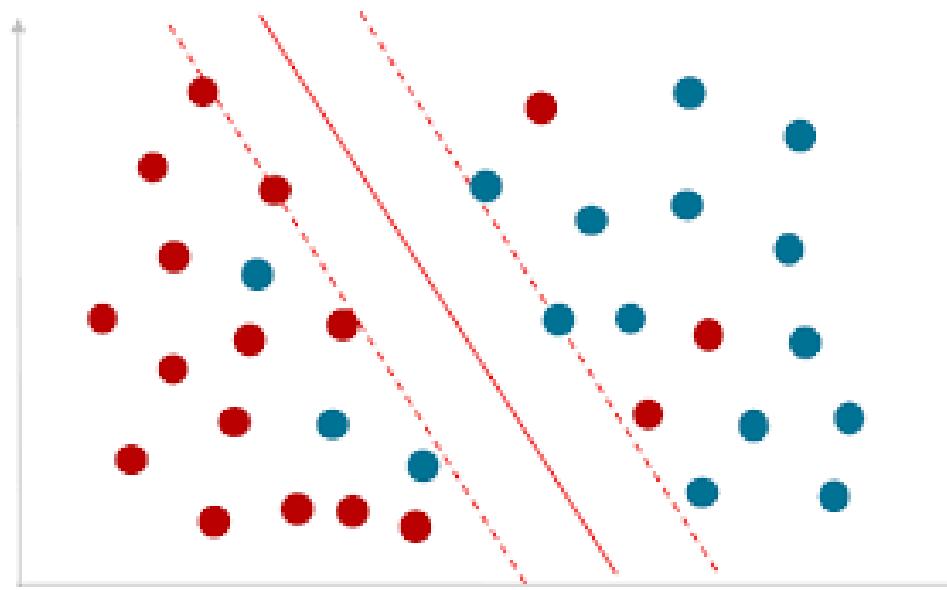


Résumé des tâches:

- Classification (avec ou sans données manquantes)
 - Régression
 - NLP = Natural language processing
 - Clustering
 - Détection d'anomalie
 - Génération de données
- ...

2. La mesure de performance, P

Comment juger de la performance de l'algorithme d'apprentissage ?

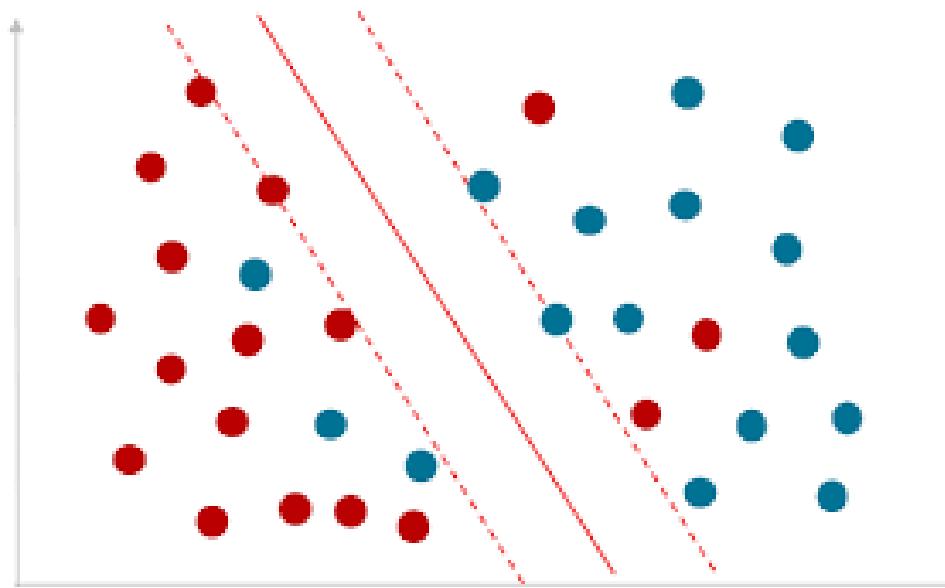


2. La mesure de performance, P

Comment juger de la performance de l'algorithme d'apprentissage ?

- Dépend de la tâche.

Quelle mesure de performance pour une classification ?

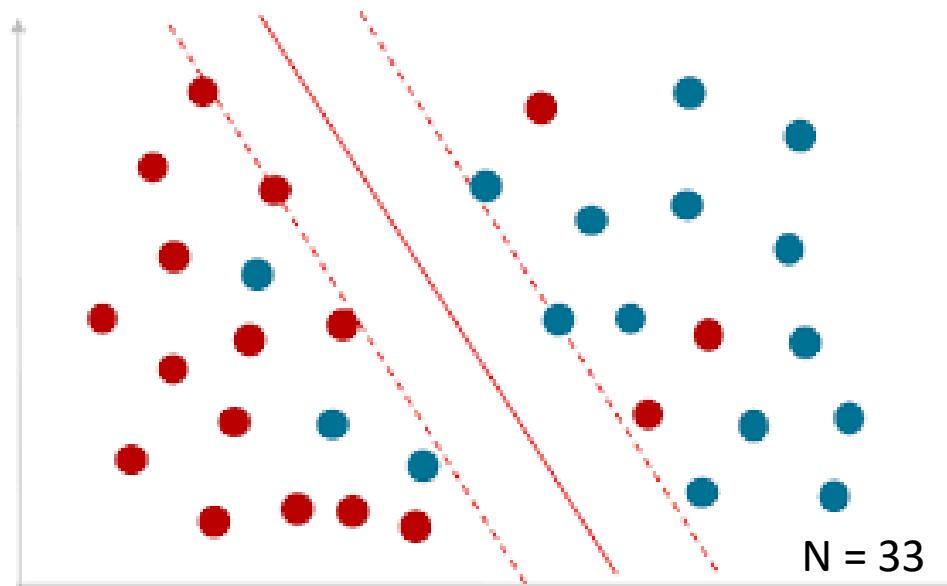


2. La mesure de performance, P

Comment juger de la performance de l'algorithme d'apprentissage ?

- Dépend de la tâche.

Quelle mesure de performance pour une classification ?



On utilisera souvent la précision/accuracy

Accuracy = proportion de points
correctement classifiés.

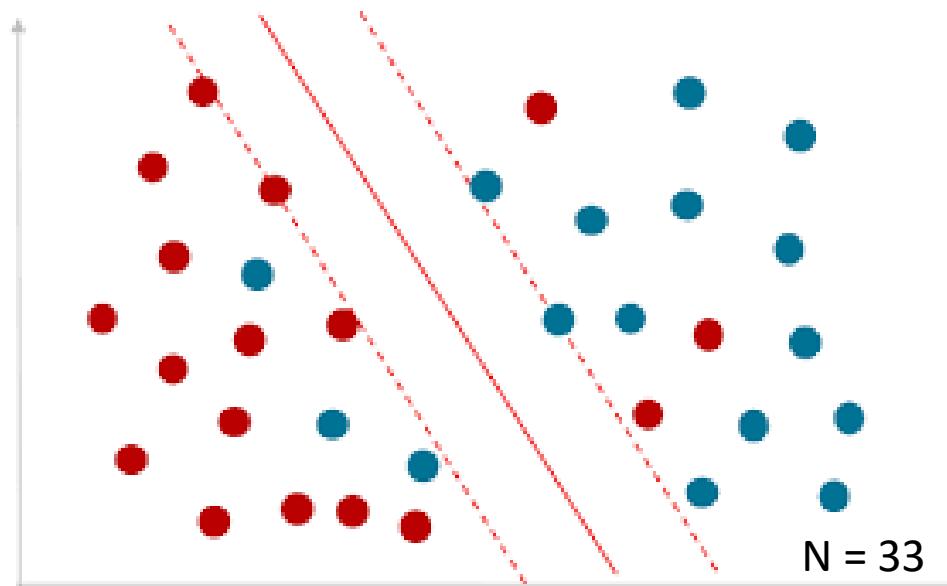
Quelle est la précision de ce modèle ?

2. La mesure de performance, P

Comment juger de la performance de l'algorithme d'apprentissage ?

- Dépend de la tâche.

Quelle mesure de performance pour une classification ?



On utilisera souvent la précision/accuracy

Accuracy = proportion de points
correctement classifiés.

Quelle est la précision de ce modèle ?

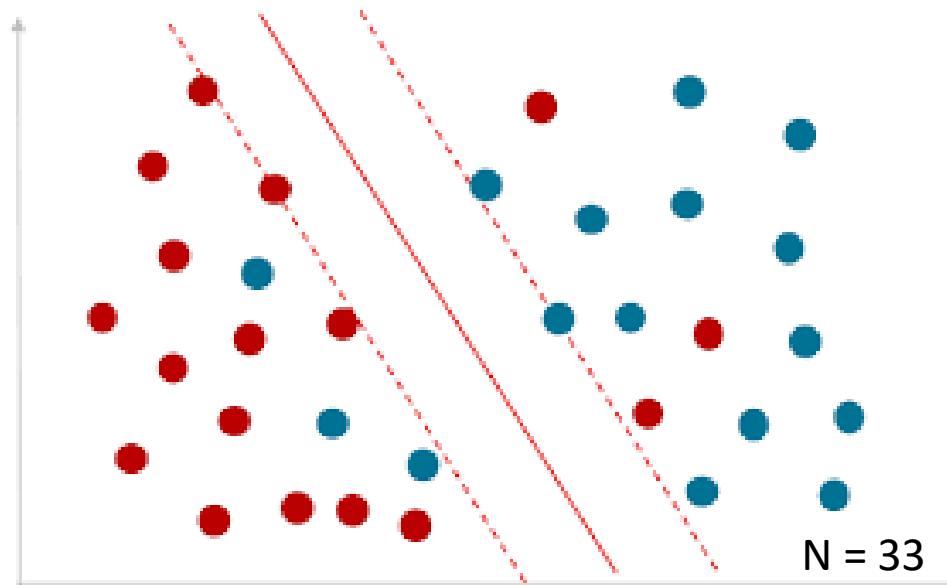
$$27/33 = 82\%$$

2. La mesure de performance, P

Comment juger de la performance de l'algorithme d'apprentissage ?

- Dépend de la tâche.

Quelle mesure de performance pour une classification ?



On utilisera souvent la précision/accuracy

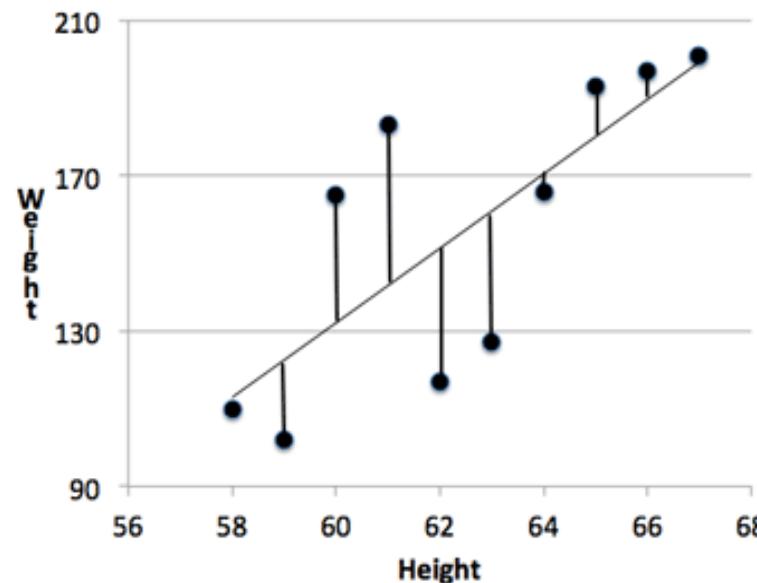
Mais pas que !
Pouvez vous en citer d'autres?

2. La mesure de performance, P

Comment juger de la performance de l'algorithme d'apprentissage ?

- Dépend de la tâche.

Quelle mesure de performance pour une regression?

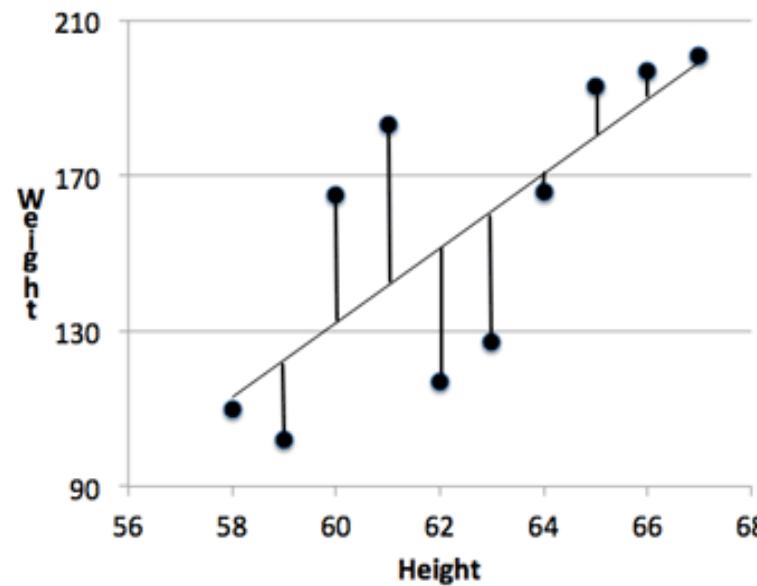


2. La mesure de performance, P

Comment juger de la performance de l'algorithme d'apprentissage ?

- Dépend de la tâche.

Quelle mesure de performance pour une regression?



On utilisera souvent la mean square error (MSE):

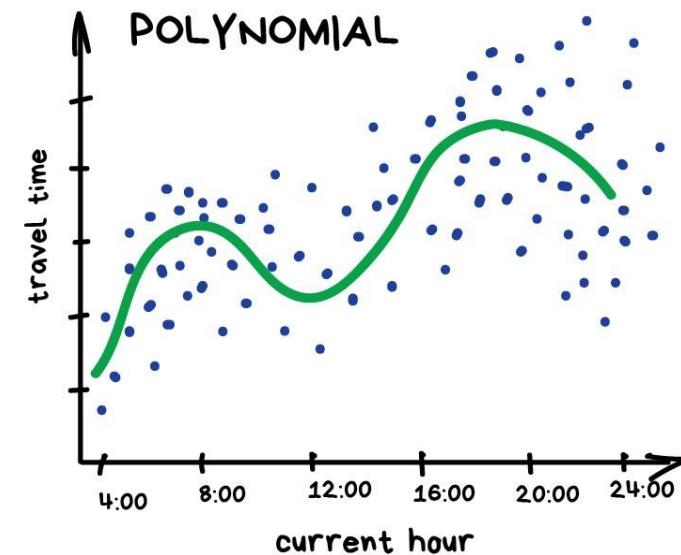
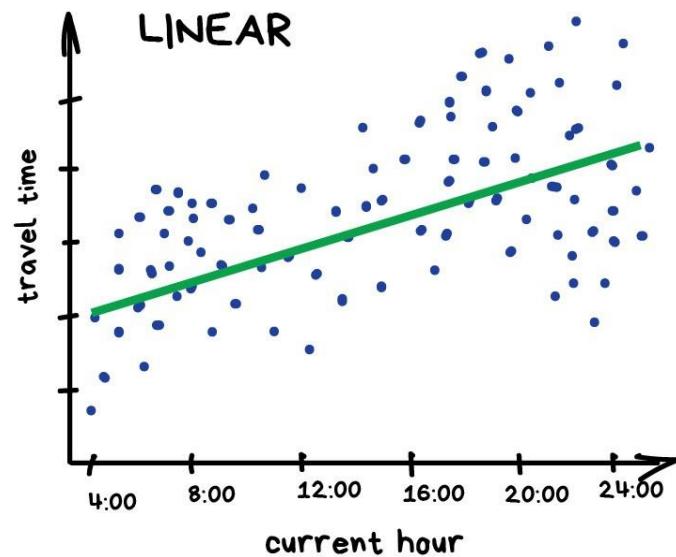
La somme des erreurs au carré

$$+ \left| \begin{array}{c} 2 \\ + \end{array} \right| \dots$$

2. La mesure de performance, P

Dépend de l'objectif: Doit on pénaliser l'erreur en moyenne ou simplement les erreurs grossières ?

PREDICT TRAFFIC JAMS



REGRESSION

2. La mesure de performance, P

Dépend de l'objectif: Doit on pénaliser plus les faux positifs ou les faux négatifs ?

- Médecine: Vaut-il mieux prédire qu'un patient sain a le cancer plutôt que de rater un patient malade et dire qu'il n'a pas le cancer ?
 - Dans le cas d'un dépistage généralisé ? (sang dans les selles)
 - Examen spécifique ? (coloscopie)

3. L'expérience, E

- ~1 trillion webpages

(<http://googleblog.blogspot.dk/2008/07/we-knew-web-was-big.html>)



- One hour of video is uploaded to youtube every second resulting in 10 years of content every day

(source: youtube)



- We have sequenced more than 1000 peoples genome of $3.8 \cdot 10^9$ base pairs

(source: K. P. Murphy "Machine Learning")

- Walmart handles more than 1 mio. transactions per hour and has databases containing more than $2.5 \cdot 10^{15}$ bytes of information

(source: K. P. Murphy "Machine Learning")



- Each night the worlds astronomy laboratories store high-resolution of the night sky of around a terabyte (10^{12})

(source: Stephen Marsland "Machine Learning An Algorithmic Perspective")

- In total, the four main detectors at the Large Hadron Collider (LHC) produced 13 petabytes (10^{15}) of data in 2010

(source: wikipedia "Big Data")

- Facebook handles 40 billion photos from its user base.

(source: wikipedia "Big Data")



- FICO Falcon Credit Card Fraud Detection System protects 2.1 billion active accounts world-wide

(source: wikipedia "Big Data")



3. L'expérience, E

- ▶ *Datum*

a characteristic or a number that may contain information about an objects, individuals, observations, populations

e.g. Age [years] = 31

- ▶ *Data*

multiple *datum* about one or multiple objects, individuals, etc

Without data \implies Without E \implies No machine learning!

3. L'expérience, E

- ▶ Multiple individuals or observations for the same quantity \Rightarrow variable, feature or attribute x
- ▶ Observed x for M individuals $x_1, \dots, x_M \Rightarrow$ feature vector \mathbf{x}

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}, \quad \text{e.g. Age in years of } M = 3 \text{ individuals } \mathbf{x}_A = \begin{bmatrix} 31 \\ 23 \\ 32 \end{bmatrix}$$

3. L'expérience, E

- Most cases we have multiple individuals **and** multiple variables
 $\implies N$ feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_N \implies$ feature matrix \mathbf{X}

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,N} \\ \vdots & \ddots & \\ x_{M,1} & & x_{M,N} \end{bmatrix}$$

- Columns are feature vectors
Rows are observation vectors $\implies \mathbf{X}$ is $M \times N$ matrix

e.g. Age in years \mathbf{x}_A and weight \mathbf{x}_W in kilos of 3 individuals

$$\mathbf{X} = [\mathbf{x}_A, \mathbf{x}_W] = \begin{bmatrix} 31 & 68 \\ 23 & 64 \\ 32 & 58 \end{bmatrix}$$

3. L'expérience, E

The data is the tuple (**X**,**y**)

- ▶ In machine learning, we assume that there is a unknown function $f(\cdot)$ linking the independent part of the observation vector \mathbf{x}_i to y_i

$$y_i = f(x_i)$$

- **One of the objectives of machine learning algorithms:** obtain an approximation $\hat{f}(\cdot)$ of $f(\cdot)$ from the data (\mathbf{X}, \mathbf{y}) such that we can obtain a reasonable prediction $\hat{y} = \hat{f}(\mathbf{x})$ for an observation \mathbf{x} which is not in data.

3. L'expérience, E

e.g. $\mathbf{X} = [\mathbf{x}_A, \mathbf{x}_W]$ and $\mathbf{y} = \begin{bmatrix} 178 \\ 173 \\ 158 \end{bmatrix}$ is height in centimeters

⇒ Predict height from age and weight.

e.g. $\mathbf{X} = [\mathbf{x}_A, \mathbf{x}_W]$ and $\mathbf{y} = \begin{bmatrix} \text{no} \\ \text{no} \\ \text{yes} \end{bmatrix}$ says if the person is diabetic or not

⇒ Predict if a person is diabetic from age and weight.

3. L'Experience E. Types de donnees

Quantitative	<ul style="list-style-type: none">▶ measurable quantities - numerical▶ mathematical functions can be applied (e.g. sum, mean)▶ comparisons are possible (e.g. =, ≠, >, <)
Qualitative	<ul style="list-style-type: none">▶ characteristics or qualities (which type/category?)▶ mathematical functions cannot be applied▶ not all comparisons are possible

3. L'Experience E. Types de donnees

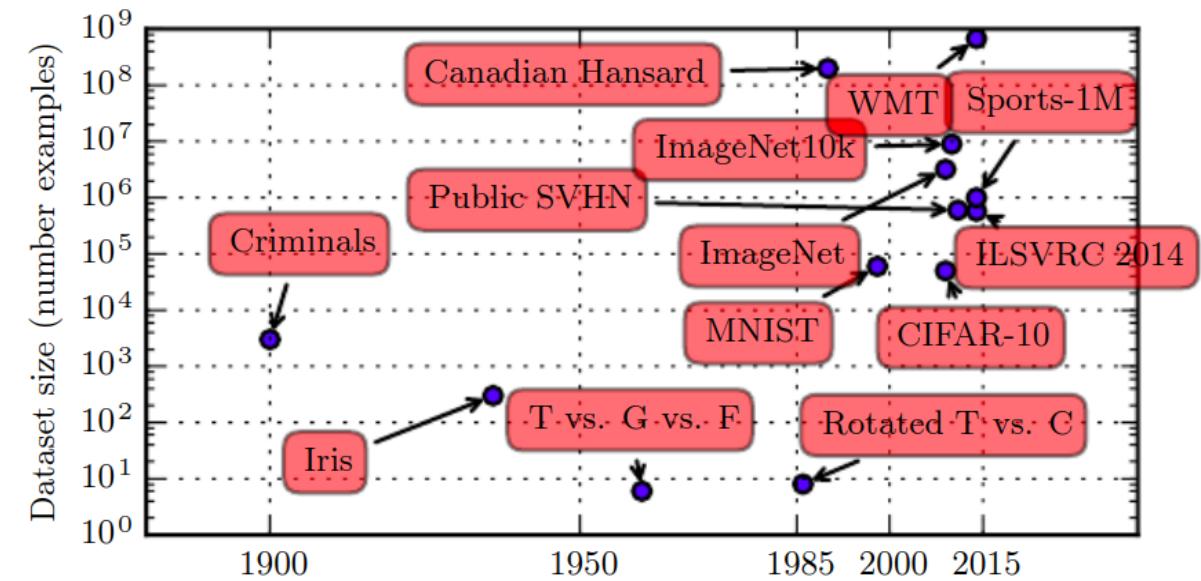
Quantitative	Continuous	▶ any value in an interval e.g. height
	Discrete	▶ only a finite number of values e.g. number of rooms in a house
Qualitative	Nominal	▶ no possible ordering e.g. diabetic? Yes/No
	Ordinal	▶ ordering is possible e.g. product quality? Bad/Good

3. L'expérience, E

- Peut être labelisé ou non selon la tache et le dataset
- **Qu'est-ce qu'un dataset ?**
Une collection d'exemples; ex: une collection d'images

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

MNIST



Evolution de la taille des datasets
(Goodfellow et al., 2017)

3. L'expérience, E

- Peut être labelisé ou non selon la tache et le dataset
- **Qu'est-ce qu'un dataset labelisé ou non labelisé?**

Un label est une valeur, une classe, ... associée aux exemples d'un dataset

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

MNIST

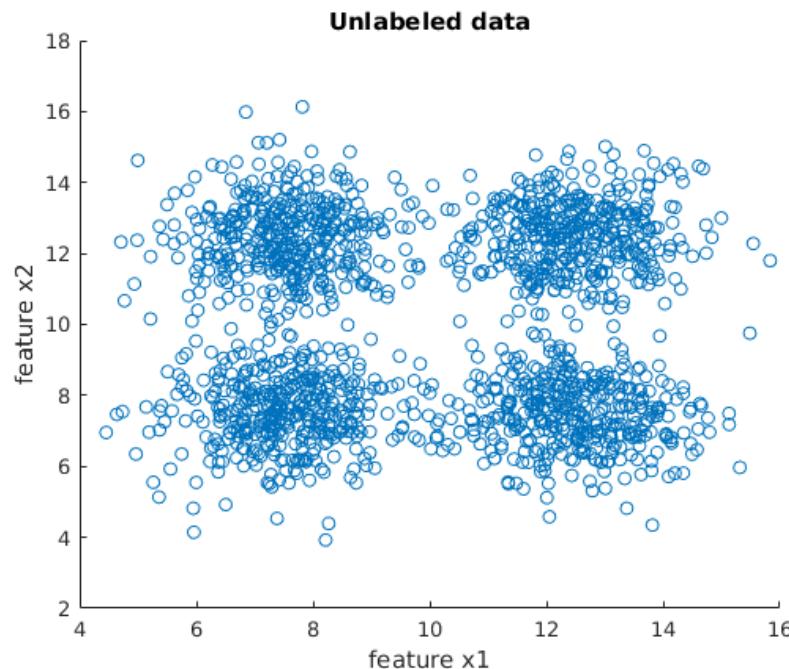
Les labels du dataset MNIST sont:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

C'est un dataset LABELISE

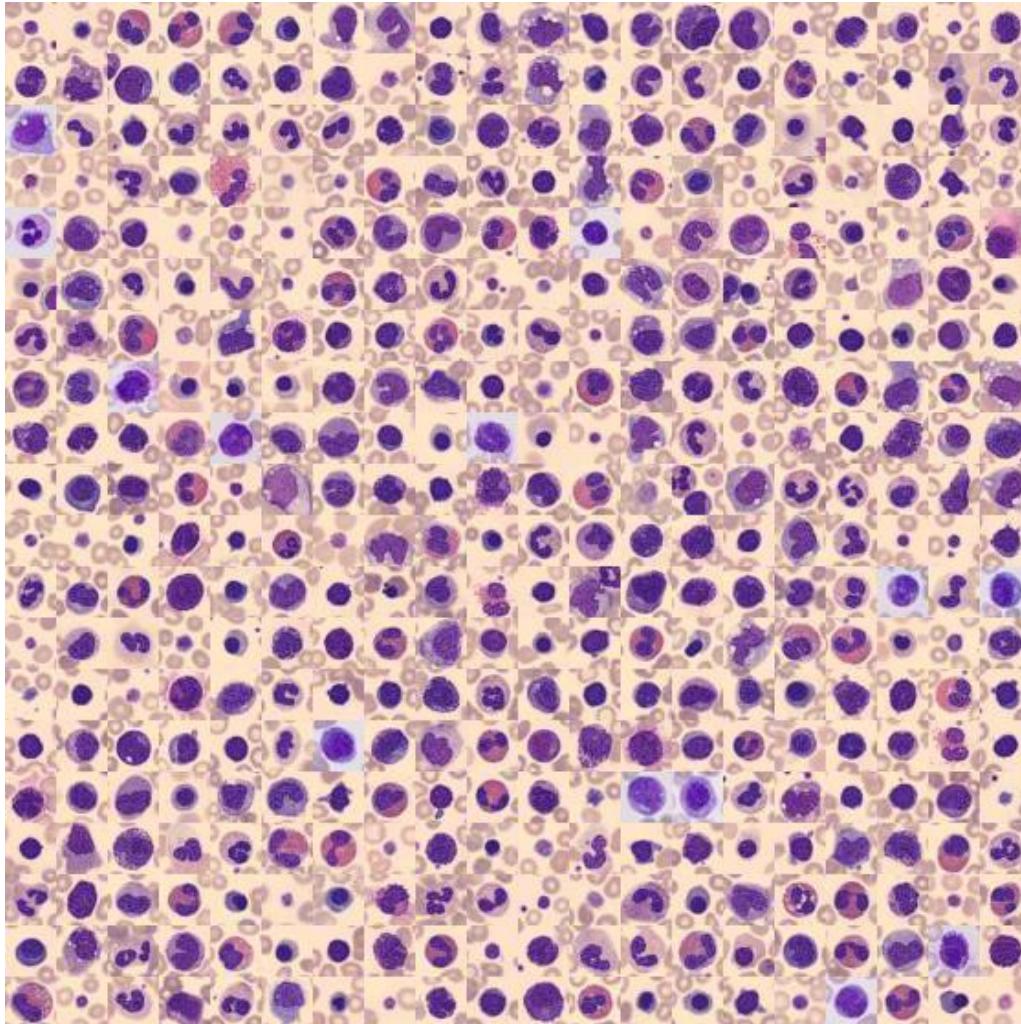
3. L'expérience, E

- Peut être supervisé ou non-supervisé selon la tache et le dataset
- **Qu'est-ce qu'un dataset labelisé ou non labelisé?**
Un label est une valeur, une classe, ... associée aux exemples d'un dataset



Les données de ce dataset sont les points. Nous n'avons pas de label.
C'est un dataset NON-LABELISE

Dataset exemple pour 2e partie de cours



BloodMNIST (Yang et al., Nature datasets 2022)

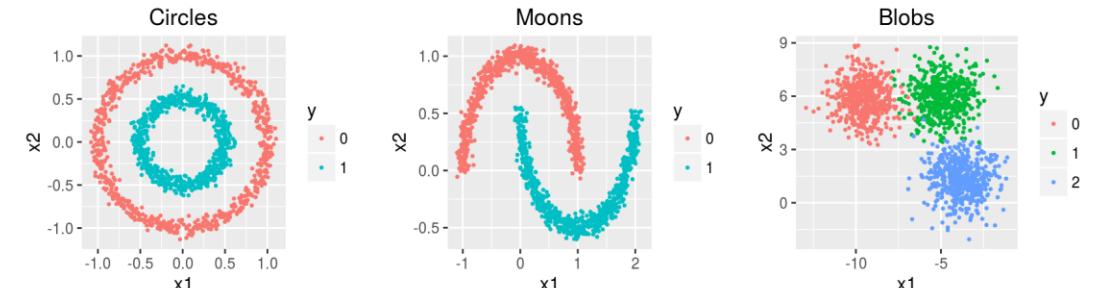
Andrea Acevedo, Anna Merino, et al., "A dataset of microscopic peripheral blood cell images for development of automatic recognition systems" , 2020.

Objectif: identification du type cellulaire

Quel type de tache est-ce ?

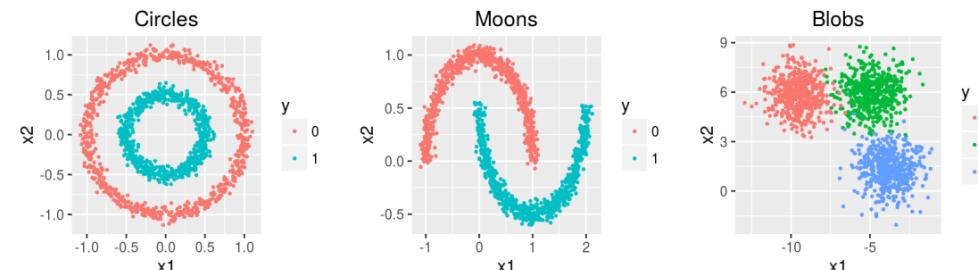
3.1. Supervised

- On connaît les labels / les valeurs (y) associée à chaque exemple (x) qu'on veut prédire avec notre modèle.
- Exemples :
 - Dépistage de cancer : Cancer vs Pas de cancer,
 - Prédiction du prix de l'immobilier,
 - Triage de mails : SPAM ou non SPAM.
- En termes de probabilités: $p(y|x)$



3.2. Unsupervised

- On veut trouver des tendances / des groupes, mais on ne sait pas à l'avance lesquels
- Exemples :
 - Regroupement de sites web similaires pour faire des suggestions,
 - Recherche de sous-groupes de tumeurs.
- En termes de probabilités: $p(x)$



4. Comment ça marche?

- On veut modéliser les probabilités $p(y|x)$ ou $p(x)$
- Ce modèle est le lien entre les données x et les labels y ou simplement entre les données entre elles.

4. Comment ça marche?

- On veut modéliser les probabilités $p(y|x)$ ou $p(x)$
- Ce modèle est le lien entre les données x et les labels y ou simplement entre les données entre elles.
- Le modèle est une simplification de la vie réelle qui repose sur des opérations et des fonctions mathématiques bien connues.

4. Comment ça marche?

- On veut modeliser les probabilités $p(y|x)$ ou $p(x)$
- Ce modèle est le lien entre les données x et les labels y ou simplement entre les données entre elles.
- Le modèle est une simplification de la vie réelle qui repose sur des opérations et des fonctions mathématiques bien connues.

$p_\theta(\text{donnees})$

$p_\theta(\text{labels}|\text{donnees})$

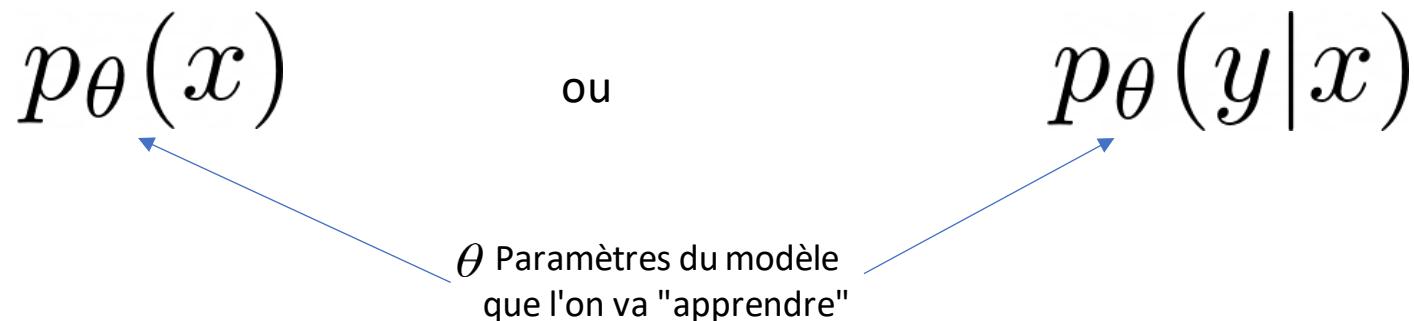
- Les opérations et les fonctions du modèle sont les paramètres de notre modèle et sont génériquement appelés θ .

4. Définir le modèle et sa fonction de perte

- En machine learning on cherche à modéliser des probabilités: $p(\text{données})$ ou $p(\text{labels}|\text{données})$.

$$p_{\theta}(x) \quad \text{ou} \quad p_{\theta}(y|x)$$

θ Paramètres du modèle que l'on va "apprendre"



- Pour trouver le paramètre qui approchera au mieux les "vraies" distributions, on crée une fonction, appelée fonction de perte qui dépend des paramètres et des données: (on la crée à partir de principes de probabilités ou de statistiques)

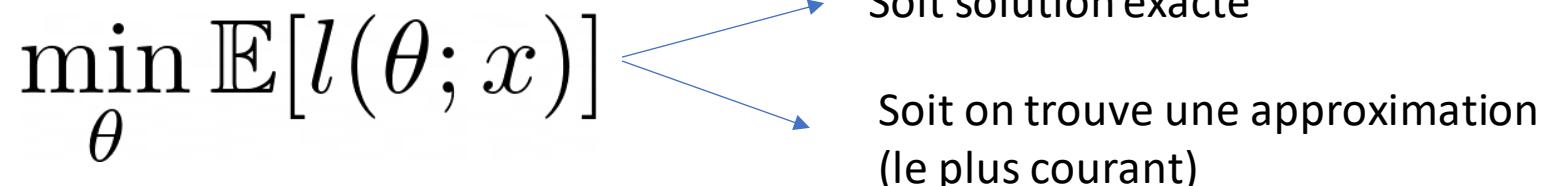
$$l(\theta; x)$$

- Le but sera de minimiser l'espérance de la fonction de perte en fonction des paramètres pour obtenir le meilleur paramètre.

$$\min_{\theta} \mathbb{E}[l(\theta; x)]$$

Soit solution exacte

Soit on trouve une approximation (le plus courant)



Comment trouve t on le minimum d'une fonction?

$$\min_{\theta} \mathbb{E}[l(\theta; x)]$$

Comment trouve t on le minimum d'une fonction?

$$\min_{\theta} \mathbb{E}[l(\theta; x)]$$

$$\nabla_{\theta} \mathbb{E}[l(\theta; x)] = 0$$

Quel est le problème ici?

$$\nabla_{\theta} \mathbb{E}[l(\theta; x)] = 0$$

Quel est le problème ici?

$$\nabla_{\theta} \mathbb{E}[l(\theta; x)] = 0$$

On ne peut pas calculer l'espérance car on ne connaît pas la distribution des données.

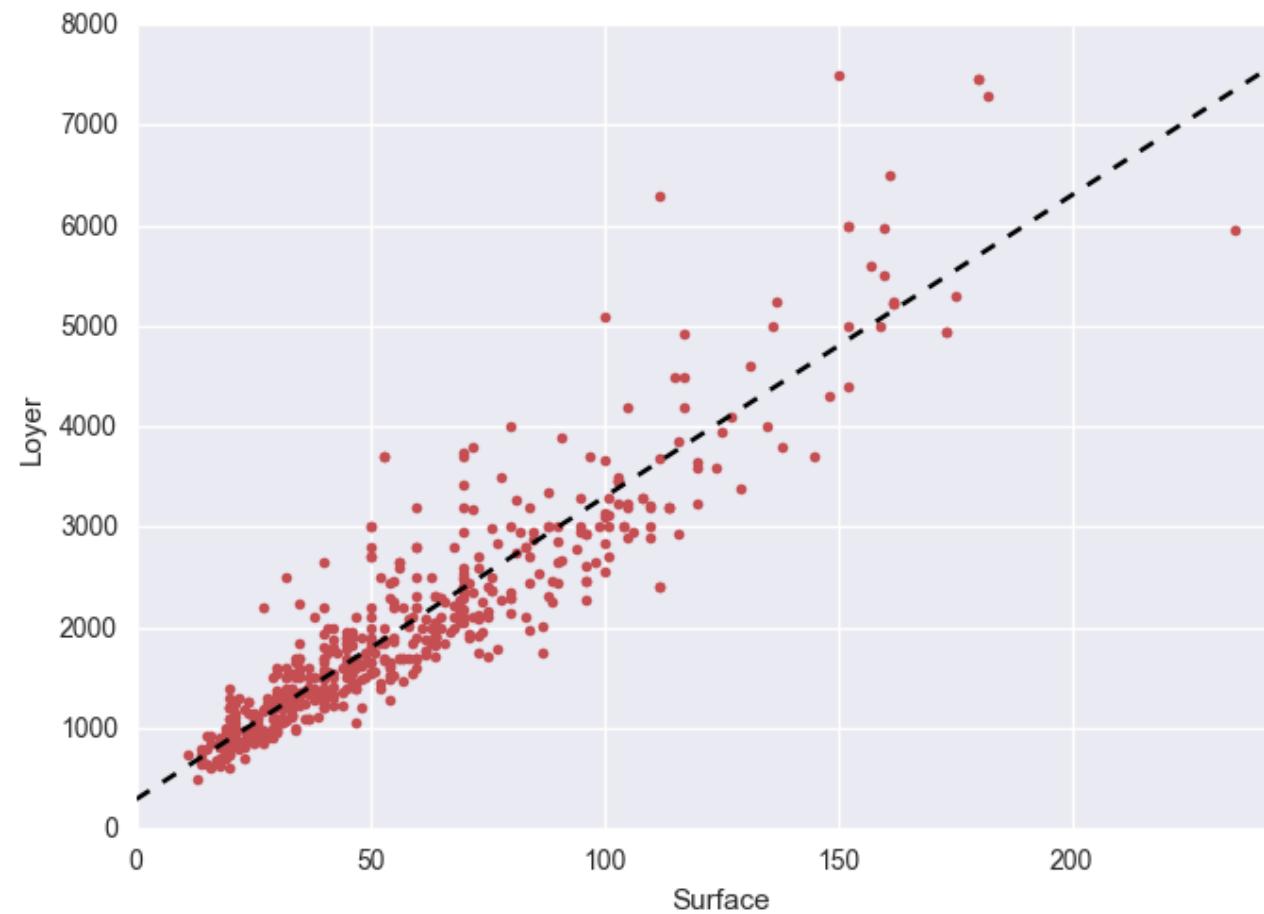
En revanche, on a notre disposition des données. On va les utiliser pour estimer l'espérance ainsi que son gradient

5. Exemple: Linear regression

- Données: Appartements x
- Variables: Surface des appartements
- Label: Loyer y
- Modèle:

$$y = f(x) = \theta_0 x$$

- Taches T ?
- Performance P ?
- Experience E ?
- Paramètres θ ?

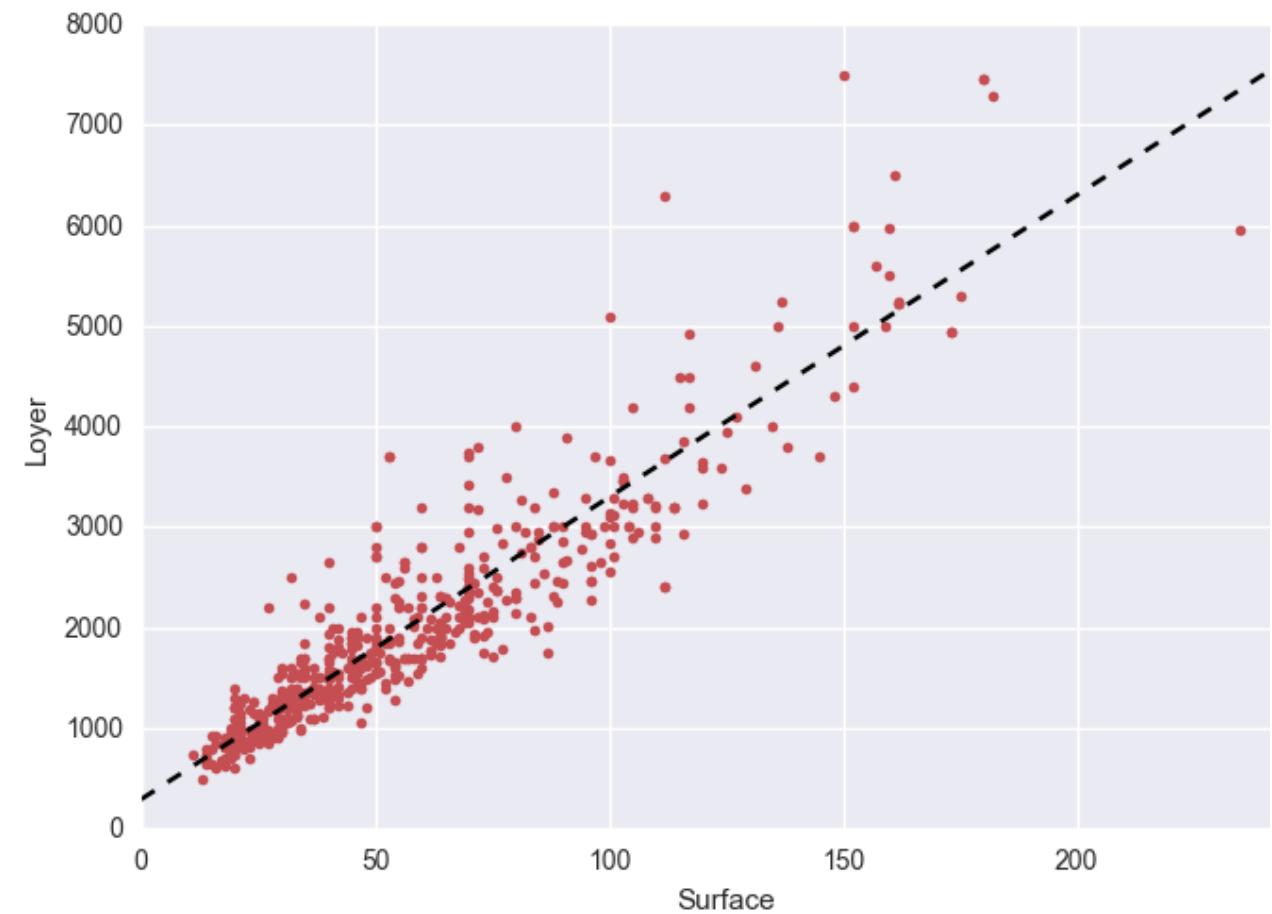


5. Exemple: Linear regression

- Données: Appartements x
- Variables : Surface de appartement
- Label: Loyer y
- Modèle:

$$y = f(x) = \theta_0 x$$

- Taches T ? Regression
- Performance P ? $MSE = \sum_{i=0}^n (f(x_i) - y_i)^2$
- Experience E ? Supervised
- Paramètres θ ? $\theta = \{\theta_0\}$

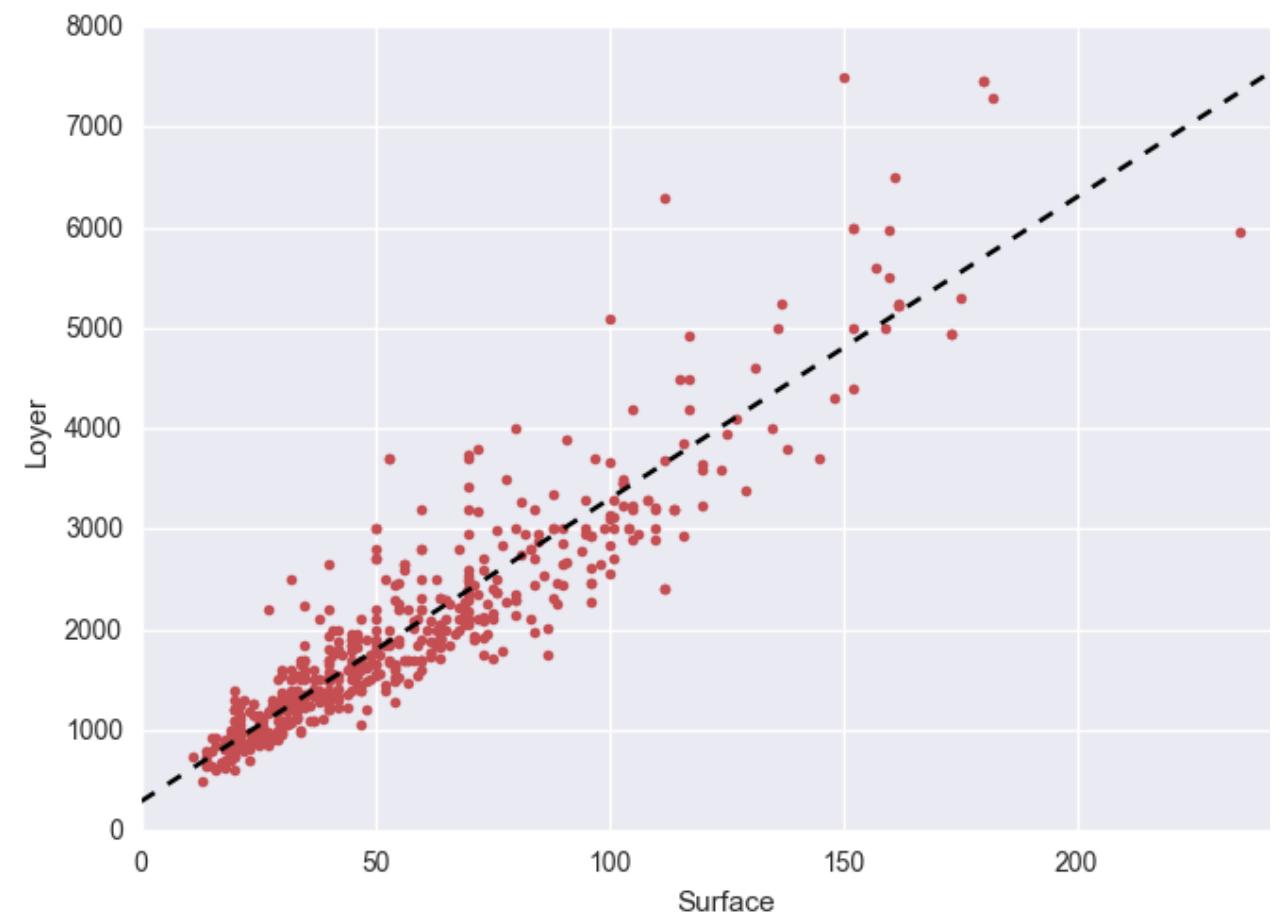


5. Exemple: Linear regression

- Objectif: minimiser la fonction de pertes

$$l(\theta, x) = MSE_{train}(\theta, x)$$

$$\begin{aligned}\mathbb{E}[l(\theta; x)] &\approx \frac{1}{n} \sum_{i=1}^n (\theta_0 * x_i - y_i)^2 \\ &= \frac{1}{n} \sum_{i=1}^n (\theta_0^2 x_i^2 - 2\theta_0 x_i y_i + y_i^2)\end{aligned}$$



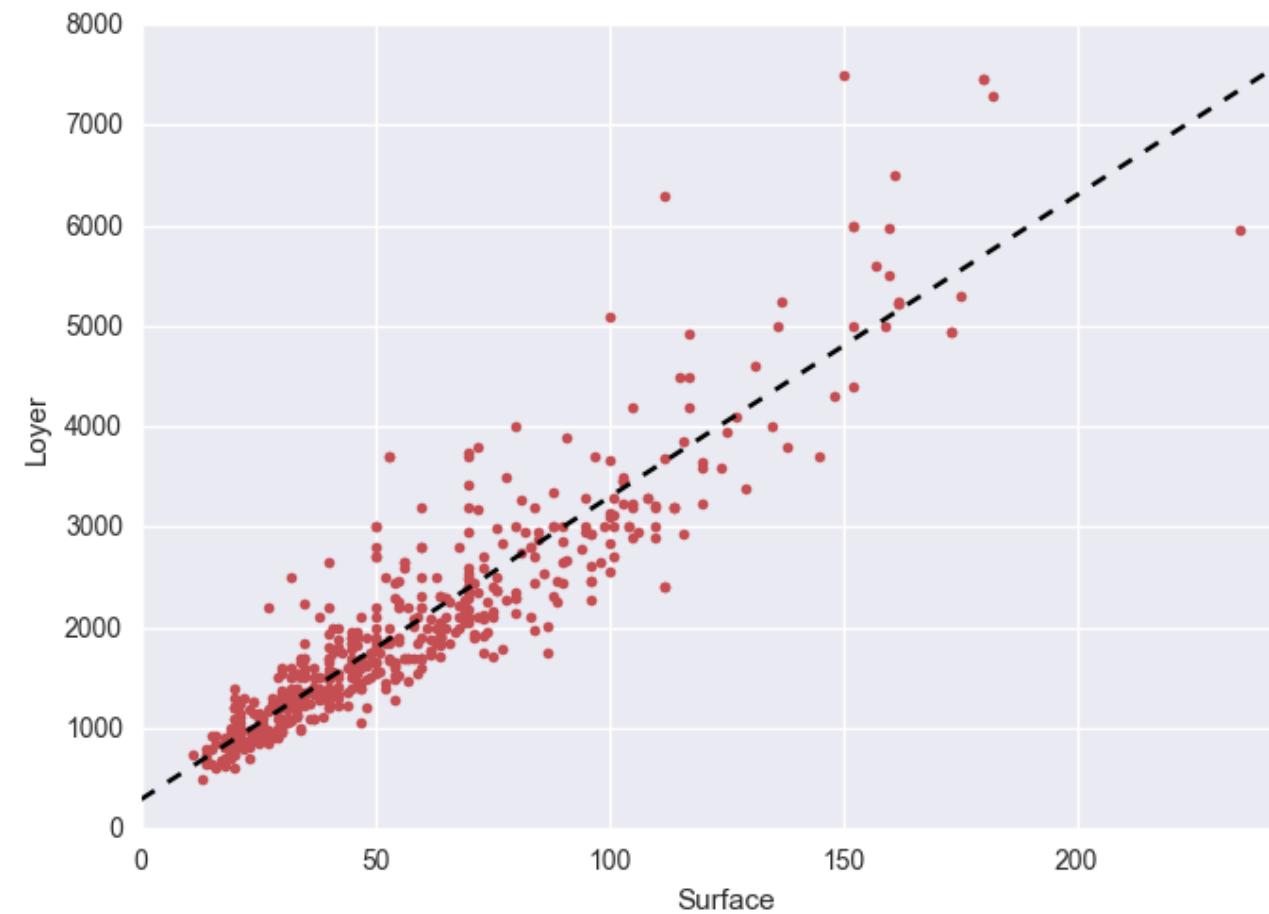
5. Exemple: Linear regression

- Objectif: minimiser la fonction de pertes

$$l(\theta, x) = MSE_{train}(\theta, x)$$

$$\begin{aligned}\mathbb{E}[l(\theta; x)] &\approx \frac{1}{n} \sum_{i=1}^n (\theta_0 * x_i - y_i)^2 \\ &= \frac{1}{n} \sum_{i=1}^n (\theta_0^2 x_i^2 - 2\theta_0 x_i y_i + y_i^2)\end{aligned}$$

Comment trouver le minimum en fonction de θ ?



5. Exemple: Linear regression

- Objectif: minimiser la fonction de pertes

$$l(\theta, x) = MSE_{train}(\theta, x)$$

$$\begin{aligned}\mathbb{E}[l(\theta; x)] &\approx \frac{1}{n} \sum_{i=1}^n (\theta_0 * x_i - y_i)^2 \\ &= \frac{1}{n} \sum_{i=1}^n (\theta_0^2 x_i^2 - 2\theta_0 x_i y_i + y_i^2)\end{aligned}$$

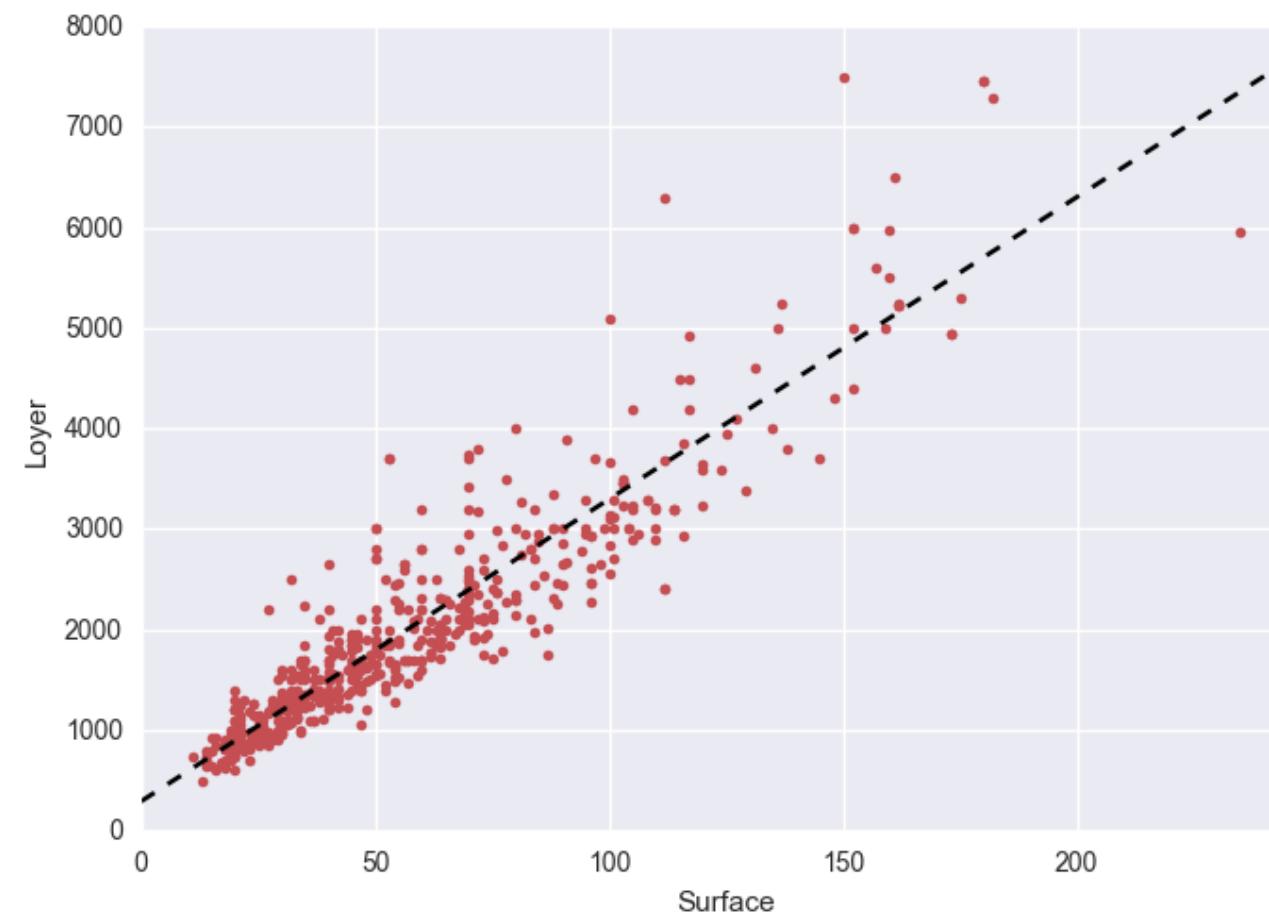
Comment trouver le minimum en fonction de θ ?

En résolvant l'équation:

$$\nabla_{\theta} \mathbb{E}[l(\theta; x)] = 0$$

On calcule:

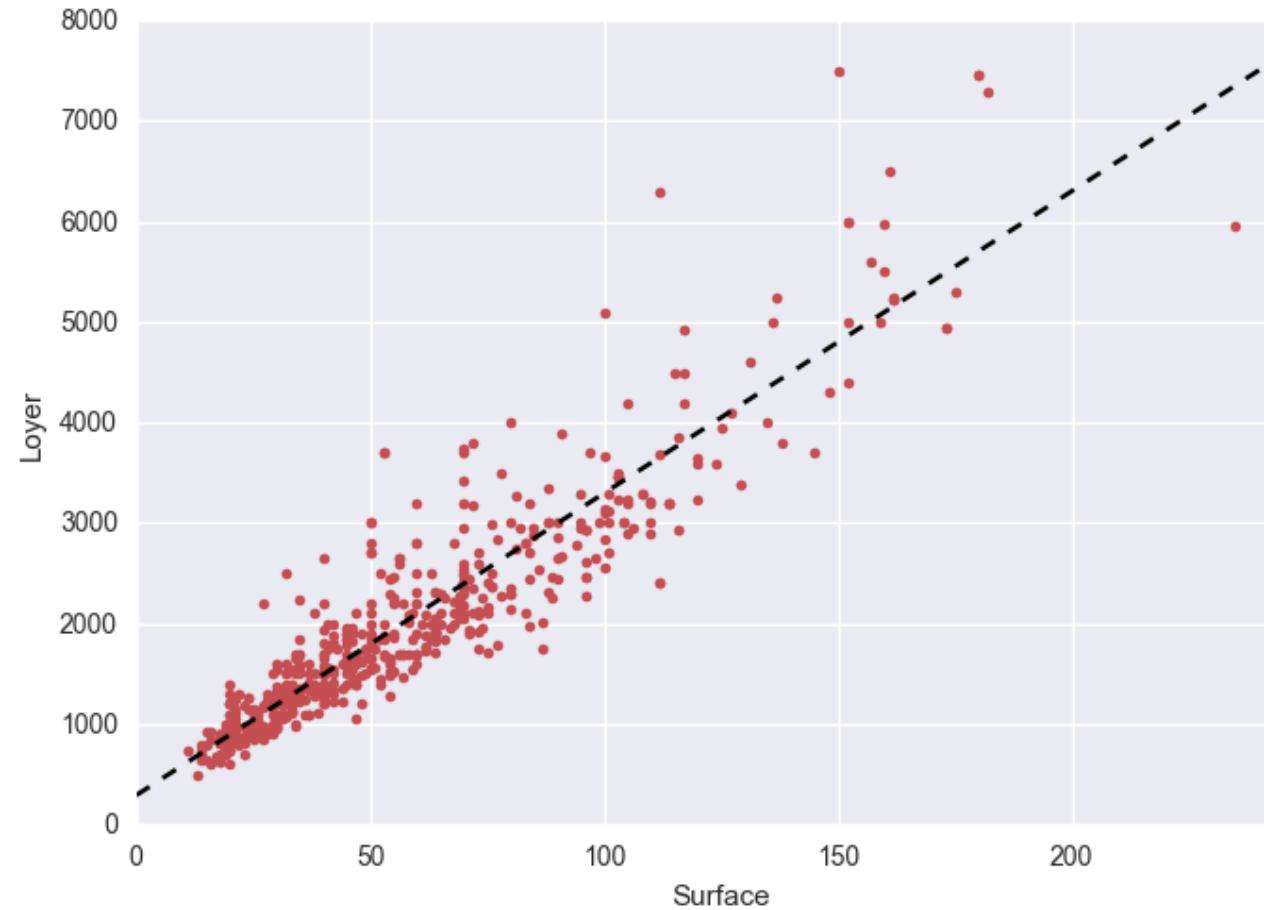
$$\nabla_{\theta} \mathbb{E}[l(\theta; x)] \approx \frac{1}{n} \sum_{i=1}^n (2\theta_0 x_i^2 - 2x_i y_i)$$



5. Exemple: Linear regression

- Objectif: minimiser la fonction de pertes

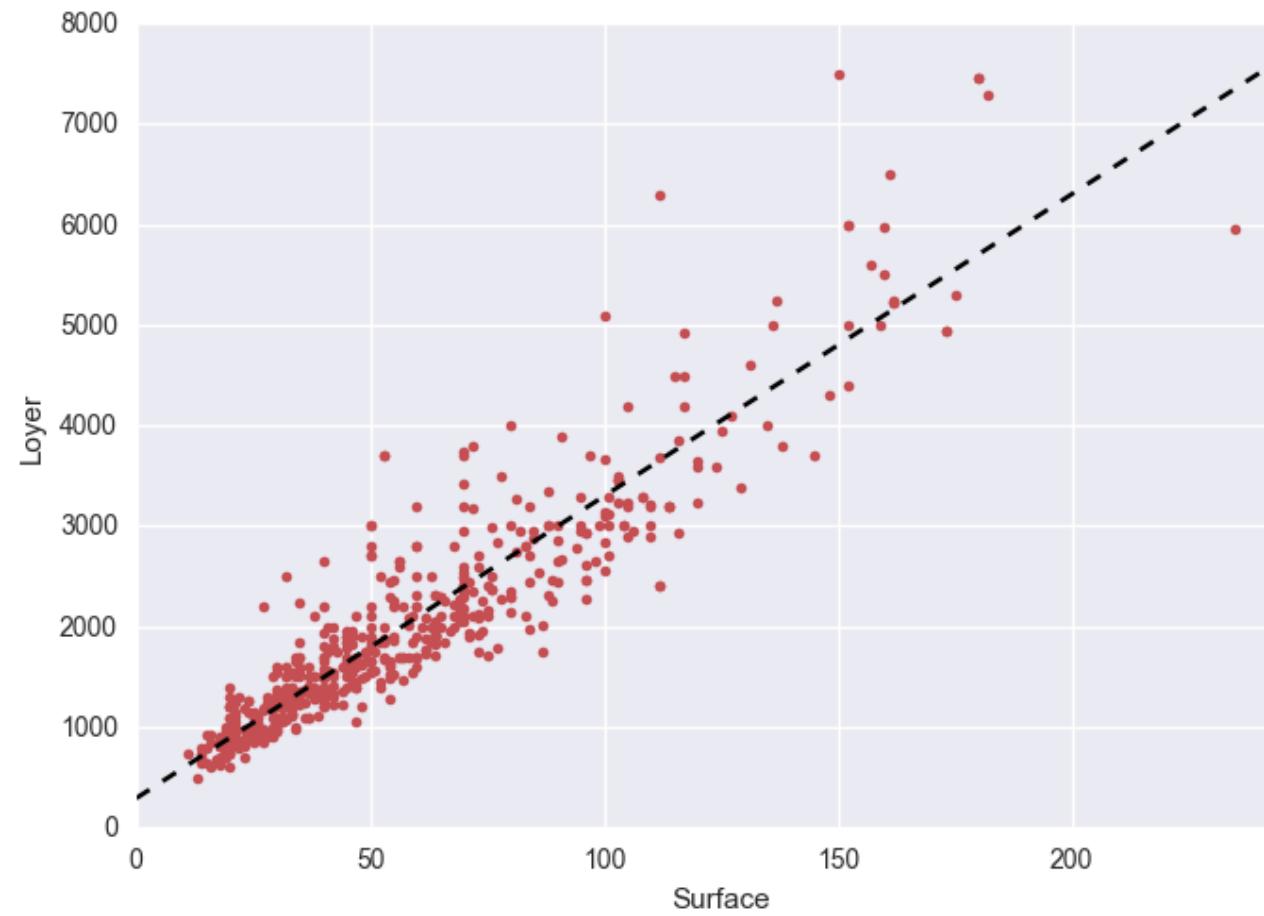
$$\begin{aligned}\nabla_{\theta} \mathbb{E}[l(\theta; x)] = 0 &\iff \frac{1}{n} \sum_{i=1}^n (2\theta_0 x_i^2 - 2x_i y_i) = 0 \\ &\iff \frac{2\theta_0}{n} \sum_{i=1}^n x_i^2 - \frac{2}{n} \sum_{i=1}^n x_i y_i = 0 \\ &\iff \theta_0 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i \\ &\iff \theta_0 = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}\end{aligned}$$



5. Exemple: Linear regression

Finalement, en minimisant la fonction de pertes,
on trouve le modèle suivant:

$$y = f(x) = \theta_0 x = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} x$$



Résumé des choses à savoir

- Qu'est ce qu'un algorithme d'apprentissage ?
- Supervised vs Unsupervised
- Qu'est-ce qu'un modèle, une fonction de perte ?

On connaît à présent qqs
bases du Machine learning

Mais on ne connaît pas encore les bonnes pratiques !

II. Capacité, Overfitting et Underfitting

1. Training and test datasets

- Généralisation:

Le modèle doit avoir une bonne performance aussi sur des données non observées. Comment faire pour s'en assurer ?

II. Capacité, Overfitting et Underfitting

1. Training and test datasets

- Généralisation:

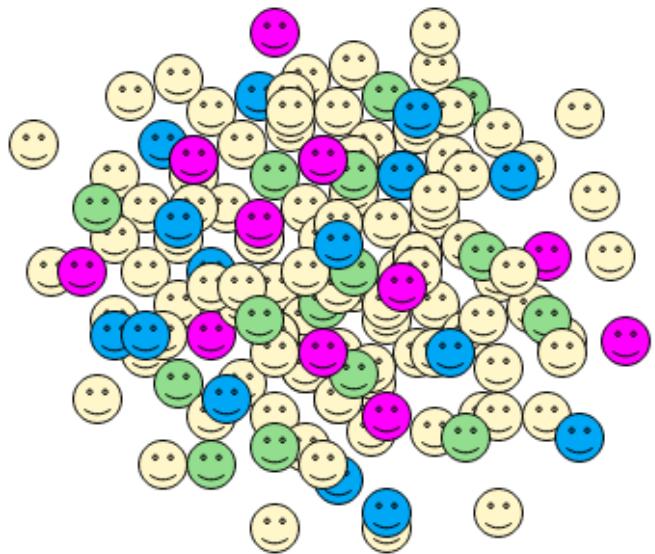
L'algorithme doit avoir une bonne performance aussi sur des données non observées. Comment faire pour s'en assurer ?

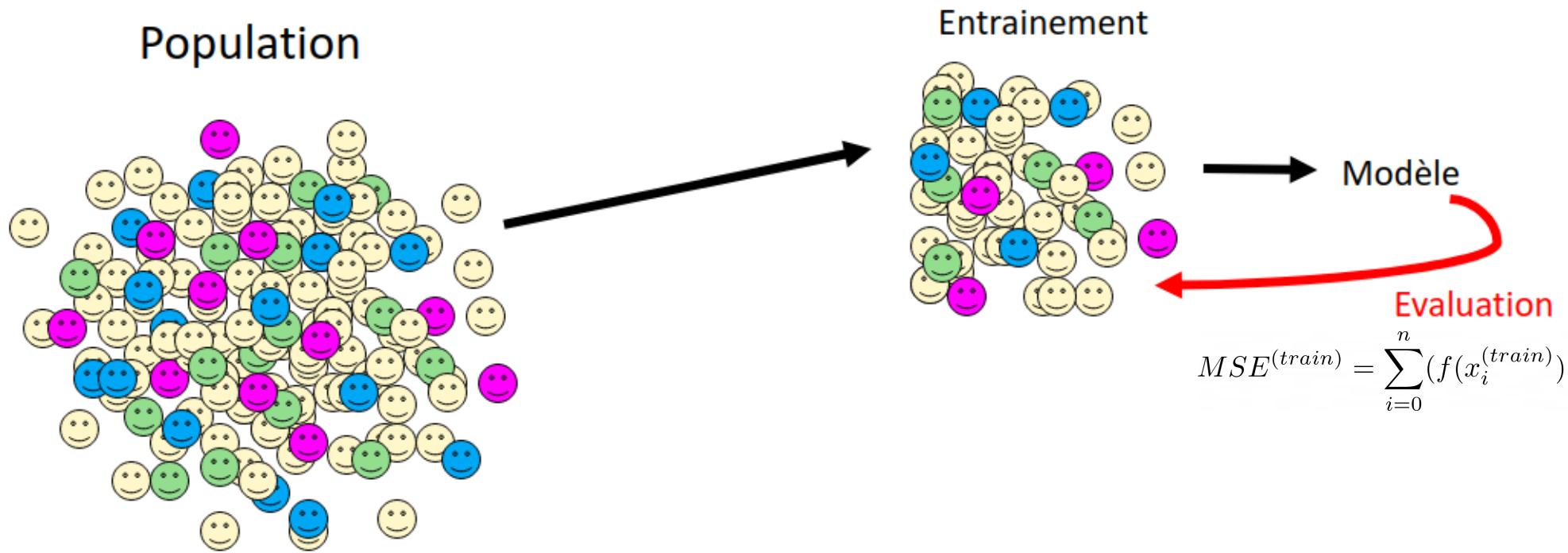
=> Training dataset et Testing dataset --> Training error et test error

$$MSE^{(train)} = \sum_{i=0}^n (f(x_i^{(train)}) - y_i^{(train)})^2$$

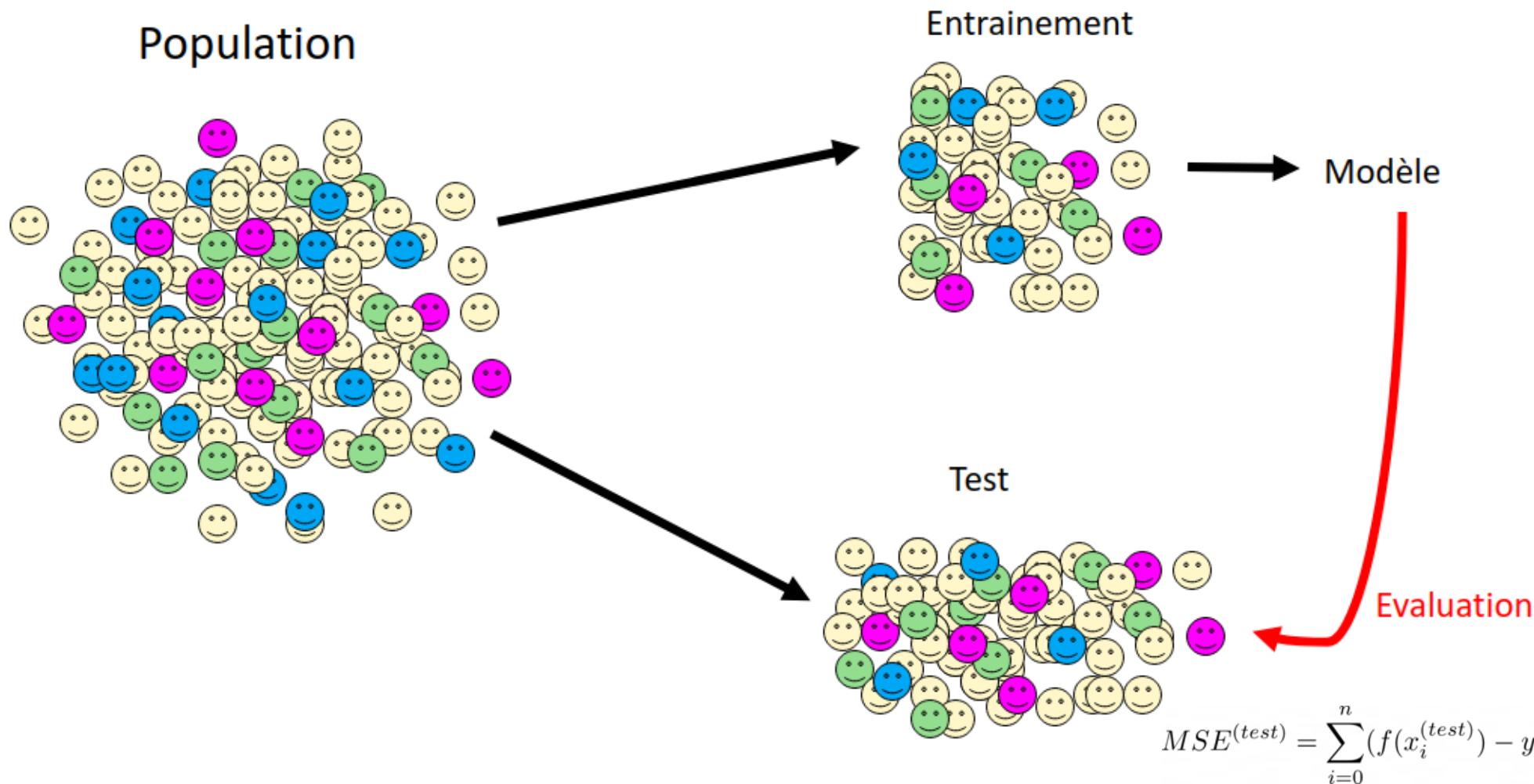
$$MSE^{(test)} = \sum_{i=0}^n (f(x_i^{(test)}) - y_i^{(test)})^2$$

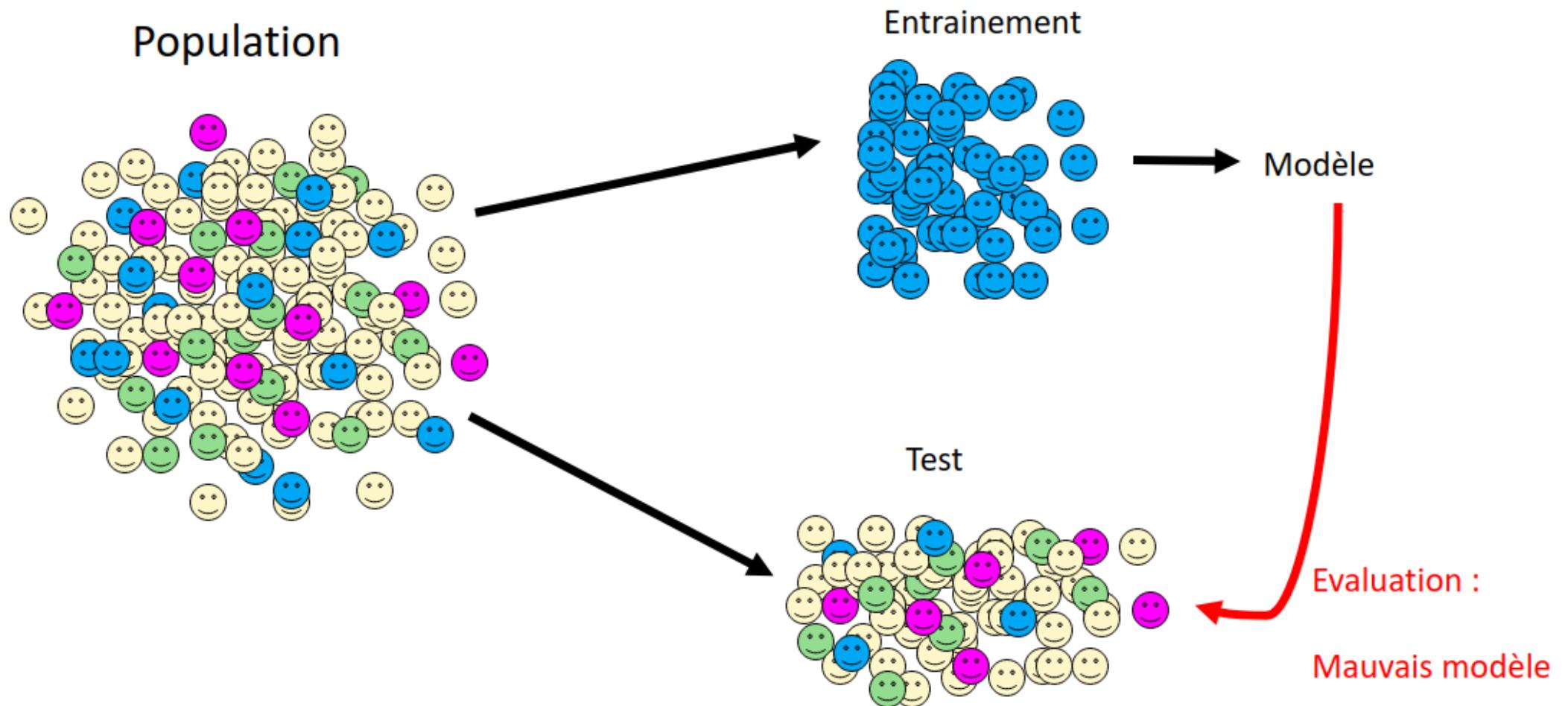
Population

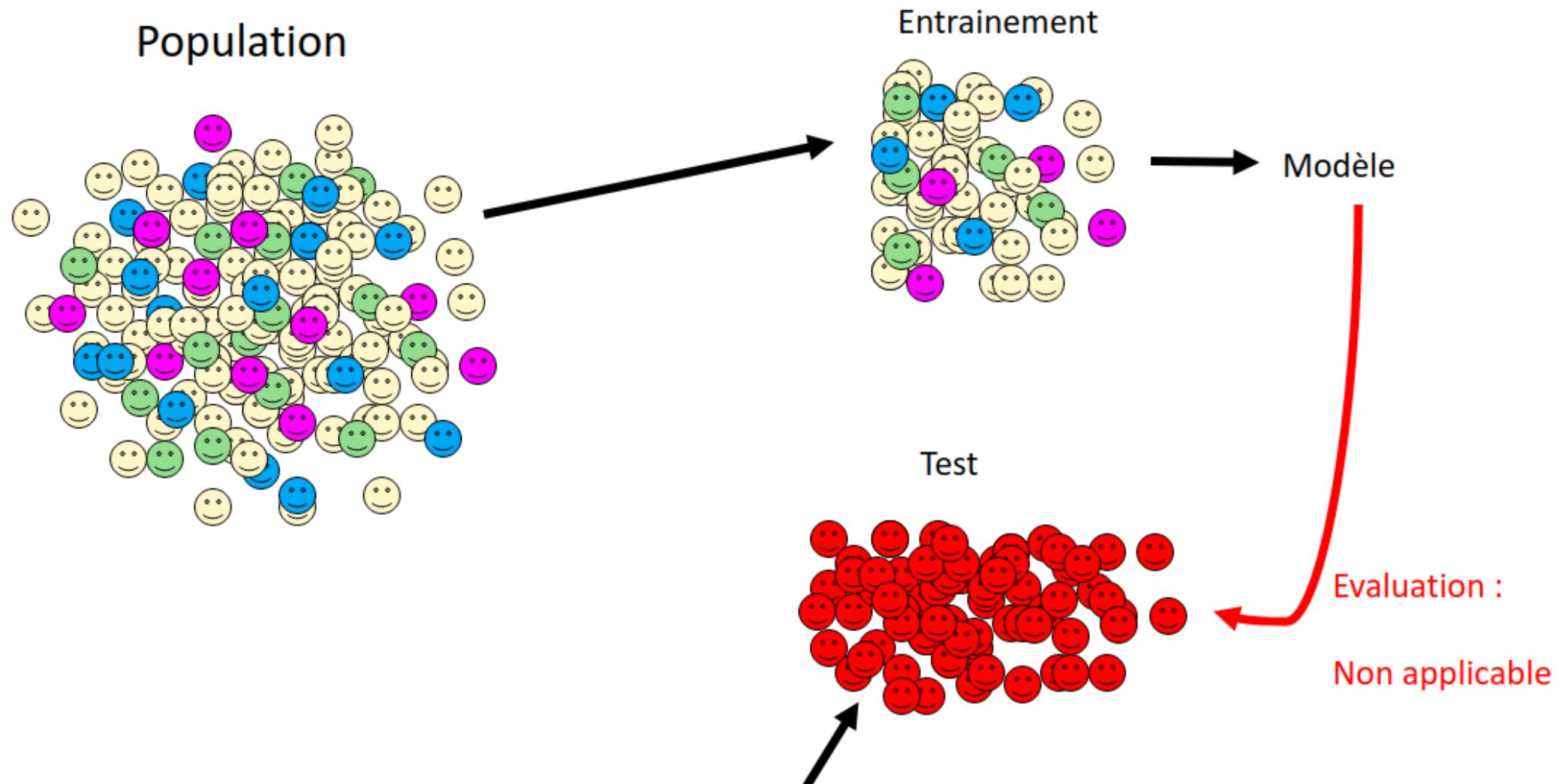




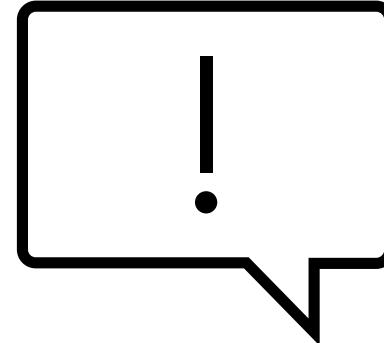
$$MSE^{(train)} = \sum_{i=0}^n (f(x_i^{(train)}) - y_i^{(train)})^2$$







2. Overfitting et underfitting



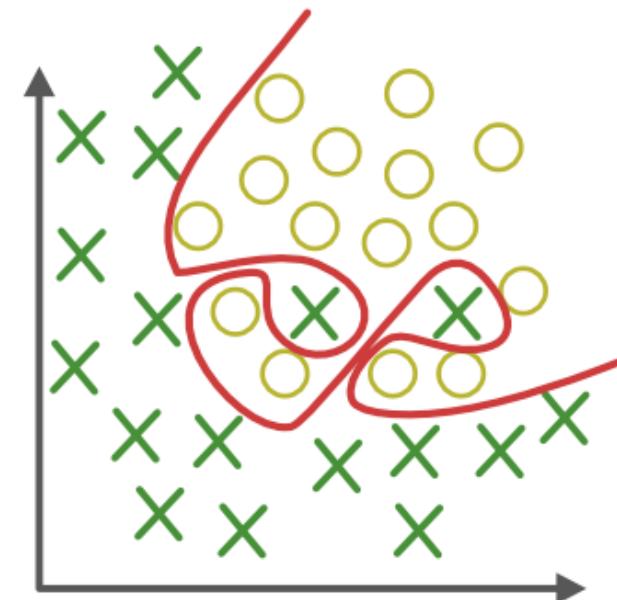
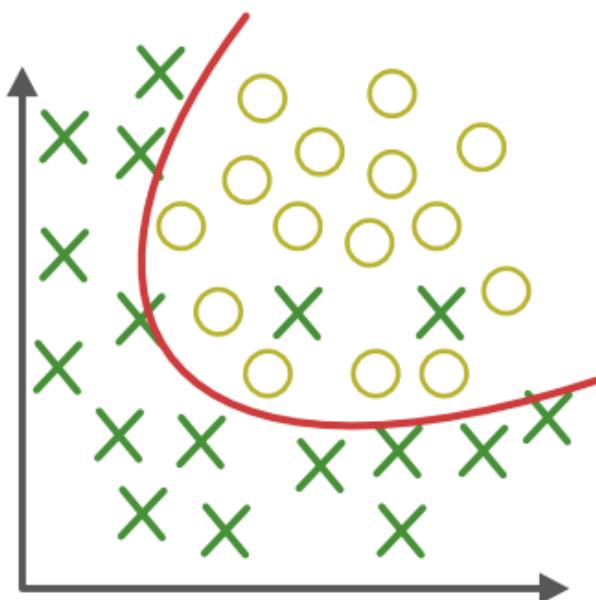
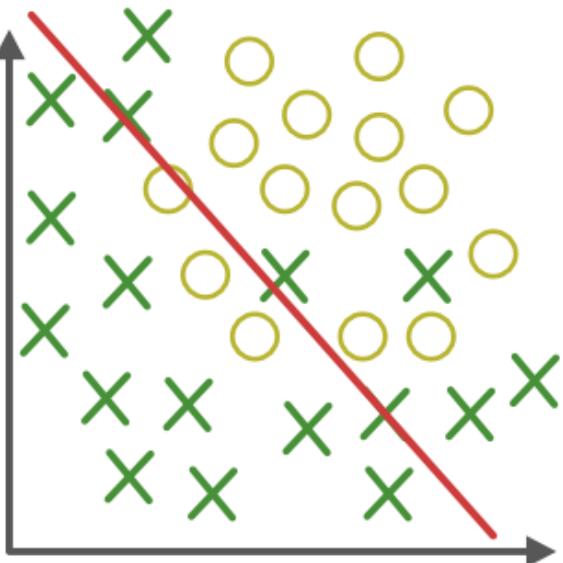
Objectifs:

- Training loss **petite**
- Différence entre training et test error **petite**

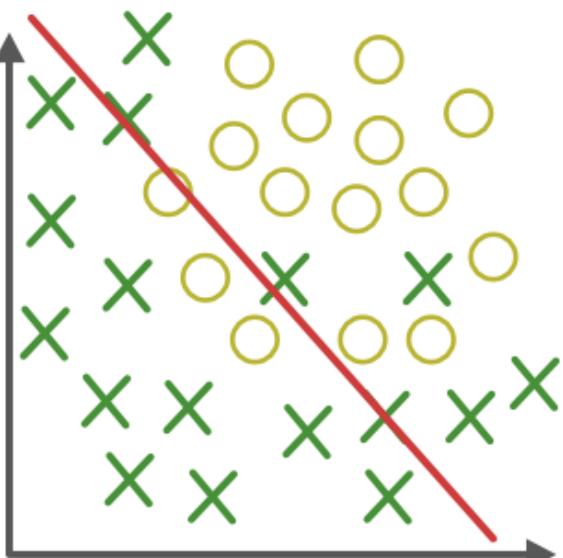
Underfitting: Training error trop élevée - Performance trop faible

Overfitting: Différence entre training et test error trop élevée

2. Overfitting et underfitting

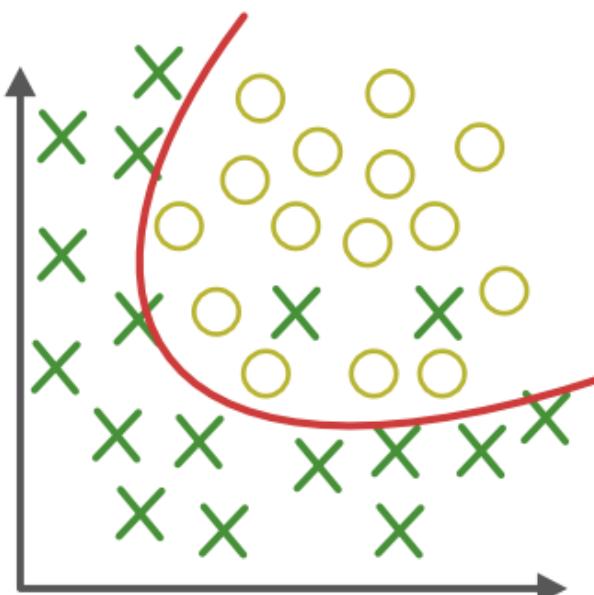


2. Overfitting et underfitting

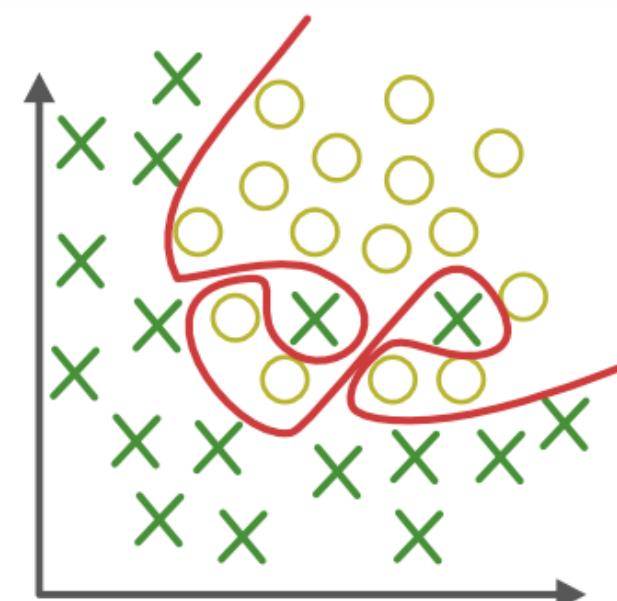


Under-fitting

(too simple to explain the variance)



Appropriate-fitting



Over-fitting

(forcefitting--too good to be true)

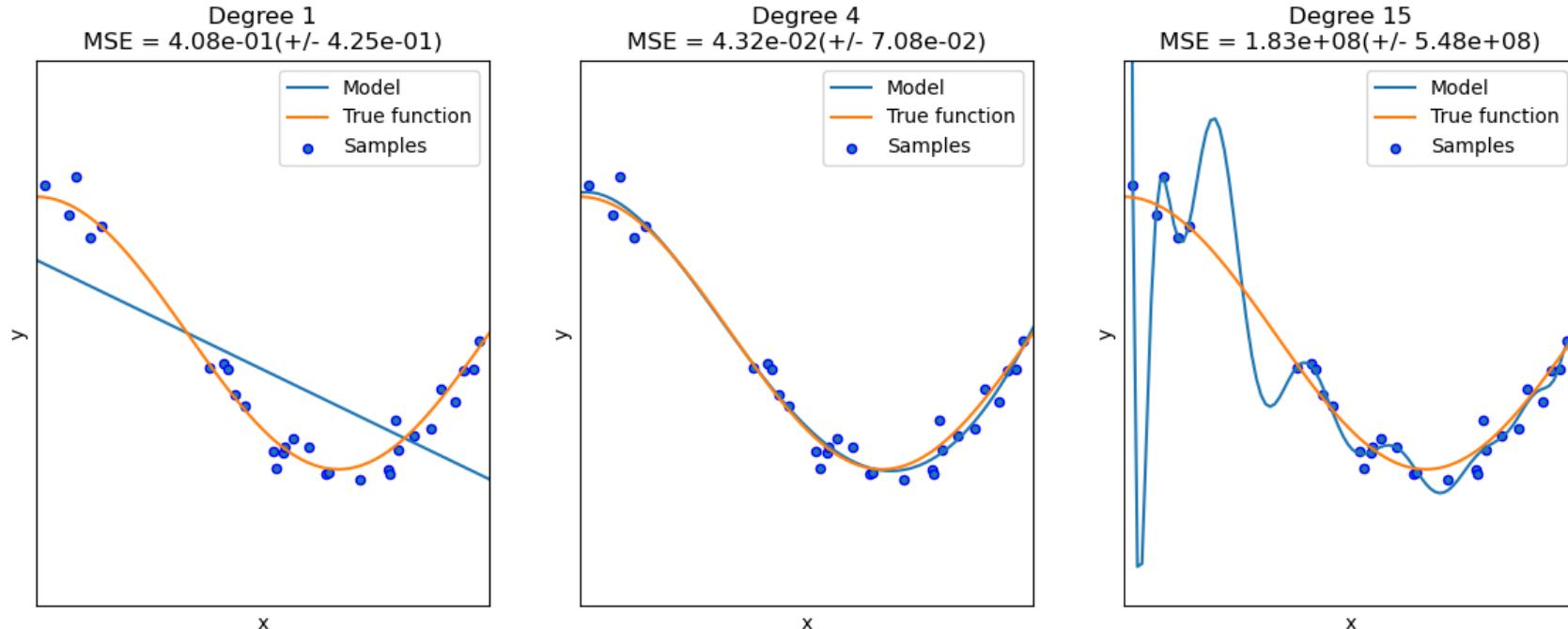
Complexité du modèle (capacité)

Regression
polynomiale

$$f(x) = \theta_0 + \theta_1 * x + \theta_2 * x^2 + \theta_3 * x^3 + \dots$$

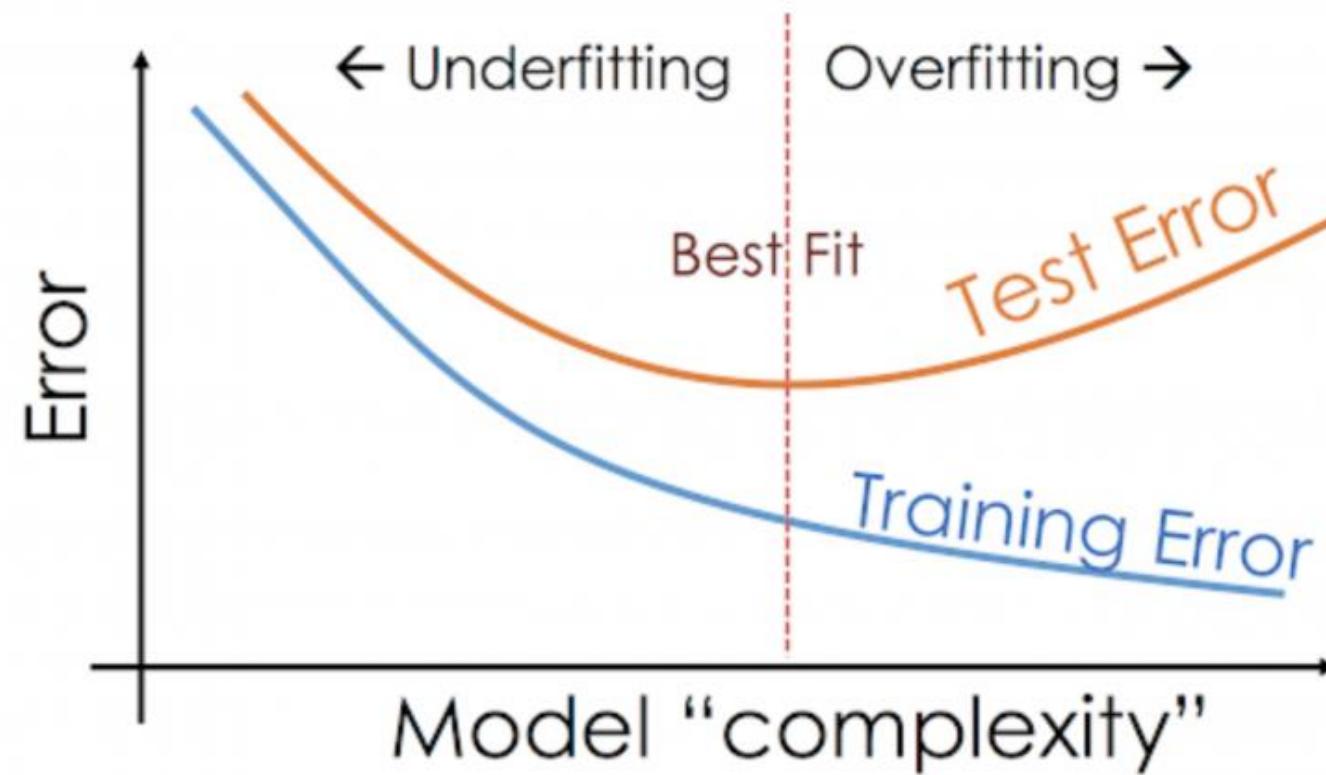
$\theta = \{ \dots \} ?$

Dans cette exemple, la complexité du modèle est équivalente au degré du polynôme choisi pour le modèle.



Quel modèle overfit ? Quel modèle underfit ?

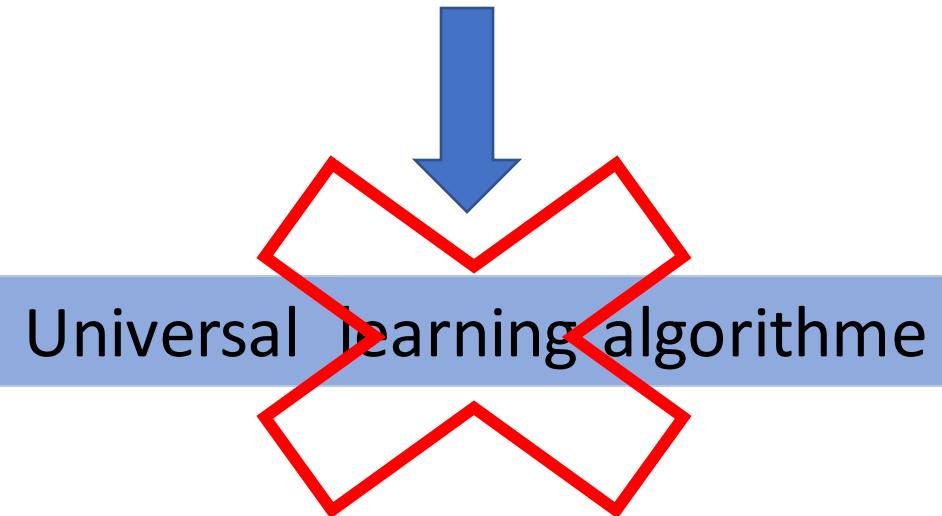
Complexité du modèle (capacité)



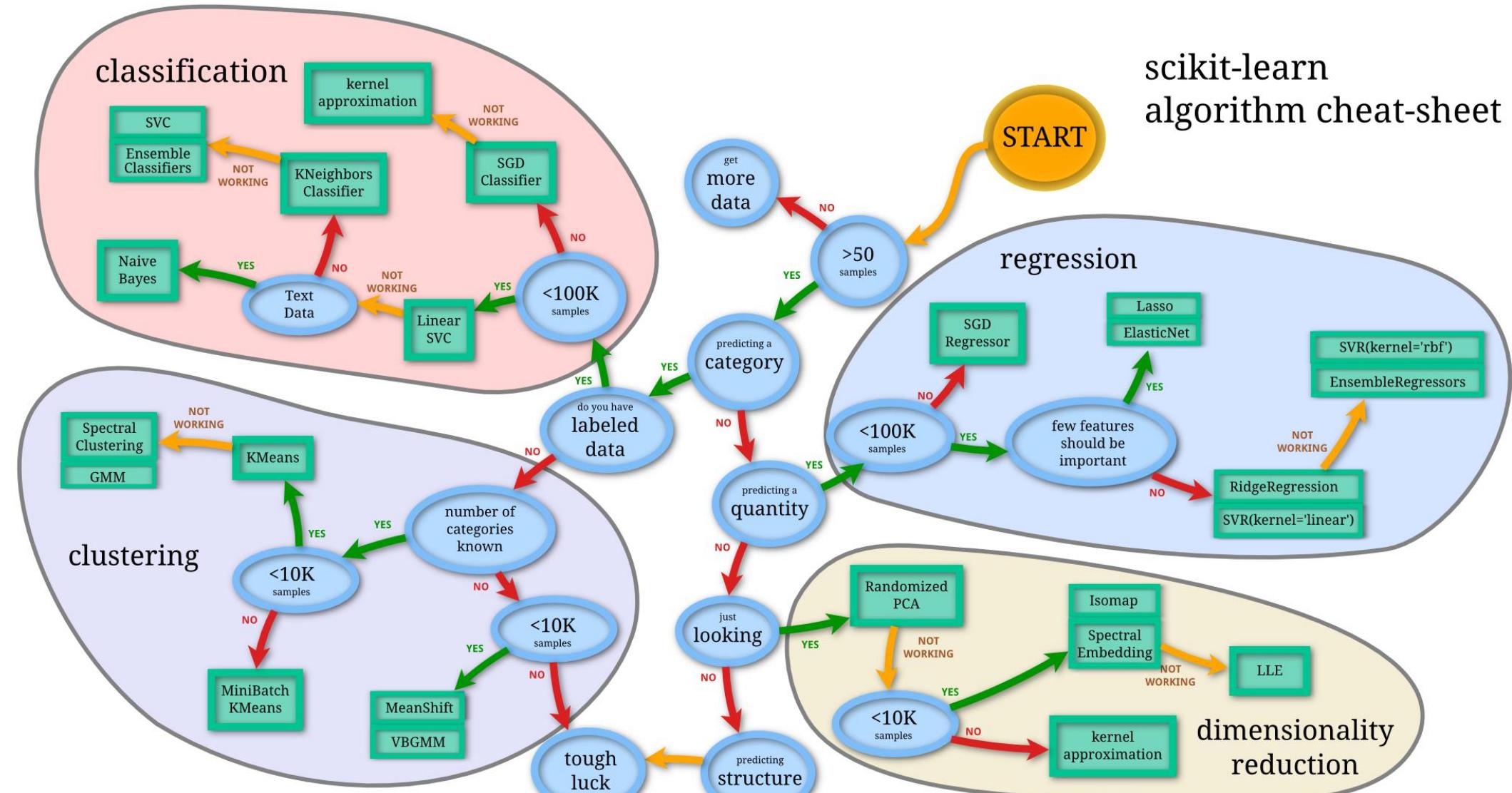
Est-il possible de créer un algorithme qui s'adapte à toutes les tâches ?

The No Free Lunch Theorem (Wolpert, 1996):

The average performance of any pair of algorithms across all possible problems is identical.



scikit-learn algorithm cheat-sheet



Back

scikit
learn

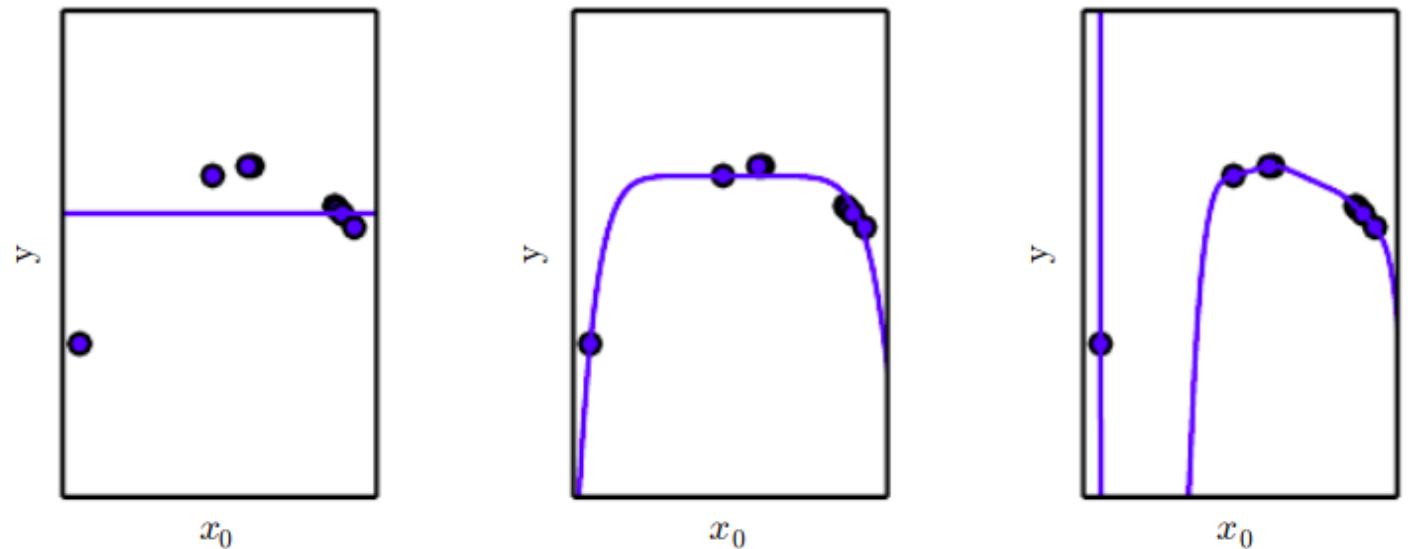
3. Fonction de perte & Régularisation

- Modèle:

$$f(x) = \theta_0 + \theta_1 * x + \theta_2 * x^2 + \theta_3 * x^3 + \dots$$

- Fonction de perte:

$$l(\theta, x) = MSE_{train}(\theta, x)$$



$$MSE^{(train)} = \sum_{i=0}^n (f(x_i^{(train)}) - y_i^{(train)})^2$$

3. Fonction de perte & Régularisation

- Modèle:

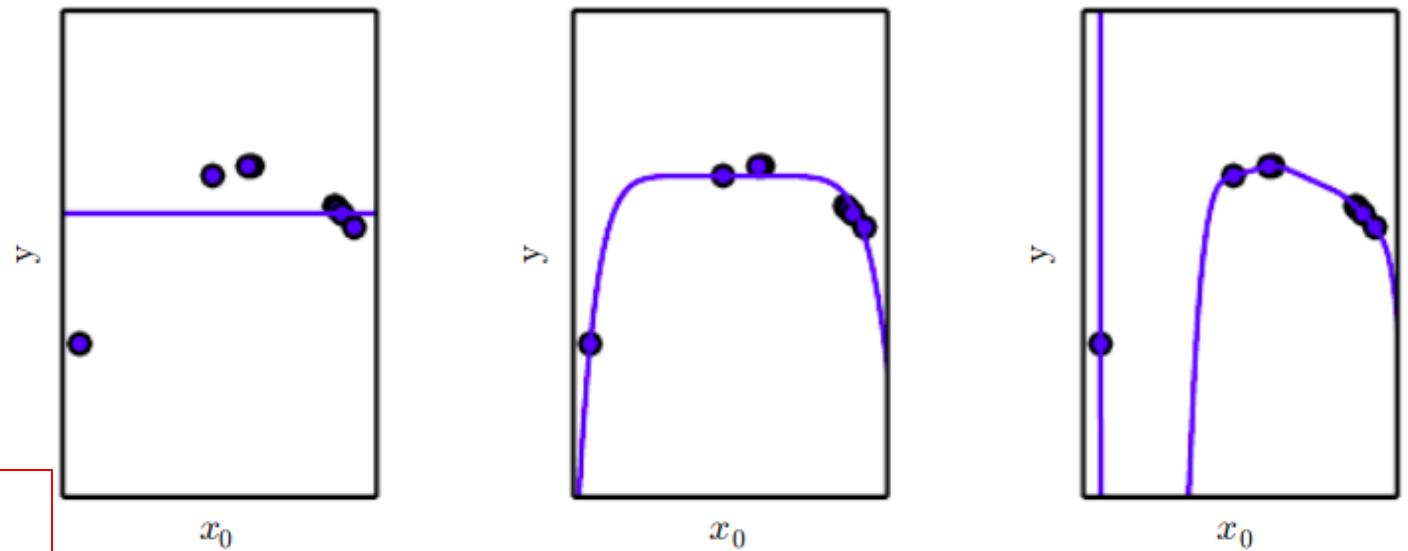
$$f(x) = \theta_0 + \theta_1 * x + \theta_2 * x^2 + \theta_3 * x^3 + \dots$$

- Fonction de perte:

$$l(\theta, x) = MSE_{train}(\theta, x)$$

Fonction de perte regularisée:

$$\begin{aligned} l(\theta, x) &= MSE_{train} + \lambda \theta^\top \theta \\ &\quad \text{Weight decay} \\ &= MSE_{train} + \lambda (\theta_0^2 + \theta_1^2 + \theta_2^2 + \dots) \end{aligned}$$



$$MSE^{(train)} = \sum_{i=0}^n (f(x_i^{(train)}) - y_i^{(train)})^2$$

Autres types de régularisation:

- Fonction de pertes:

$$l(\theta, x)$$

- Fonction de pertes + Regularization Ridge:

$$l(\theta, x) + \lambda(\theta_0^2 + \theta_1^2 + \theta_2^2 + \dots)$$

- Fonction de pertes + Regularization Lasso:

$$l(\theta, x) + \lambda(\theta_0 + \theta_1 + \theta_2 + \dots)$$

- Fonction de pertes + Regularization mixte (Elastic net)

$$l(\theta, x) + \lambda \left(\alpha(\theta_0 + \theta_1 + \theta_2 + \dots) + \frac{1 - \alpha}{2} (\theta_0^2 + \theta_1^2 + \theta_2^2 + \dots) \right)$$

Régularisation

DEF: La régularisation est n'importe quelle modification de l'algorithme d'apprentissage que l'on fait dans l'intention de réduire l'erreur de test mais pas l'erreur d'entraînement.

Rappel: No free lunch Theorem

Il n'y a pas d'algorithme meilleur que les autres, et en particulier, il n'y a pas de régularisation meilleure que les autres.

4. Hyper-paramètres et datasets de validation

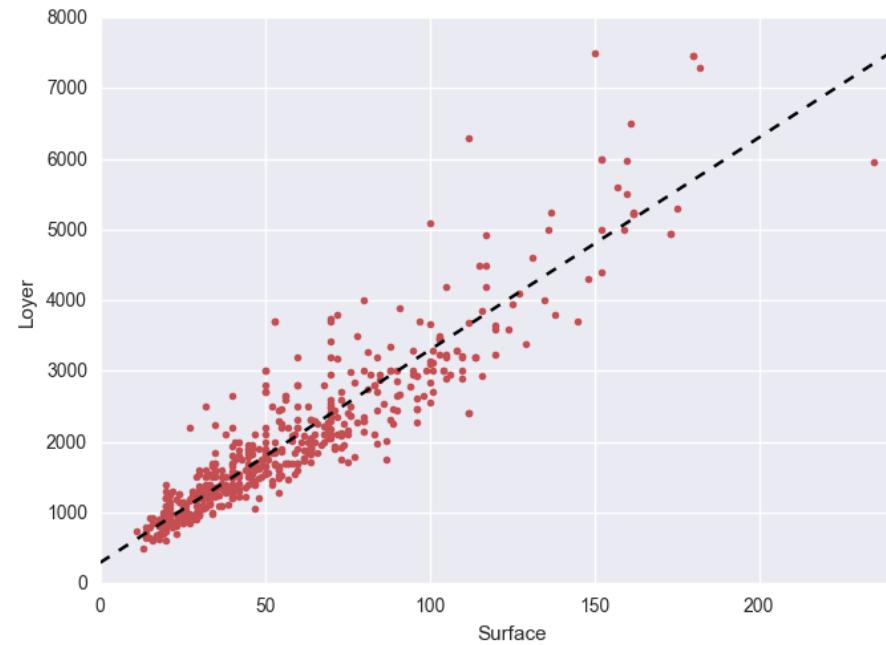
DEF: Un hyper-paramètre est un paramètre que l'algorithme n'apprend pas puisque son optimisation est en pratique très difficile, et ne dépend en général pas des données observées.

4. Hyper-paramètres et datasets de validation

DEF: Un hyper-paramètre est un paramètre que l'algorithme n'apprend pas puisque son optimisation est en pratique très difficile, et ne dépend en général pas des données observées.

Combien d'hyper-paramètres dans une régression linéaire ?

$$y = f(x) = \theta_0 x$$



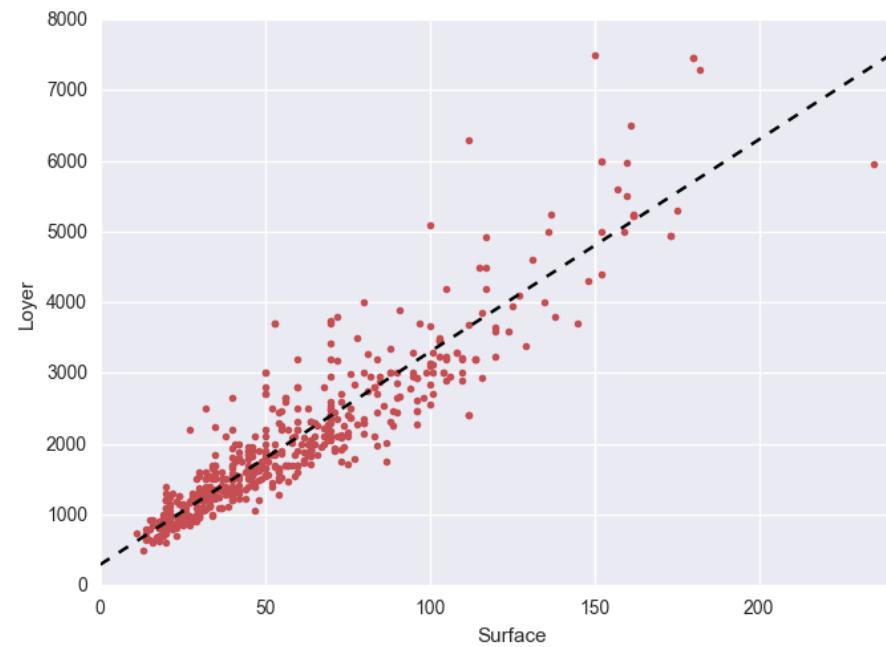
4. Hyper-paramètres et datasets de validation

DEF: Un hyper-paramètre est un paramètre que l'algorithme n'apprend pas puisque son optimisation est en pratique très difficile, et ne dépend en général pas des données observées.

Combien d'hyper-paramètres dans une régression linéaire ?

$$y = f(x) = \theta_0 x$$

Nb hyperparamètres = 0

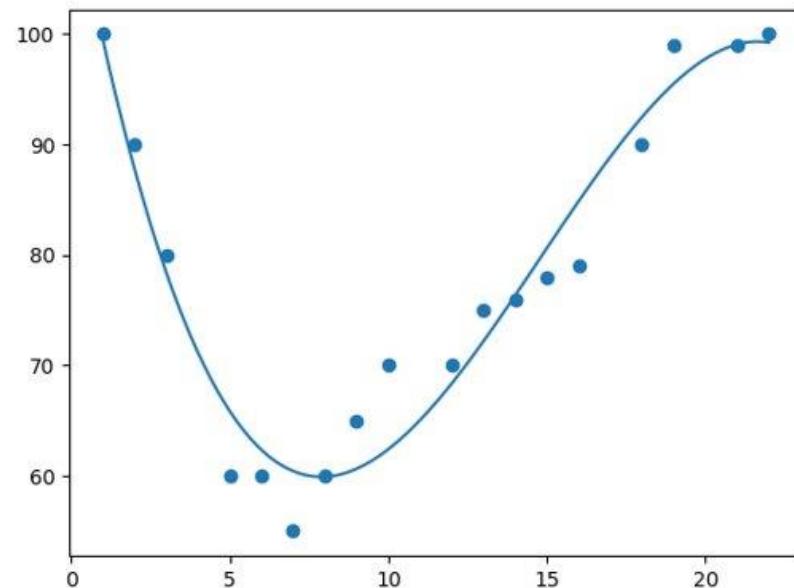


4. Hyper-paramètres et datasets de validation

DEF: Un hyper-paramètre est un paramètre que l'algorithme n'apprend pas puisque son optimisation est en pratique très difficile, et ne dépend en général pas des données observées.

Combien d'hyper-paramètres dans une régression polynomiale ?

$$y = f(x) = \sum_{i=0}^k \theta_i x^i = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_k x^k$$



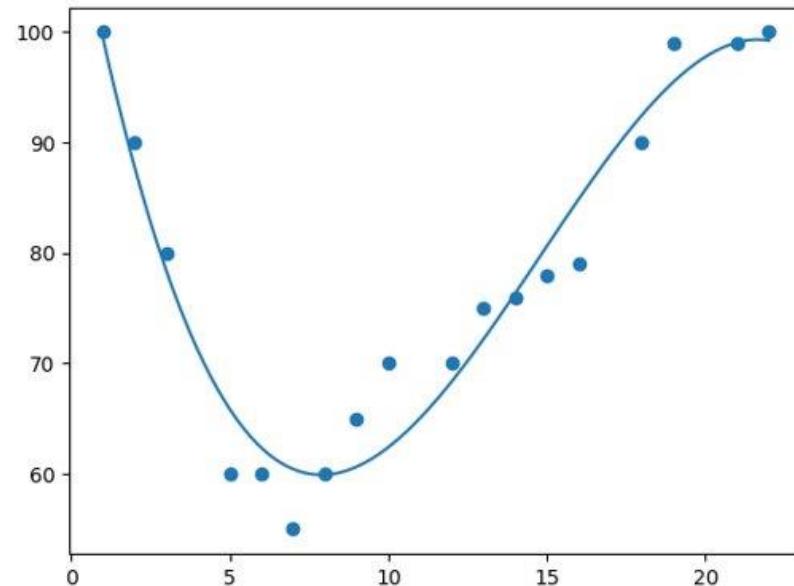
4. Hyper-paramètres et datasets de validation

DEF: Un hyper-paramètre est un paramètre que l'algorithme n'apprend pas puisque son optimisation est en pratique très difficile, et ne dépend en général pas des données observées.

Combien d'hyper-paramètres dans une régression polynomiale ?

$$y = f(x) = \sum_{i=0}^k \theta_i x^i = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_k x^k$$

Nb hyperparamètres = 1 : k



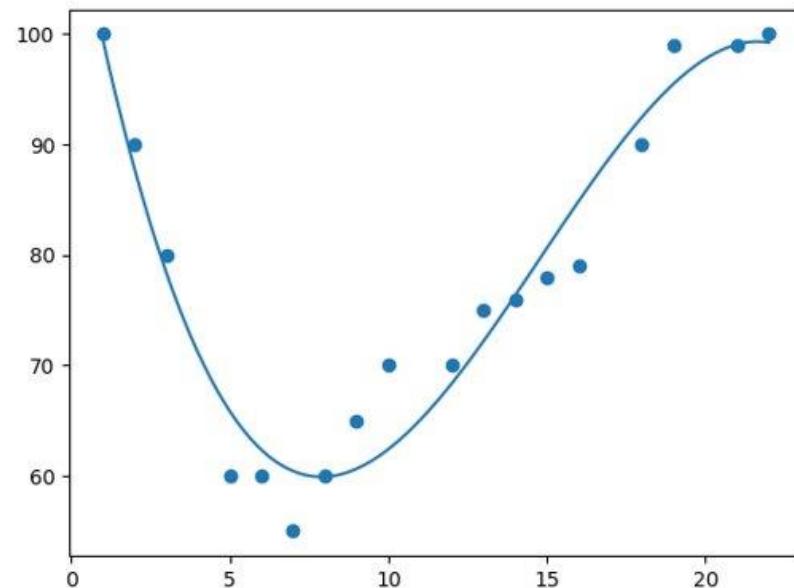
4. Hyper-paramètres et datasets de validation

DEF: Un hyper-paramètre est un paramètre que l'algorithme n'apprend pas puisque son optimisation est en pratique très difficile, et ne dépend en général pas des données observées.

Combien d'hyper-paramètres dans une regression polynomiale avec une ridge regression ?

$$y = f(x) = \sum_{i=0}^k \theta_i x^i = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_k x^k$$

$$l(\theta, x) + \lambda(\theta_0^2 + \theta_1^2 + \theta_2^2 + \dots)$$



4. Hyper-paramètres et datasets de validation

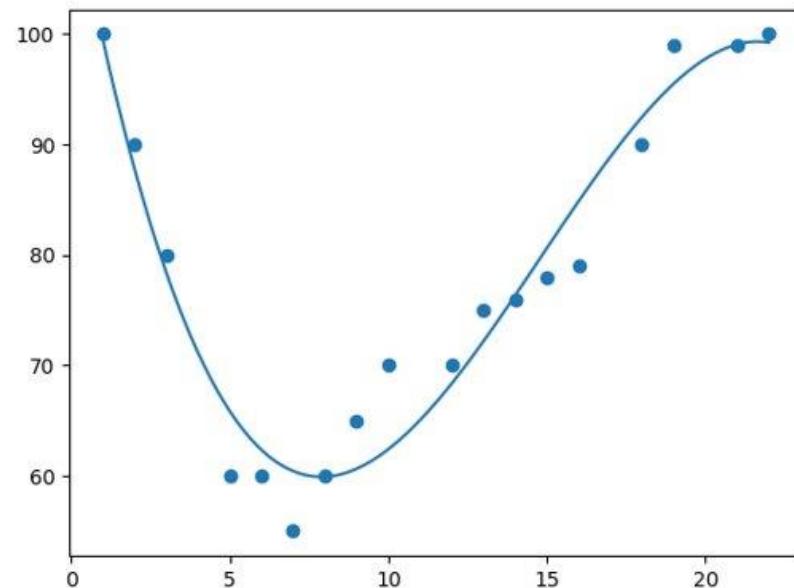
DEF: Un hyper-paramètre est un paramètre que l'algorithme n'apprend pas puisque son optimisation est en pratique très difficile, et ne dépend en général pas des données observées.

Combien d'hyper-paramètres dans une regression polynomiale avec une ridge regression ?

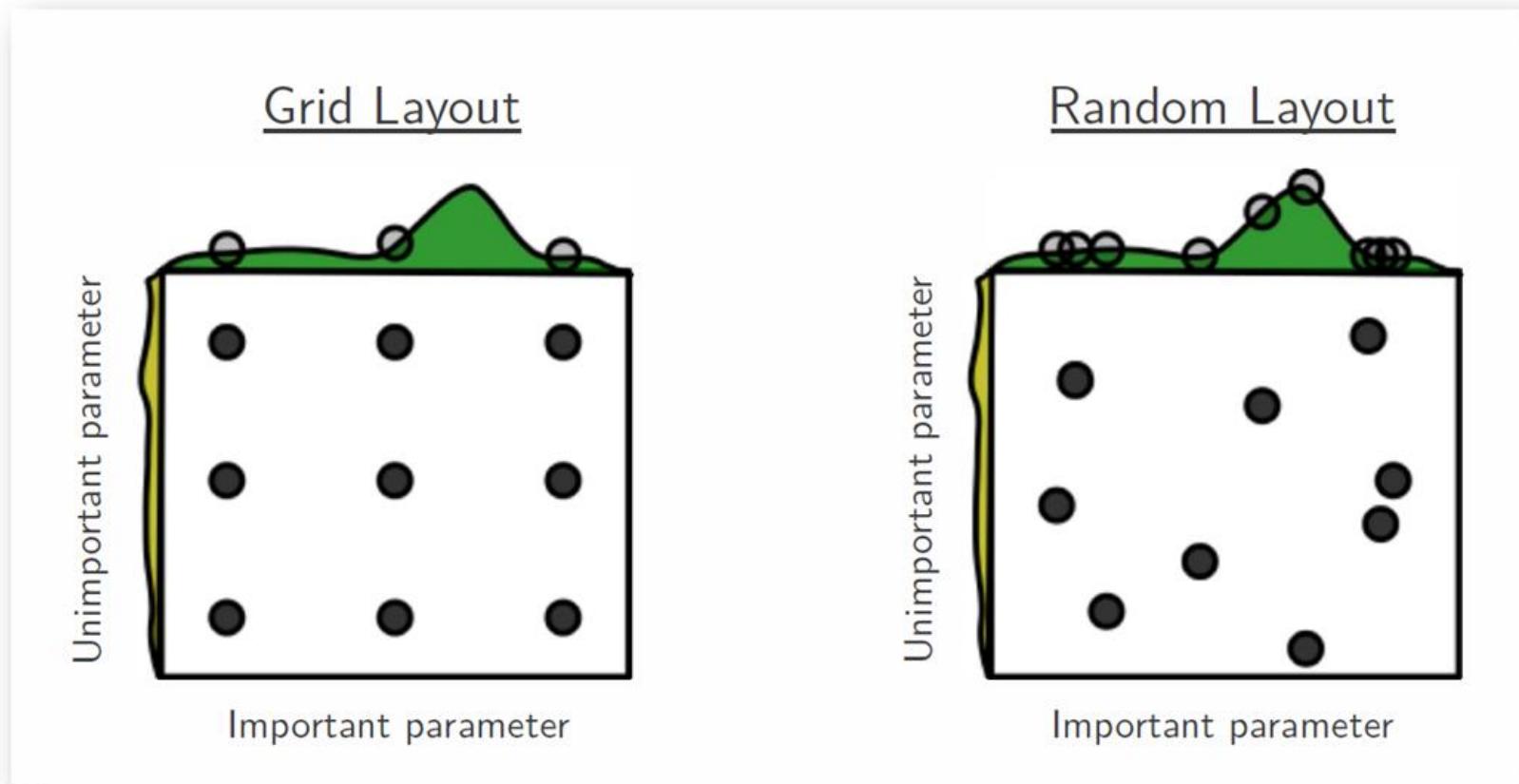
$$y = f(x) = \sum_{i=0}^k \theta_i x^i = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_k x^k$$

$$l(\theta, x) + \lambda(\theta_0^2 + \theta_1^2 + \theta_2^2 + \dots)$$

Nb hyperparamètres = 2 : k, lambda



Comment choisir les hyperparamètres ?

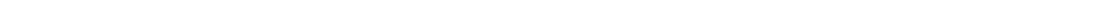
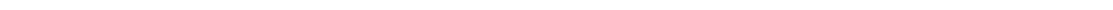


Hyper-parametres tuning

Cross-validation (k-fold)

Permet de trouver les hyper paramètres les plus adéquats pour la généralisation de notre modèle sous l'hypothèse que le dataset est suffisamment gros.

Ex: 4-fold cross validation

1. 
 2. 
 3. 
 4. 

Tous les patients ont été utilisés 3 fois pour l'entraînement et 1 fois pour l'évaluation

Permet d'évaluer le modèle sur l'ensemble de la cohorte d'entraînement

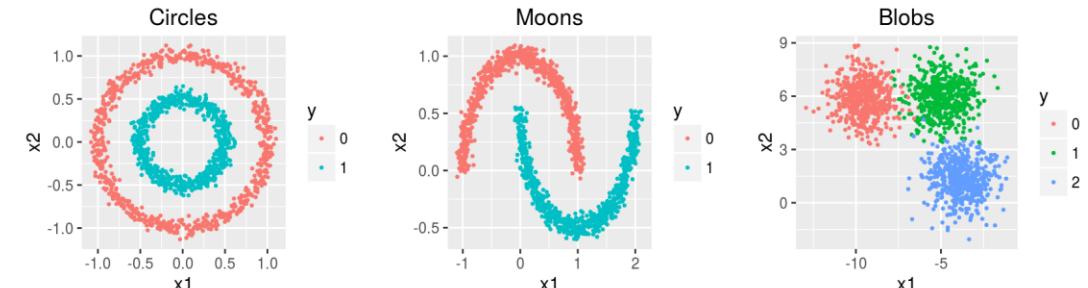
A retenir

- Dataset de train et de test
- Overfitting/ Underfitting
- Qu'est-ce qu'un hyperparamètre?
- Qu'est-ce qu'une régularisation?
- Cross-validation ?

III. Supervised learning

1. Definition

- On connaît les labels / les valeurs (y) associée à chaque exemple (x) qu'on veut prédire avec notre modèle.
- Exemples :
 - Dépistage de cancer : Cancer vs Pas de cancer,
 - Prédiction du prix de l'immobilier,
 - Triage de mails : SPAM ou non SPAM.
- En termes de probabilités: $p(y|x)$

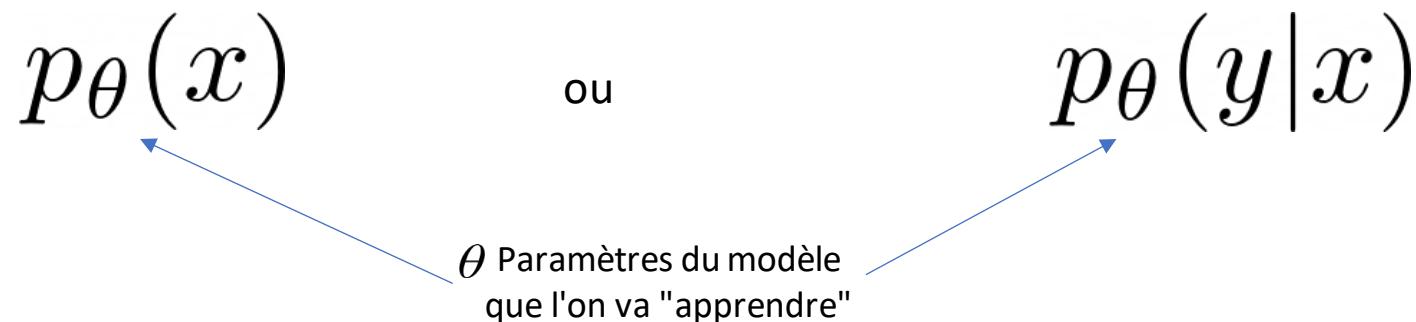


2. Définir le modèle et sa fonction de perte

- En machine learning on cherche à modéliser des probabilités: $p(\text{données})$ ou $p(\text{labels} | \text{données})$.

$$p_{\theta}(x) \quad \text{ou} \quad p_{\theta}(y|x)$$

θ Paramètres du modèle que l'on va "apprendre"



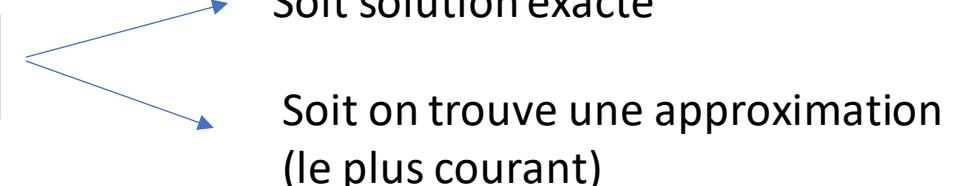
- Pour trouver le paramètre qui approchera au mieux les "vraies" distributions, on crée une fonction, appelée fonction de perte qui dépend des paramètres et des données: (on la crée à partir de principes de probabilités ou de statistiques)

$$l(\theta; x)$$

- Le but sera de minimiser l'espérance de la fonction de perte en fonction des paramètres pour obtenir le meilleur paramètre.

$$\min_{\theta} \mathbb{E}[l(\theta; x)]$$

Soit solution exacte
Soit on trouve une approximation (le plus courant)



3. Ex: Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$y = 0$

Si l'email est un spam, $y=1$. Sinon $y=0$.

La regression logistique est une méthode faite pour estimer la probabilité qu'un email soit un spam ou non à partir de données observées.

Ou plus généralement, qu'une donnée appartient à une classe ou non.

$x =$

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$y = 1$

Autrement dit, on cherche à estimer:

$$p(1|x)$$

3. Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$y = 0$

La regression logistique se base sur
l'hypothèse que $p(1|x)$ prend la forme:

$$p(1|x) = \sigma(\theta_0 x)$$

$x =$

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$y = 1$

3. Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$$y = 0$$

$x =$

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$$y = 1$$

La regression logistique se base sur l'hypothèse que $p(1|x)$ prend la forme:

$$p(1|x) = \sigma(\theta_0 x)$$



On reconnaît la régression linéaire

3. Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$$y = 0$$

$x =$

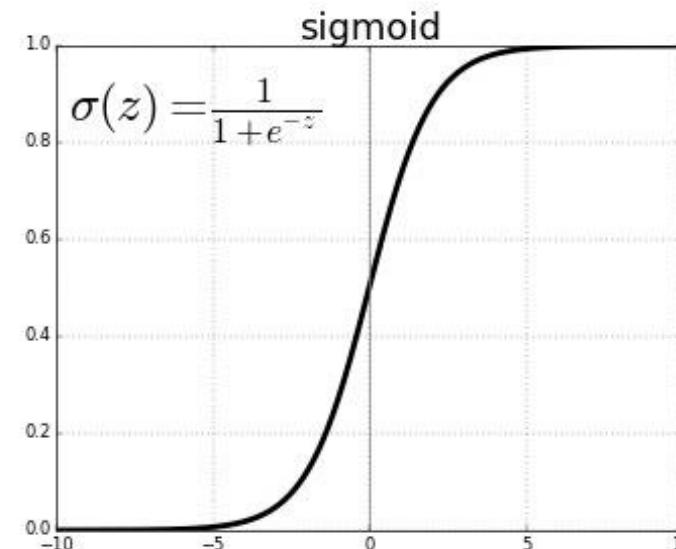
viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$$y = 1$$

La regression logistique se base sur l'hypothèse que $p(1|x)$ prend la forme:

$$p(1|x) = \sigma(\theta_0 x)$$



3. Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$$y = 0$$

La regression logistique se base sur l'hypothèse que $p(1|x)$ prend la forme:

$$p(1|x) = \sigma(\theta_0 x)$$

$x =$

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$$y = 1$$

La fonction de pertes à minimiser va s'écrire:

$$\begin{aligned} \mathbb{E}[l(\theta; x)] &\approx \sum_{i=1}^n y_i \log(p(1|x_i)) + (1 - y_i) \log(p(0|x)) \\ &= \sum_{i=1}^n y_i \log(\sigma(\theta_0 x_i)) + (1 - y_i) \log(1 - \sigma(\theta_0 x_i)) \end{aligned}$$

On l'appelle la Binary Cross-entropy

3. Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$$y = 0$$

La regression logistique se base sur l'hypothese que $p(1|x)$ prend la forme:

$$p(1|x) = \sigma(\theta_0 x)$$

$x =$

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$$y = 1$$

On ne connaît pas la solution exacte de ce problème donc on va trouver une solution approchée en utilisant une technique **classique** en Machine Learning: la **descente de gradient**.

$i=1$

4. Descente de gradient

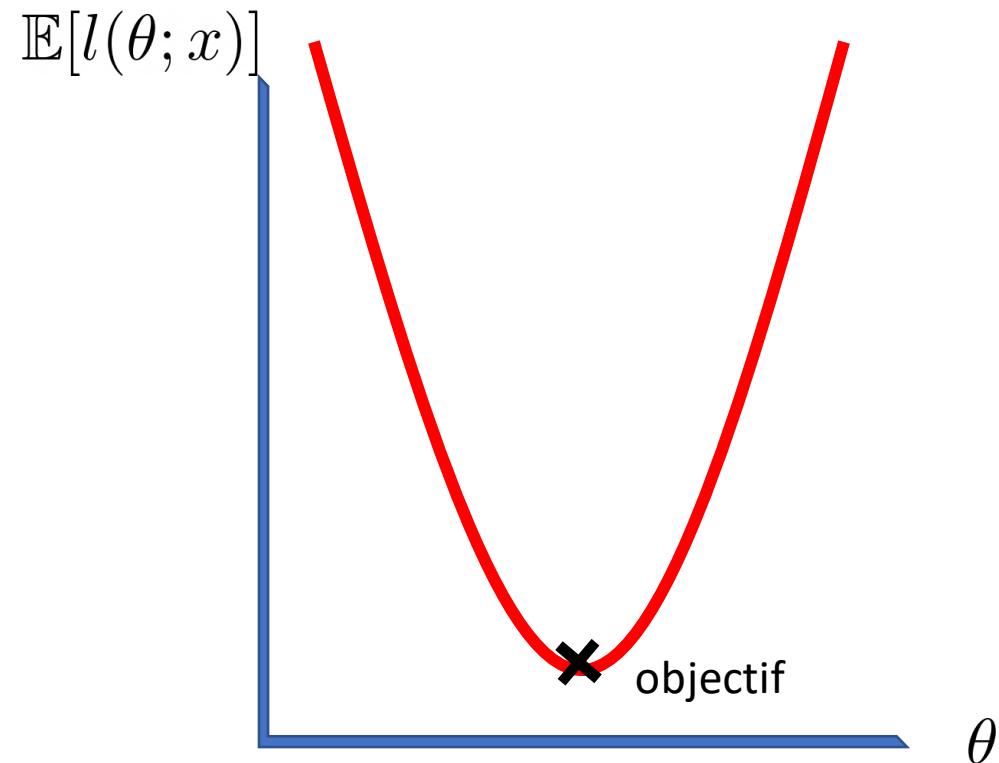
Aparté : Descente de Gradient

- La descente de gradient est un algorithme **itératif** qui permet de trouver une approximation des meilleurs paramètres de notre model.
- Pour chaque paramètre de notre model:
 - On commence avec des paramètres arbitraires (souvent aléatoire) θ_0
 - Puis à chaque itération, on met à jour simultanément tous les paramètres en suivant la formule suivante :

$$\theta_{t+1} = \theta_t - \lambda \nabla_{\theta} \mathbb{E}[l(\theta; x)]$$

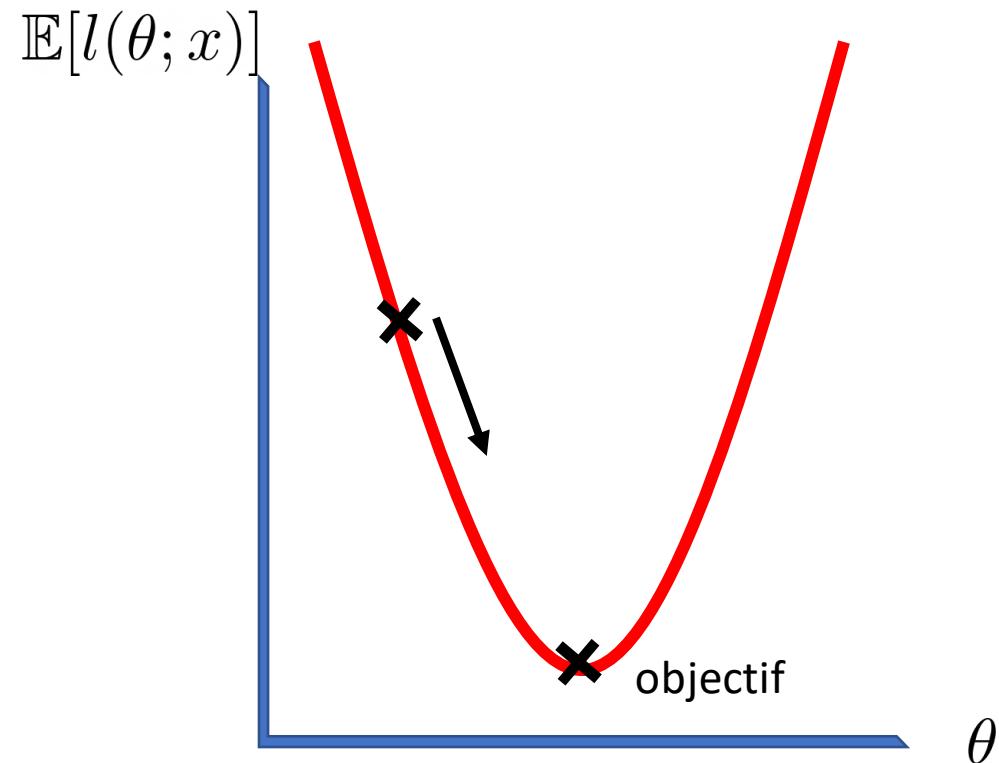
- Arrêter quand $\nabla_{\theta} \mathbb{E}[l(\theta; x)] \approx 0$

Descente de Gradient



$$\boxed{\nabla_{\theta} \mathbb{E}[l(\theta; x)]}$$

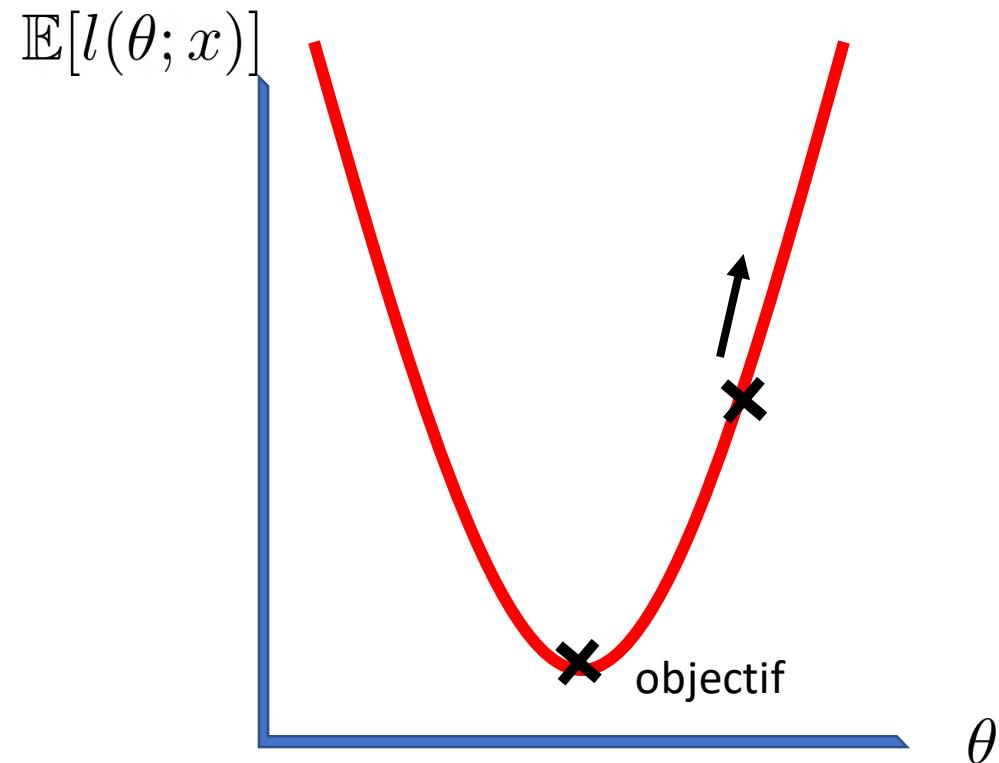
Descente de Gradient



$$\boxed{\nabla_{\theta} \mathbb{E}[l(\theta; x)]}$$

Valeur < 0

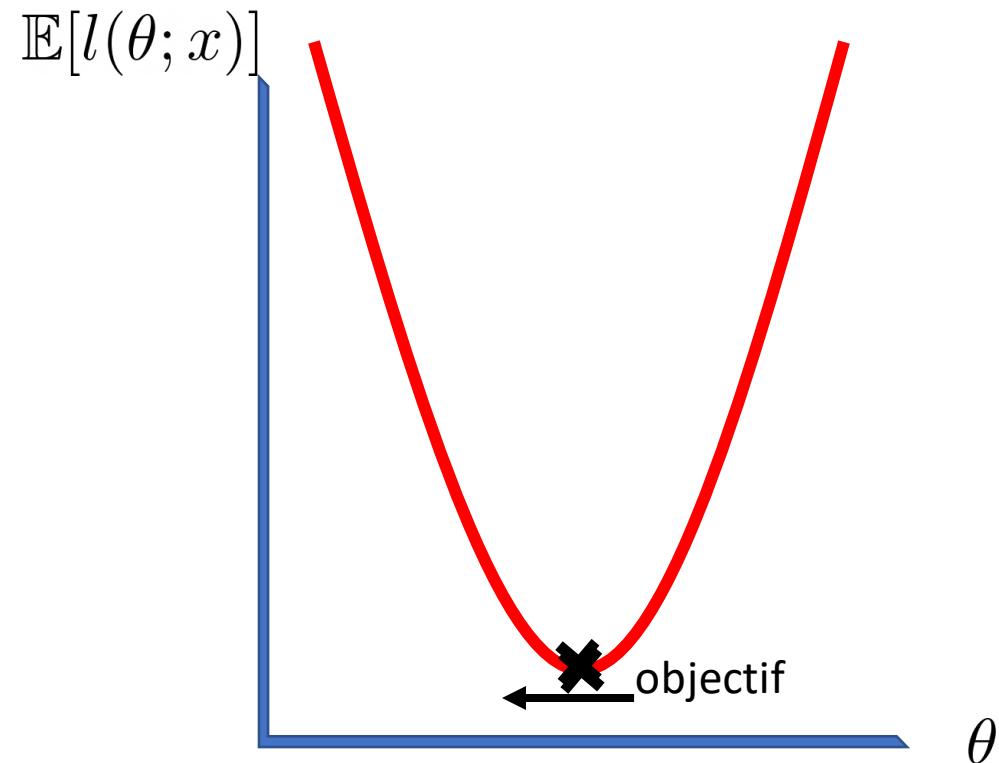
Descente de Gradient



$$\boxed{\nabla_{\theta} \mathbb{E}[l(\theta; x)]}$$

Valeur > 0

Descente de Gradient

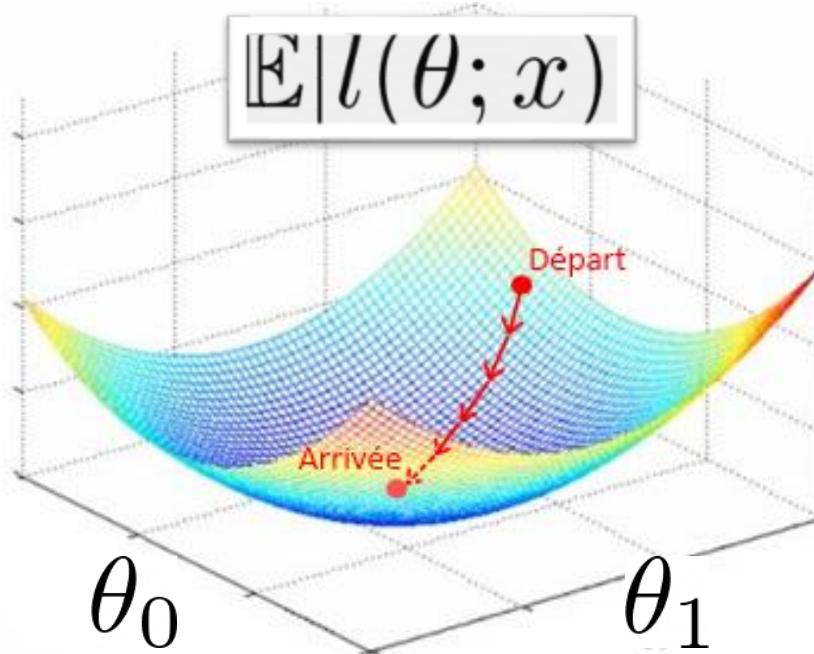


$$\boxed{\nabla_{\theta} \mathbb{E}[l(\theta; x)]}$$

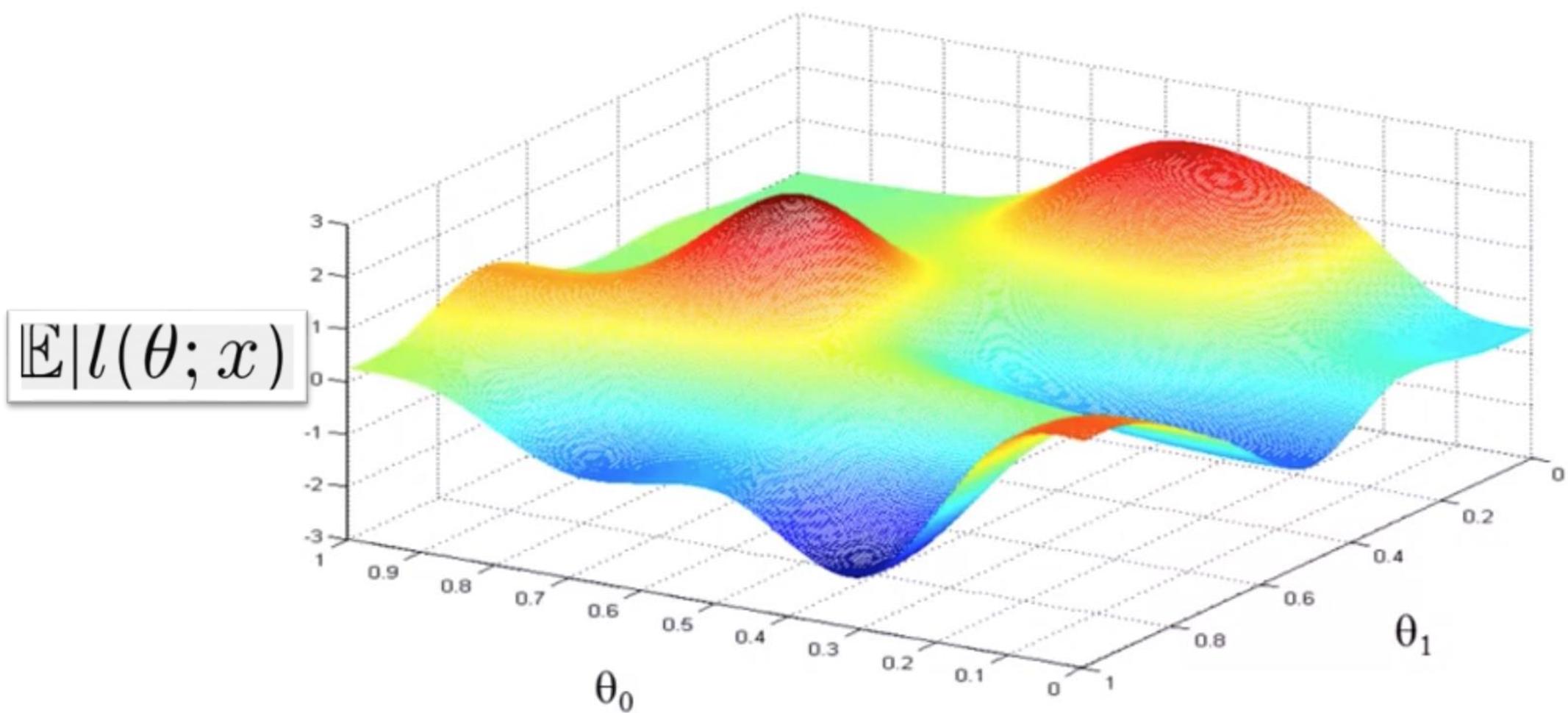
Valeur = 0

Descente de Gradient

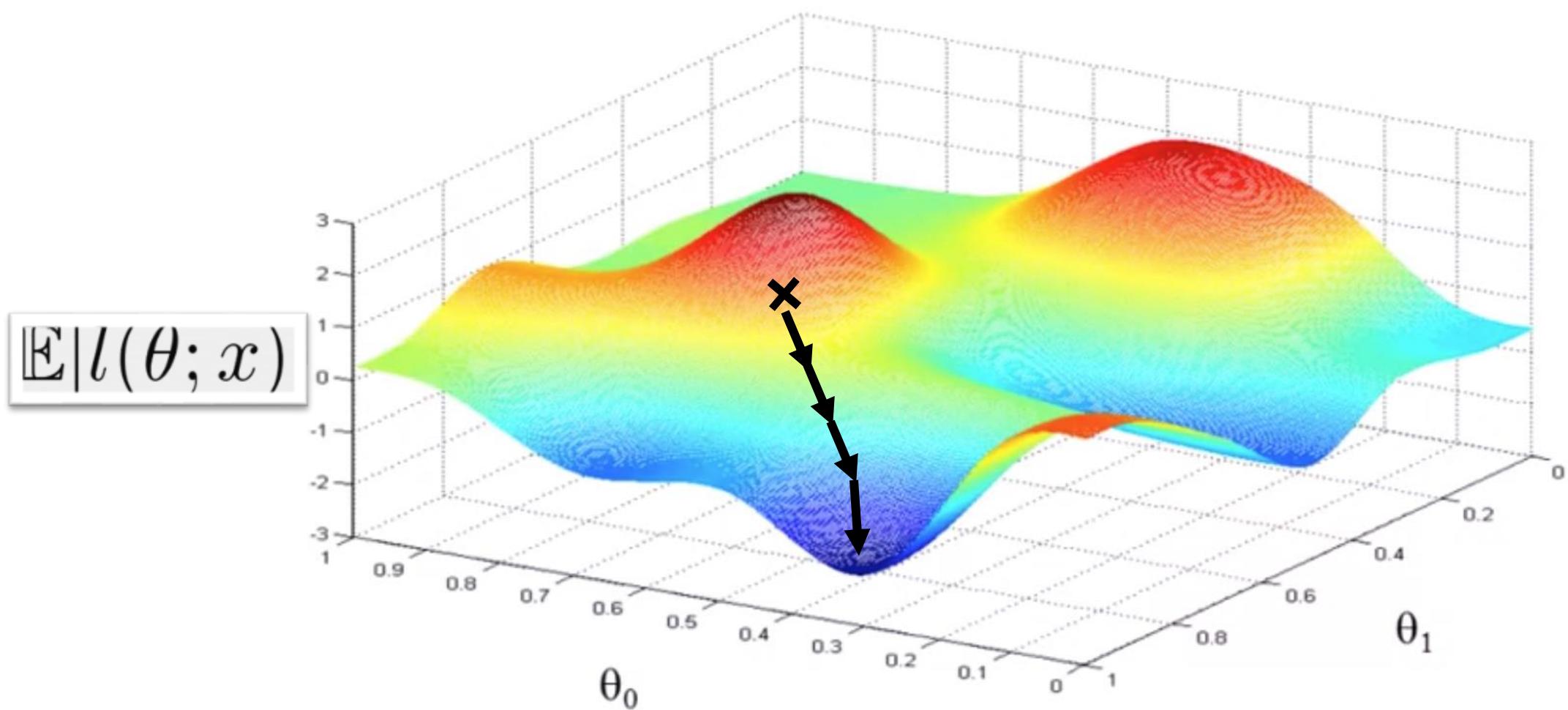
$$\theta_{t+1} = \theta_t - \lambda \nabla_{\theta} \mathbb{E}[l(\theta; x)]$$



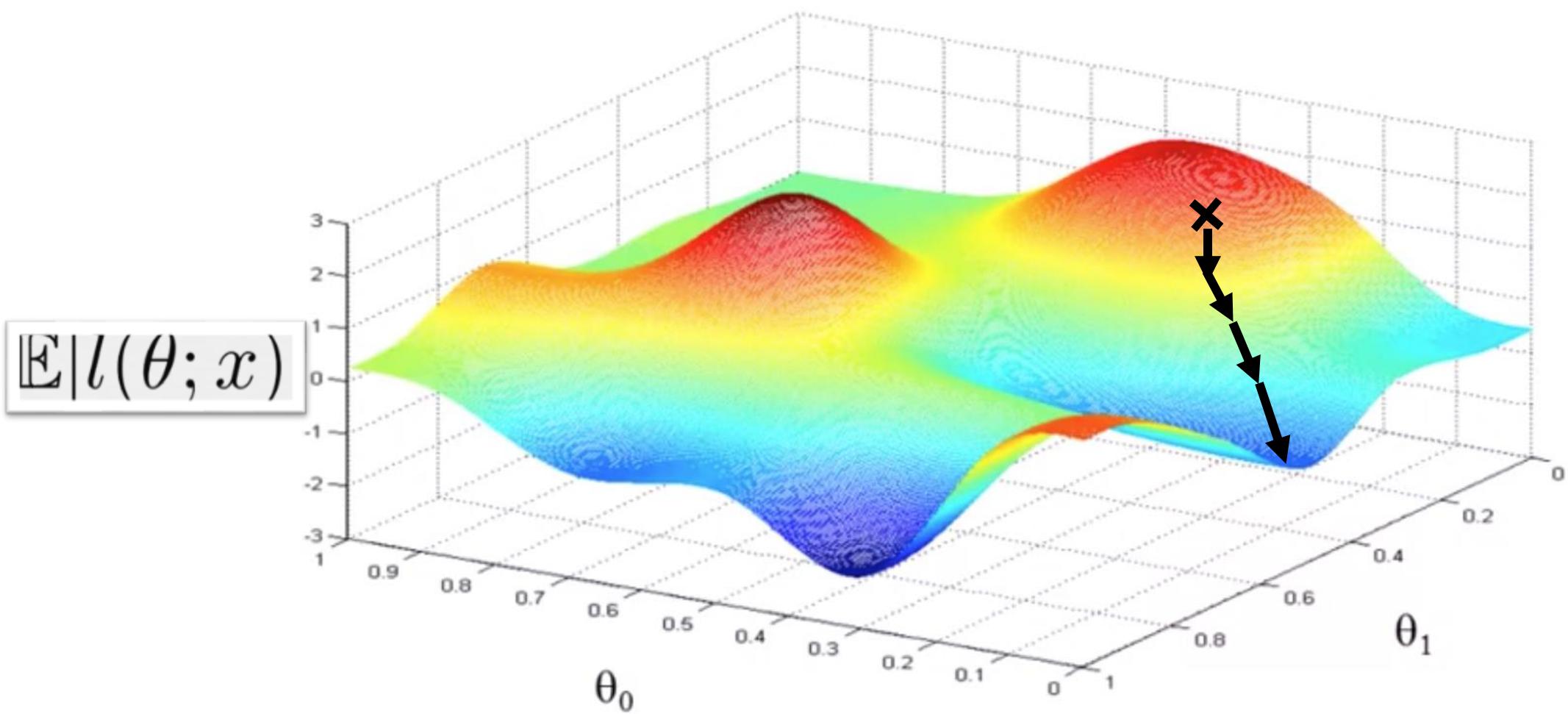
Descente de Gradient



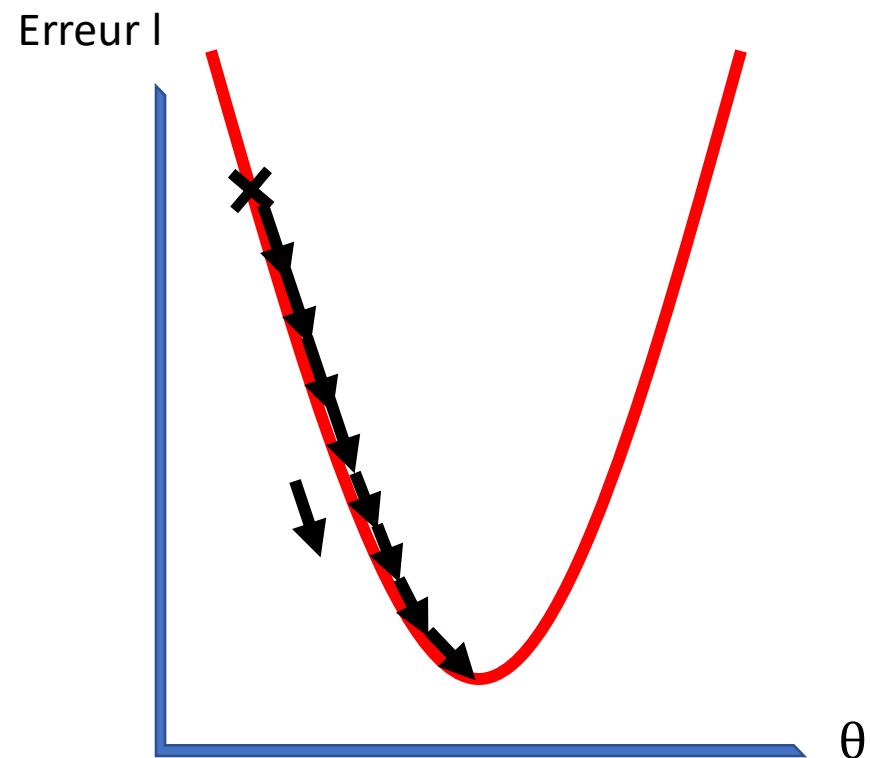
Descente de Gradient



Descente de Gradient



Taux d'apprentissage

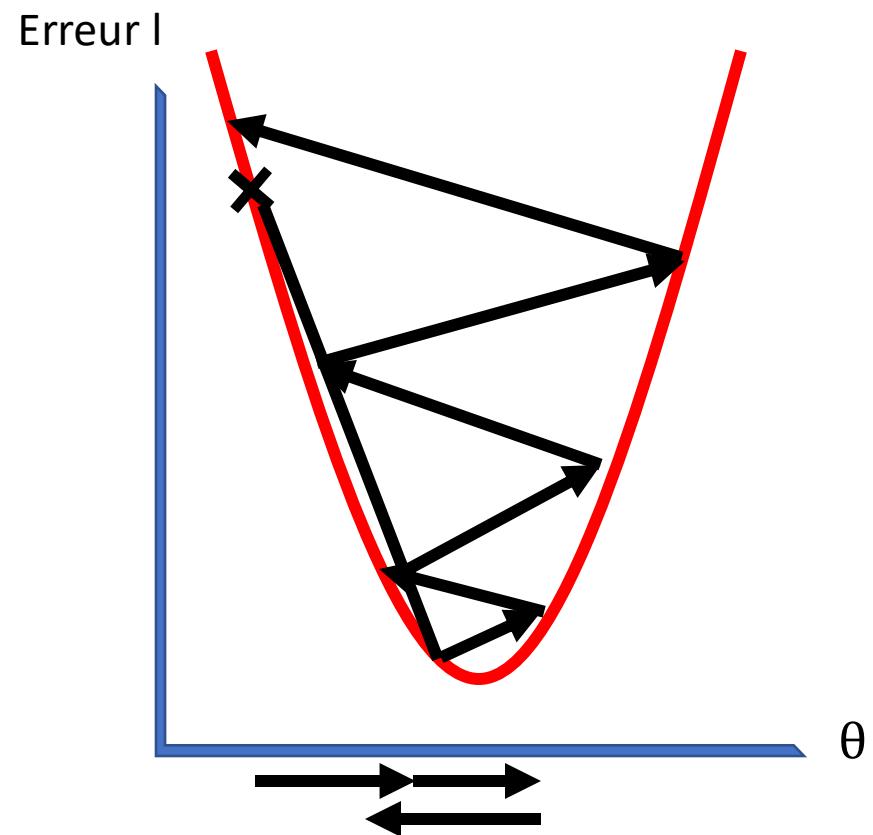


$$\theta_{t+1} = \theta_t - \lambda \nabla_{\theta} \mathbb{E}[l(\theta; x)]$$

λ trop petit : petites variations de θ

Beaucoup de calculs pour arriver au minimum

Taux d'apprentissage



λ trop grand: grandes variations de θ
Risque : pas de convergence voir même divergence

λ trop grand: grandes variations de θ
Risque : pas de convergence voir même divergence

4. Retour à la regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$$y = 0$$

La regression logistique se base sur l'hypothèse que $p(1|x)$ prend la forme:

$$p(1|x) = \sigma(\theta_0 x)$$

$x =$

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$$y = 1$$

On a fait la descente de gradient --> on obtient le meilleur paramètre

On a, à présent, un modèle de $p(1|x)$. Comment prendre la décision finale de classer un mail x comme un spam ou non ?

En d'autre terme, quelle sera notre règle de décision ?

4. Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$y = 0$

$x =$

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$y = 1$

La regression logistique se base sur l'hypothese que $p(1|x)$ prend la forme:

$$p(1|x) = \sigma(\theta_0 x)$$

Comment prendre la decision finale ?

Si $p(1|x) > \frac{1}{2}$, $y = 1$

Sinon, $y = 0$

4. Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$y = 0$

La regression logistique est un **classifieur binaire**.

Comment evaluer ses performances ?

$x =$

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$y = 1$

Take home messages

- Les briques de l'algorithme d'apprentissage (T, E, P)
- La construction du modèle
- La fonction de pertes et la descente de gradient
- Overfitting et underfitting + Regularization et hyperparametres.

Approches statistiques du Machine Learning – partie 2

Hugo Schmutz, doctorant 3IA



Inria



Plan du cours

IV. Modèles et algorithmes d'apprentissage supervisés

1. Regression linéaire
- 1.bis. Regression logistique
2. k-NN
3. Decision tree
4. Random forest
- 5.Comparaison

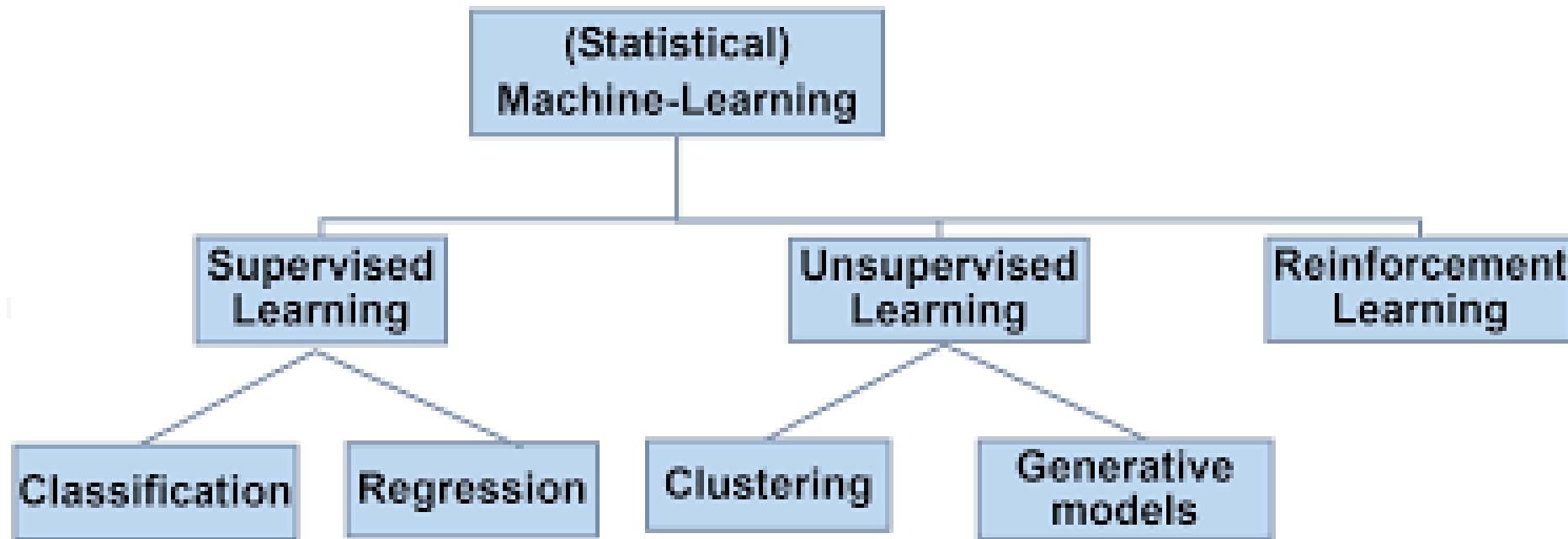
V. Modèles et algorithmes d'apprentissage non-supervisés

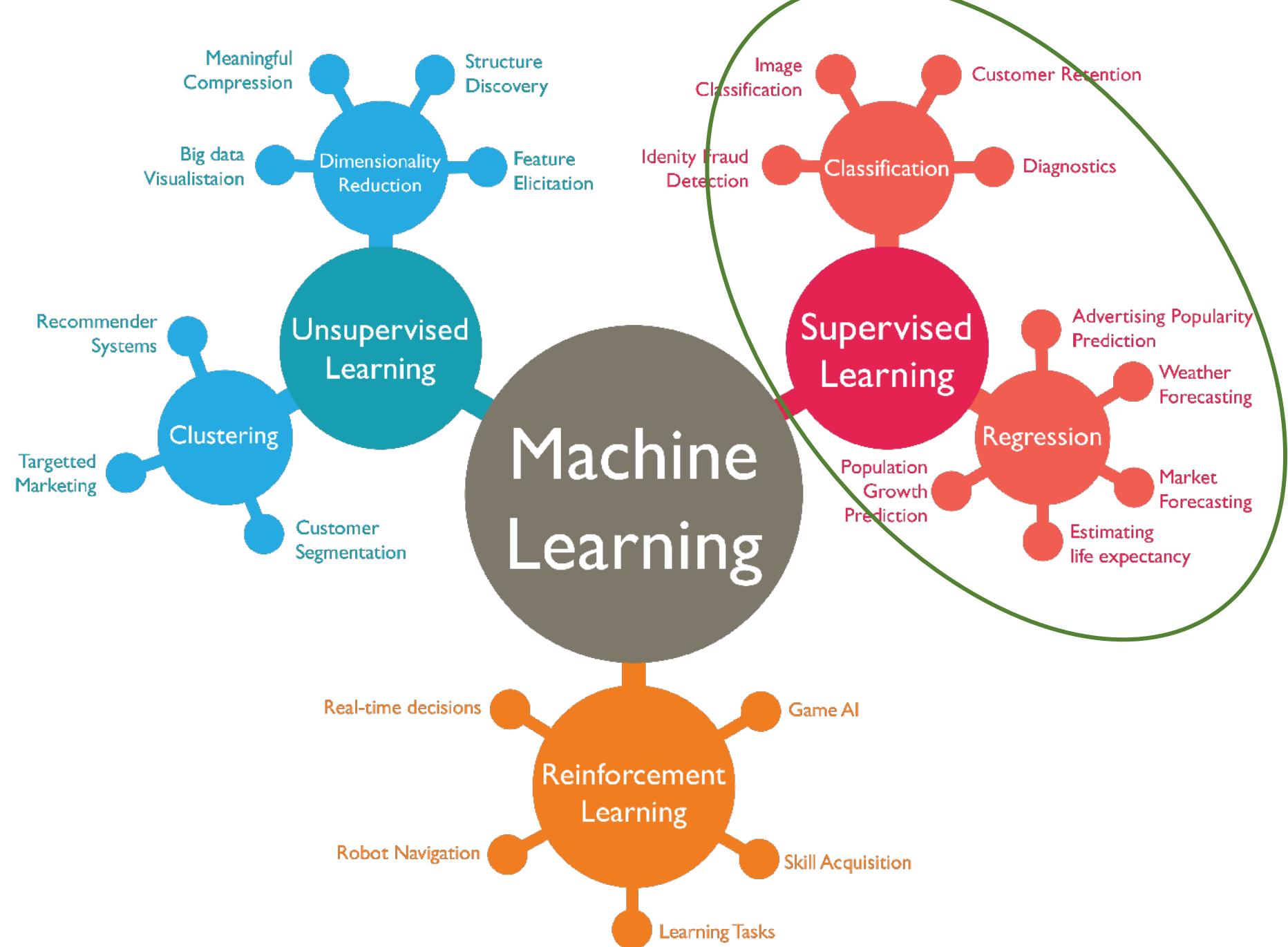
1. Analyse par composantes principales
2. k-means

VI. Vers le Deep Learning

1. La malédiction de la dimension
2. Perceptron
3. Multi-layer perceptron = Réseaux de neurones
4. Convolutional neural network (Deep Learning)

IV. Modèles et algorithmes d'apprentissage supervisés

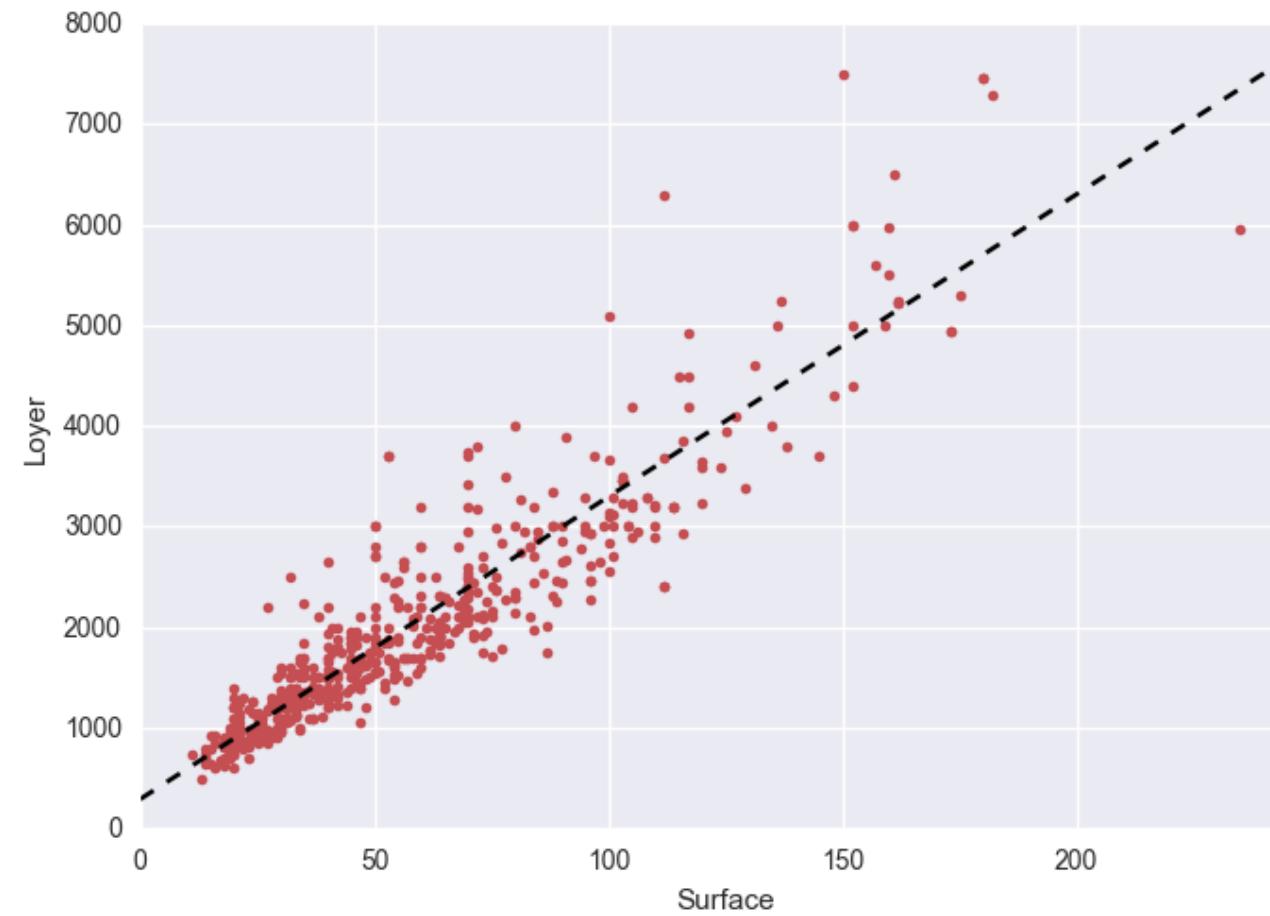




1. Rappel: Linear regression

Finalement, en minimisant la fonction de pertes,
on trouve le modèle suivant:

$$y = f(x) = \theta_0 x = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} x$$



1. Regression linéaire (Version probabilistique)

Changement de modélisation:

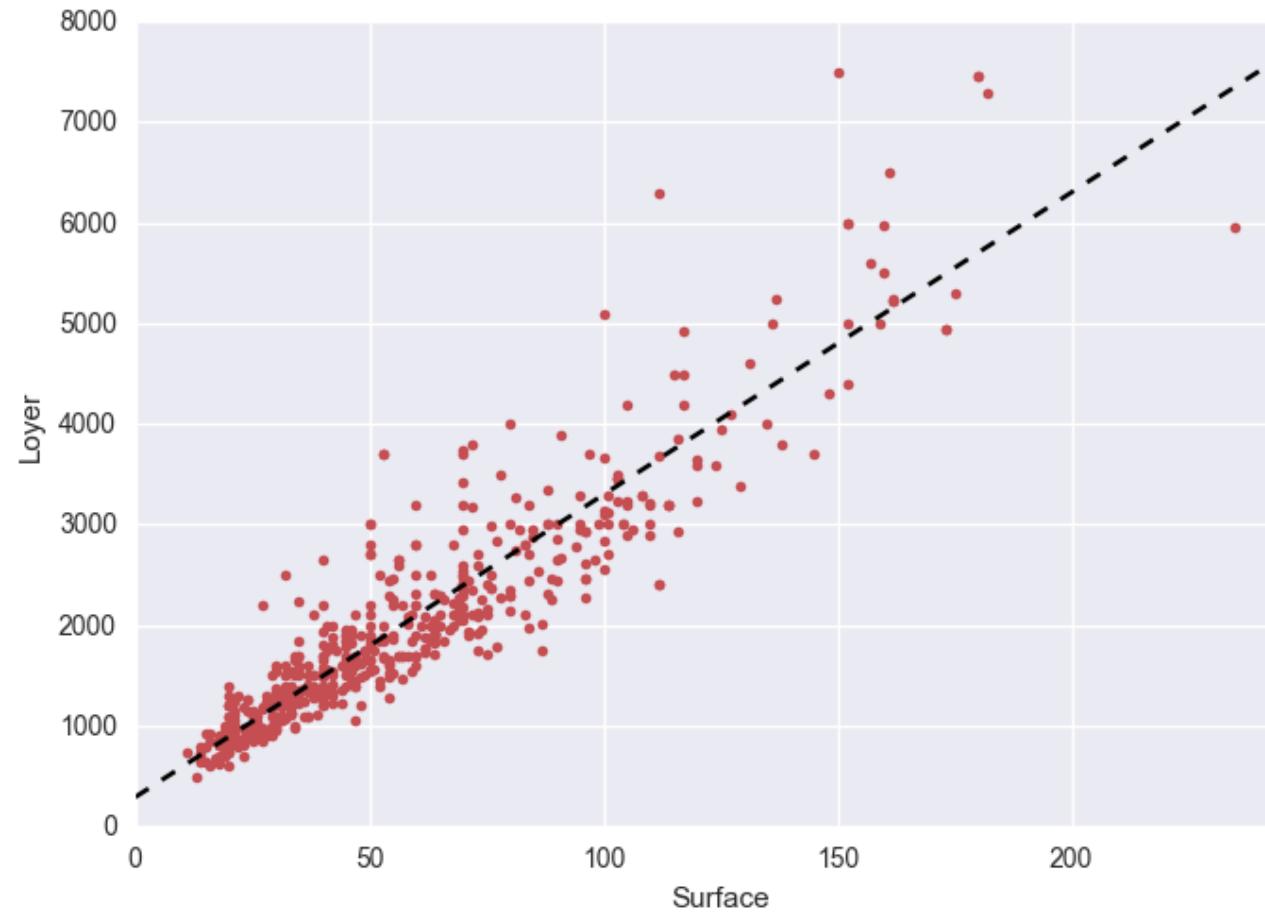
$$y = f(x) + \epsilon$$

Avec le bruit:

$$\epsilon \sim \mathcal{N}(0, 1)$$

Et:

$$f(x) = \theta_0 x$$



1. Régression linéaire (Version probabilistique)

Changement de modélisation:

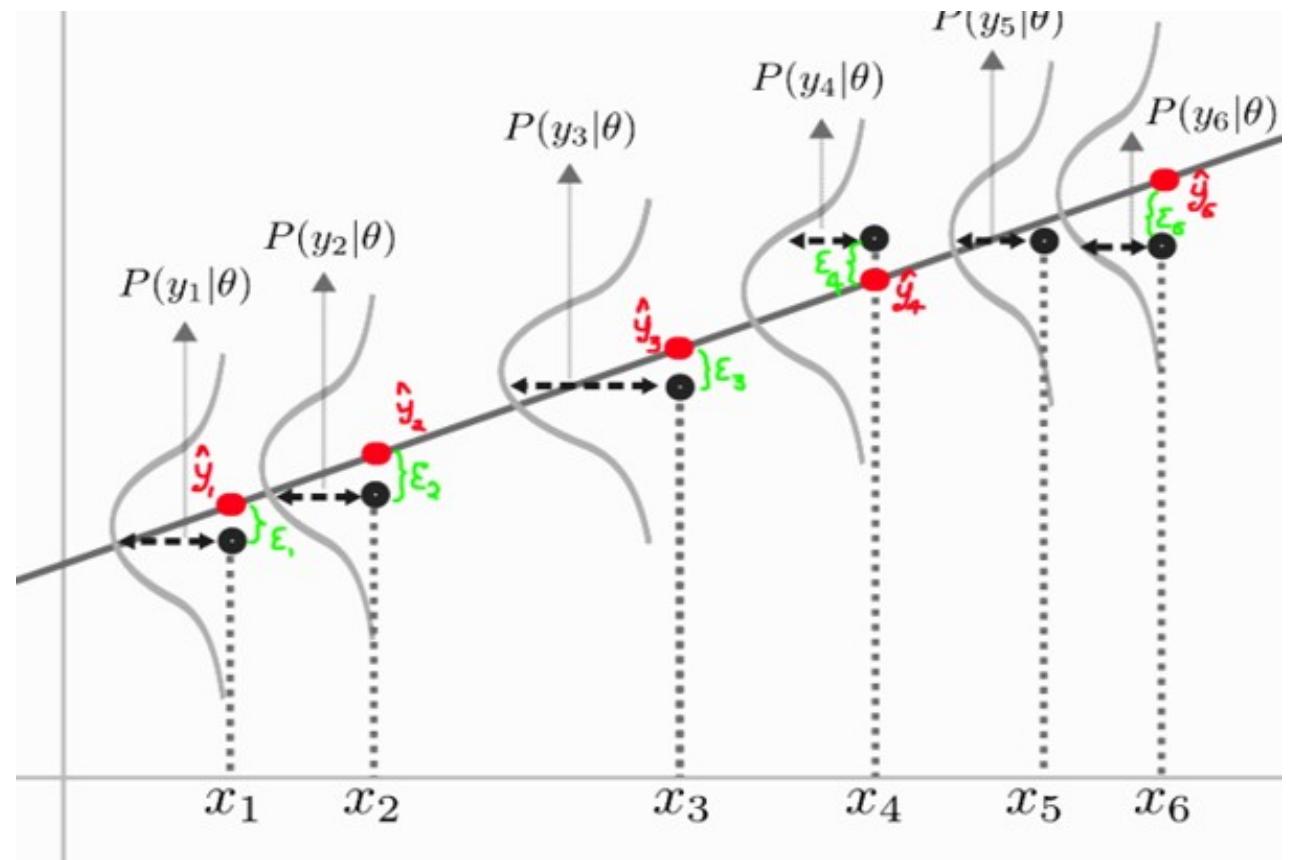
$$y = f(x) + \epsilon$$

Avec le bruit:

$$\epsilon \sim \mathcal{N}(0, 1)$$

Et:

$$f(x) = \theta_0 x$$



1. Regression linéaire (Version probabiliste)

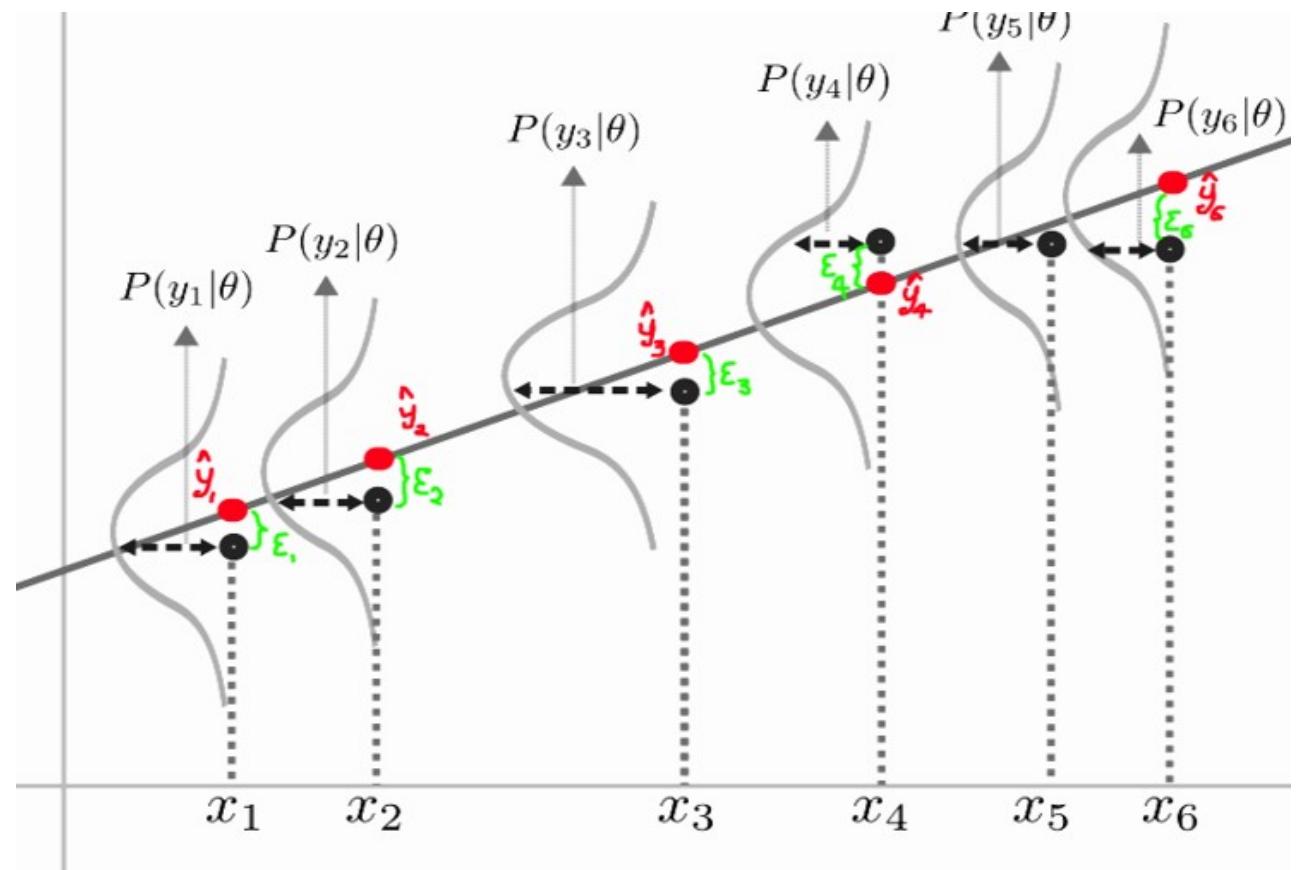
Changement de modélisation:

$$y = f(x) + \epsilon$$

Donc:

$$y|x \sim \mathcal{N}(\theta_0 x, 1)$$

$$P(y|x) = \frac{1}{\sqrt{2\pi}} e^{-(y-\theta_0 x)^2/2}$$



1. Regression linéaire

- **Avantages**
 - **Model simple:** Relation la plus simple possible entre les données et les labels.
 - **Tres rapide et léger :** pas de calculs compliqués comme la solution est exacte.
 - **Interprétabilité:** Comprendre l'influence relative des variables sur la prédiction et le "sens" d'influence (i.e. positif ou négatif)
- **Inconvénients**
 - **TROP simple:** Les relations entre les variables sont en général bien plus compliquées que ça.
 - **Tres affecté par les données isolées.**
 - **Hypothèse de l'indépendance des variables.** Ne tient jamais en pratique
 - **Incapable de détecter l'importance des variables:** deux features très corrélées vont se partager les poids.

1.bis: Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$y = 0$

Si l'email est un spam, $y=1$. Sinon $y=0$.

La regression logistique est une méthode faite pour estimer la probabilité qu'un email soit un spam ou non à partir de données observées.

Ou plus généralement, qu'une donnée appartient à une classe ou non.

$x =$

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$y = 1$

Autrement dit, on cherche à estimer:

$$p(1|x)$$

1.bis: Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$$y = 0$$

La regression logistique se base sur l'hypothèse que $p(1|x)$ prend la forme:

$$p(1|x) = \sigma(\theta_0 x)$$

$x =$

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$$y = 1$$

1.bis: Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$$y = 0$$

La regression logistique se base sur
l'hypothèse que $p(1|x)$ prend la forme:

$$p(1|x) = \sigma(\theta_0 x)$$



$x =$

viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$$y = 1$$

On reconnaît la regression
linéaire

1.bis: Regression logistique

$x =$

hey	... 1829
I'm	... 1710
no	... 1191
where	... 1012
you	... 985
speak	... 873
learn	... 747
one	... 739

good letters

$$y = 0$$

$x =$

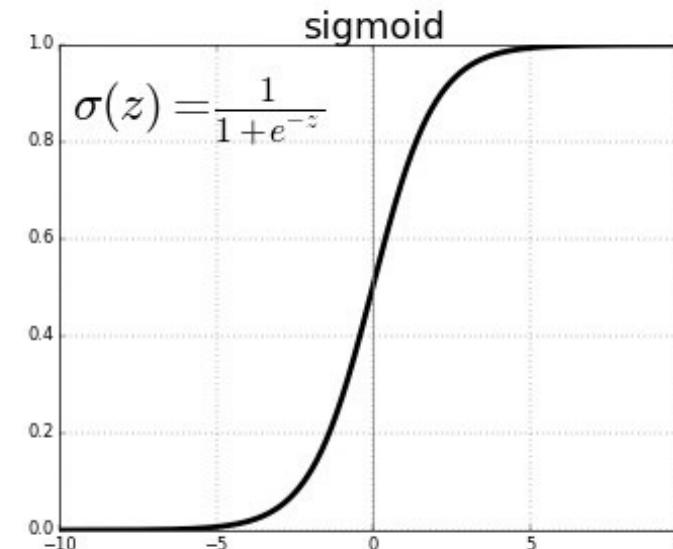
viagra	... 1552
casino	... 1492
100%	... 1320
credit	... 1184
sale	... 985
press	... 873
free	... 747
enlarge	... 739

spam letters

$$y = 1$$

La regression logistique se base sur l'hypothèse que $p(1|x)$ prend la forme:

$$p(1|x) = \sigma(\theta_0 x)$$



1.bis: Regression logistique - Evaluation

La matrice de confusion

		Réponse de l'expert	
		p	n
Réponse du classifer	Y	Vrai Positif	Faux Positif
	N	Faux Négatif	Vrai Négatif

1.bis: Regression logistique - Evaluation

La matrice de confusion

		Réponse de l'expert	
		p	n
		Vrai Positif	Faux Positif
Réponse du classifieur	Y	Vrai Négatif	Faux Négatif
	N	Faux Positif	Vrai Négatif

$$\text{Accuracy} = \frac{VP + VN}{VP + FP + VN + FN}$$

1.bis: Regression logistique - Evaluation

La matrice de confusion

		Réponse de l'expert
		p n
		Vrai Positif Faux Positif
Réponse du classifier	Y	Vrai Négatif
	N	Faux Négatif

$$Accuracy = \frac{VP + VN}{VP + FP + VN + FN}$$

$$Precision = \frac{VP}{VP + FP}$$

How many selected items
are relevant ?

$$Rappel = \frac{VP}{VP + FN}$$

How many relevant items
are selected ?

1.bis: Regression logistique - Evaluation

La matrice de confusion

		Réponse de l'expert	
		p	n
		Vrai Positif	Faux Positif
Réponse du classifier	Y	Vrai Négatif	Faux Négatif
	N	Faux Négatif	Vrai Négatif

$$\text{Accuracy} = \frac{VP + VN}{VP + FP + VN + FN}$$

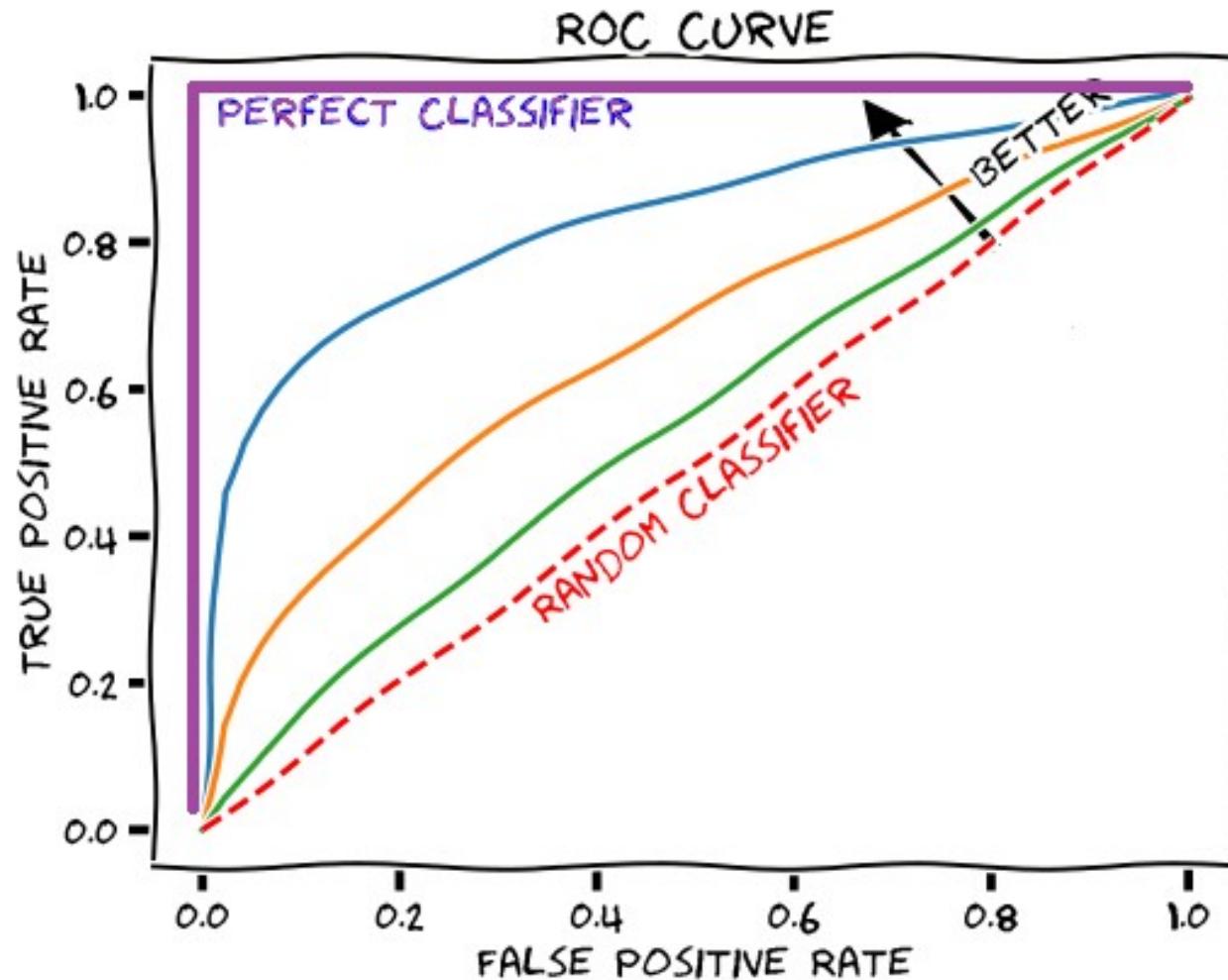
$$\text{Spécificité} = \frac{VN}{VN + FP}$$

How many negative selected people are truly negative ? Ex: How many healthy people are identified as not having the condition ?

$$\text{Rappel} = \frac{VP}{VP + FN} = \text{Sensibilité}$$

How many relevant items are selected ? Ex: How many sick people are correctly identified as having the condition ?

1.bis. Roc Curves:



- Est-ce le meilleur choix ?

Si $p(1|x) > \frac{1}{2}$, $y = 1$

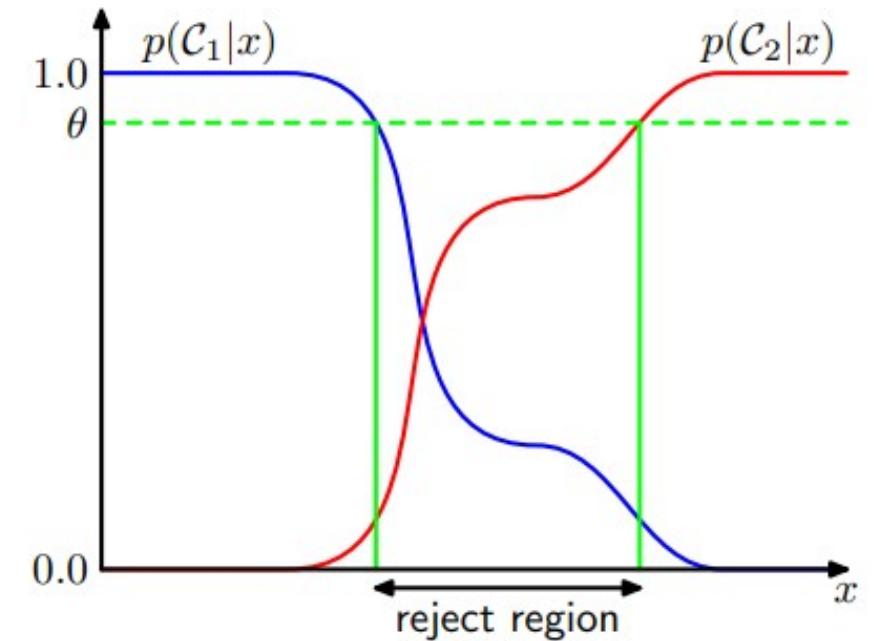
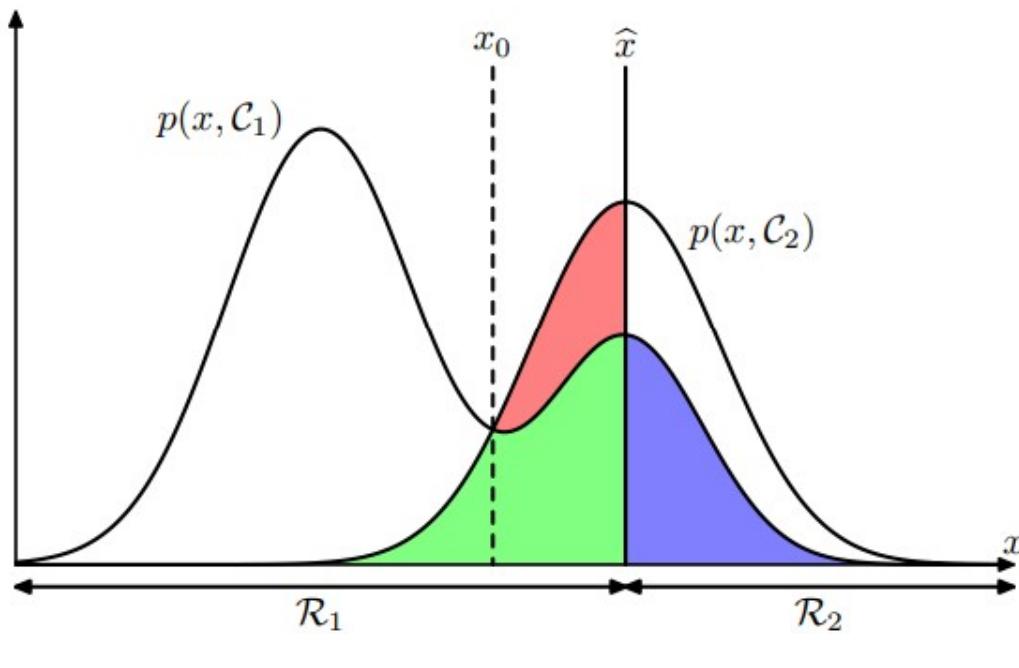
Sinon, $y = 0$

- Que se passe-t'il lorsque l'on est dans la situation $p(1|x) \sim \frac{1}{2}$? Peut-on légitimement classer x dans la classe 1 ?

Decision theory (Bishop, 2006)

Maintenant on connaît $p(1|x)$ et $p(0|x)$

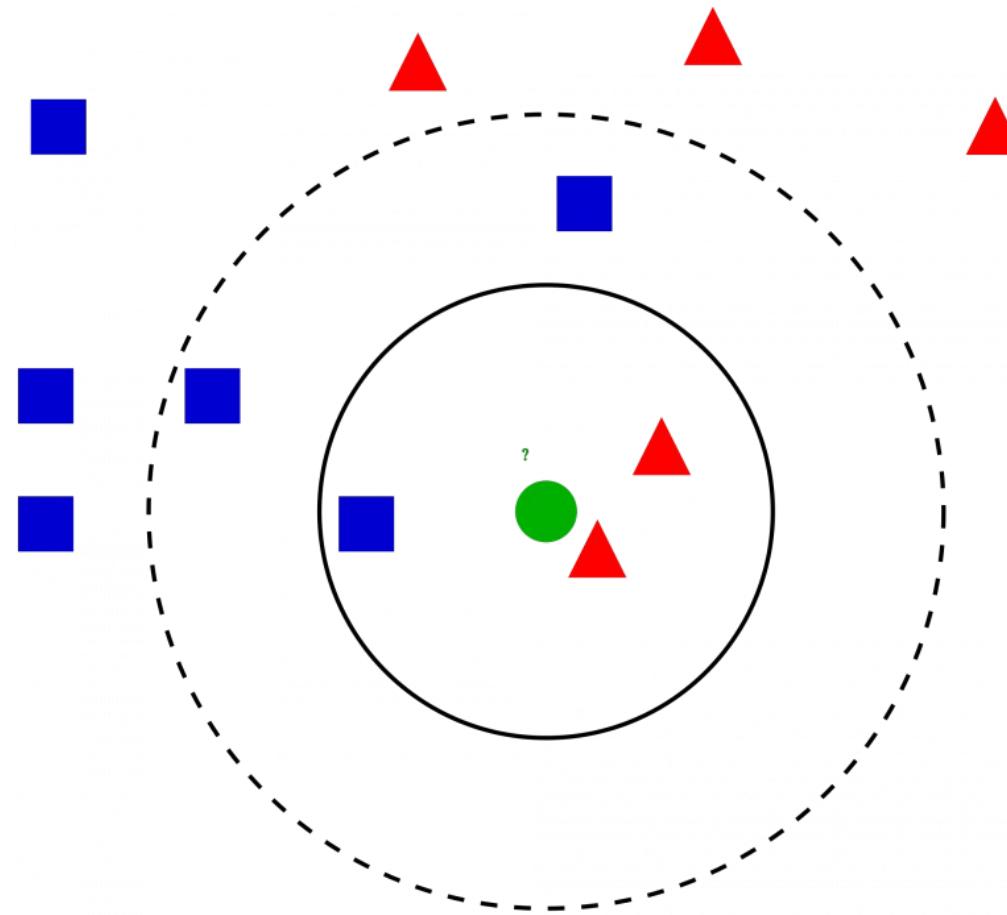
Comment prend on la decision finale de classer x dans la classe 0 ou 1 ?



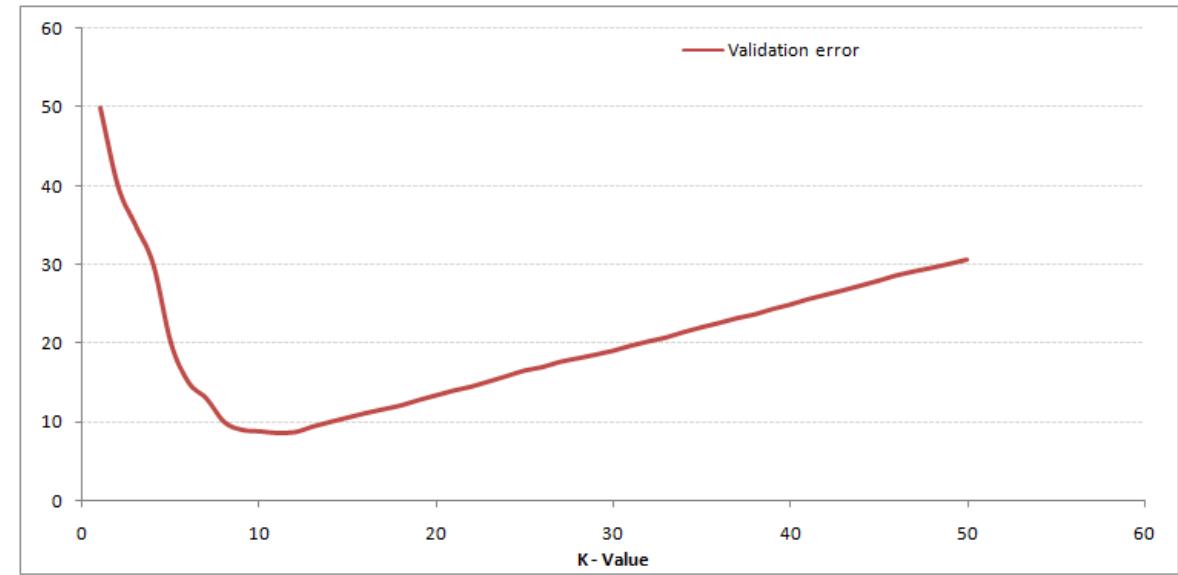
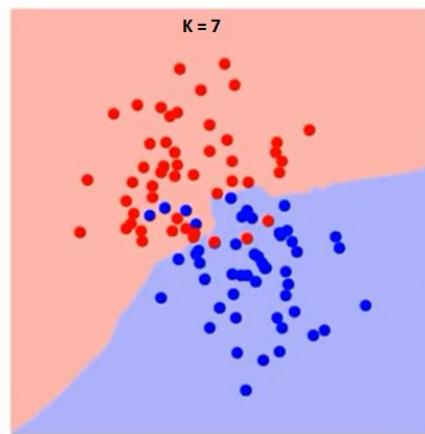
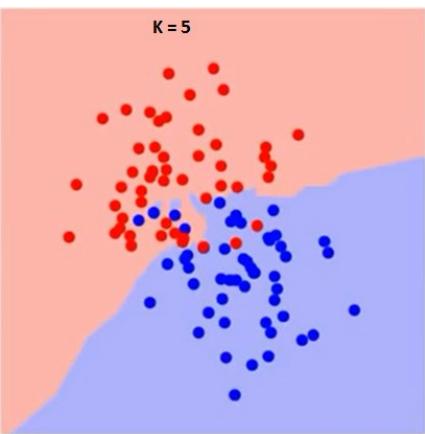
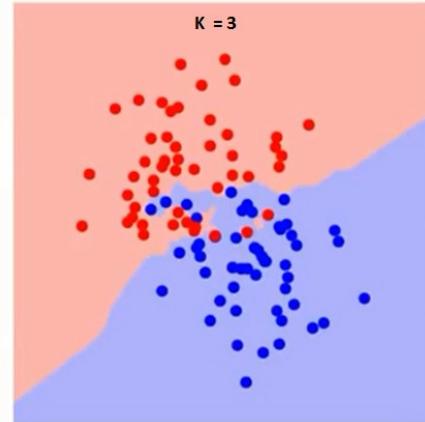
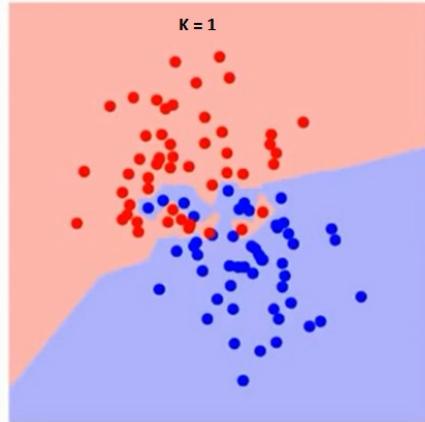
1. Regression logistique

- **Avantages**
 - **Model simple:** Relation la plus simple possible entre les données et les labels.
 - **Très rapide et léger :** pas de calculs compliqués comme la solution est exacte.
 - **Interprétabilité:** Comprendre l'influence relatif des variables sur la prédiction et le "sens" d'influence (i.e. positif ou négatif)
 - **Performance:** Très bonne performance sur les datasets linéairement séparable
- **Inconvénients**
 - **TROP simple:** Les relations entre les variables sont en général bien plus compliquées que ça.
 - **Très affecté par les données isolées.**
 - **Hypothèse de l'indépendance des variables.** Ne tient jamais en pratique
 - **Incapable de détecter l'importance des variables:** deux features très corrélées vont se partager les poids.
 - **Overfitting:** Sur les gros dataset ou sur les très petits datasets.
A retenir: si le nombre d'observations est plus faible que le nombre de variable, la logistique regression overfit.

2. Les k plus proches voisins



3. Les k plus proches voisins



3. Les k plus proches voisins

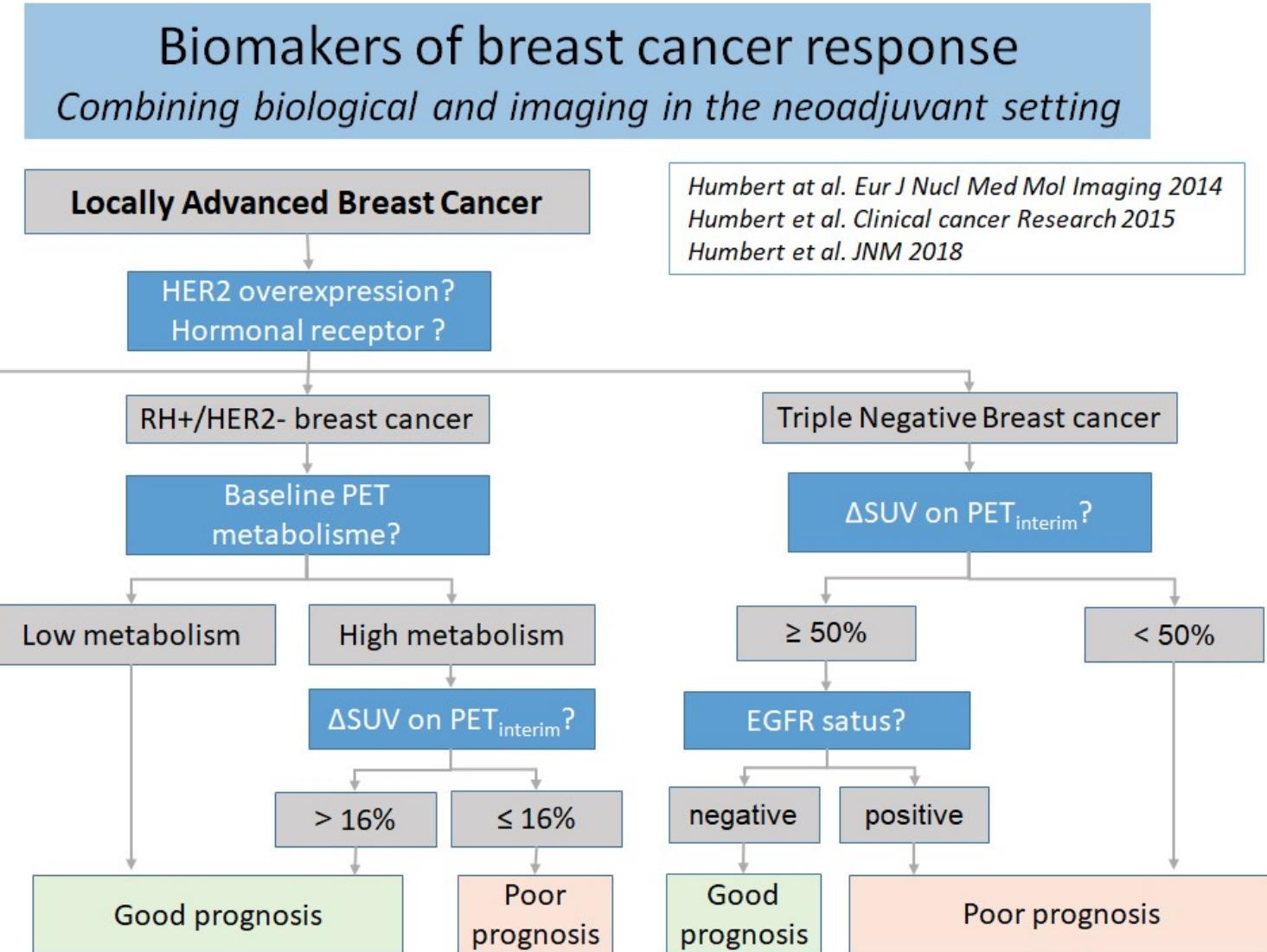
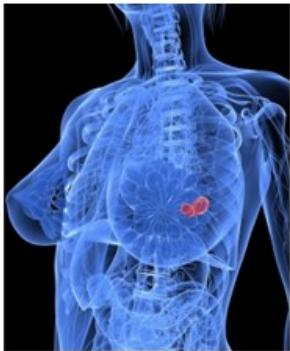
Avantages

- **K-NN est intuitif:** facile à comprendre, implémenter
- **Aucunes hypothèses sur les données:** algorithme non-paramétrique
- **Pas d'étapes d'entraînement:** étiqueter les nouvelles données à partir des anciennes
- **Évolue constamment:** si on rajoute des données
- **Facile à implémenter pour les problèmes multi-classes:** La plupart des classificateurs binaires sont difficiles à transformer en classificateur multi-classes.
- **Classification et Regression.**
- **Un seul Hyper Paramètre + choix de la distance:**

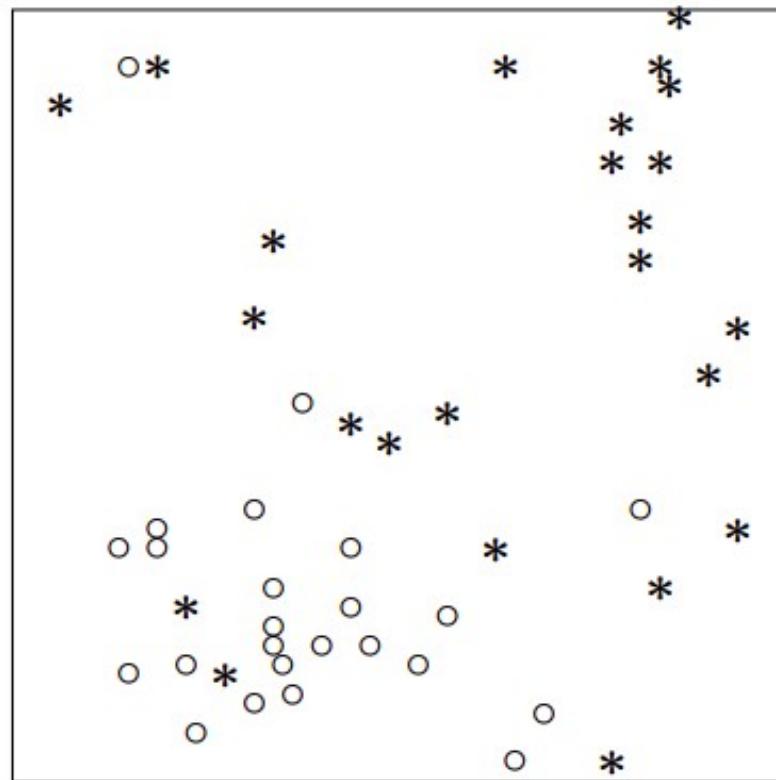
Inconvénients

- **K-NN est lent:** à chaque nouveau data-point, on doit calculer toutes les distances entre les points
- **Malediction de la dimension (Curse of Dimensionality):** marche bien sur des petites dimensions
- **Besoin absolu de variables homogènes.**
- **Quel est le K (nb de voisins) optimal ?**
- **Les classes non équilibrées posent problème:** Si une majorité des points du set d'entraînement sont d'une certaine classe
- **Sensible aux données isolées**

3. Arbre de decisions

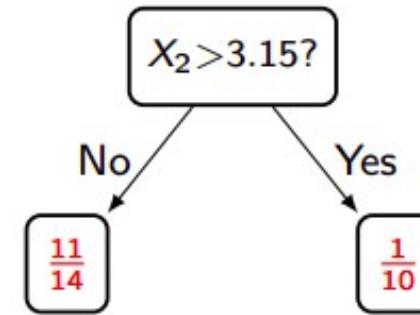
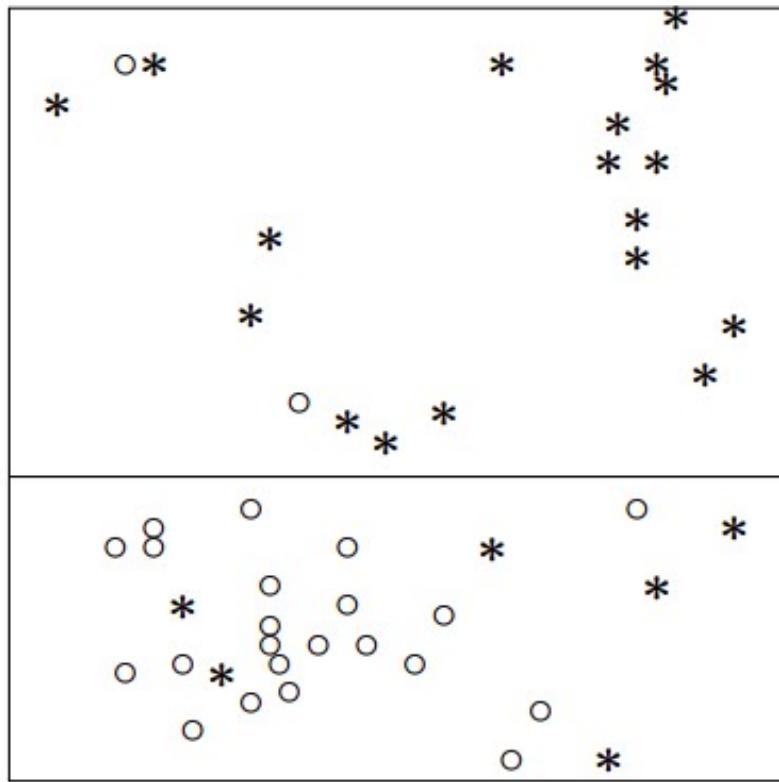


Construire un arbre de decision

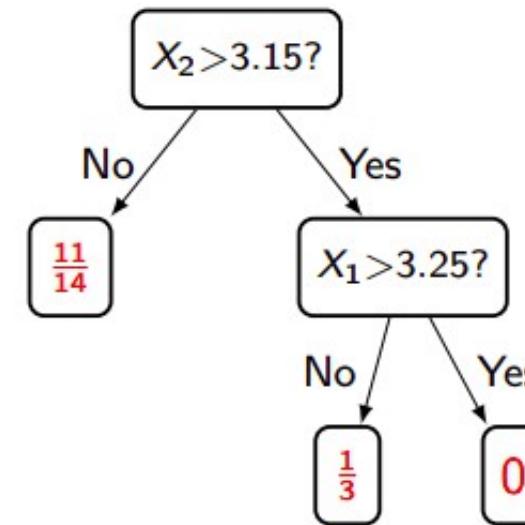
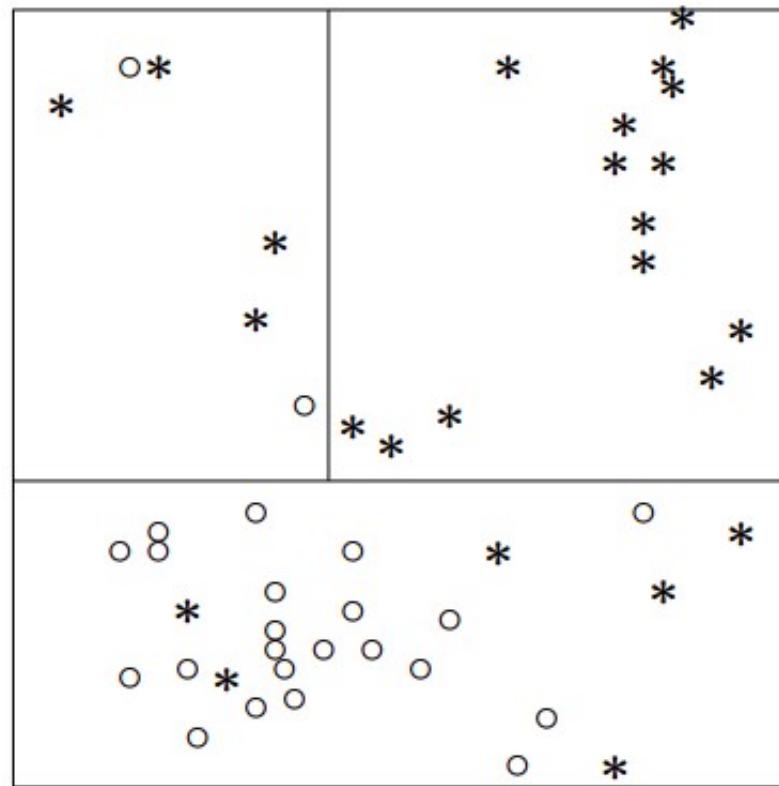


$$\boxed{\frac{1}{2}}$$

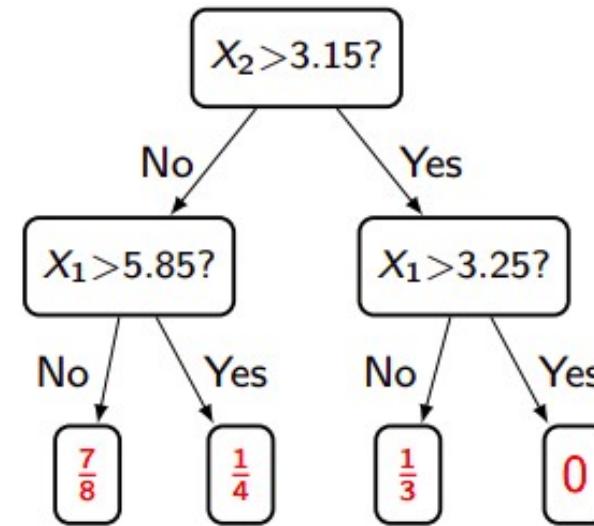
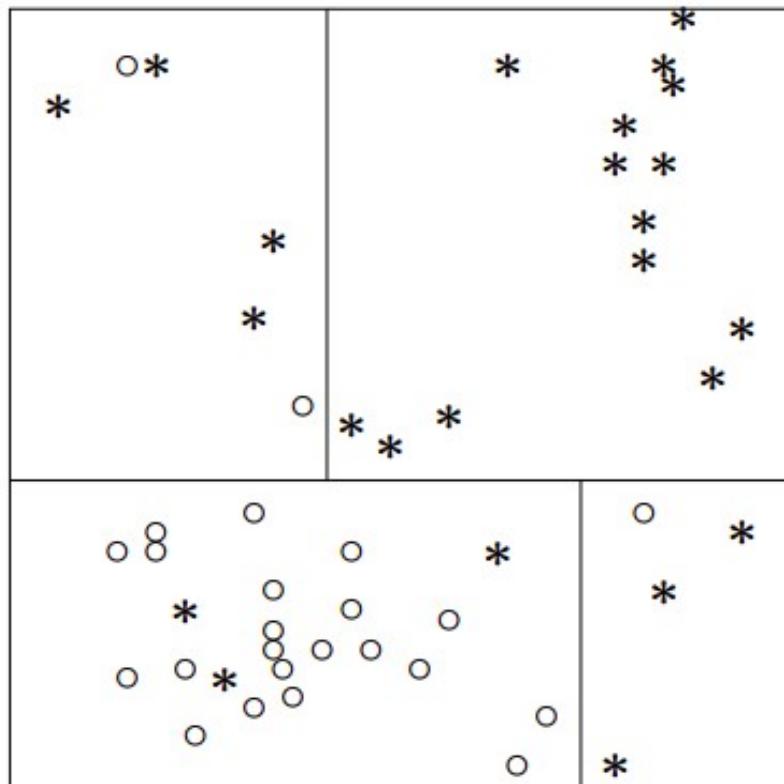
Construire un arbre de decision



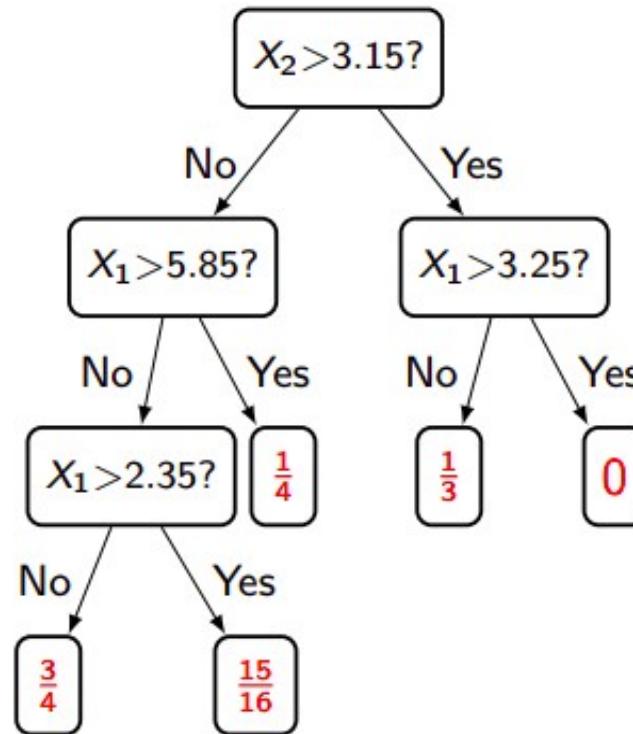
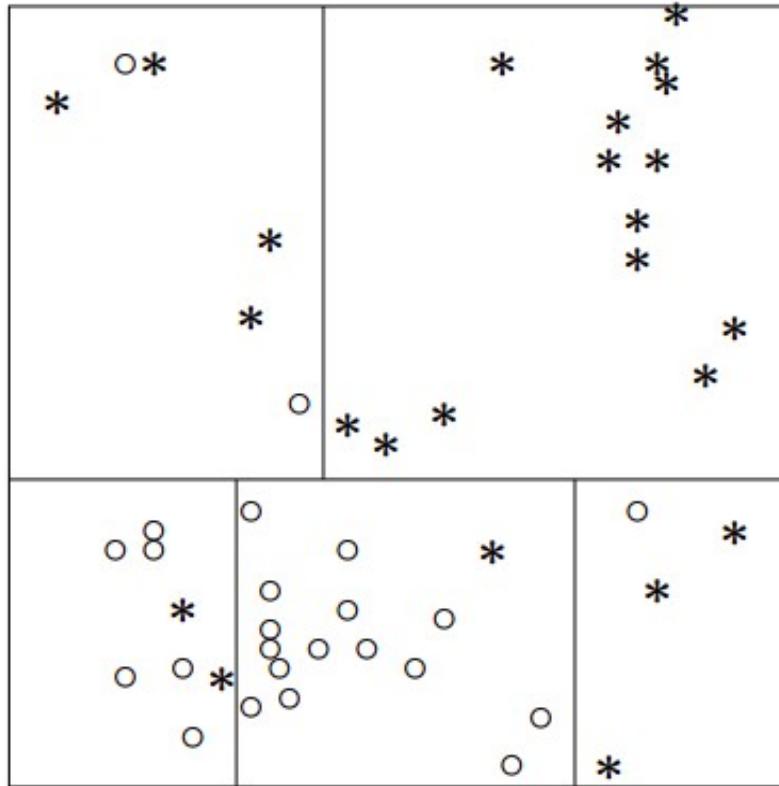
Construire un arbre de decision



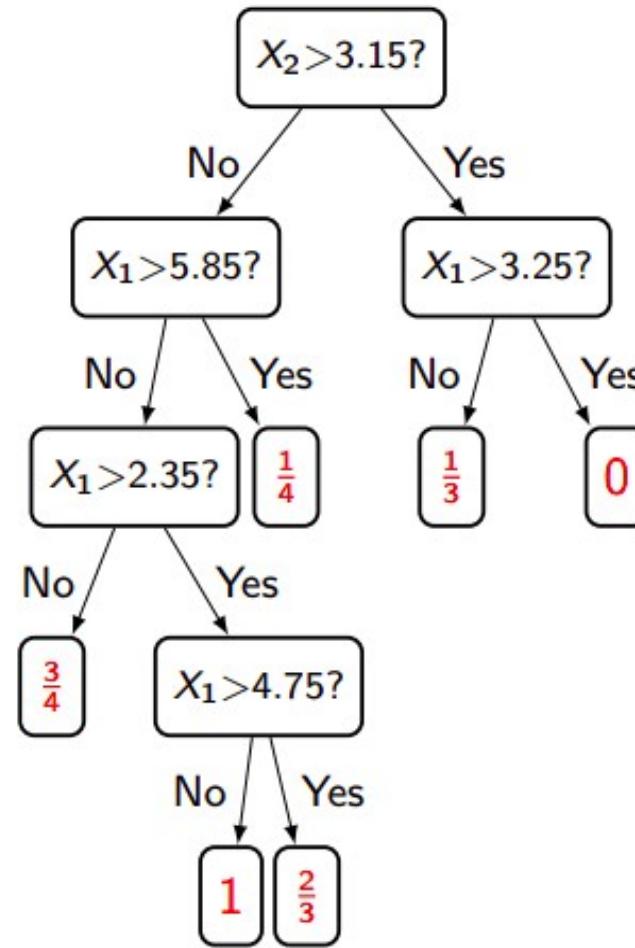
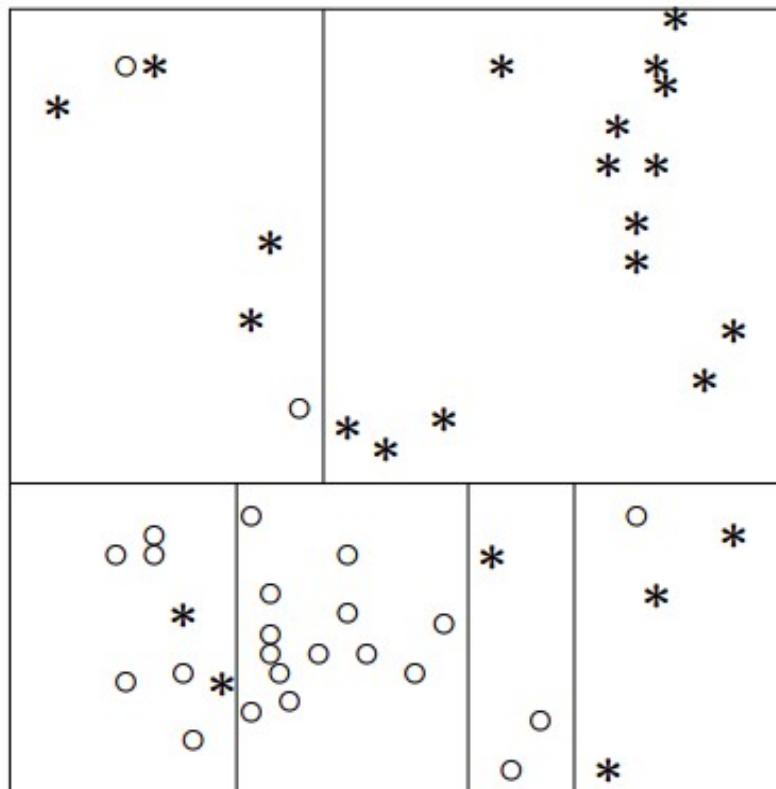
Construire un arbre de decision



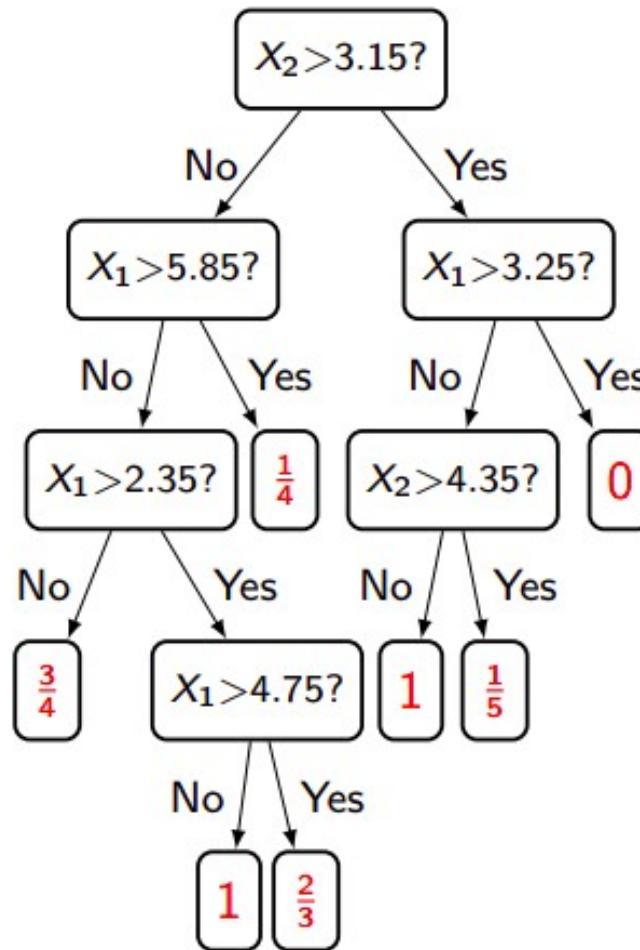
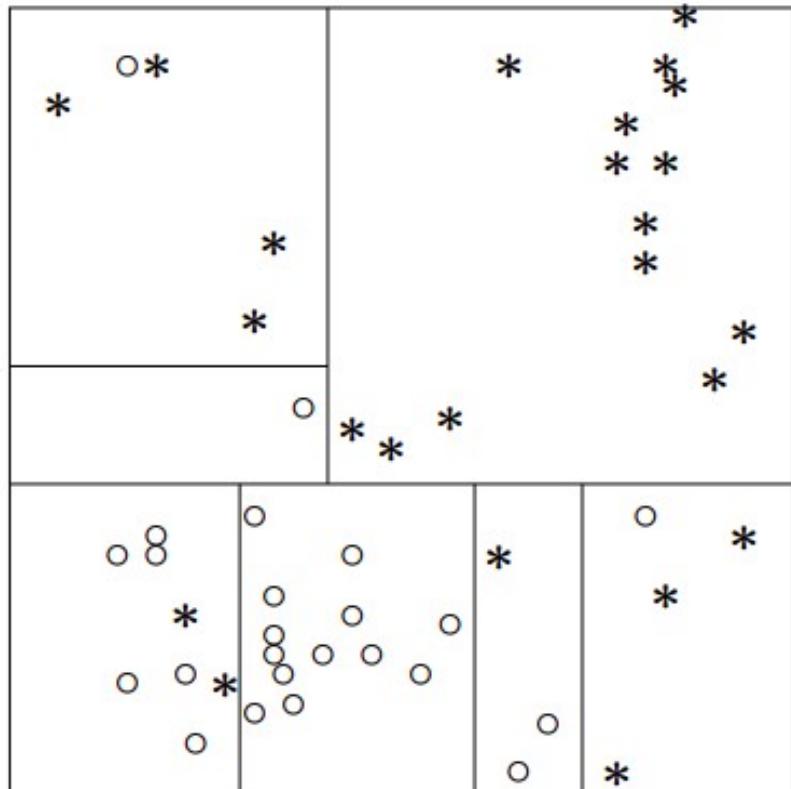
Construire un arbre de decision



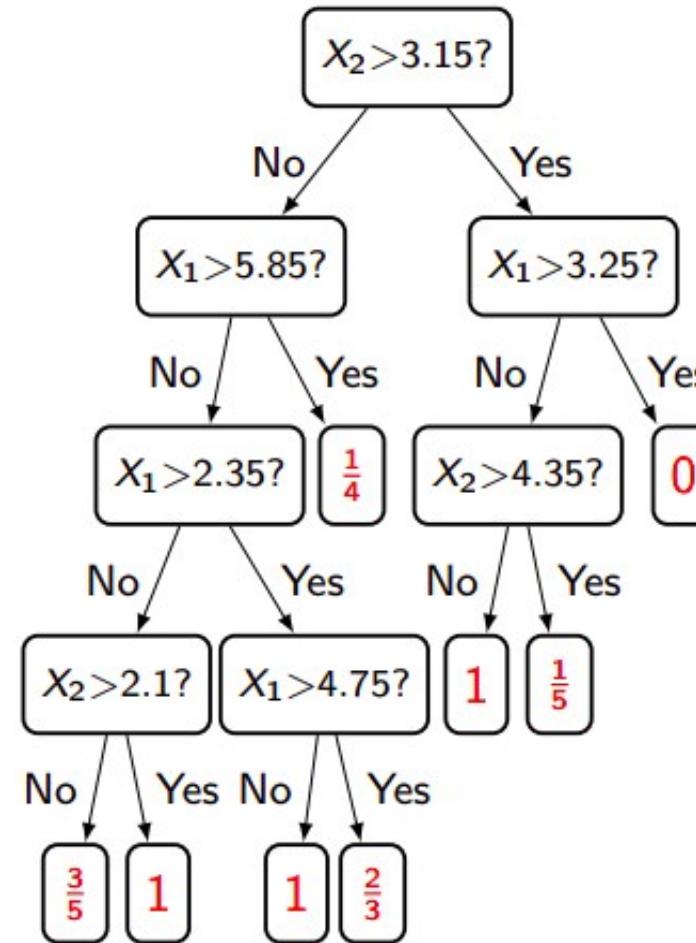
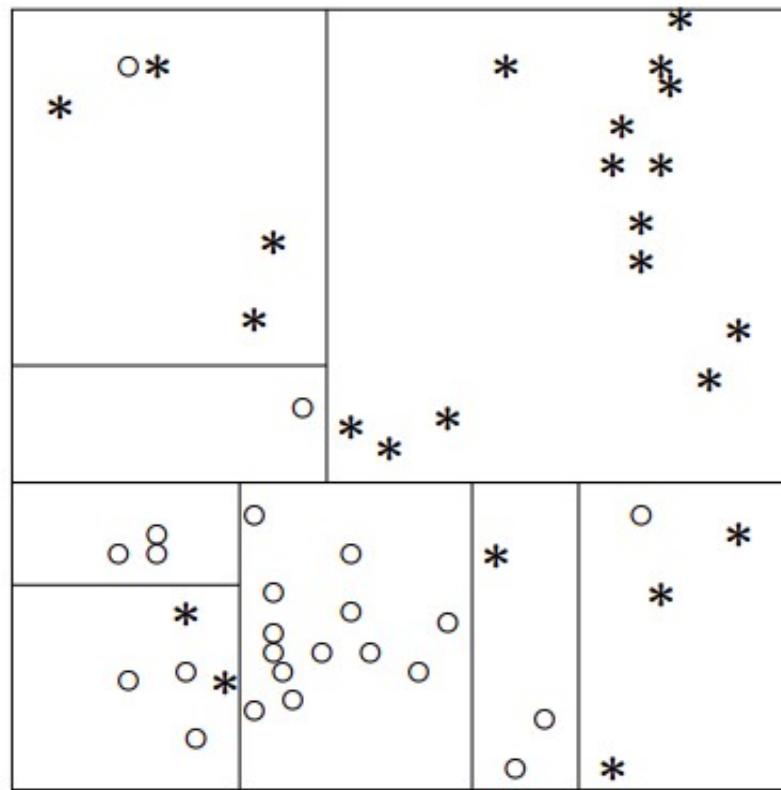
Construire un arbre de decision



Construire un arbre de decision



Construire un arbre de decision



3. Arbre de decisions

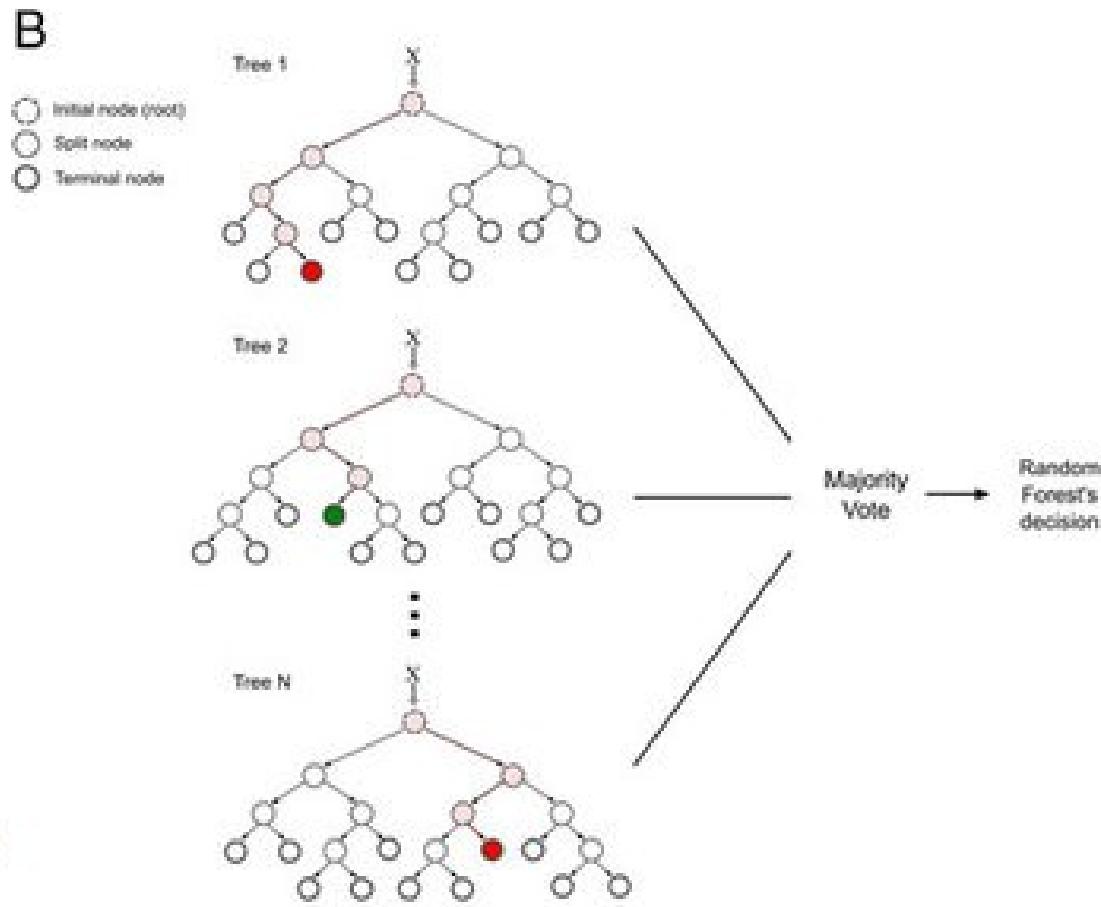
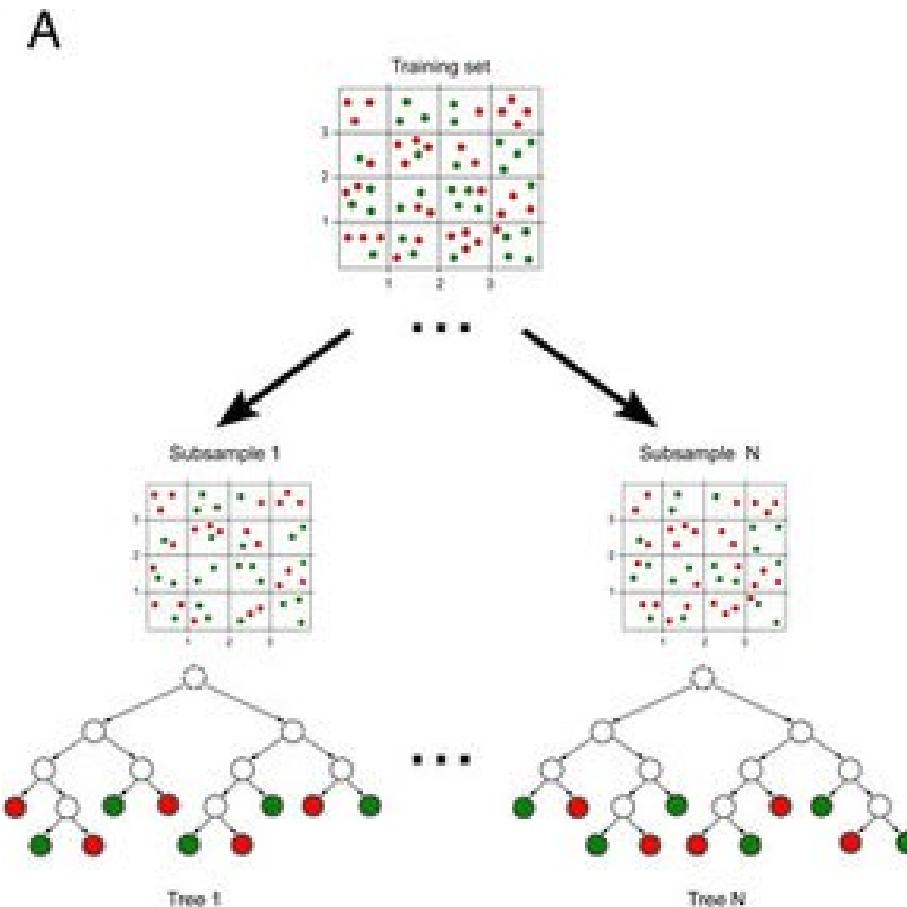
Avantages

- Gere facilement les variables non utiles.
- **Missing data:** Peut classifier malgres les donnees manquantes
- **Interpretable:** La classification est tres facile a interprete
- **Algorithme compacte:** nombre de noeuds << Dimensions
- **Rapide a tester**

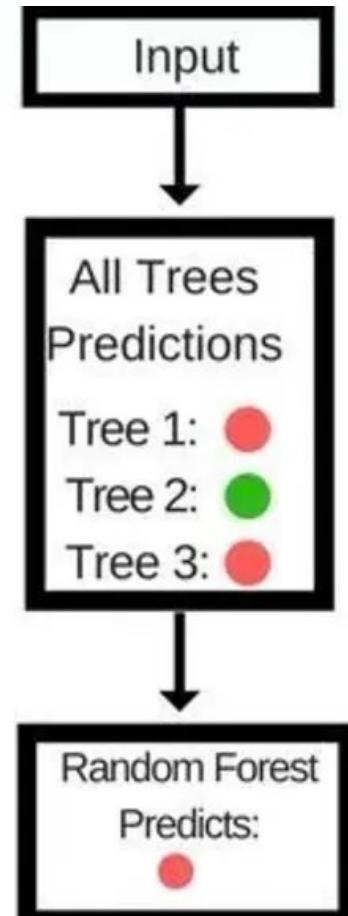
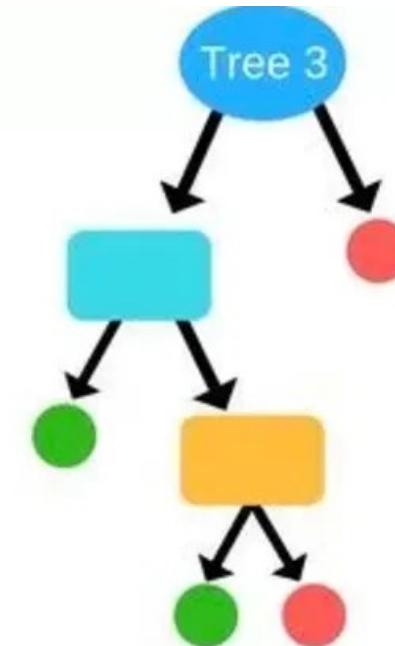
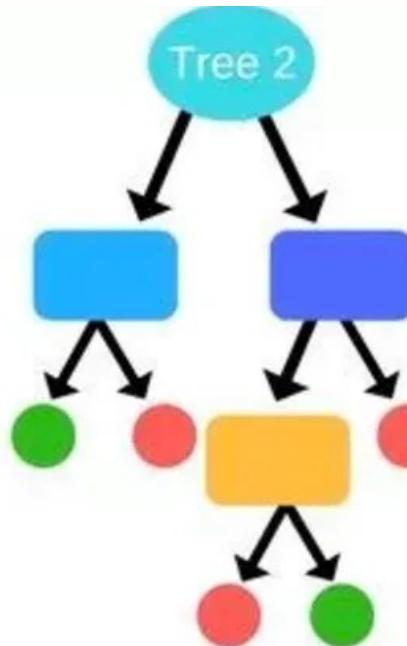
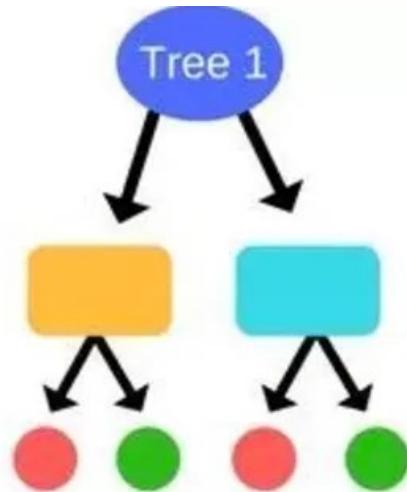
Inconvénients

- **Overfit** tres souvent
- Permet de diviser l'espace uniquement selon des plans alignes avec les axes
- **Greedy:** Ne trouve pas necessairement le meilleur arbre, pas d'etape d'optimisation
- **Haute variabilite et peu robuste**

4. Random Forest



4. Random Forest



Introduction To Random Forest Algorithm

4. Random Forest

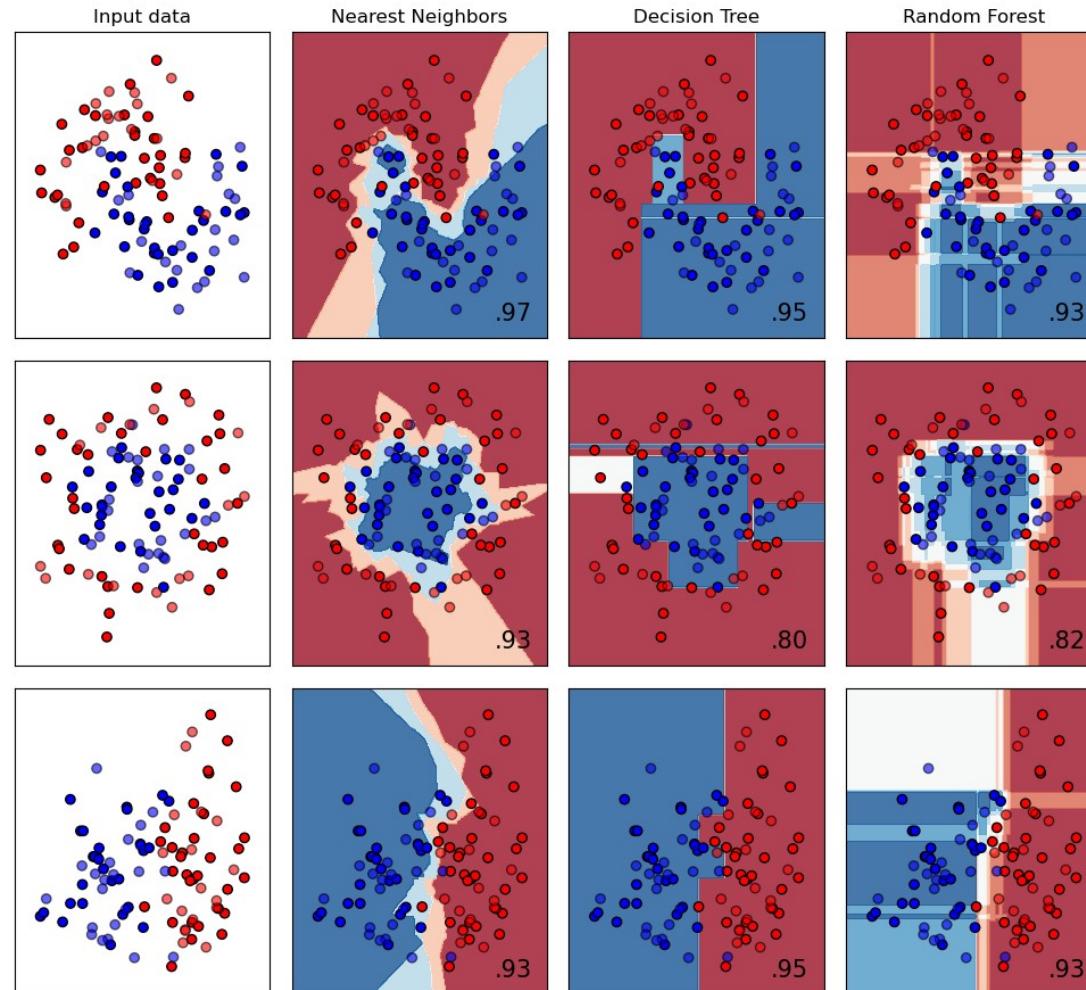
Avantages

- Robuste aux donnees isolees
- Marche bien avec les donnees non lineaire
- Peu de risque d'overfitting
- Efficace avec les gros datasets
- Interpretable

Inconvénients

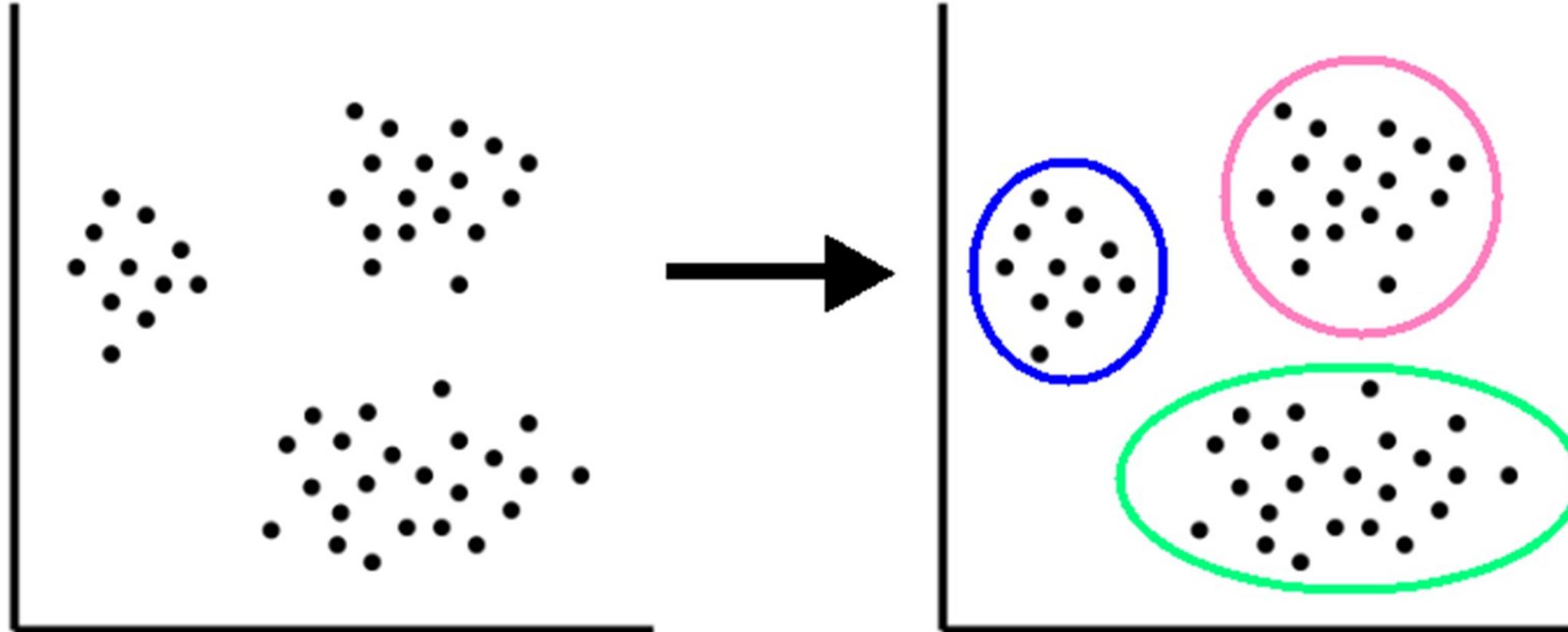
- Mauvais avec les variables categoriques
- Lent a entrainer
- Mauvais avec les donnees clairsemées.

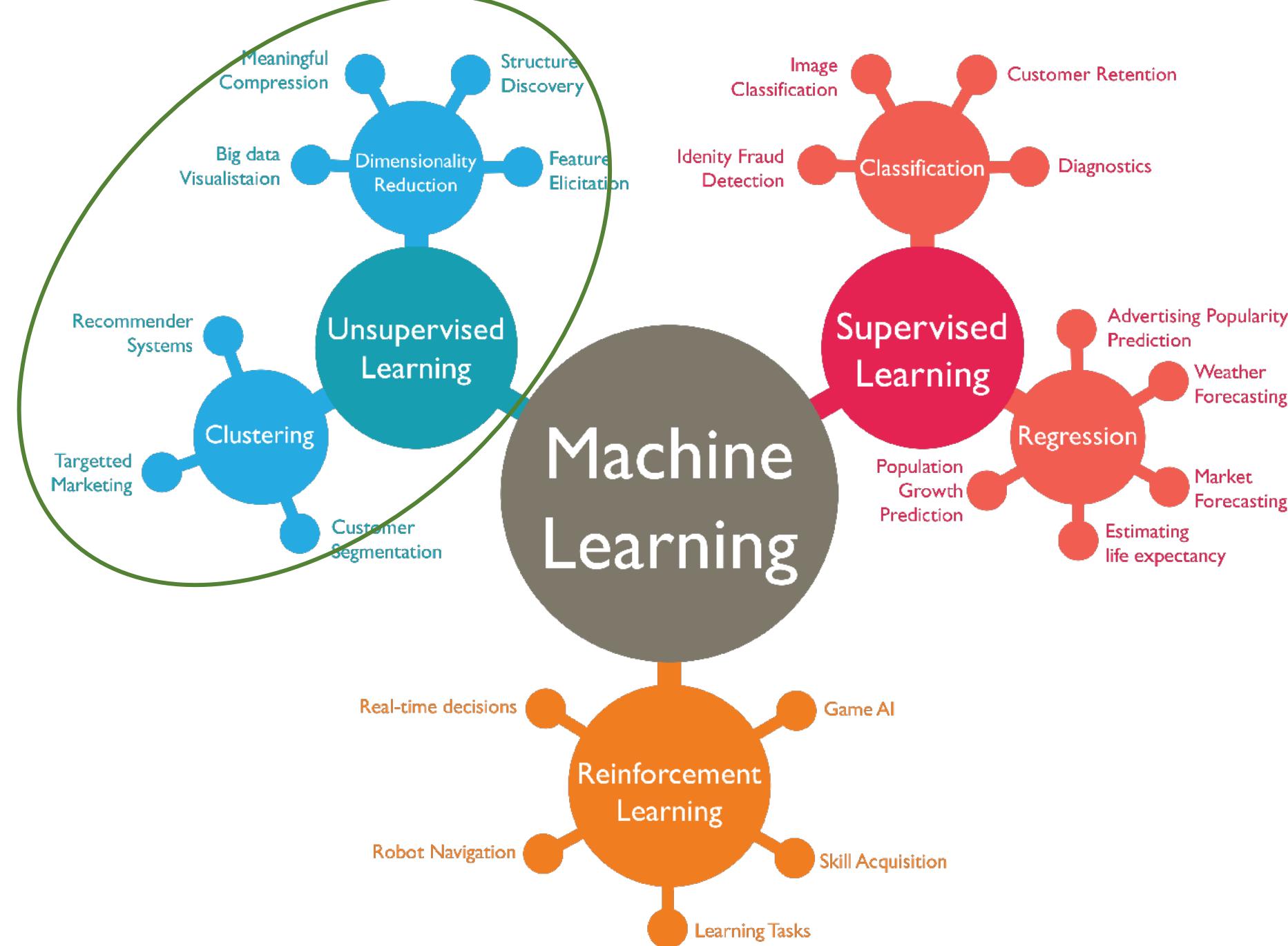
5. Comparaison



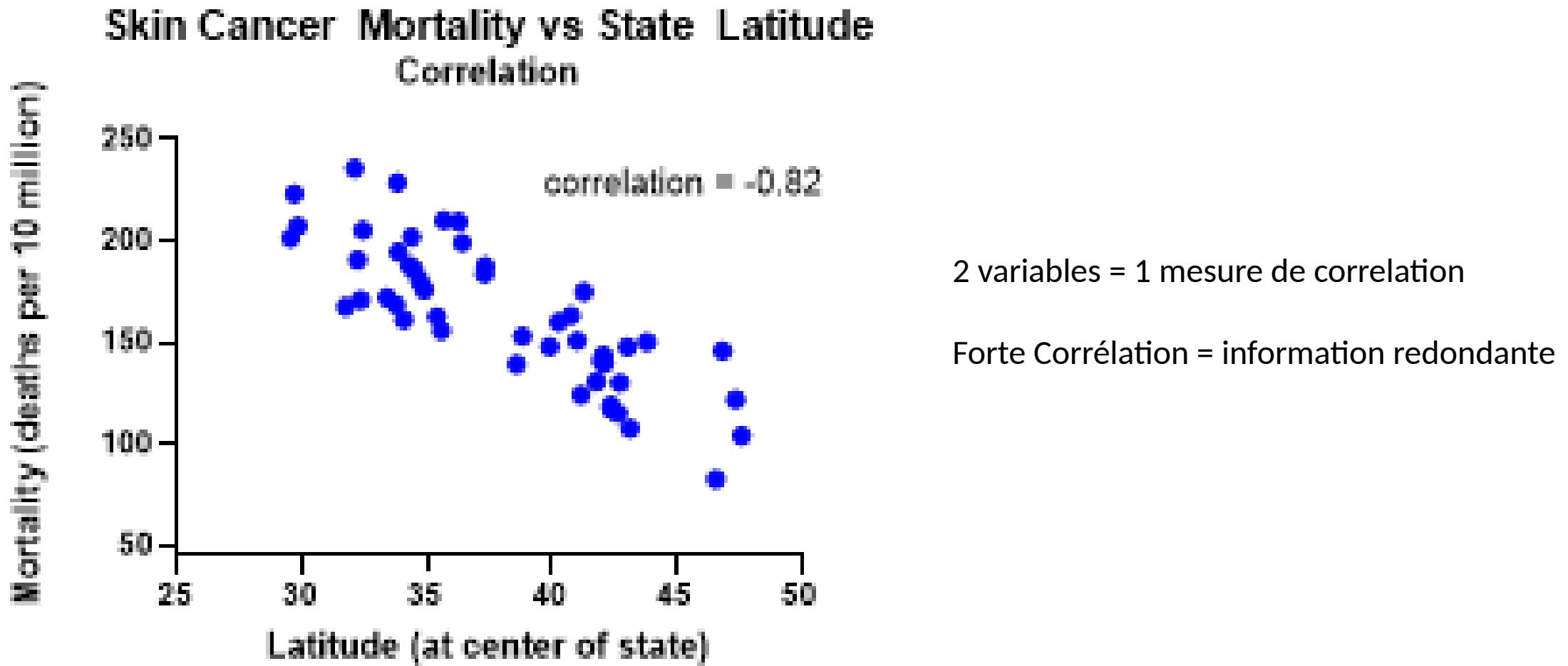
V. Modèles et algorithmes d'apprentissage non-supervisé

difficulté d'obtenir des données labélisées, biais humain et subjectivité de la labellisation

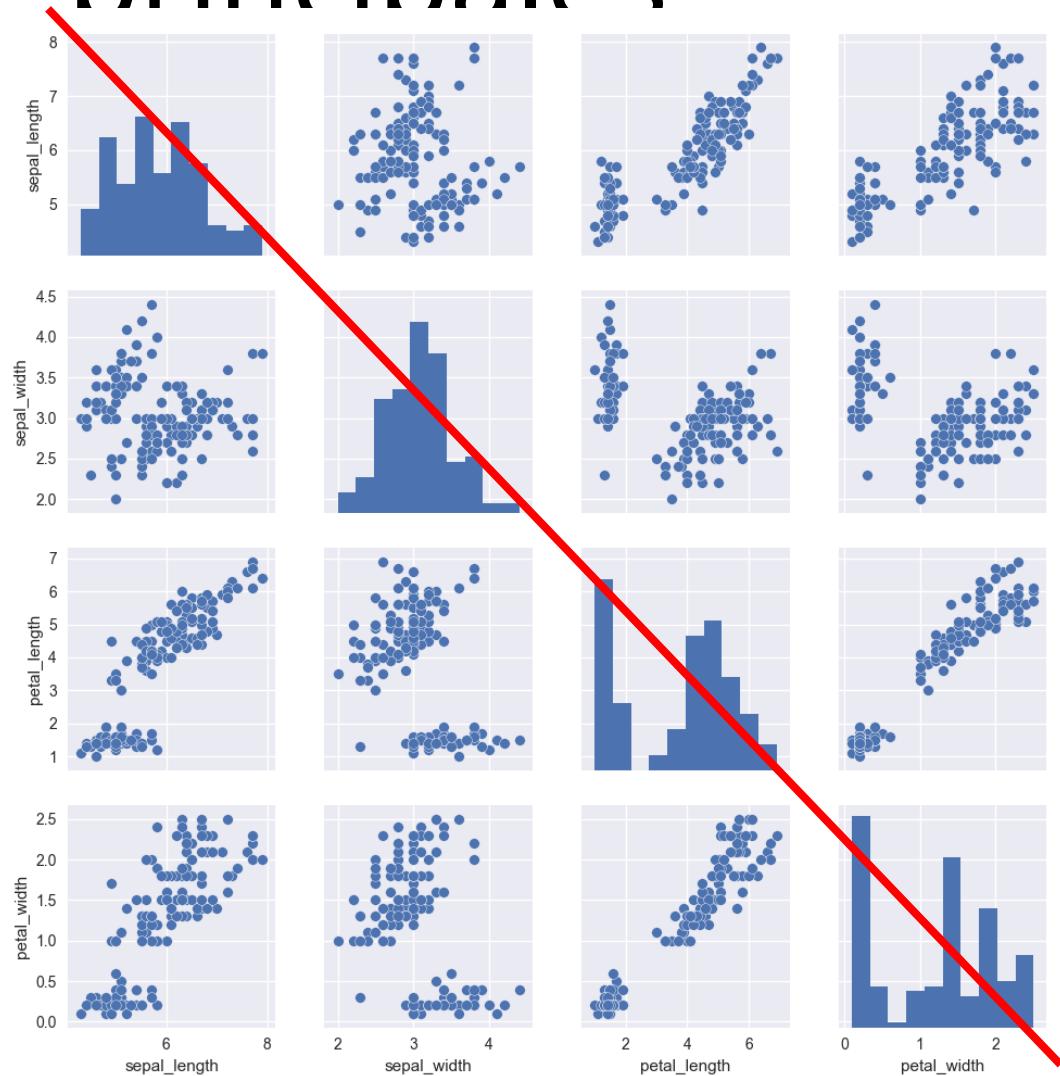




1. Analyse par composantes principales



1. Analyse par composantes principales



4 variables = 6 mesure de correlation

n variables = $n(n-1)/2$ mesure de correlation
100 variables = 4950 mesure de correlation

1. Analyse par composantes principales

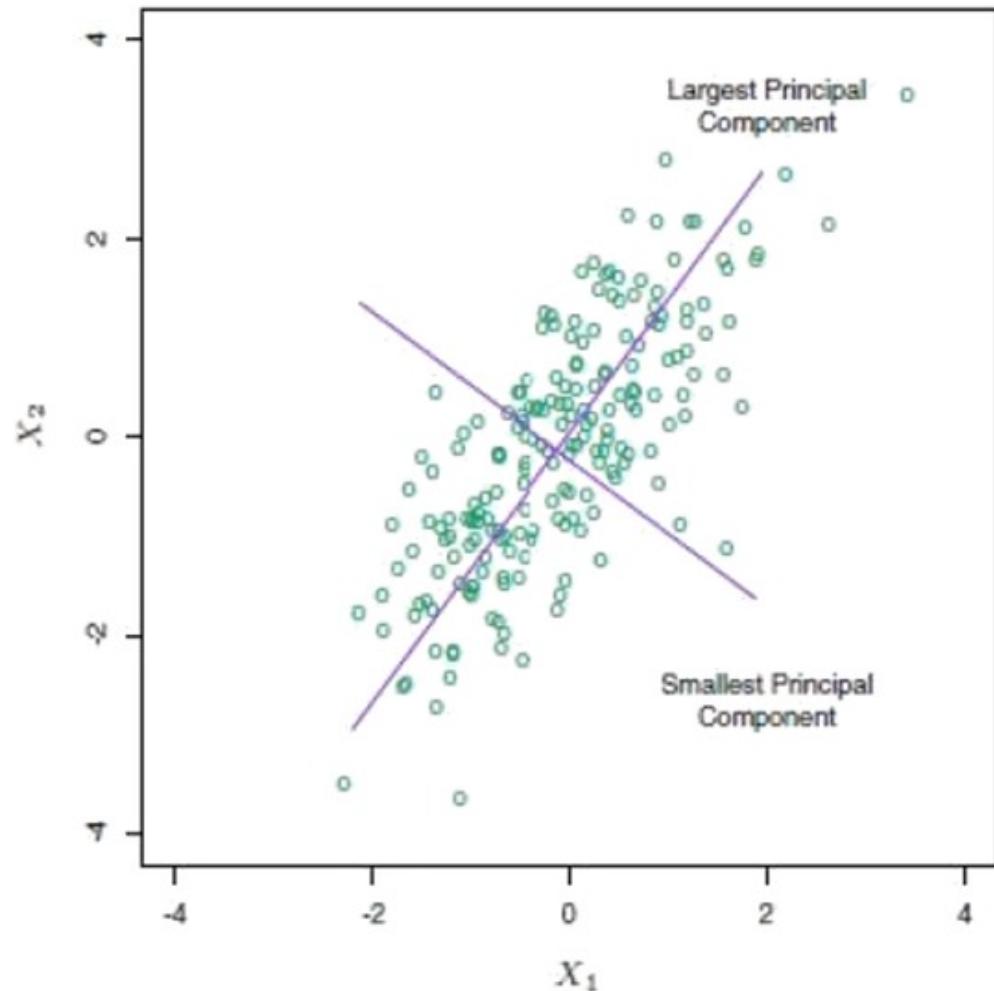
Idées: Transformer des variables corrélées entre elles en nouvelles variables décorrélées les unes des autres.

Les « composantes principales ».

Output: Nouvelles variables décorrélées et ordonnées par ordre d'importance.

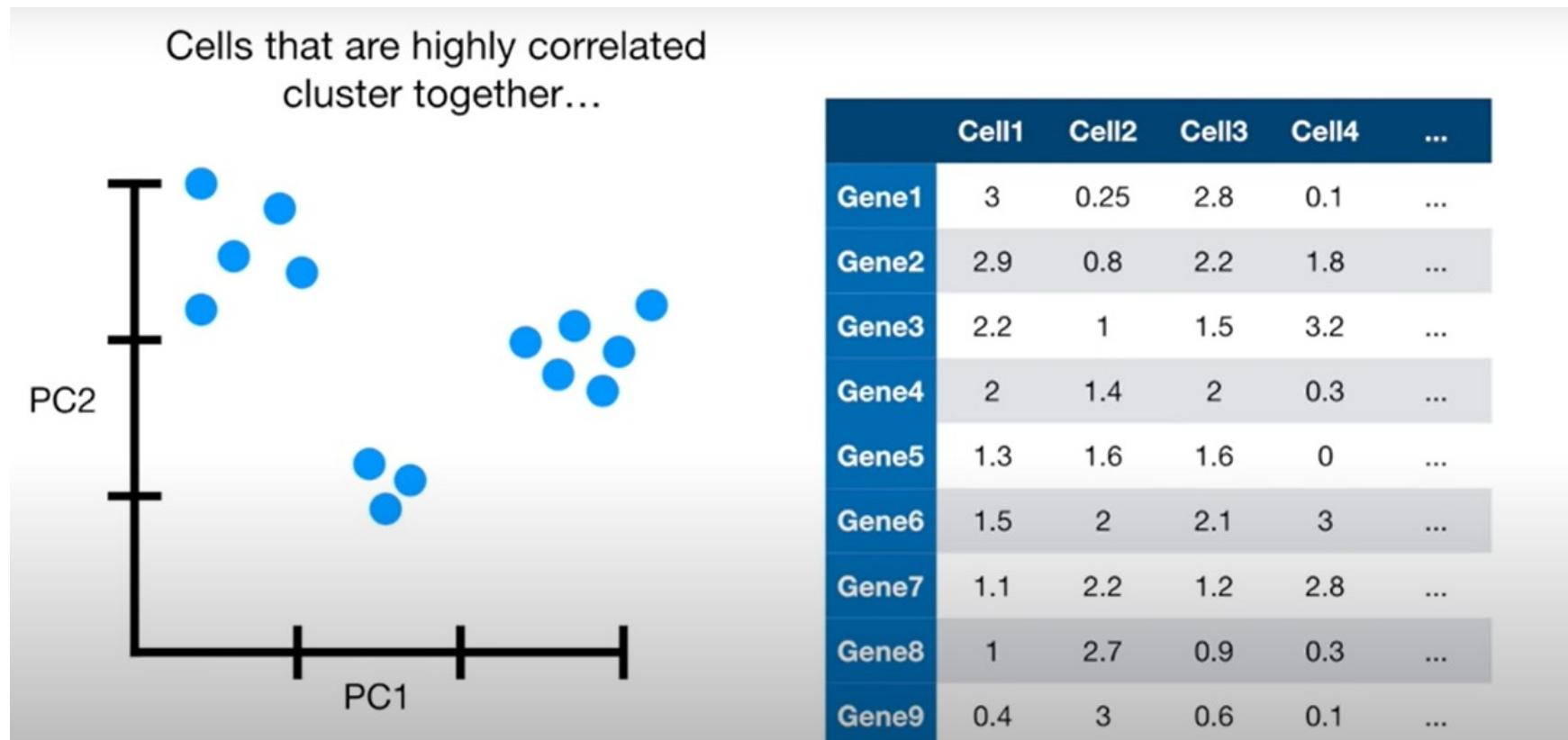
En pratique:

- permet de réduire le nombre de variables (i.e. render l'information moins redondantes)
- Clustering



1. Analyse par composantes principales

- Exemple: StatQuest: PCA main ideas in only 5 minutes!!! (https://www.youtube.com/watch?v=HMOI_IkzW08)



1. Analyse par composantes principales

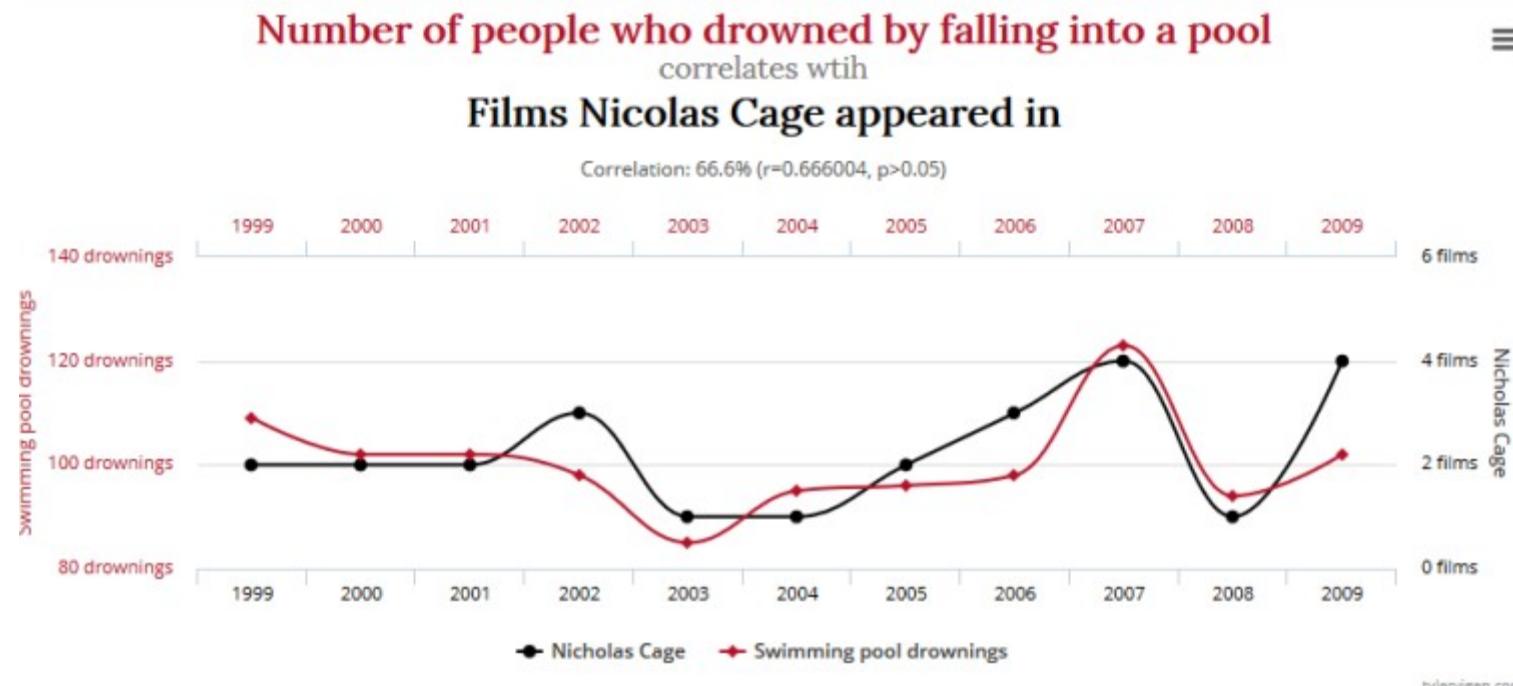
Avantages

- Enlève les variables corrélées
- Permet d'améliorer les performances d'autres types de méthodes sensibles aux données corrélées
- Réduit les possibilités d'overfitting
- Améliore la visualization de données en haute-dimension.

Inconvénients

- Les variables deviennent moins interprétables.
- Attention: une standardization des données est requise avant une PCA
- Pertes d'informations
- Nous avons seulement accès à la corrélation des variables dans un échantillon.

1. Analyse par composantes principales



a sources: Centers for Disease Control & Prevention and Internet Movie Database

tylervigen.com

1. Analyse par composantes principales

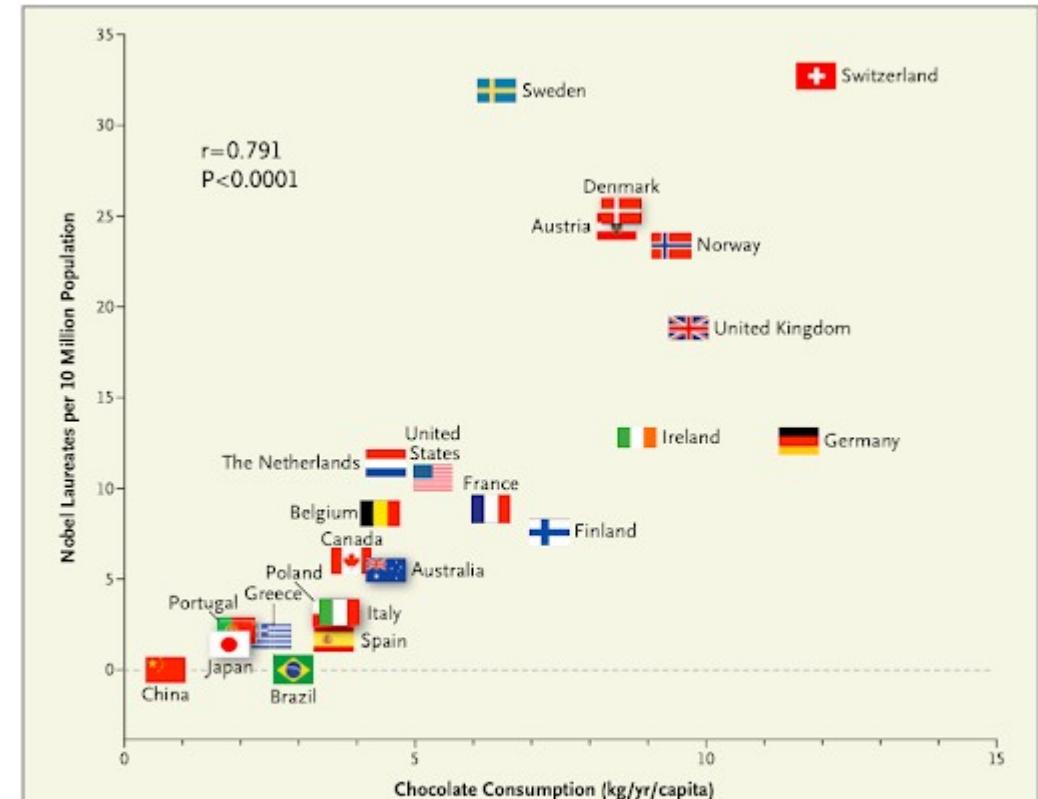
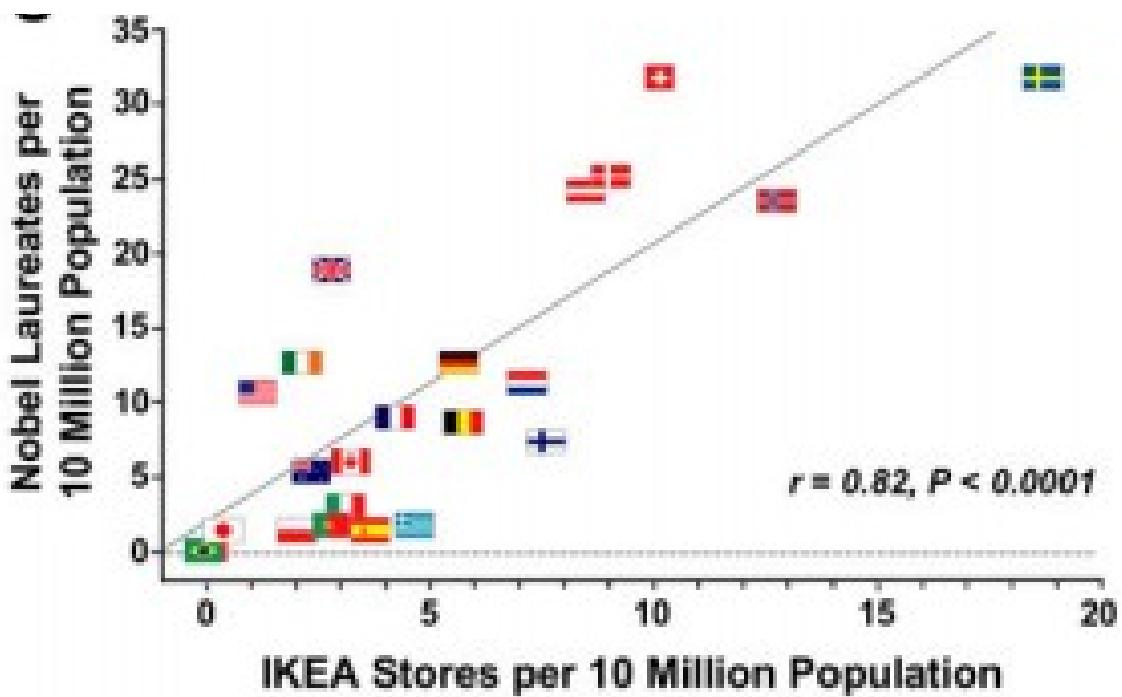


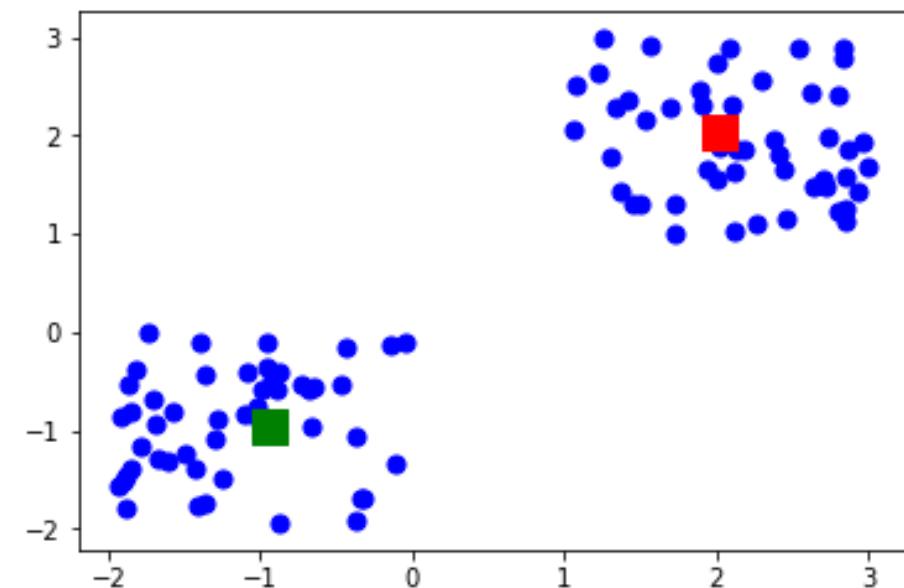
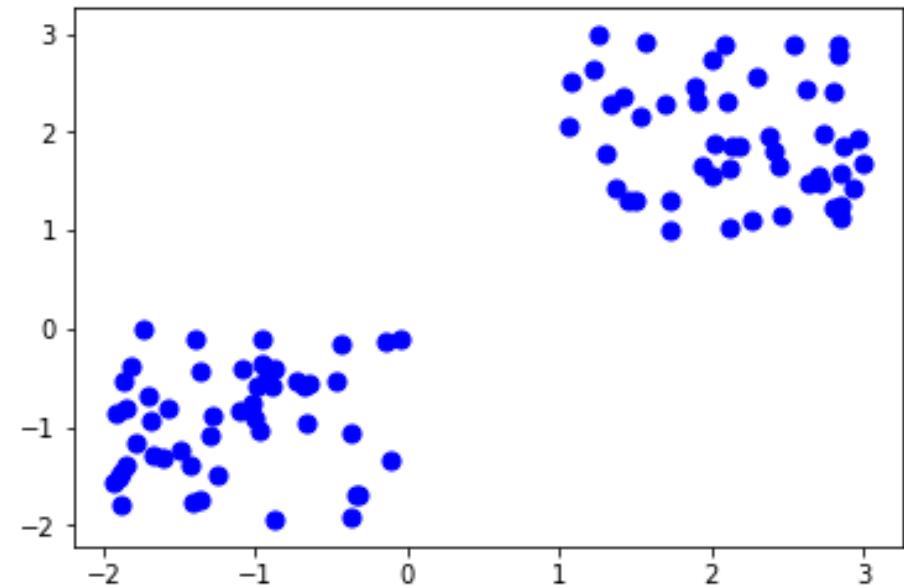
Figure 1. Correlation between Countries' Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.

2. k-means clustering

Idées: On définit un cluster par son centroid (le centre imaginaire du cluster). Etant données k (un hyper-parameter), on cherche les centroids de chaque cluster itérativement.

Un data point est dans le cluster du centroid le plus proche

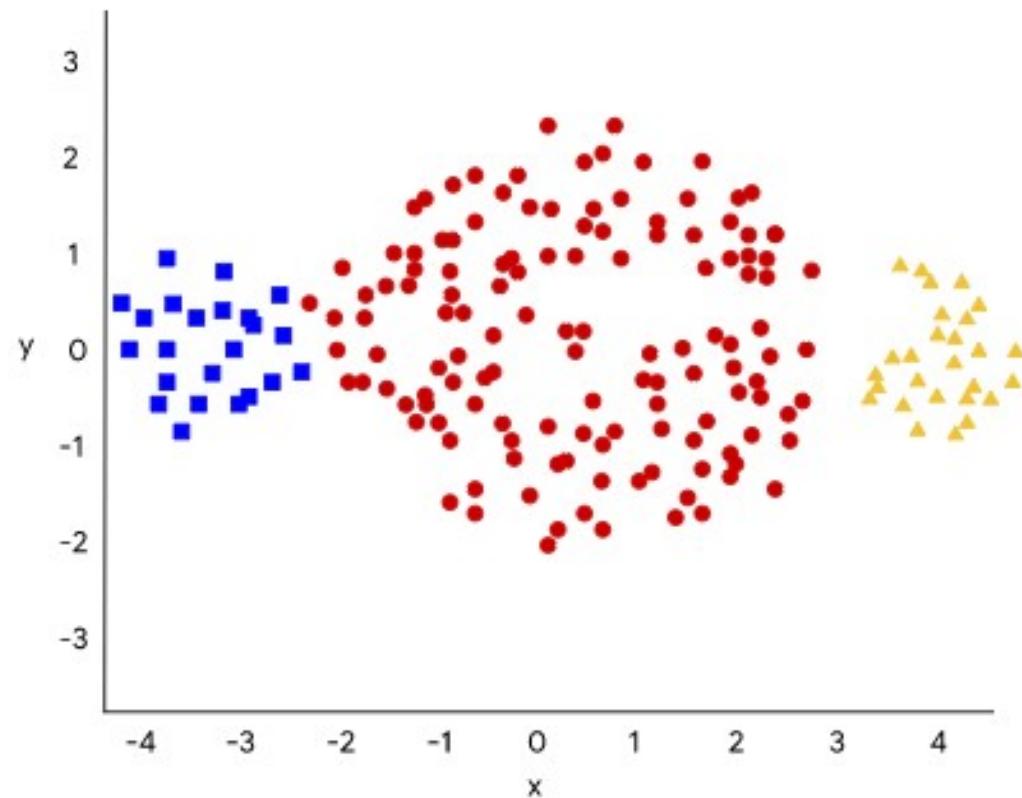
Output: k points (les centroids de chaque cluster)



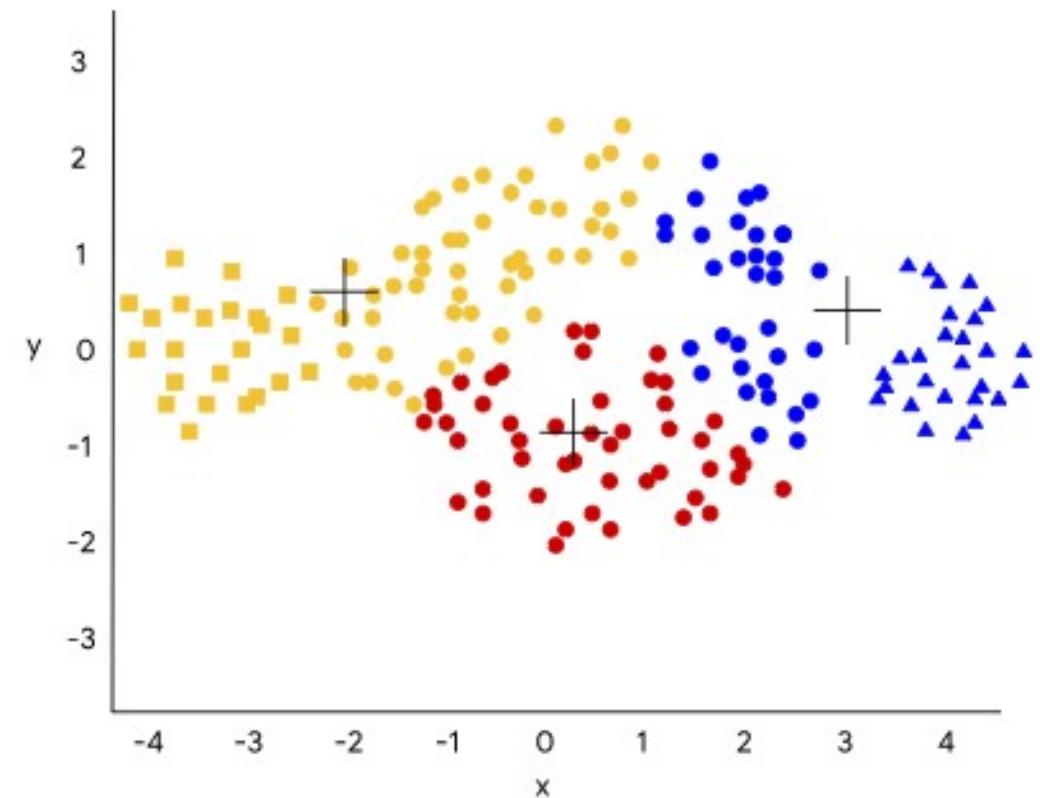
2. k-means clustering

Démo

Clustering intuitif



Clustering k-means



2. k-means clustering

Avantages

- Facile à implémenter
- Adapter pour traiter des gros datasets
- Converge toujours bien
- k-means++ pour trouver les centroids initiaux

Inconvénients

- Le choix de k n'est pas simple
- Résultats dépendent de l'initialisation
- Inéfficace sur les datasets déséquilibré

VI. Vers le Deep Learning

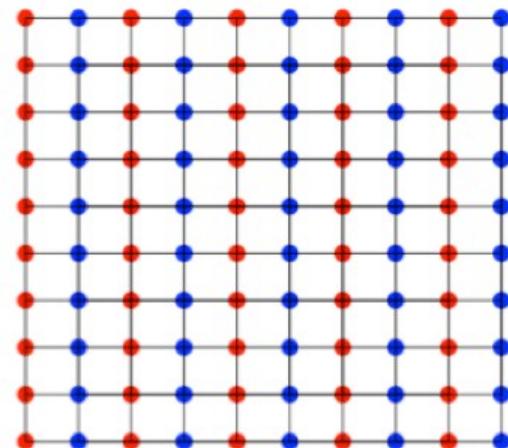
- The curse of dimensionality
- Perceptron
- Multi-layer Perceptron
- Convolutional Neural Network

1. The curse of dimensionality

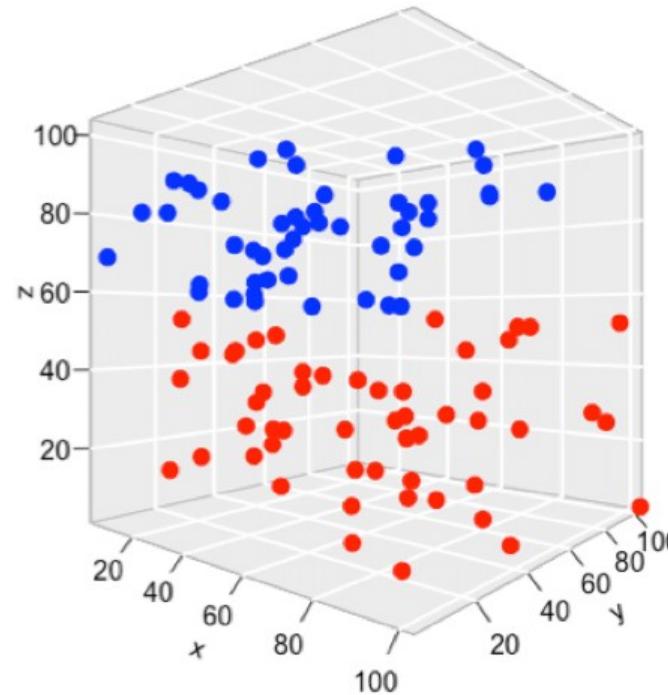
Le machine learning est basée sur des statistiques. Ainsi pour être robuste, nous avons besoin d'explorer un maximum de configuration possible.



(A) 1-D



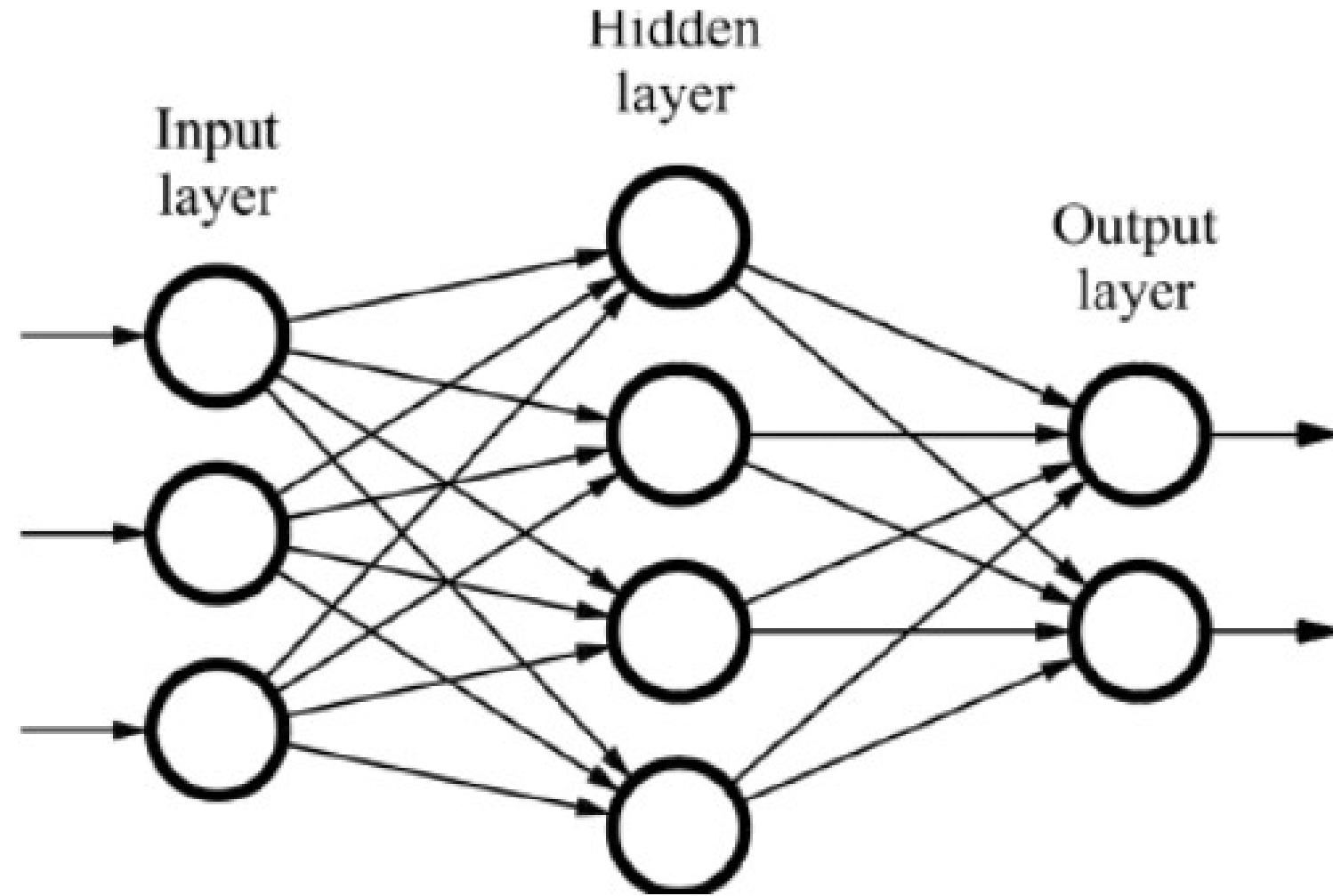
(B) 2-D



(C) 3-D

- 1D: 10 points pour couvrir l'espace
- 2D: 100 points pour couvrir l'espace
- 3D: 1000 points pour couvrir l'espace
- n-D: Combien de points pour couvrir l'espace ?

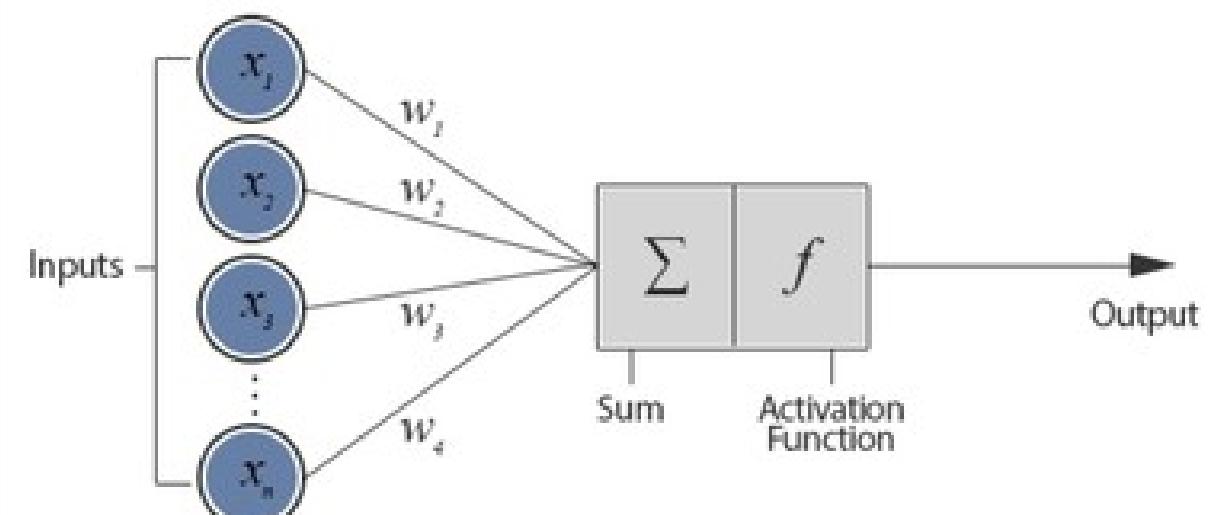
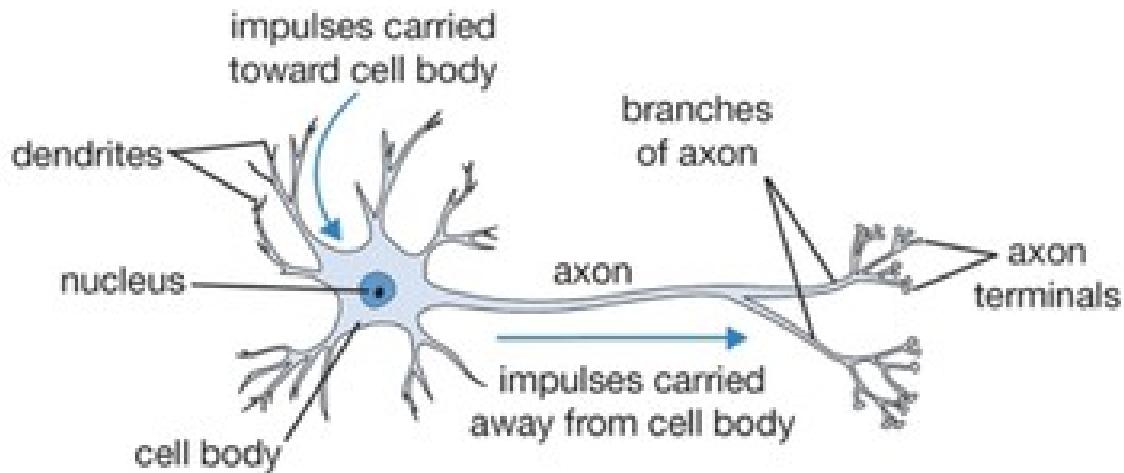
2. Réseaux de neurones



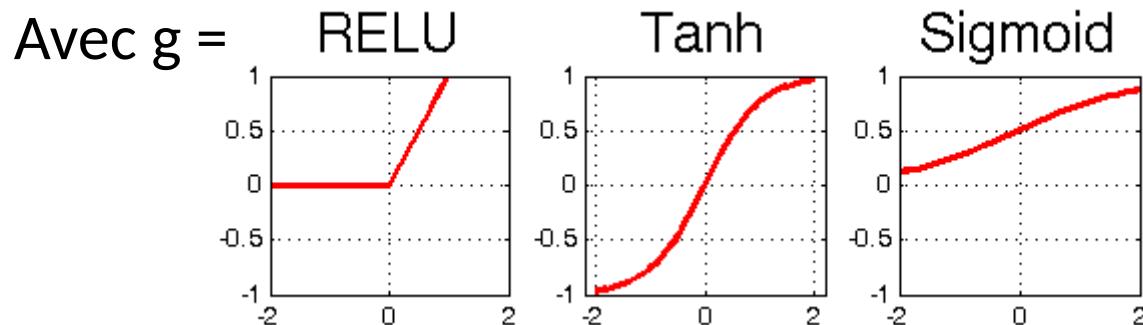
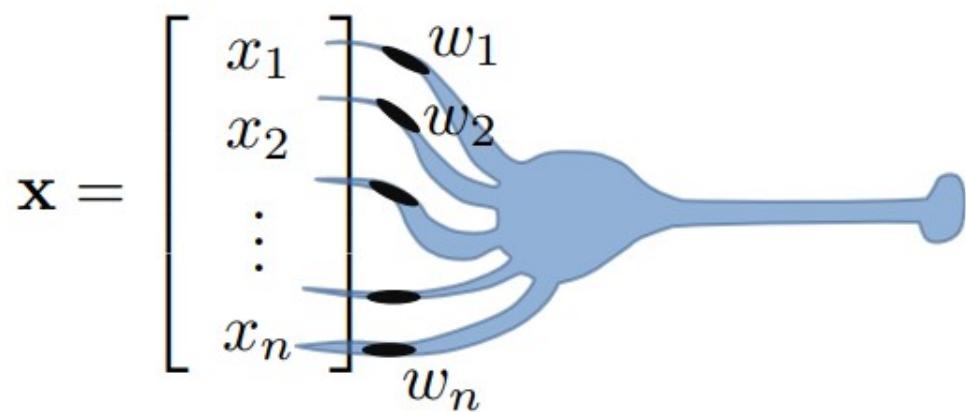
2.a. Perceptron

Rosenblatt, 1957

Biological Neuron versus Artificial Neural Network



2.a.Perceptron



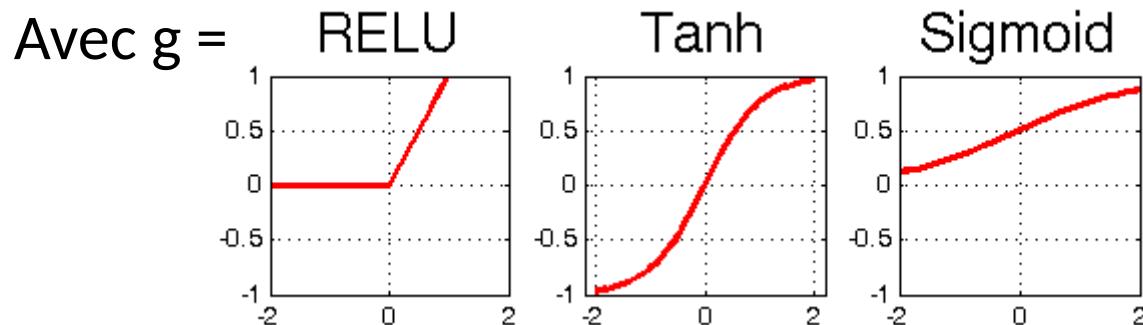
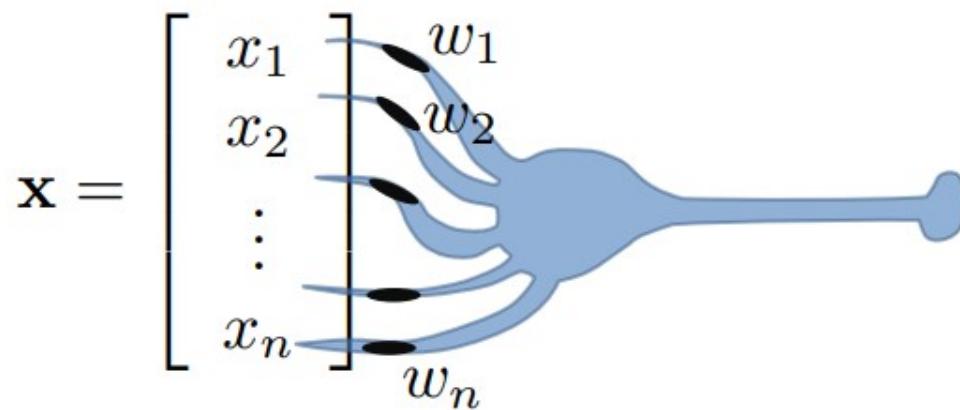
Comme une regression linéaire

$$\text{output} = g(w^T x + b)$$

Fonction d'activation

Nombre de paramètres ?

2.a. Perceptron

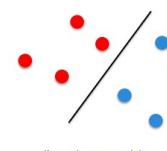


Comme une regression linéaire

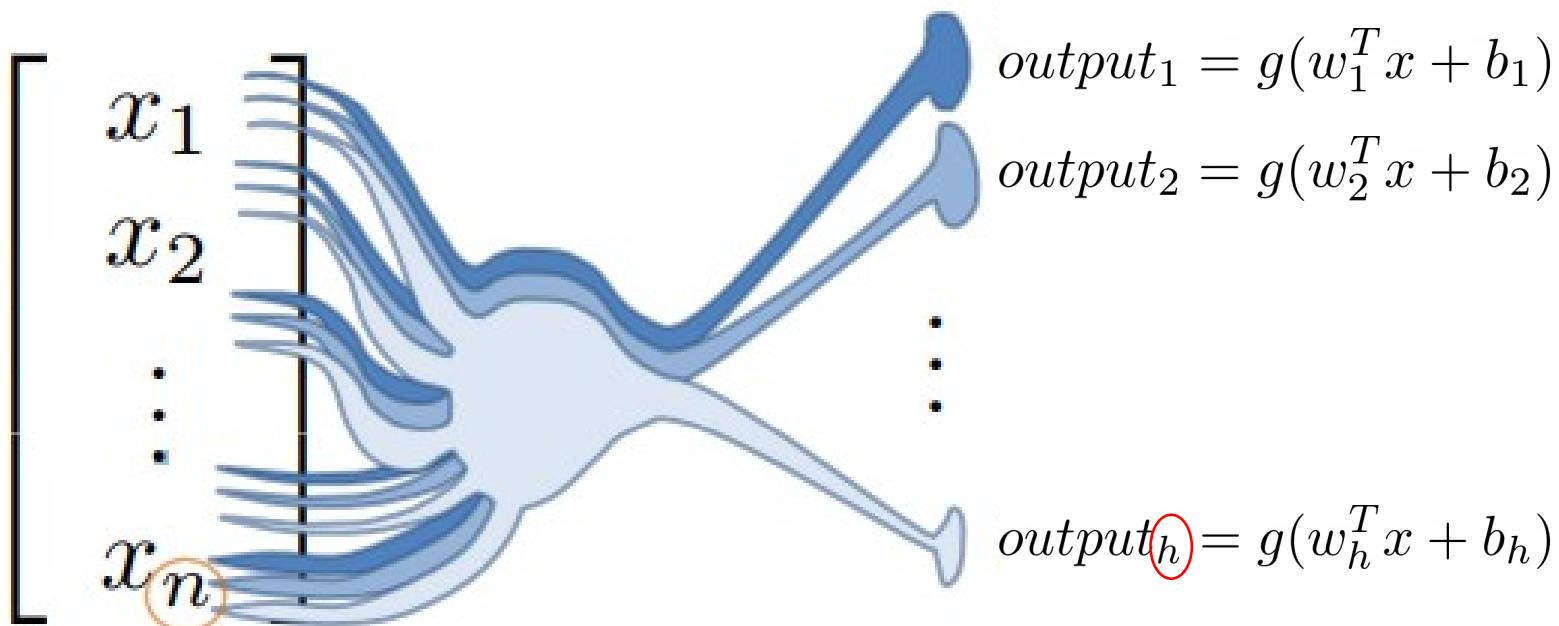
$$\text{output} = g(w^T x + b)$$

Fonction d'activation

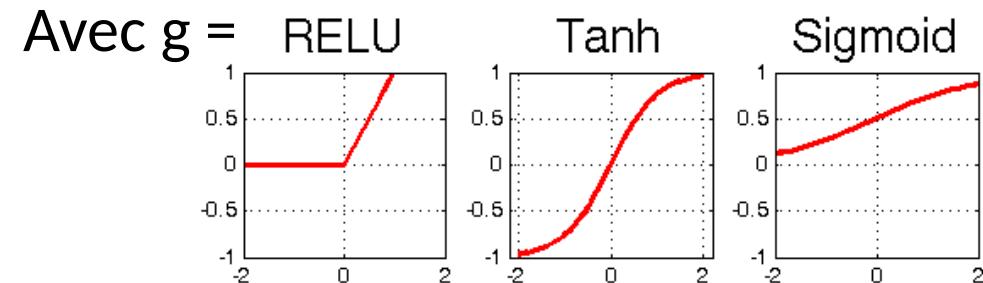
Pour l'instant, cela ressemble énormément à la regression linéaire. Ce type de modèle ne permet pas de gérer des problèmes non linéaires:



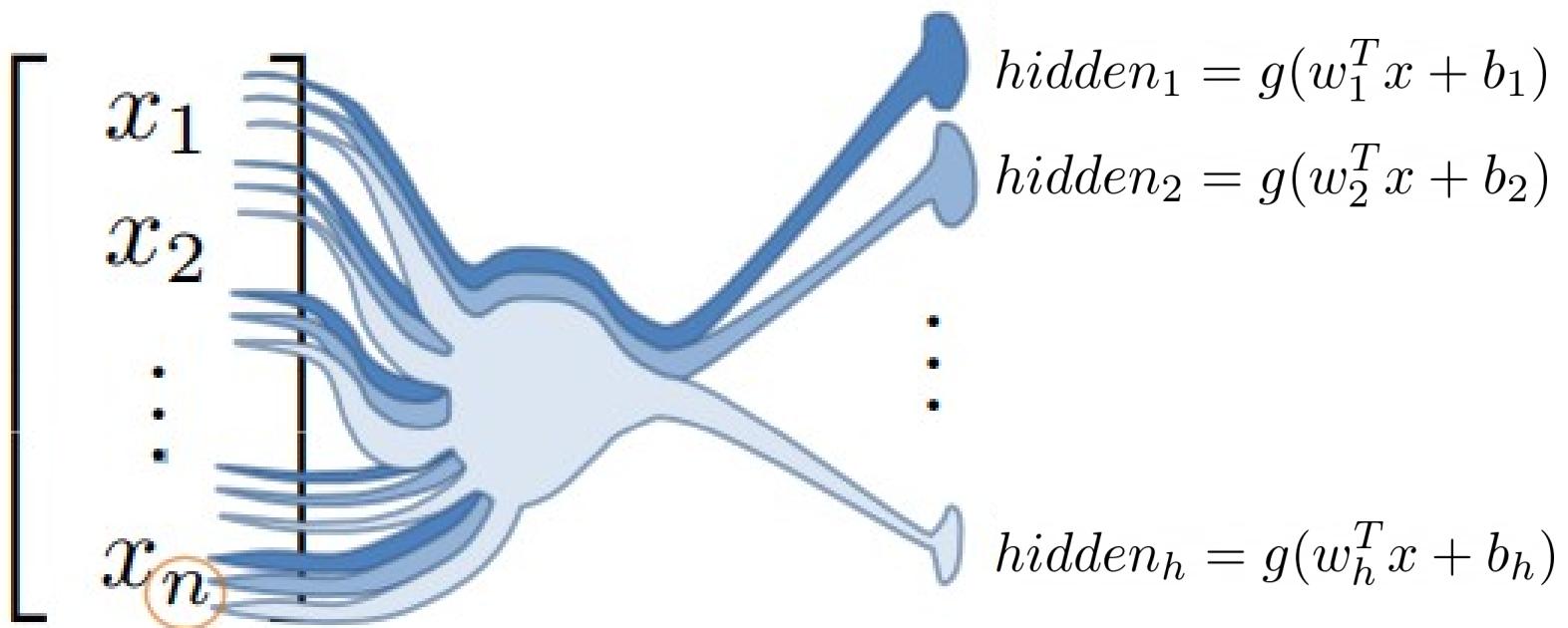
2.b. Multi-layer Perceptron



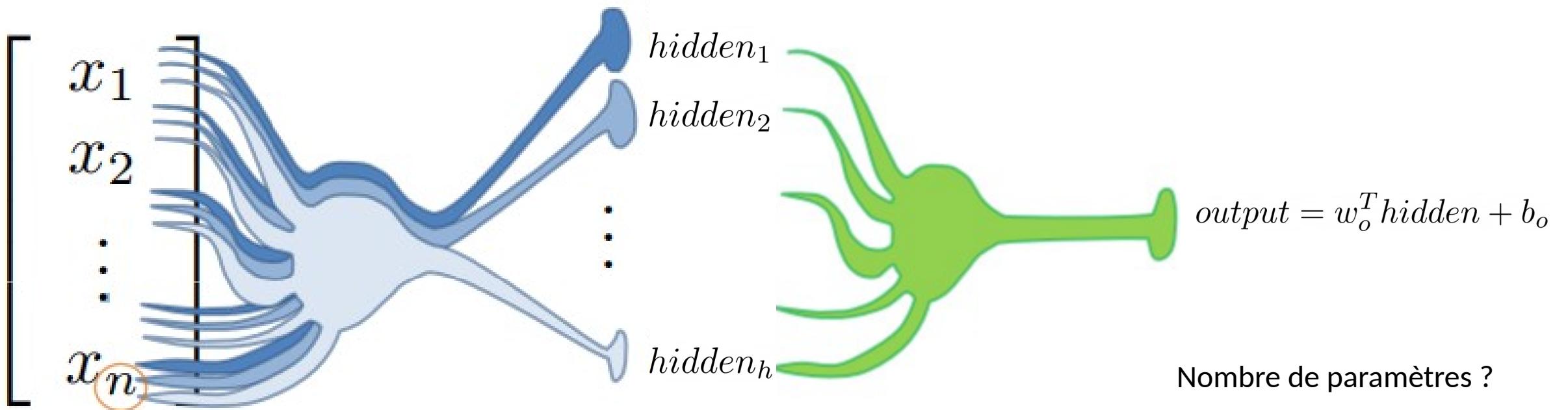
Nombre de paramètres ?



2.b. Multi-layer Perceptron

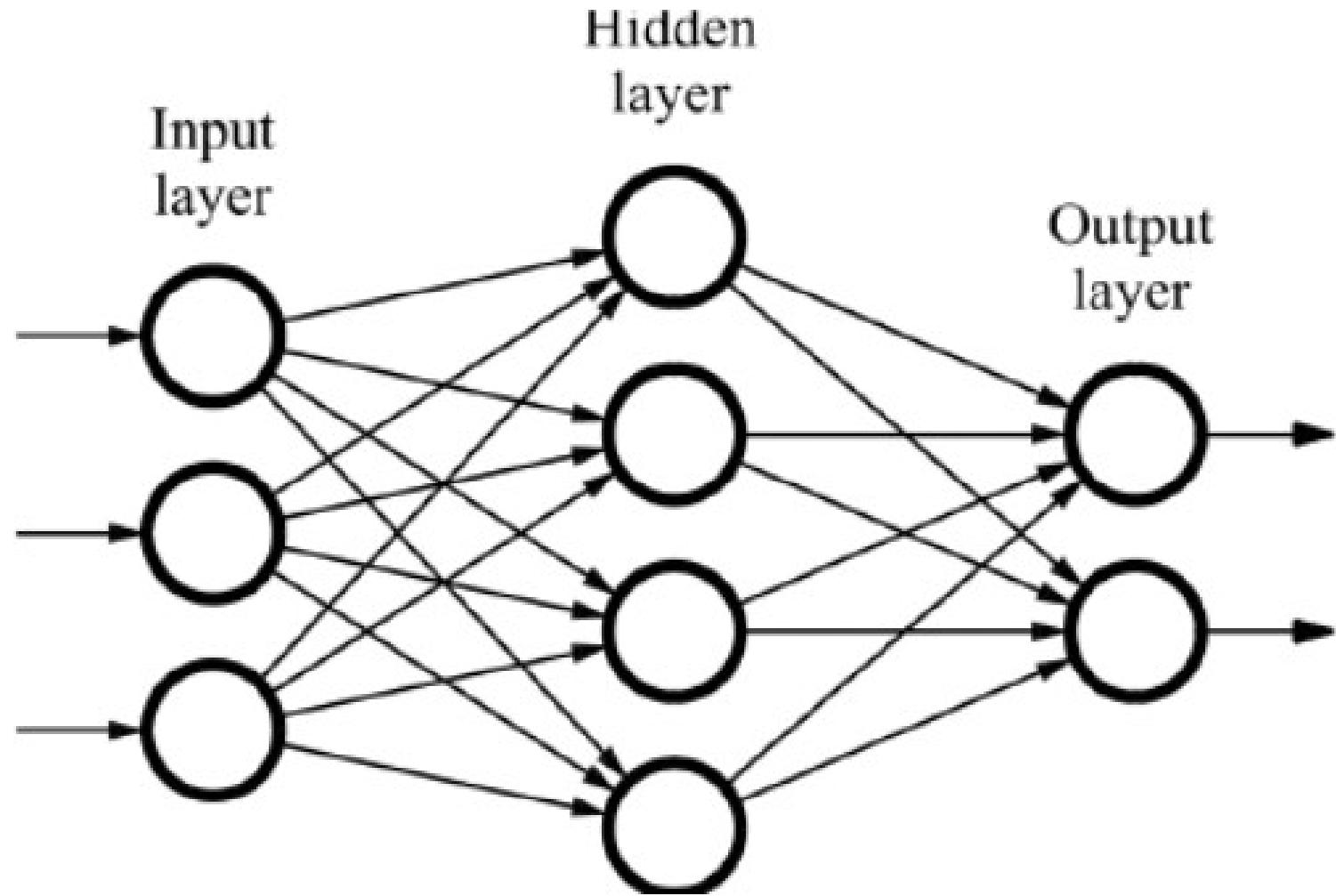


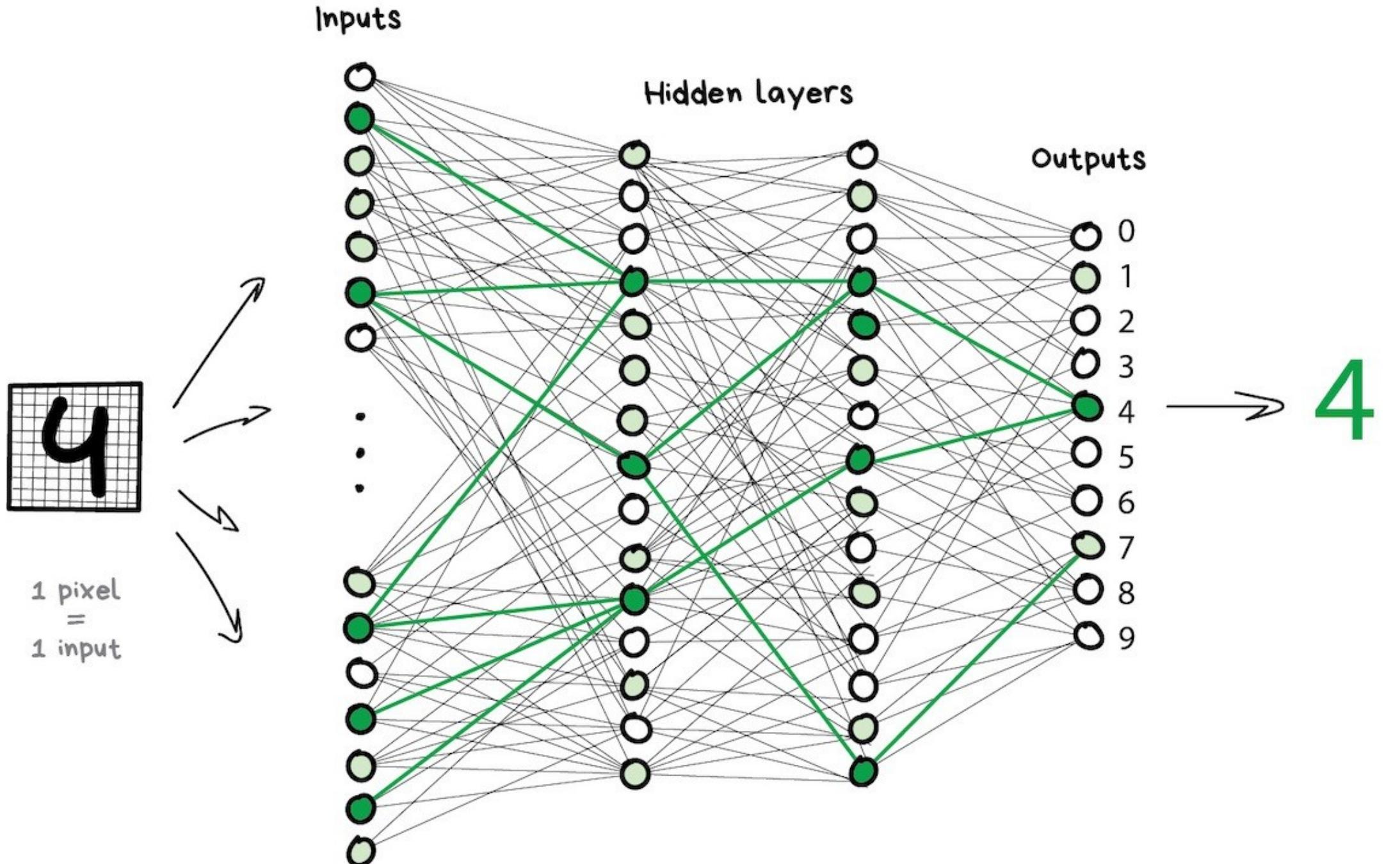
2.b. Multi-layer Perceptron



Un multi-layer perceptron peut résoudre des problèmes non linéaires

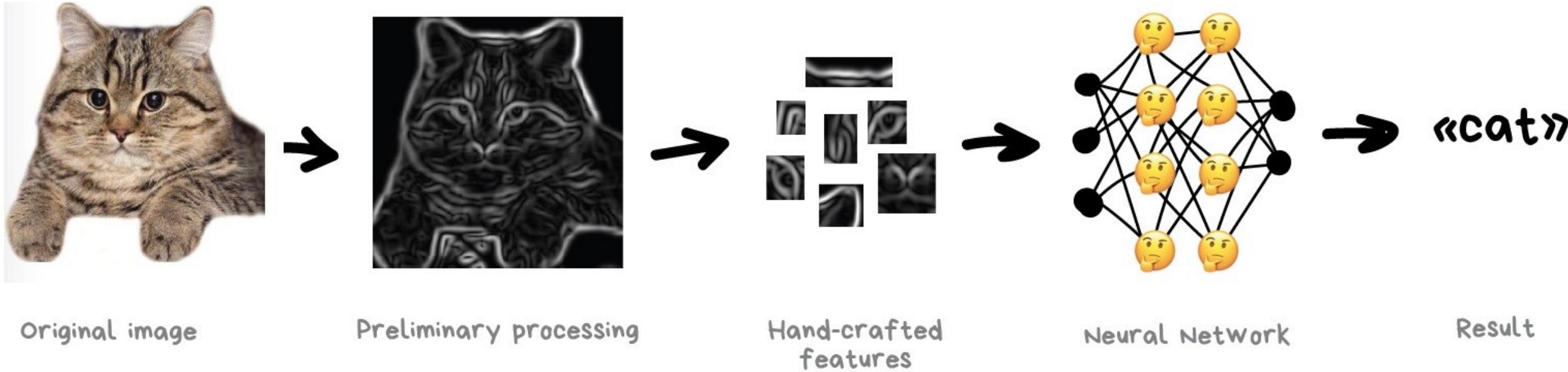
Neural network





MULTILAYER PERCEPTRON (MLP)

5. Convolutional Neural Network



Workflow de Machine Learning traditionnel: on crée les variables de nos données à la main

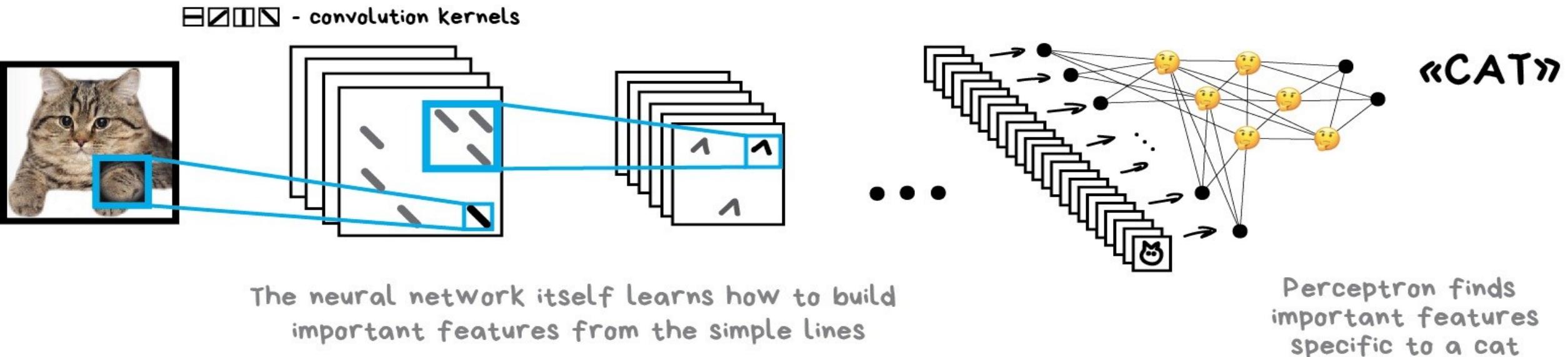
Exemple: on va indiquer où sont les oreilles et la queue.

Problèmes: Que se passe-t-il si les oreilles du chat ne sont pas sur la photo ?

L'homme ne reconnaît pas un chat d'un chien par la position ni la forme des oreilles.

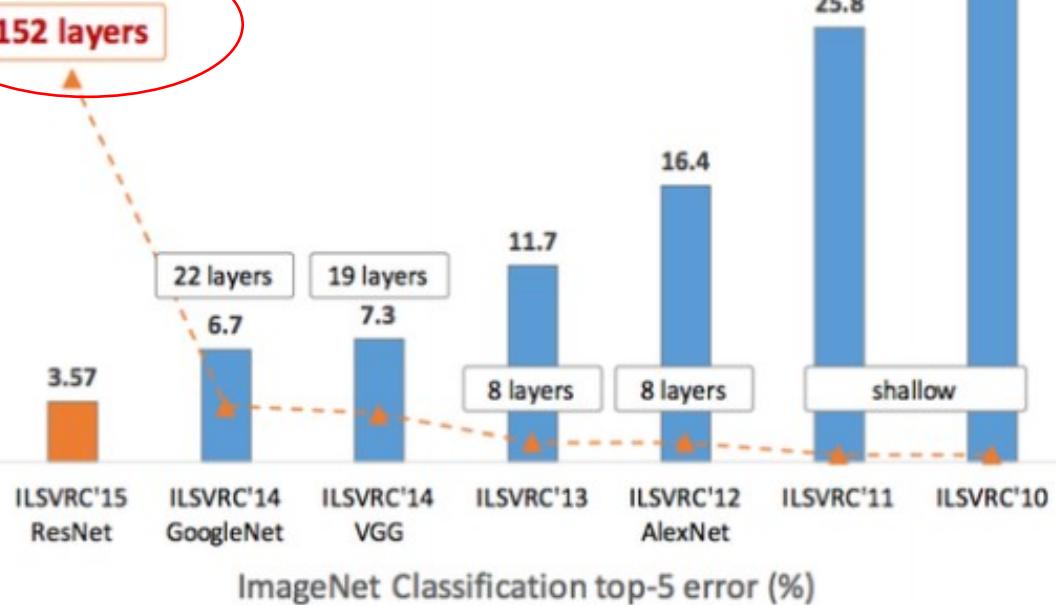
5. Convolutional Neural Network

On veut que la “machine” crée elle même ses variables



ResNet [He et al, CVPR 2016]

Revolution of Depth



Year	Model name	Accuracy (%)	Size (MB)	MACs (G)	#Params (M)	Time (ms)
2012	AlexNet	56.48	237.89	0.72	61.10	0.46
2014	VGG19	72.38	461.10	0.40	20.08	1.12
2014	Inception_v3	79.35	5212.60	1.53	6.25	5.80
2016	ResNet18	76.82	721.60	0.56	11.22	1.28
2016	ResNet50	78.07	4233.60	1.30	23.70	4.56
2016	ResNet101	77.35	6409.60	2.52	42.70	7.38
2016	ResNet152	78.28	9097.60	3.74	58.34	10.54
2018	DenseNet121	78.46	5025.60	0.90	7.05	5.23
2018	DenseNet169	75.56	6111.60	1.07	12.64	6.75
2018	DenseNet201	76.87	7947.60	1.38	18.28	9.24

Take home messages

Gardez en tête le nom des différents algorithme de Machine learning, leurs avantages et inconvénients principaux.

La descente de gradient

Réseaux de neurons

Réseaux de neurones convolutionnels

Conseil de lecture

Pour un blog très bien fait sur les bases du Machine et du deep learning:
vas3k.com

Livre disponible en ligne qui a grandement inspiré mon cours: *Deep Learning*, Goodfellow

Artificial Intelligence—The Revolution Hasn't Happened Yet, Michael I. Jordan

Coursera, [Ai for medecine](#), Andrew Ng