
Introduction à la science des données

Introduction

- Dans ce cours, nous commencerons par retracer brièvement **l'histoire de l'IA**
- Ensuite nous nous intéresserons aux différentes branches et la terminologie
- Enfin nous aborderons les **concepts clés** du machine learning par le prisme de **la régression linéaire simple puis multiple**

Sommaire

- L’histoire de l’IA
- Les termes relatifs à l’IA
- Les différents types d’apprentissage
- La régression linéaire simple
- La régression linéaire multiple

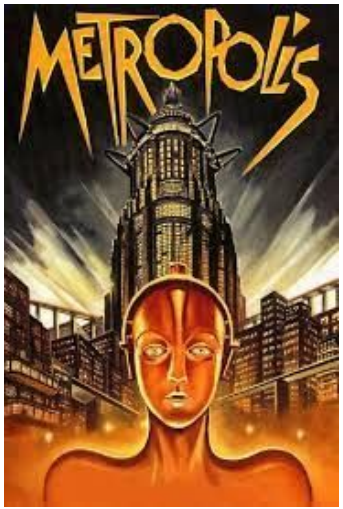
Une brève histoire de l’IA

SIMPLON.CO

Une brève histoire de l'IA

- Dans la première partie du XXe siècle, les auteurs de science fiction ont familiarisé le monde avec le concept de **robot doté d'intelligence artificiel**

Tin Man, personnage du Magicien d'Oz, 1900



Maria, personnage de Metropolis, 1927

Une brève histoire de l'IA

- En 1950, dans son papier **Computing Machinery and Intelligence**, **Alan Turing** pose les bases de l'IA. Il aborde notamment comment entraîner des machines intelligentes et comment tester leur intelligence.
- Turing créa un test qui gardera son nom : **le test de Turing**
- Ce test consiste à faire communiquer un humain avec un ordinateur et un humain. Si la personne qui engage la conversation n'est pas capable d'identifier lequel de ses interlocuteurs est la machine, on considère que le test a été réussi.
  Note : Turing est également connu pour avoir participé au décryptage d'enigma pendant la seconde guerre mondiale.

Une brève histoire de l'IA

- Malheureusement les travaux de Turing n'ont pas donné d'impulsion majeur dans la recherche en IA à causes de deux raisons :
  - Les ordinateurs **ne peuvent pas stocker des informations**, seulement exécuter des commandes
  - Les ordinateurs sont très **chers**

Une brève histoire de l'IA

- En 1956, un événement majeur eut lieu dans l'histoire de l'IA : **Dartmouth Summer Research Project on Artificial Intelligence.**
- Durant cette conférence, le premier algorithme d'IA fut présenté : **Logic Theroist**. Un programme pour mimer les compétences de résolution de problème des humains.

Une brève histoire de l’IA

- Grâce à cette conférence, les **20 années suivantes furent prospères pour l’IA.**
- Les ordinateurs furent capables de stocker des informations, ils sont également moins chers, plus rapides et plus accessibles.
- Les attentes en IA étaient très élevées, Marvin Minsky déclara en 1970, que dans 3 à 8 ans, il y aurait une machine dotée d’une intelligence artificielle équivalente à celle d’un humain moyen.
- Suite à cet optimisme élevé, les scientifiques se sont rendu compte que les ordinateurs n’ étaient pas dotés d'une capacité de calcul suffisamment performante pour répondre aux enjeux. Les investissements baissèrent jusqu’à 1980.

Une brève histoire de l'IA

- En 1980, l'engouement reprit grâce aux progrès dans l'**algorithmie**
- Deux concepts apparurent à cette époque :
  - Le **deep learning** ou **apprentissage profond**
  - Les **systèmes experts** qui reproduisent le processus de décision d'un humain expert dans un domaine. Ces systèmes étaient très utilisés dans l'industrie. Le gouvernement japonais investit 400 millions de dollars dans ce domaine à l'époque. Toutefois, les performances de ces algorithmes n'étaient pas à la hauteur des attentes.

Une brève histoire de l'IA

Quelques événements marquants depuis :

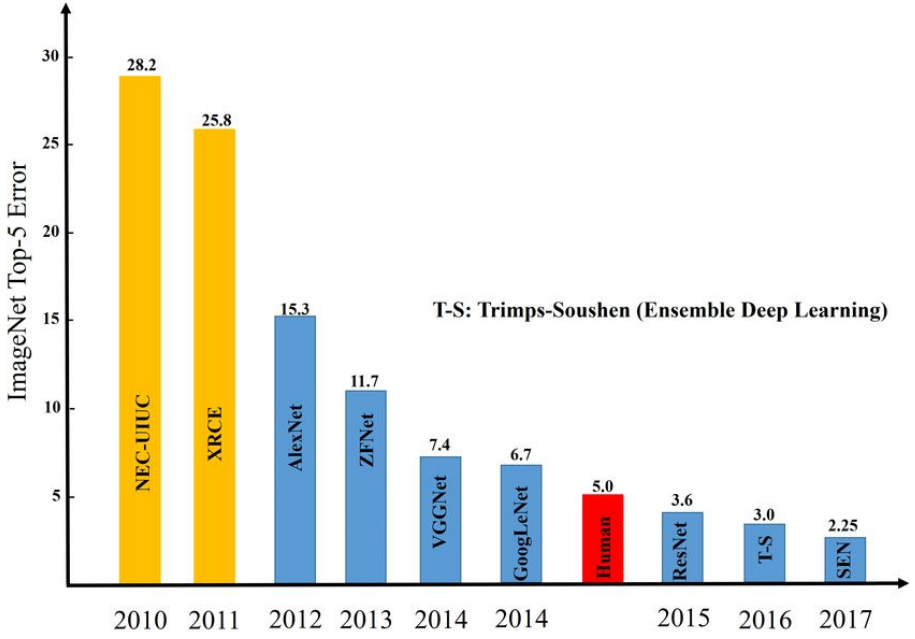
- En 1997, le programme **Deep Blue** d'IBM battit le champion d'échecs Gary Kasparov
- En 2016, se fût au tour d'**Alpha Go** de battre le champion de jeu de Go chinois, Ke Jie

ImageNet

ImageNet est une compétition en reconnaissance d'image lancée en 2007, durant cet événement, les principaux acteurs de l'IA dans le monde s'affrontent en essayant de **minimiser** leurs **erreurs de prédictions**.

L'objectif est de **prédire** correctement **la classe** d'une image. La classe c'est ce qui est représenté sur l'image : un chien, un bateau, etc. Durant cette compétition, il y a **1000** classes possibles, ce qui rend la tâche complexe.

ImageNet

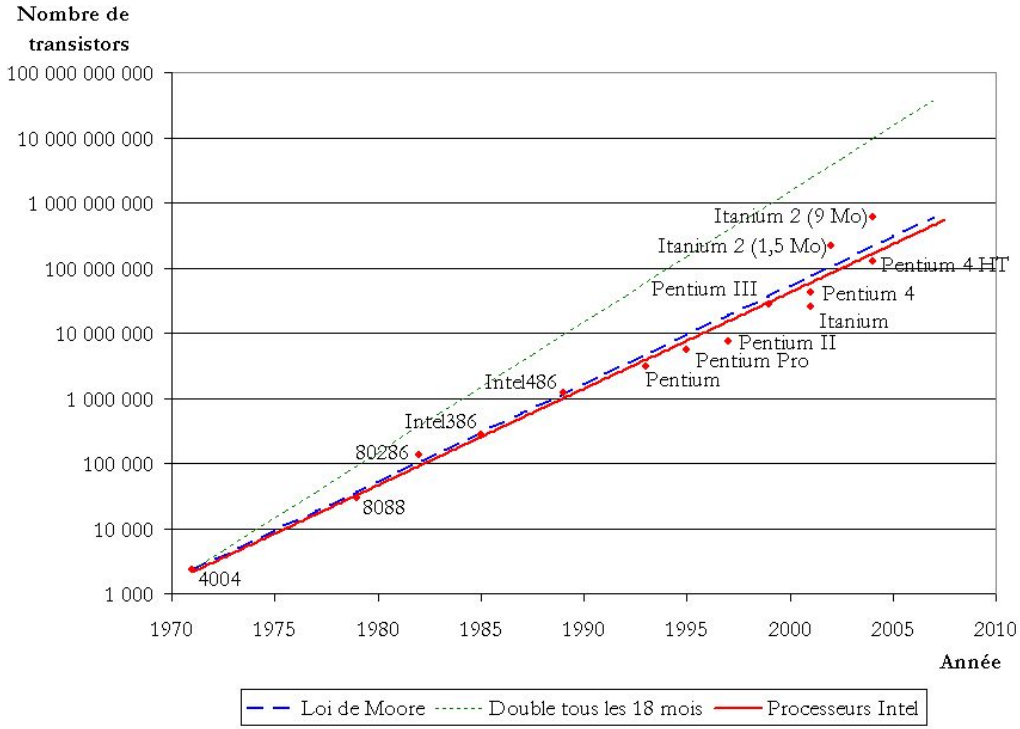


Le **deep learning** (réseaux de neurones) a émergé dans les années 80 mais a longtemps été mis de côté à cause des capacités de calcul nécessaire pour qu'il soit performant.

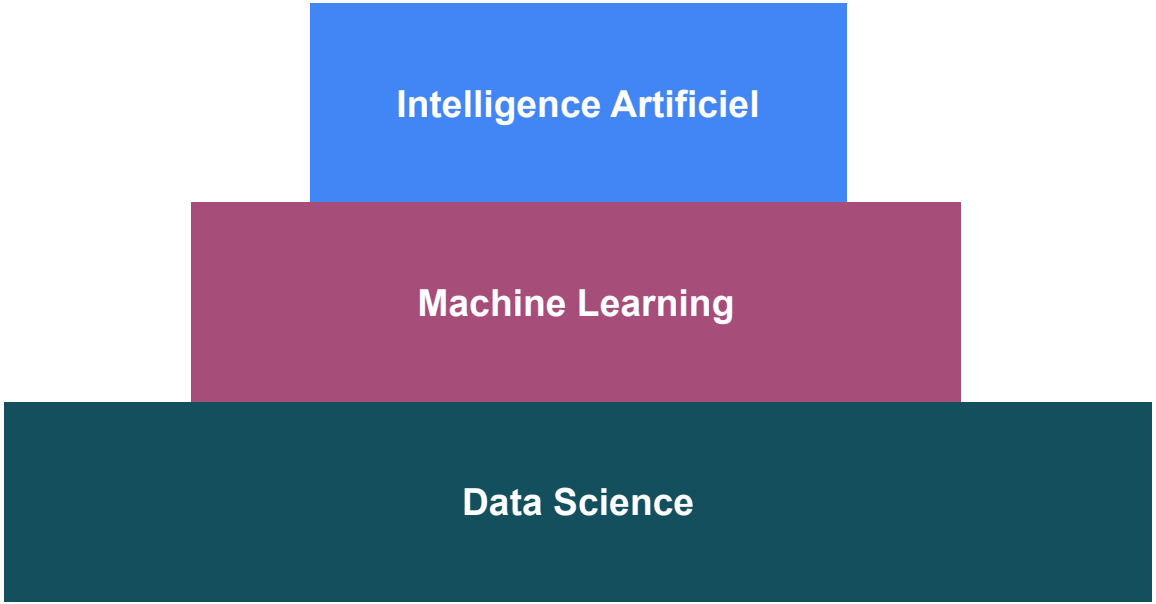
Il est revenu sur le devant de la scène en 2012 avec le réseau de neurones AlexNet.

La loi de Moore

- Les progrès de l'intelligence artificielle sont étroitement liés à la loi de Moore.
- Cette loi indique que la **complexité des transistors double tous les 18 mois**. Par extension, on dit que la capacité de calcul des ordinateurs double tous les deux ans.



Les termes relatifs à l'IA

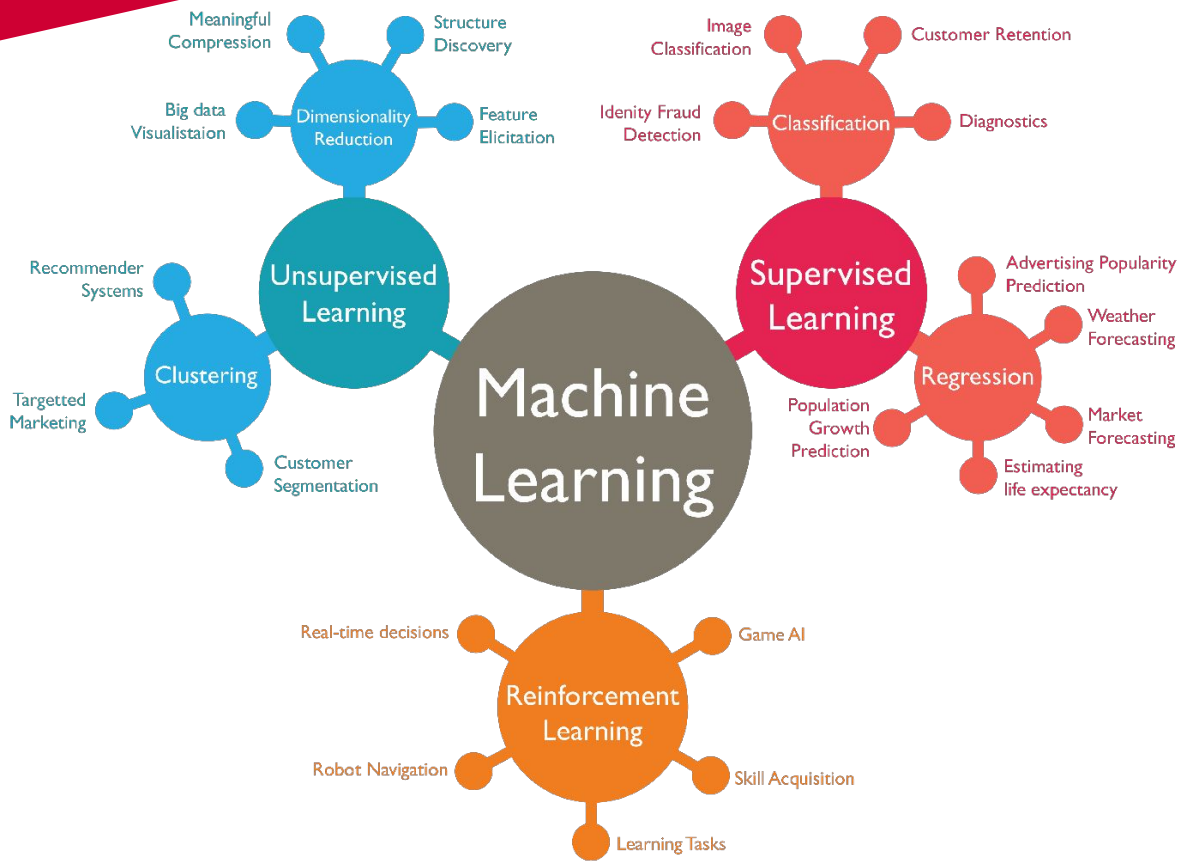


Exemple

Le machine learning

SIMPLON
.CO

Mapping des différentes branches du ML



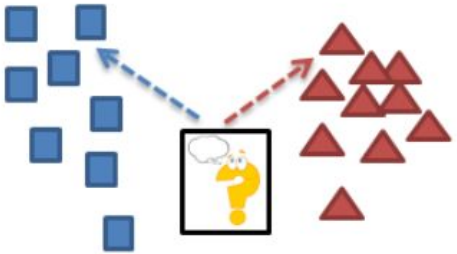
Les différents types d'apprentissage

Apprentissage supervisé	Apprentissage non-supervisé
• Les données sont **étiquetées** • Ces étiquettes sont utilisées pour pénaliser ou récompenser notre algorithme	• **Aucune étiquette** • L'algorithme va se baser sur la structure générale des données

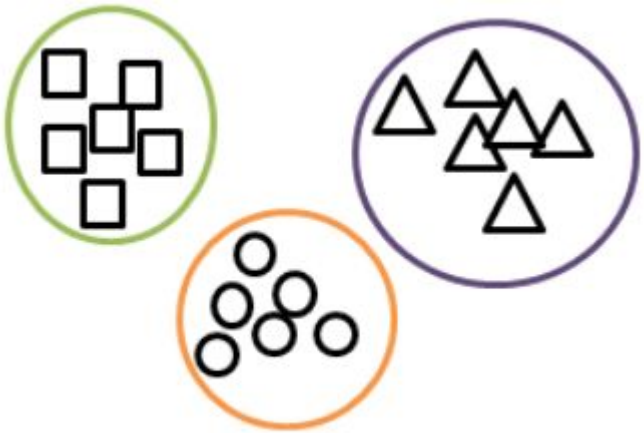
L'**apprentissage supervisé** est bien **plus développé** de nos jours que l'apprentissage non-supervisé. Cependant, la majorité des données ne sont pas étiquetées. Pouvoir exploiter ces jeux de données à l'aide d'algorithme d'apprentissage non-supervisé est un enjeu pour les années à venir.

Exemple d’apprentissage supervisé

- On souhaite entraîner un algorithme qui puisse prédire si l’**image contient un chat ou un chien**
- Afin de pouvoir entraîner cet algorithme, il faut détenir au préalable un jeu de données d’image avec des étiquettes [chien] et [chat]
- Durant l'entraînement du modèle nous allons utiliser une **fonction coût** afin de favoriser ou pénaliser les prédictions



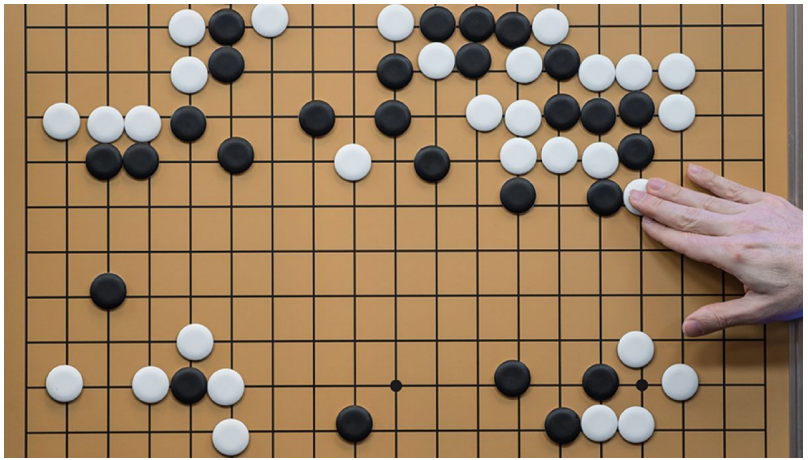
Exemple d’apprentissage non-supervisé



- Vous disposez d’une **base de données client** qui contient des informations à propos de leurs habitudes de consommation (panier moyen, fréquence d’achat, catégorie de produits préférée, etc.)
- A partir de ces informations, vous allez chercher à définir différents segments de client.
- Pour obtenir ces segments vous pouvez utiliser un **algorithme de clustering.**
- Les données ne sont pas étiquetées, l’algorithme va comprendre la structure générale des données.

La 3e branche : Apprentissage par renforcement

- L’objectif de cette méthode est d’**entraîner un agent** qui va prendre une **séquence de décisions**.
- L’algorithme va être sanctionné ou récompensé en fonction des décisions qu’il a prises. Le but est de **maximiser le total des récompenses**.
- AlphaGo est un exemple de reinforcement learning. Sa spécificité étant que l’agent est son propre professeur.



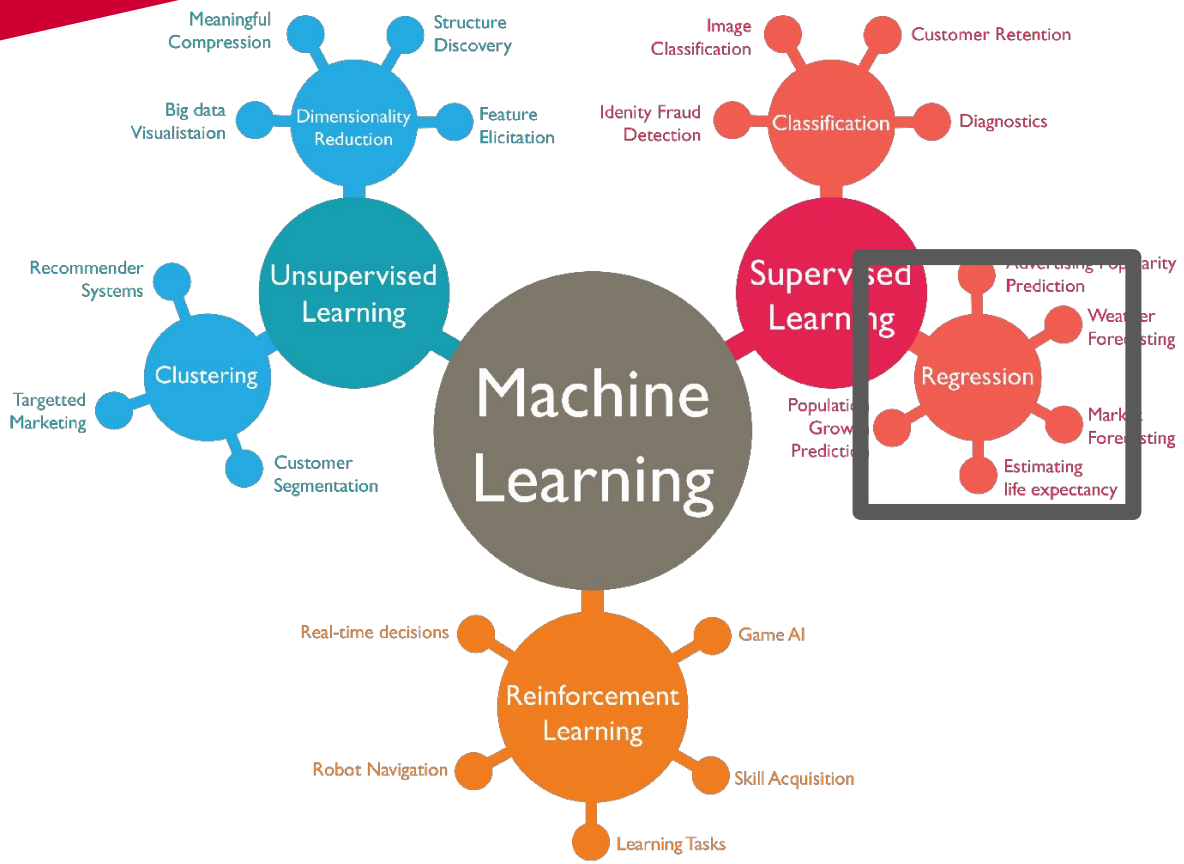
Apprentissage supervisé : Régression vs Classification

Régression	Classification
• L'output de cet algorithme est un **nombre** • Exemple : Prédire le prix d'une maison	• L'output de cet algorithme est la prédiction d'une **classe** • Par exemple dans notre algorithme pour prédire si l'image contient un chat ou un chien, l'output est un vecteur 2x1 • Voici le vecteur de sortie pour la prédiction d'un chien : $\begin{bmatrix}1\\0\end{bmatrix}$ • et d'un chat : $\begin{bmatrix}0\\1\end{bmatrix}$

La régression linéaire à une seule variable

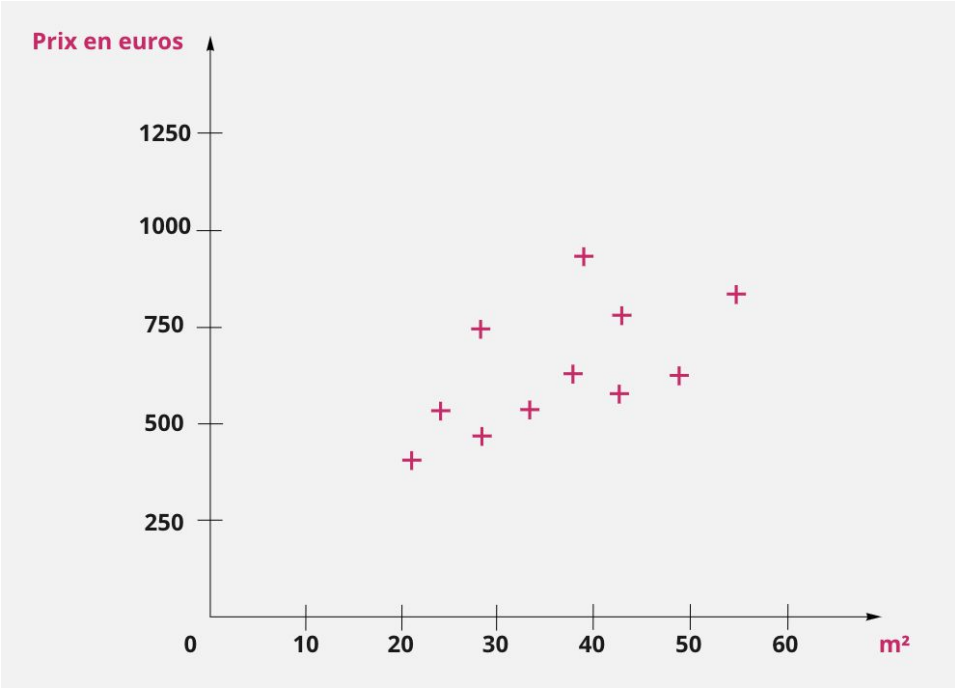
SIMPLON
.CO

Où sommes-nous ?



Prédire le prix du loyer d'un logement

1 Lien whiteboard

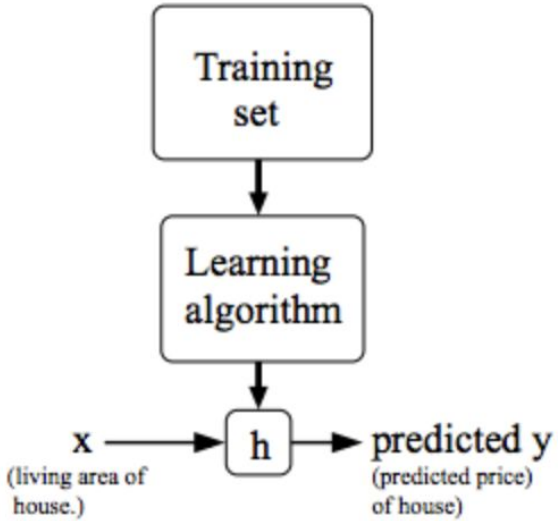


Prédire le prix du loyer d'un logement

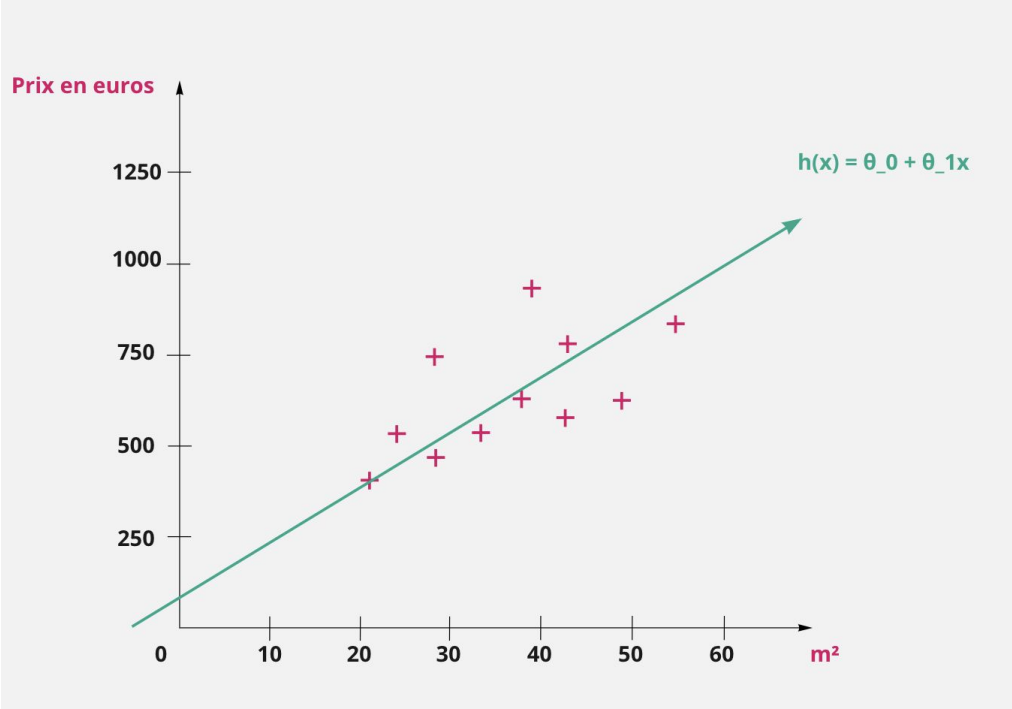
Taille en m² (x)	**Prix en € (y)**
25	380
35	423
55	689
…	…
m	

x : variable (feature)
y : cible (target)
m : le nombre d'exemples d'entraînement (training examples)
n : nombre de variables

La modélisation



Régression linéaire à une variable (univariée)



Exemples de valeur de théta

2 Lien whiteboard

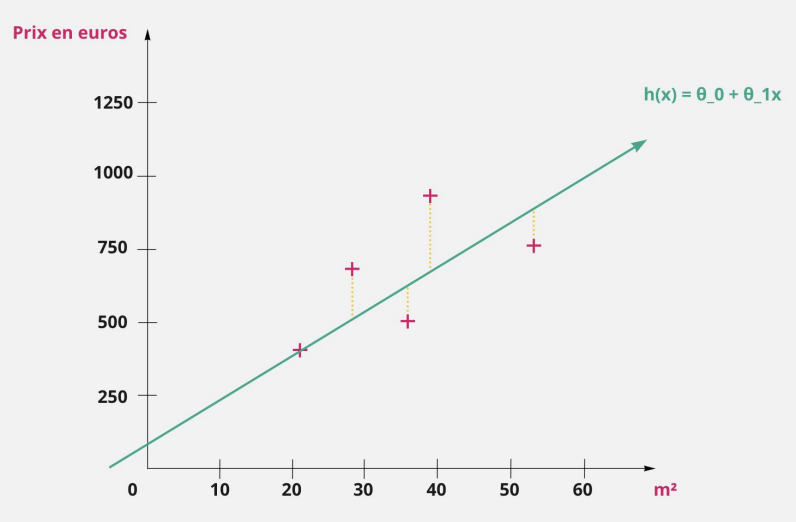
La fonction coût

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

$$\hat{y} = h_\theta(x) = \theta_0 + \theta_1 x$$

Le terme **m** sert à faire la moyenne
Le terme ½ est ajouté comme une astuce afin de faciliter les calculs de la descente du gradient par la suite

Distance entre les observations et les prédictions



La fonction coût

L’objectif est de minimiser la fonction :

$$J(\theta_0, \theta_1)$$

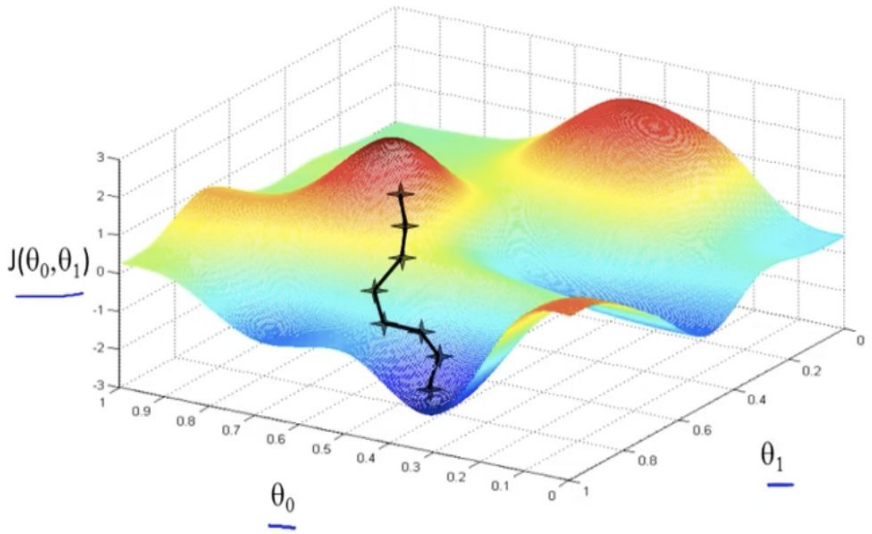
Cette fonction est appelée *Mean Squared Error (**MSE**)*, *Squared Error function* et en français on parlera plutôt de la **méthode des moindres carrés**.

C’est la métrique principale que la fonction va essayer de minimiser.

Intuition sur la fonction coût

Lien whiteboard

Descente du gradient : Intuition



dérivé

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

alpha : coefficient d'apprentissage (learning rate)

Hyperparamètre : alpha

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

Un hyperparamètre, c'est un paramètre que l'algorithme ne peut pas apprendre par lui-même.

Il faut tester l'algorithme en faisant varier cette hyperparamètre

Descente du gradient : implémentation

Correct: Simultaneous update

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

Incorrect:

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_1 := \text{temp1}$$

Descente du gradient : intuition

4 Lien whiteboard

Descente du gradient : Choisir l'alpha

5 Lien whiteboard

Descente du gradient : pour la régression linéaire

repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

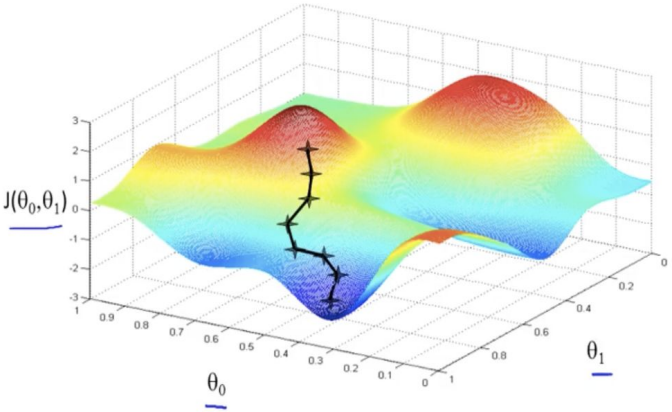
$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left((h_\theta(x_i) - y_i) x_i \right)$$

}

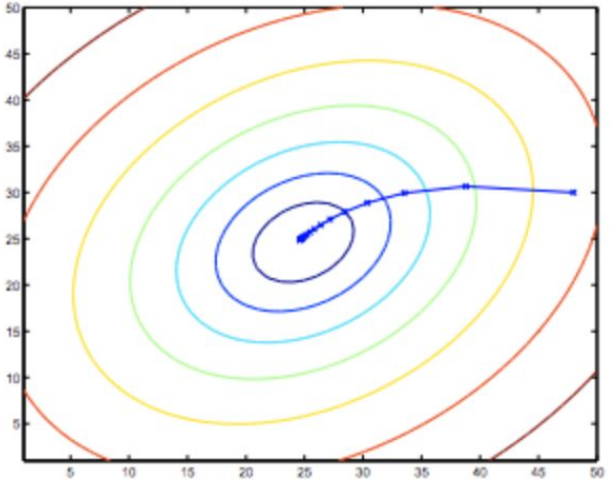
Cette méthode itère sur tout le jeu de données et est appelée le **batch gradient descent**

Descente du gradient : pour la régression linéaire

Descente du gradient



Contour's plot



La descente du gradient peut être sujette à la convergence dans des minimums locaux tandis que le problème d'optimisation posé pour la régression linéaire n'a qu'un seul minimum. En d'autre mot la **fonction coût** de la régression linéaire est **convexe**.

Descente du gradient : implémentation

6 Lien whiteboard

R^2 : coefficient de détermination

Afin d'évaluer les performances de votre algorithme, vous pouvez utiliser une autre métrique : le **coefficient de détermination** ou **R^2**.

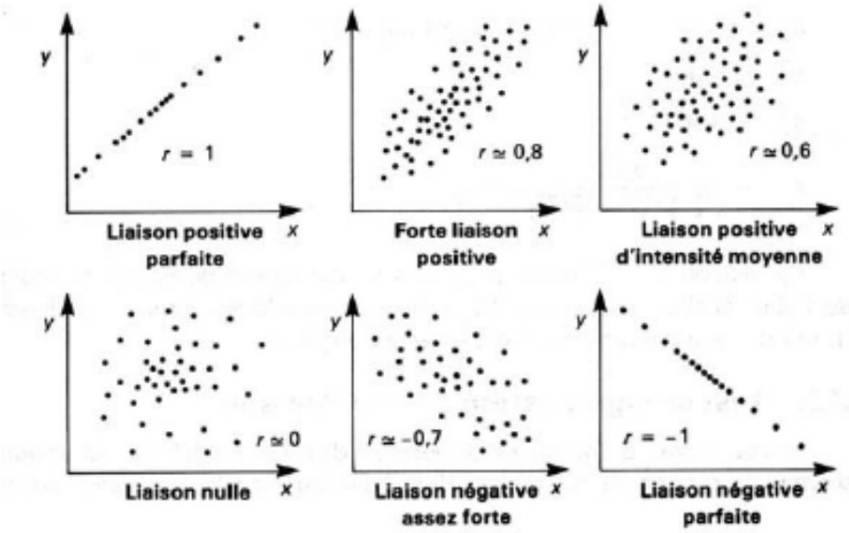
Cette métrique est comprise entre 0 et 1 explique la proportion de la variance de Y (**variable à expliquer** comme le prix d'une prix d'une maison) expliquée par X (**variable explicative**, le nombre de m²)

Plus cette métrique s'approche de 1, plus votre modélisation sera qualitative.

Coefficient de Pearson. Relation entre les variables

Le R^2 est lié à une autre métrique connue pour mesurer la relation entre deux variables : le coefficient de Pearson.

Cette mesure est désigné par la lettre **r.** Le coefficient de détermination est le **carré** du coefficient de corrélation.



Coefficients de corrélation relatifs à différents nuages de dispersion

La régression linéaire à plusieurs variables

SIMPLON.CO

Descente du gradient : implémentation

7 Lien whiteboard

Forme de la fonction h(x)

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_n x_n$$

$$h_\theta(x) = \begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x$$

$$x_0 = 1$$

Forme de la fonction h(x)

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

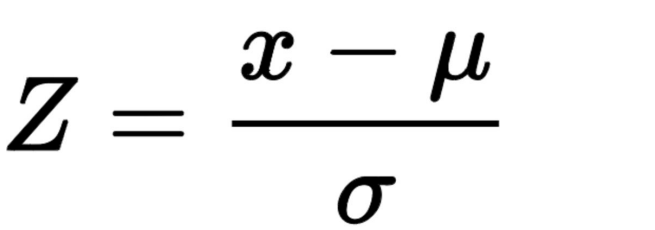
$$\dots$$

Feature scaling

Comment le feature scaling peut accélérer la convergence de la descente du gradient ?

8 Lien whiteboard

Feature scaling : standardisation



Si on soustrait la moyenne à chaque exemple, la moyenne sera égale à 0

Centrage et réduction

Si on divise chaque exemple par l'écart-type, l'écart-type sera égale à 1

Galilée et la tour de Pise

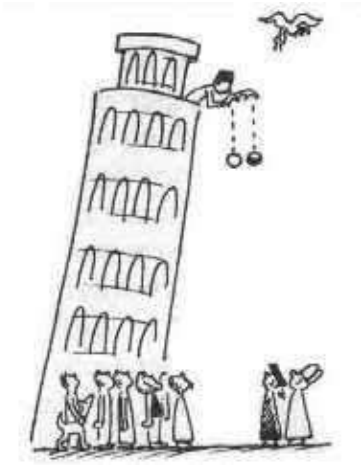
Imaginons **Galilée** grimpant sur la tour de Pise,

Il s'arrête au premier étage et jette une pierre

Il réitère son expérience à chaque étage

Grâce à ses **observations**, il en déduit une loi unique qui lie x, le temps de chute à y, la hauteur où est lâché le cailloux.

La formule qui lie x à y est **y = 0,5g*x**2**



Parfois la fonction qui lie plusieurs événements n'est pas une droite

Au passage, Galilée a ainsi jeté les bases de la méthode scientifique et de la physique : Induire des lois à partir d'observations, et prédire des phénomènes à partir de ces lois. C'est ce que fait ni plus ni moins, l'apprentissage automatique.

Variable et régression polynomial

Housing prices prediction

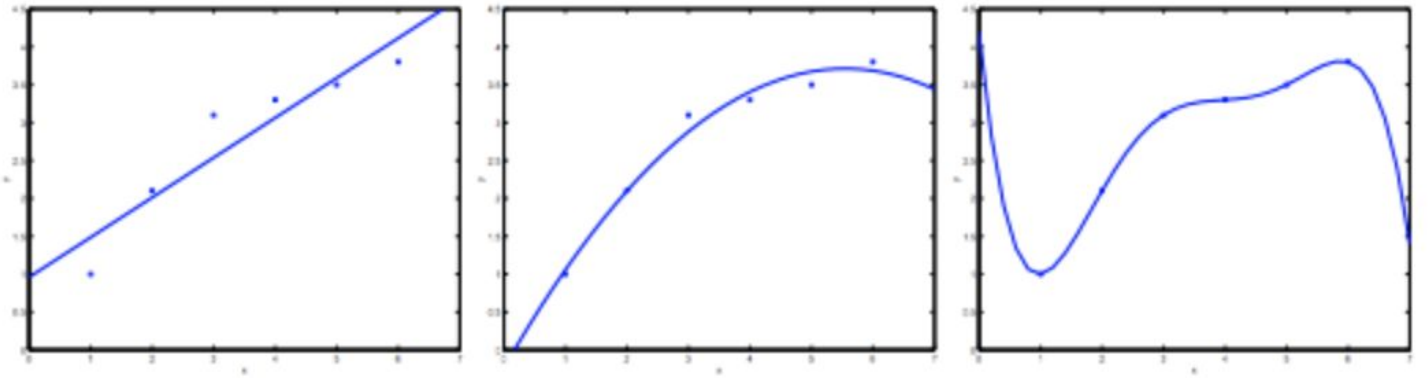
$$h_\theta(x) = \theta_0 + \theta_1 \times frontage + \theta_2 \times depth$$

Création d'une nouvelle feature (feature engineering)
Il est possible de créer une nouvelle variable **area** qui sera égale à **frontage x depth**

9 Lien whiteboard

Overfitting et la régularisation

Underfitting | Balanced | Overfitting



$\theta_0 + \theta_1 x$ $\theta_0 + \theta_1 x + \theta_2 x^2$ $\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

- **Underfitting** : Biais est élevé
- **Overfitting** : La variance est élevée. C'est-à-dire que la fonction va très bien fit le jeu de données mais les généralisations ne seront pas performantes

Overfitting et la régularisation

Comment gérer le problème de l'overfitting ?

- Réduire le nombre de features

$x_1 =$ size of house
$x_2 =$ no. of bedrooms
$x_3 =$ no. of floors
$x_4 =$ age of house
$x_5 =$ average income in neighborhood
~~$x_6 =$ kitchen size~~
$\vdots$
x_{100}

- Utiliser la régularisation

La régularisation

Imaginons que nous souhaitons rendre cette fonction plus quadratique :

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

- Si nous souhaitons baisser l'influence des paramètres $\theta_3 x^3$ et $\theta_4 x^4$ sans les supprimer, nous pouvons modifier la fonction coût :

$$min_\theta \underbrace{\frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{\text{fonction coût classique}} \underbrace{+ 1000 \cdot \theta_3^2 + 1000 \cdot \theta_4^2}_{\text{ajout des paramètres dans la fonction}}$$

- Afin de minimiser la fonction, les paramètres $\theta_3 x^3$ et $\theta_4 x^4$ devront être proches de 0

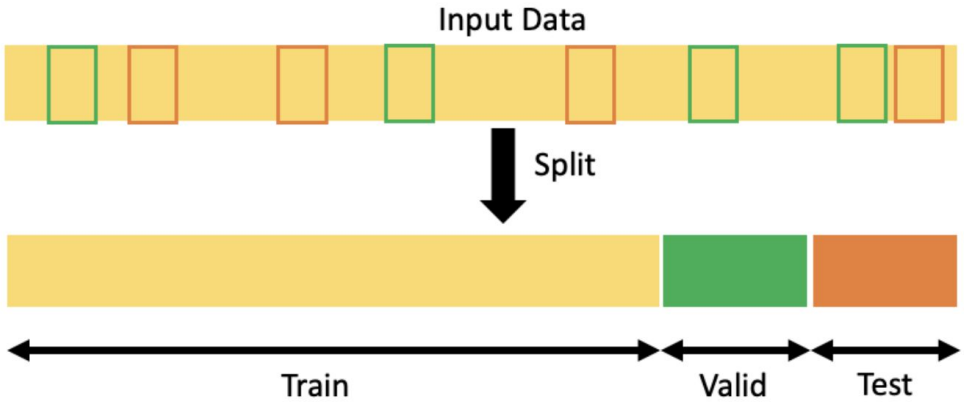
Régularisation

$$min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2$$

Le terme λ sera un autre hyperparamètre pour contrôler l'influence de la régularisation

Séparation des données

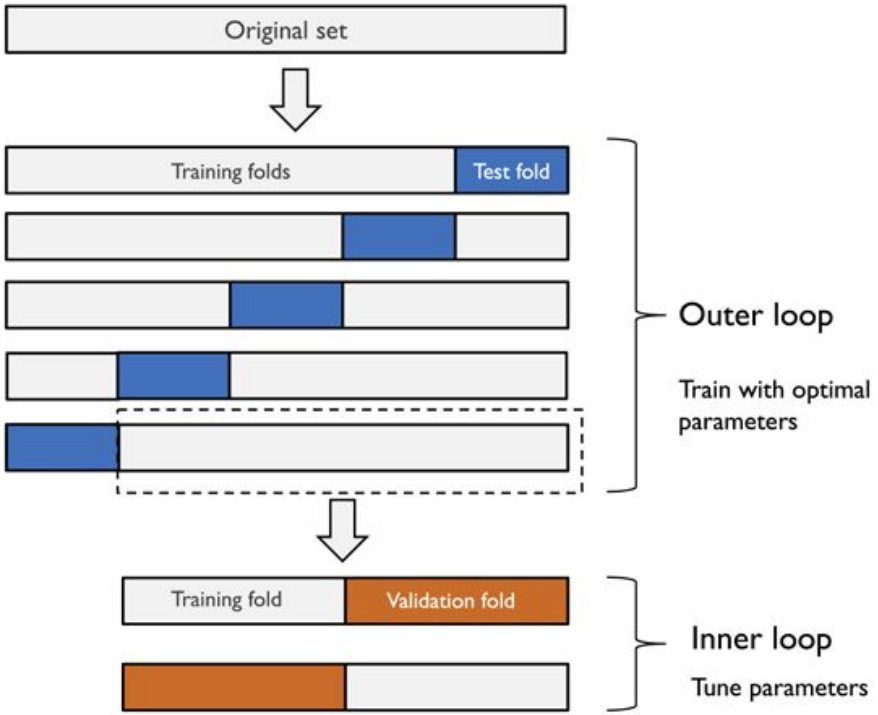
Afin de réduire l'overfitting et d'avoir des métriques de performances plus précises, il faut séparer le jeu de données en 3 parties :



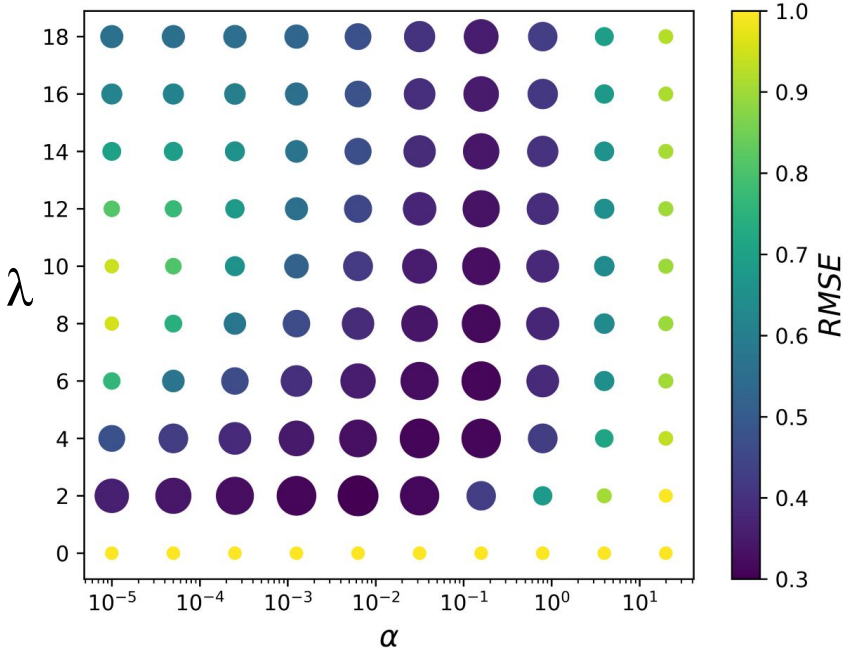
Train : entraîner l'algorithme
Validation : sélection des modèles et des hyperparamètres
Test : pour évaluer l'algorithme sur des observations qu'il n'a jamais rencontré

Validation croisée

L'objectif est de faire varier les folds et d'effectuer une moyenne des métriques sur les différents folds



Trouver les bons hyperparamètres (grid search)



L'objectif est de faire varier les hyperparamètres afin d'obtenir les meilleurs résultats

Exercices

https://github.com/jeremy-vangansberg/exo_reg

Pour aller plus loin

Veille :

- Normal Equation
- Régularisation L1 et L2 (ridge, lasso)

Resources

Ce cours est inspiré du MOOC Machine learning sur Coursera

Pour aller plus loin :

https://openclassrooms.com/fr/courses/4525281-realisez-une-analyse-exploratoire-de-donnees

https://openclassrooms.com/fr/courses/4525326-realisez-des-modelisations-de-donnees-performantes

https://eric.univ-lyon2.fr/~ricco/cours/slides/regularized_regression.pdf