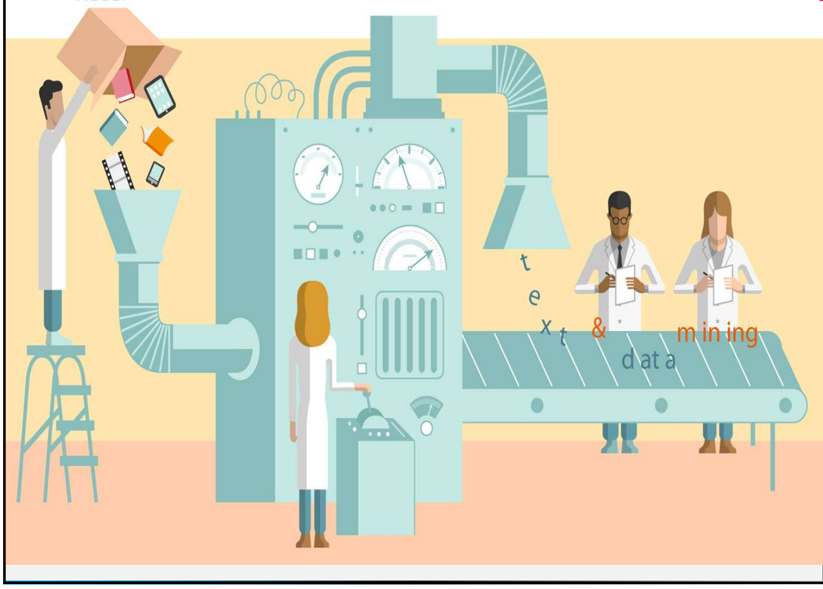
Université Mohammed V
Faculté des Sciences
Rabat

Royaume du Maroc
Université Mohammed V de Rabat
Faculté des Sciences
Département d'Informatique

IPSS
Intelligent Processing
Systems & Security

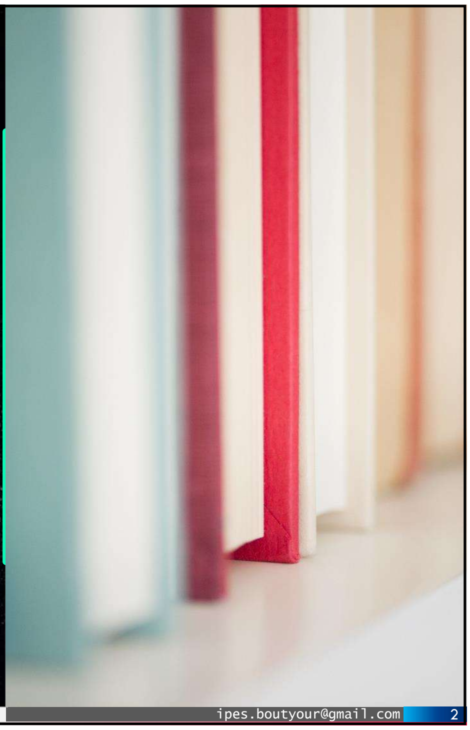


Data Mining & Machine Learning

Master IPS
Faculté des sciences – Rabat
Université Mohamed V

1

Machine Learning : Introduction

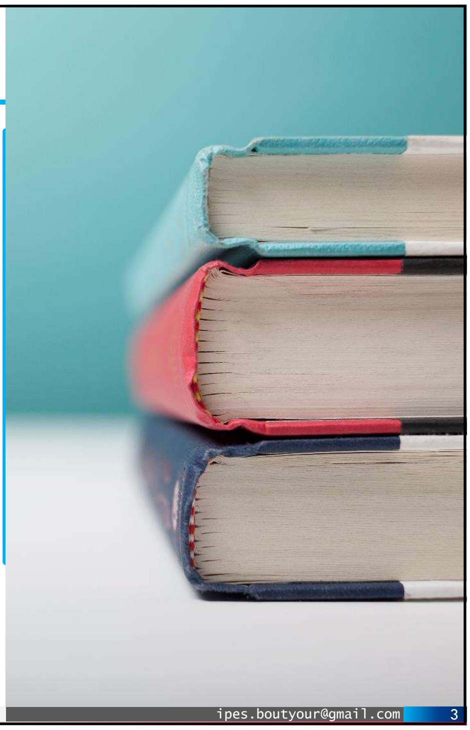


2

Avant-propos

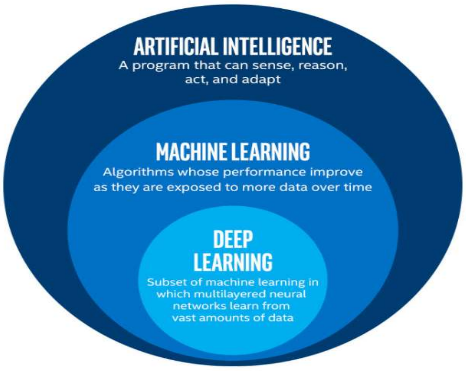
Syllabus (Machine Learning)

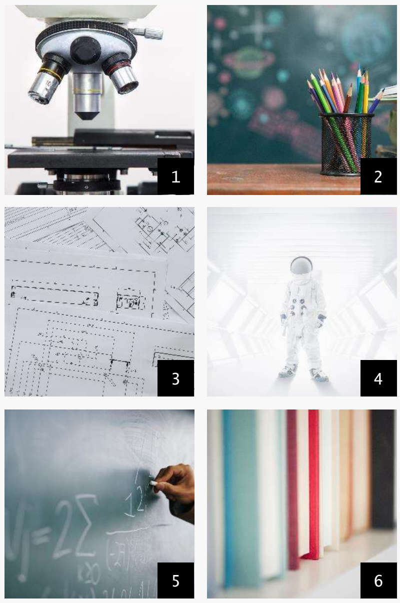
- Introduction
- Apprentissage supervisé
  - Régression linéaire
  - Régression logistique
  - KNN
  - Naïve Bayes
  - Random Forest
  - SVM
  - Réseaux de neurones
- Apprentissage non supervisé



Introduction

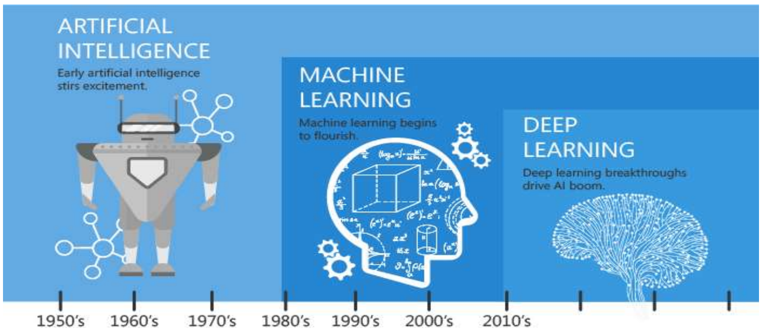
- Le **machine learning** est l'une des technologies récentes les plus en vogue cette dernière décennie
- C'est une technologie de l'Intelligence Artificielle (IA)

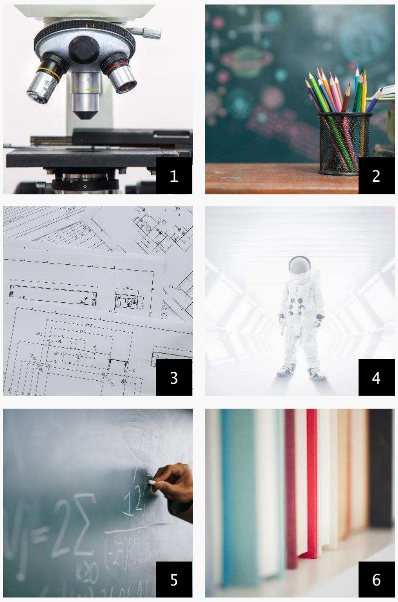




Introduction

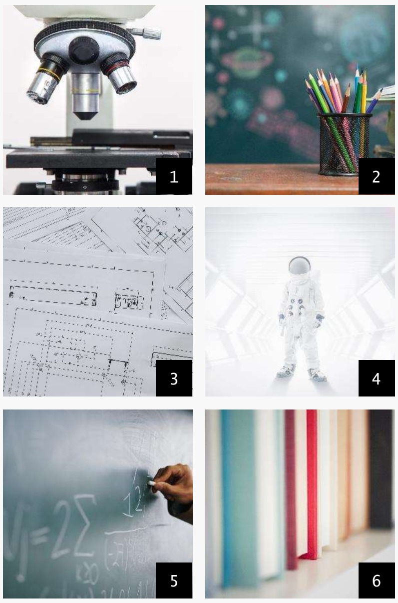
- Le **machine learning** est l'une des technologies récentes les plus en vogue cette dernière décennie
- C'est une technologie de l'Intelligence Artificielle (IA)





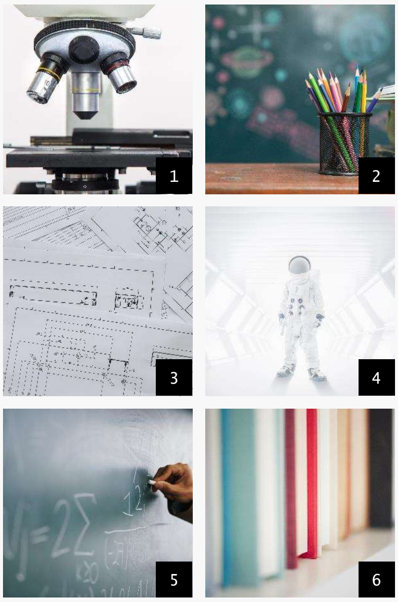
Introduction

- Le **machine learning** est l'une des technologies récentes les plus en vogue cette dernière décennie
- C'est une technologie de l'Intelligence Artificielle (IA)
- Il est omniprésent dans l'utilisation quotidienne de l'informatique :
  - Détection de spam ; Suggestion de vidéos sur les plateformes vidéo ; Suggestion de produits à acheter sur les sites e-commerce ; Reconnaissance faciale des photos uploadées sur les réseaux sociaux…



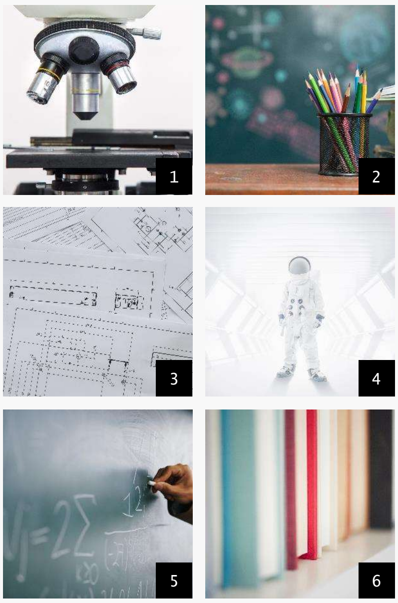
Introduction

- Le **machine learning** est l'une des technologies récentes les plus en vogue cette dernière décennie
- C'est une technologie de l'Intelligence Artificielle (IA)
- Il est omniprésent dans l'utilisation quotidienne de l'informatique :
  - Détection de spam ; Suggestion de vidéos sur les plateformes vidéo ; Suggestion de produits à acheter sur les sites e-commerce ; Reconnaissance faciale des photos uploadées sur les réseaux sociaux…



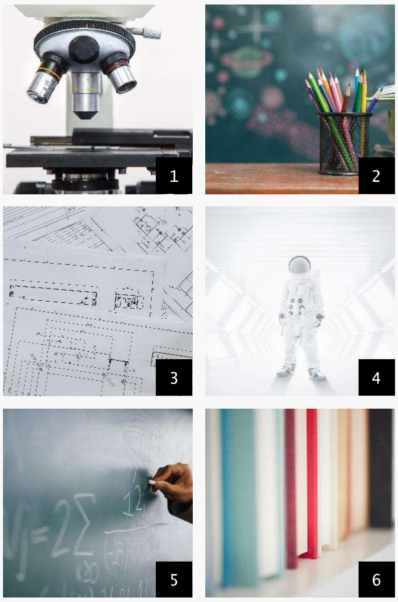
Introduction

- Le **machine learning** est l'une des technologies récentes les plus en vogue cette dernière décennie
- C'est une technologie de l'Intelligence Artificielle (IA)
- Il est omniprésent dans l'utilisation quotidienne de l'informatique.
- Il a pour objectif de développer des algorithmes qui imitent l'apprentissage du cerveau humain.
- Il touche à presque tous les secteurs d'activité



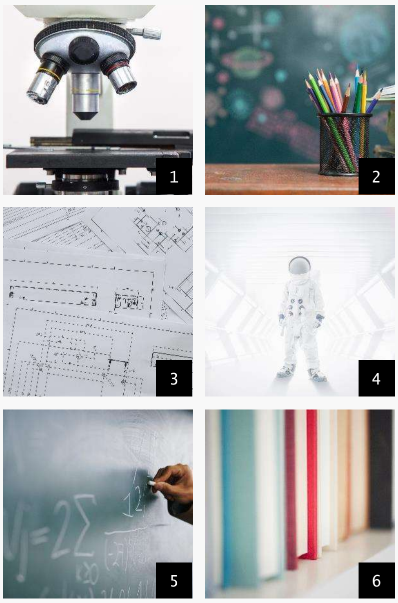
Historique

- Le concept de Machine Learning n'est pas récent
- Dans les années 1950, Arthur Samuel a développé un programme informatique de jeu de dames pas via une programmation explicite mais en le programmant à jouer contre lui-même à des milliers de parties de jeu.
- Le programme a appris ce qui faisait gagner et ce qui faisait perdre → il a appris « tout seul » à jouer aux dames mieux que son développeur.



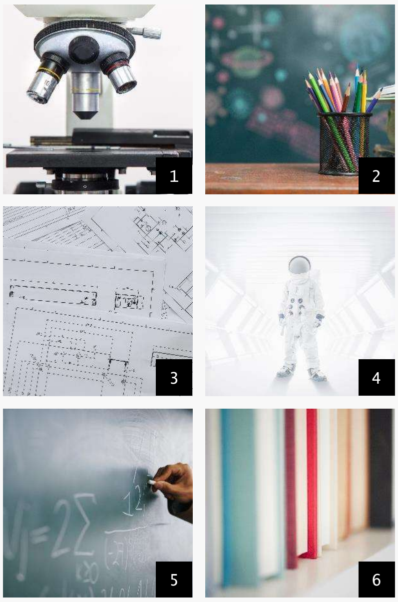
Définition

- **Arthur Samuel :** Domaine d'études qui donnent aux ordinateurs la capacité d'apprendre sans qu'ils ne soient explicitement programmés (1959)
- **Tom Mitchell :** Un programme informatique apprend automatiquement à réaliser une tâche à partir de l'expérience, si sa performance à réaliser la tâche s'améliore avec l'expérience (1997).



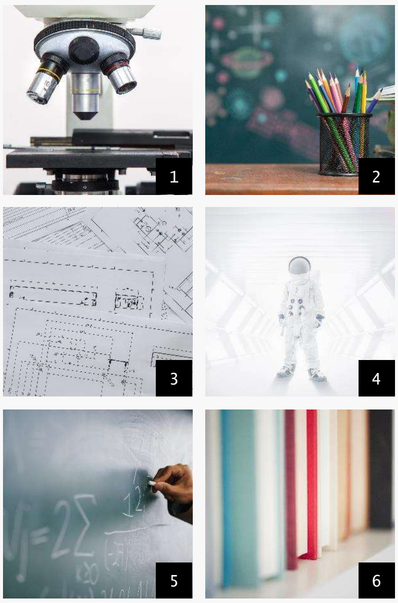
Motivations

- **Pourquoi le ML?**
  - Grand volume de données à exploiter
  - Disponibilité des ressources matérielles et de calcul
  - Certains problèmes sont trop complexes pour être explicitement programmés.



Limitations

- **Quel modèle appliquer et quand ?**
  - Nécessité de tester plusieurs modèle avant de trouver « le bon »
- **Qualité / taille de l'ensemble d'apprentissage**
- **Risque de sur-apprentissage**



Types de Machines Learning

Différents types d'apprentissage

- Apprentissage supervisé
- Apprentissage non supervisé
- Apprentissage semi-supervisé
- Apprentissage par renforcement (Reinforcement learning)
- Systèmes de recommandation (Recommander Systems)

Types de Machines Learning

Apprentissage supervisé

- Le plus fréquemment utilisé
- L'apprentissage se fait en se basant sur des données labélisées (i.e. pour lesquelles la valeur à prédire est déjà connue)
  - **Régression :** la variable à prédire est quantitative continue
  - **Classement :** la variable à prédire est qualitative

Types de Machines Learning

Apprentissage supervisé

- **Régression :** Les algorithmes de régression sont utilisés pour résoudre les problèmes de régression dans lesquels il existe une relation linéaire entre les variables d'entrée et de sortie. Ils sont utilisés pour prédire des variables de sortie continues, telles que les tendances du marché, les prévisions météorologiques, etc.
- Quelques algorithmes de régression populaires sont donnés ci-dessous :
  - Régression linéaire, Régression polynomiale, Régression rigide, Régression lasso, Réseaux de neurones, Arbres de décision.

Types de Machines Learning

Apprentissage supervisé

- **Classement :** Les algorithmes de classement sont utilisés pour résoudre les problèmes de classement dans lesquels la variable de sortie est catégorique, comme "Oui" ou "Non", Homme ou Femme, Rouge ou Bleu, etc. Les algorithmes de classement prédisent les catégories présentes dans l'ensemble de données (Spam detection, Email Filtering, etc)
- Quelques algorithmes de classement populaires sont donnés ci-dessous :
  - KNN, Naïve Bayes, Régression logistique, SVM, Réseaux de neurones, Arbres de décision.

Types de Machines Learning

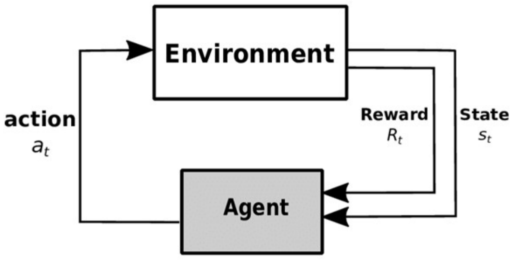
Apprentissage non supervisé

- L'apprentissage se fait directement sur des données non labélisées
- Objectif : découvrir une structure dans les données
  - **Classification :** regroupement des données en clusters
  - **Recherche de pattern :** découverte de « motif/pattern » dans les données
  - **Réduction de dimensions** (analyse factorielle)

Types de Machines Learning

Apprentissage par renforcement

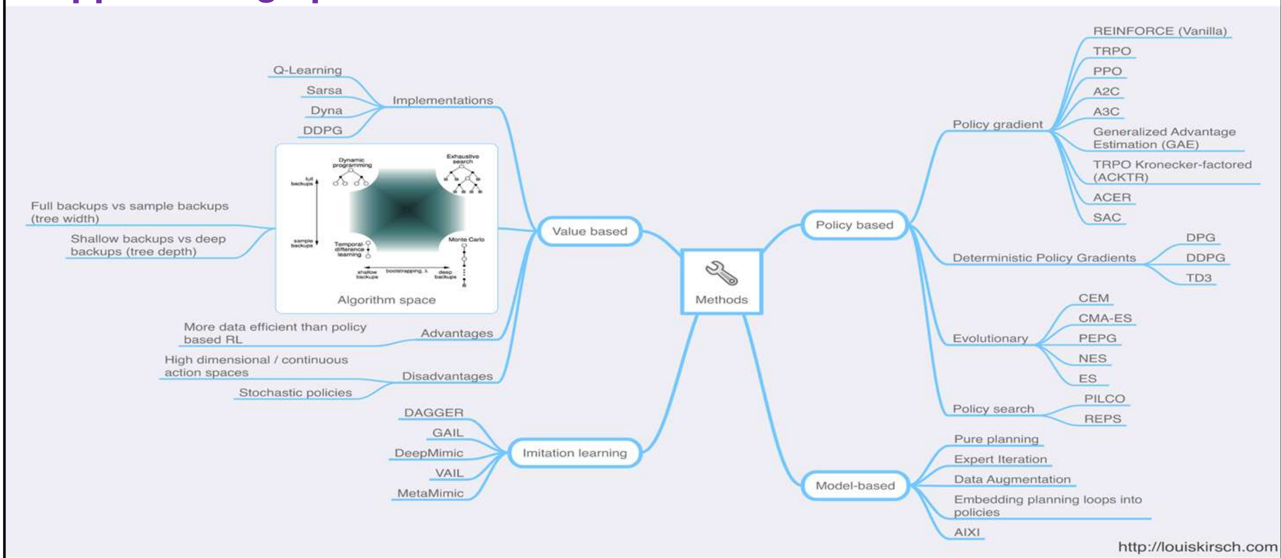
- Un agent autonome est mis dans un environnement
- L'environnement donne à l'agent une récompense ou une « punition » en fonction de ses actions.
- L'agent apprend à travers ses expériences répétitives les actions à faire / ne pas faire de sorte à optimiser la récompense dans le temps
- Algorithmes: Q-learning, SARSA, Dyna, DDPG, MetaMimic, DAGGER



A Map of Reinforcement Learning | Louis Kirsch

Types de Machines Learning

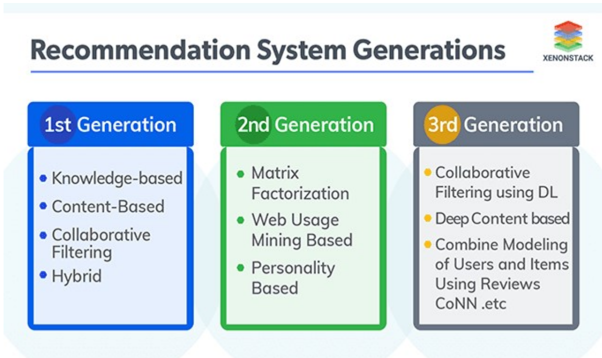
Apprentissage par renforcement



Types de Machines Learning

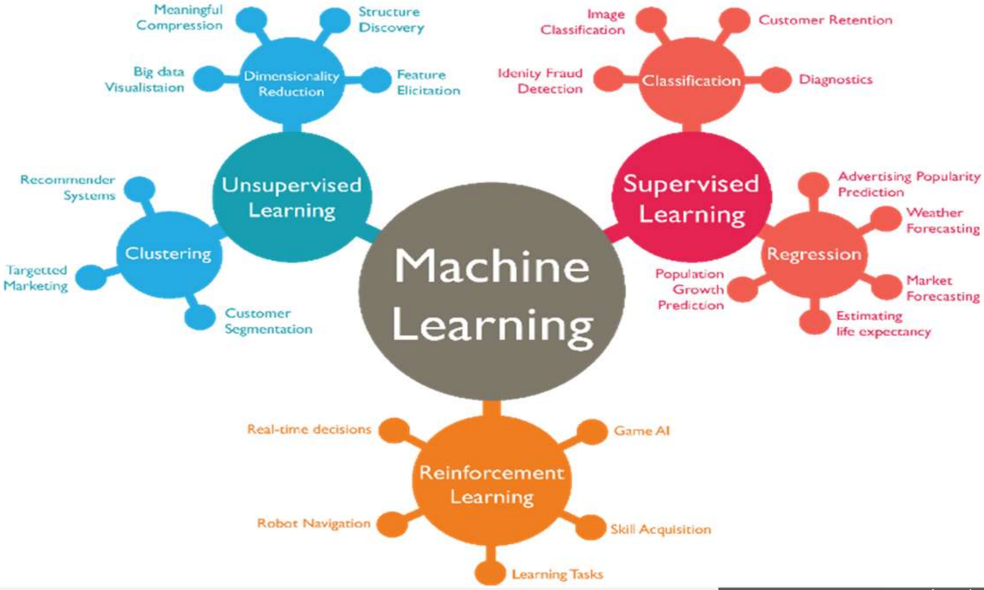
Systèmes de recommandation

- Filtrage des données de telle sorte à ne présenter que celles qui seraient intéressantes pour l'utilisateur en fonction de son profil, historique, comportement des utilisateurs similaires,…



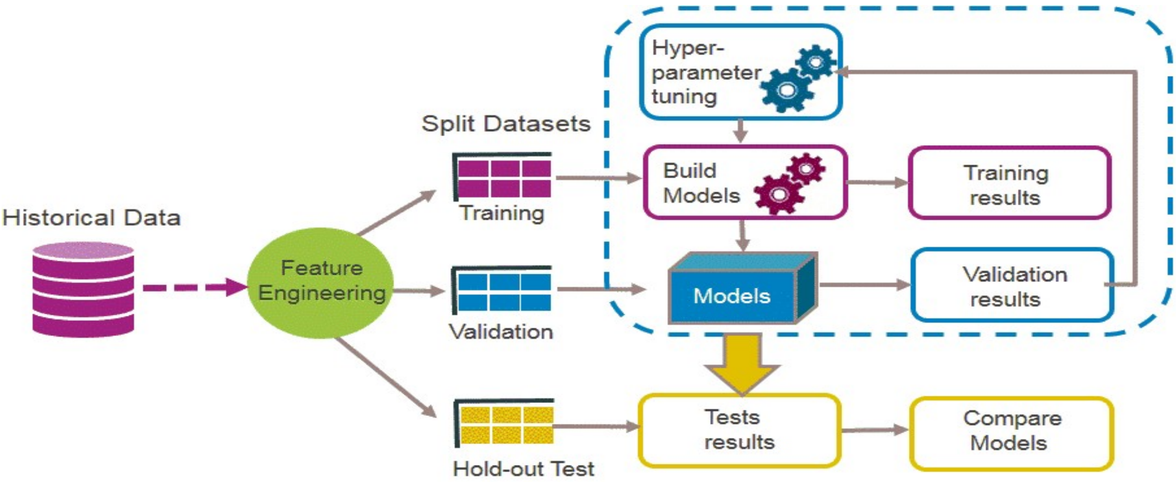
Next Generation Recommender Systems Overview (xenonstack.com)

Types de Machines Learning



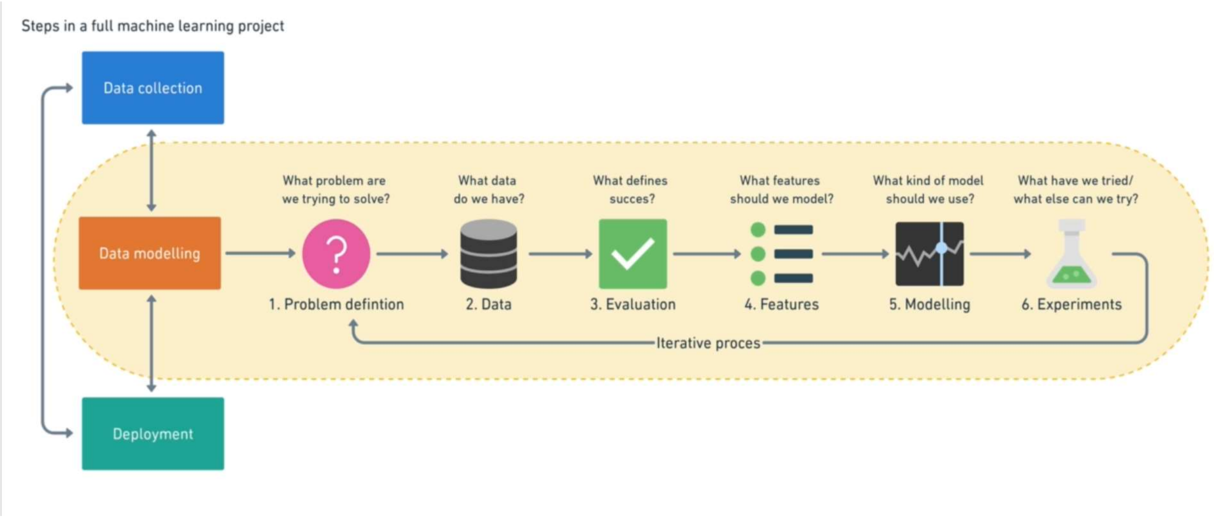
Processus du Machine Learning

Processus d'un modèle machine learning



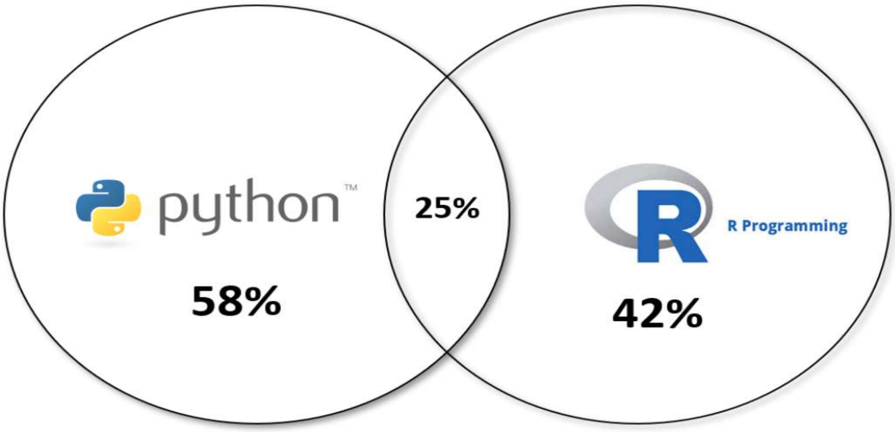
Processus du Machine Learning

Processus d'un modèle machine learning



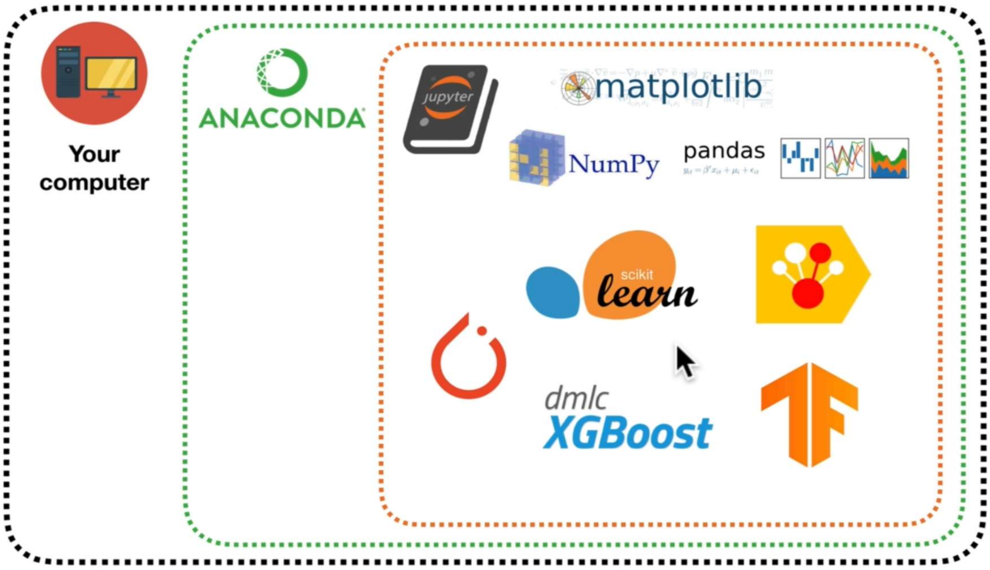
Processus du Machine Learning

Langage de programmation



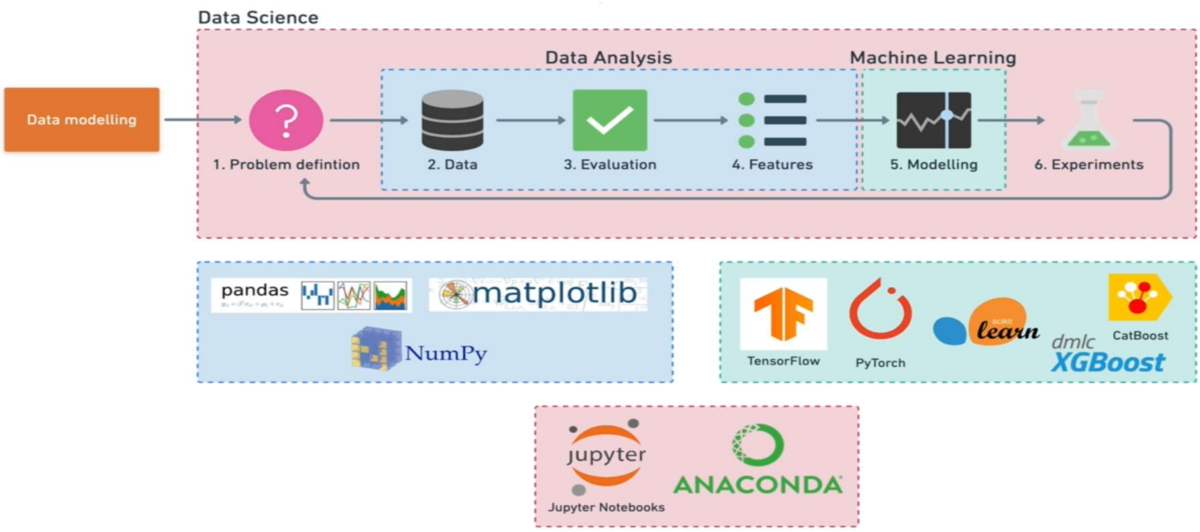
Based on a survey in 2019 on engineering students, 58% showed interest in learning Python, 42% showed interest in R Programming, and 25% in both the programming languages.

Processus du Machine Learning



Processus du Machine Learning

Environnement Python



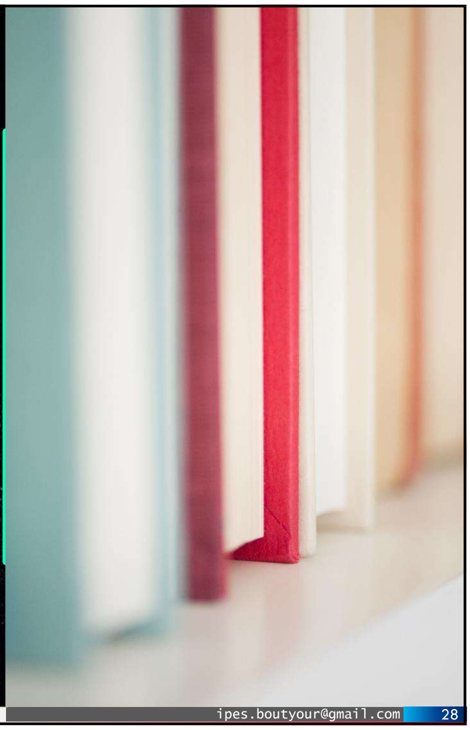
Processus du Machine Learning

Machine Learning frameworks

- **Scikit-learn** : est un framework convivial qui contient une grande variété d'outils utiles : modèles de classification, de régression et de regroupement, ainsi que des outils de prétraitement, de réduction de la dimensionnalité et d'évaluation.
- **TensorFlow (TF)** : est un cadre d'apprentissage automatique de bout en bout de **Google** qui vous permet d'effectuer un éventail extrêmement large de tâches en aval. Avec TF2.0 et les versions plus récentes, l'efficacité et la commodité ont été améliorées.
- **Keras** : est construit au-dessus de **TensorFlow**, ce qui en fait une enveloppe pour l'apprentissage profond. Il est incroyablement convivial et facile à prendre en main. La modularité de ses blocs de réseaux neuronaux et le fait qu'il soit écrit en Python, ce qui facilite le débogage, constituent de solides atouts.
- **PyTorch** : est le concurrent direct de **TensorFlow**, développé par **Facebook**, et est largement utilisé dans les projets de recherche. Il permet une personnalisation presque illimitée et est bien adapté à l'exécution d'opérations tensorielles sur les GPU (en fait, TensorFlow l'est aussi).

Chapitre 1

Machine Learning: Régression



Régression

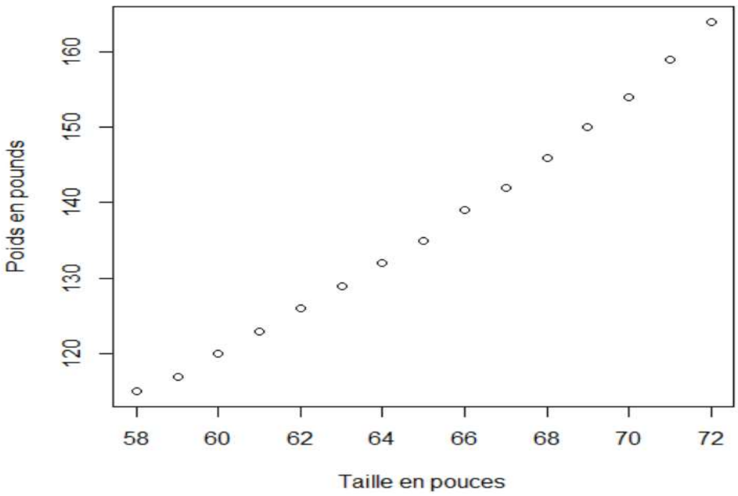
Types de régression :

- La régression sert à trouver la relation d'une variable par rapport à une ou plusieurs autres.
- Il existe plusieurs algorithmes pour la régression:
  - Régression linéaire
  - Régression polynomiale
  - Régression logistique
  - Régression quantile
  - etc.

Régression linéaire

Problématique :

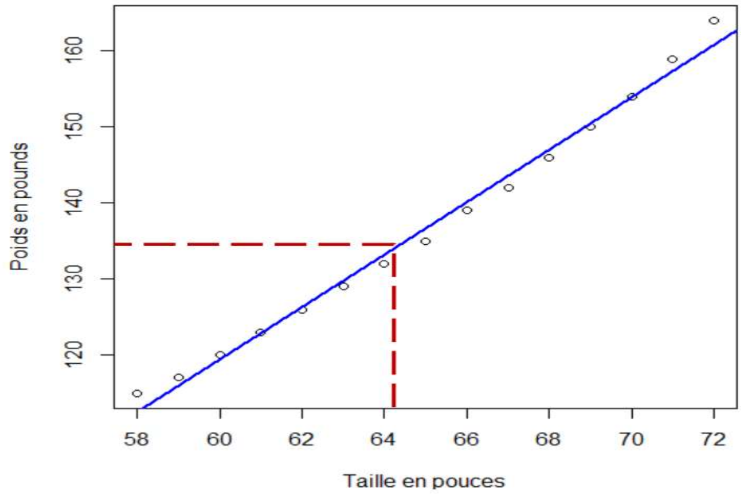
Exemple



Régression linéaire

Problématique :

Exemple



Régression linéaire

Problématique :

- On dispose d'un ensemble d'apprentissage contenant m observations pour lesquelles les valeurs des variables **x** et **y** sont déjà connues.
- On cherche à prédire la valeur d'une variable continue **y** en fonction d'une variable continue **x** : $y = f(x) = a\ x + b$

Régression linéaire

Problématique :

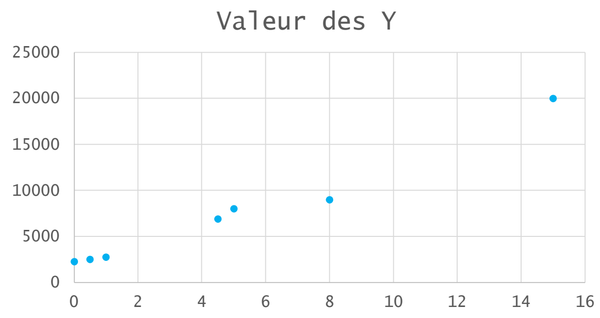
- On dispose d'un ensemble d'apprentissage contenant m observations pour lesquelles les valeurs des variables **x** et **y** sont déjà connues.
- On cherche à prédire la valeur d'une variable continue **y** en fonction d'une variable continue **x** : y =f(x) = a x + b

X	Y
0.5	2500
0	2250
1	2750
5	8000
8	9000
4.5	6900
15	20000

Régression linéaire

Problématique :

- On dispose d'un ensemble d'apprentissage contenant m observations pour lesquelles les valeurs des variables **x** et **y** sont déjà connues.
- On cherche à prédire la valeur d'une variable continue **y** en fonction d'une variable continue **x** : y =f(x) = a x + b

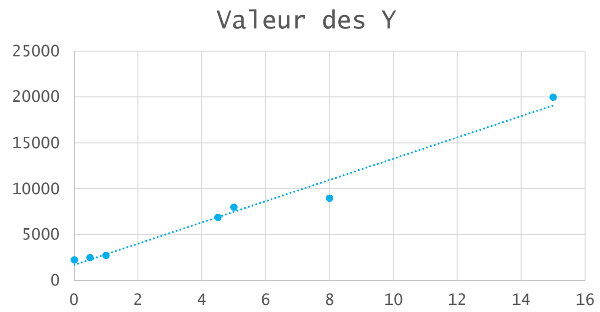


X	Y
0.5	2500
0	2250
1	2750
5	8000
8	9000
4.5	6900
15	20000

Régression linéaire

Problématique :

- On dispose d'un ensemble d'apprentissage contenant m observations pour lesquelles les valeurs des variables **x** et **y** sont déjà connues.
- On cherche à prédire la valeur d'une variable continue **y** en fonction d'une variable continue **x** : **y =f(x) = a x + b**

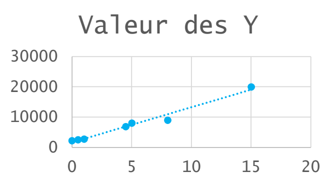


x	y
0.5	2500
0	2250
1	2750
5	8000
8	9000
4.5	6900
15	20000

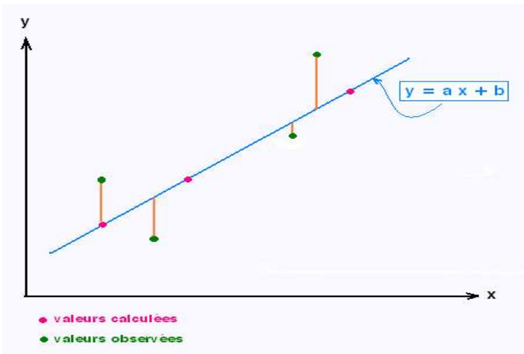
Régression linéaire

Problématique :

- On dispose d'un ensemble d'apprentissage contenant m observations pour lesquelles les valeurs des variables **x** et **y** sont déjà connues.
- On cherche à prédire la valeur d'une variable continue **y** en fonction d'une variable continue **x** : **y =f(x) = a x + b**



x	y
0.5	2500
0	2250
1	2750
5	8000
8	9000
4.5	6900
15	20000

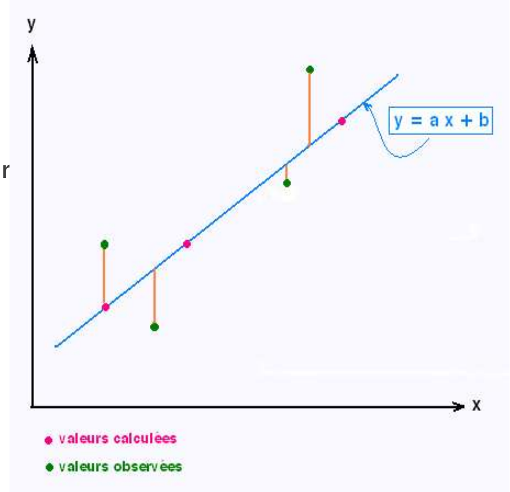


Régression linéaire

Fonction hypothèse:

- Définir la fonction hypothèse h : $h_\theta = \theta_0 + \theta_1 . x$ où θ_0 et θ_1 sont les paramètres du modèle régressif.
- **Objectif :**

  $h_\theta(\mathbf{x})$ doit être une bonne approximation de la valeur réelle de **y**.
- **Comment procéder?**



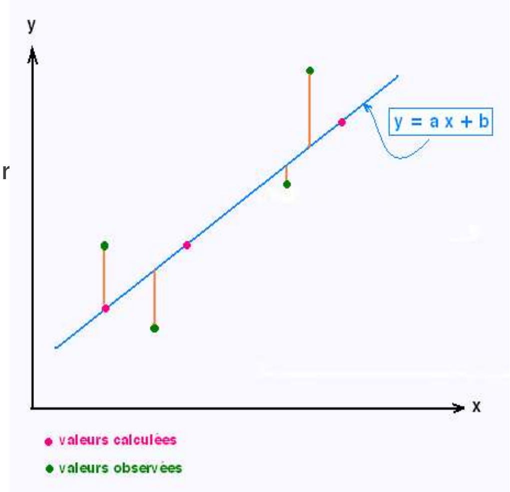
Régression linéaire

Fonction hypothèse:

- Définir la fonction hypothèse h : $h_\theta = \theta_0 + \theta_1 . x$ où θ_0 et θ_1 sont les paramètres du modèle régressif.
- **Objectif :**

  $h_\theta(\mathbf{x})$ doit être une bonne approximation de la valeur réelle de **y**.
- **Comment procéder?**

  - Minimiser les écarts entre la droite de régression et la valeur réelle.



Régression linéaire

Fonction hypothèse:

- Définir la fonction hypothèse h :

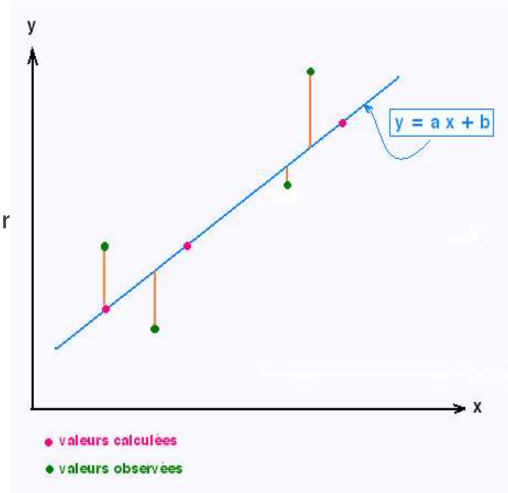
  $h_\theta = \theta_0 + \theta_1 x$ où θ_0 et θ_1 sont les paramètres du modèle régressif.

- **Objectif :**

  $h_\theta(\mathbf{x})$ doit être une bonne approximation de la valeur réelle de **y**.

- **Comment procéder?**
  - Minimiser les écarts entre la droite de régression et la valeur réelle (Erreurs quadratiques):

    Minimiser $\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2$ → **Fonction coût J**



Régression linéaire

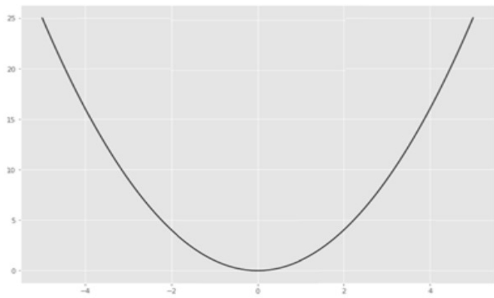
Fonction coût:

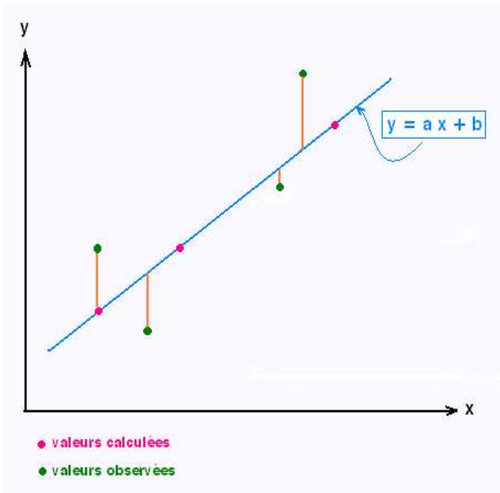
- Définir la fonction coût **J** :

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

- Minimiser J(θ_0,θ_1) :

θ_0=0





Régression linéaire

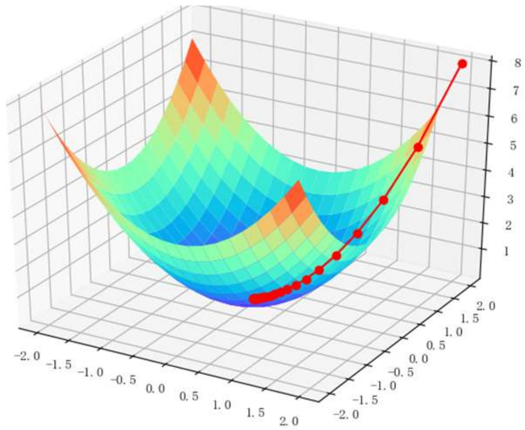
Fonction coût:

- Définir la fonction coût **J** :

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

- Minimiser J(θ_0, θ_1) :

  La descente du gradient (Gradient Descent)



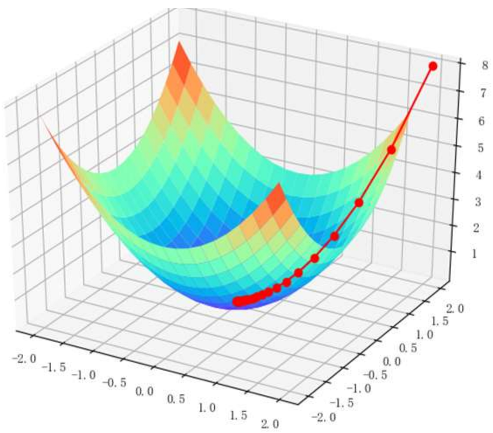
Régression linéaire

Descente du gradient:

- Algorithme :

Initialiser avec θ_j au hasard (j = 0, 1)
Répéter
$\theta_0 \leftarrow \theta_0 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_0}$
$\theta_1 \leftarrow \theta_1 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_1}$
Jusqu'à convergence

- α est le taux d'apprentissage ($\alpha \in]0, 1]$)
  - Si trop petit, lent à converger
  - Si trop grand, on risque de manquer la solution optimale.



Régression linéaire

Descente du gradient:

- Algorithme :

Initialiser avec θ_j au hasard (j = 0, 1)

Répéter

$$\theta_0 \leftarrow \theta_0 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_0}$$

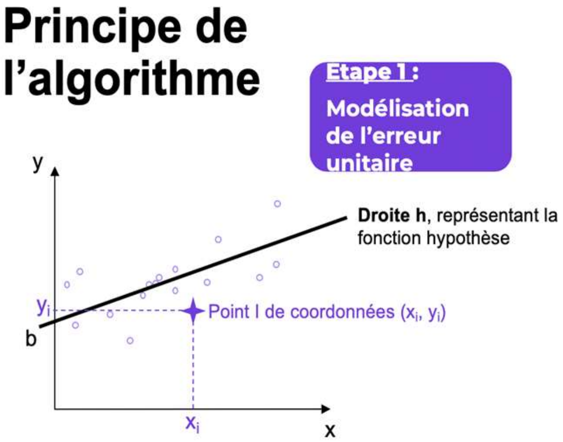
$$\theta_1 \leftarrow \theta_1 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_1}$$

Jusqu'à convergence

Avec :

$$\frac{\partial J(\theta_0,\theta_1)}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$\frac{\partial J(\theta_0,\theta_1)}{\partial \theta_1} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x^{(i)}$$



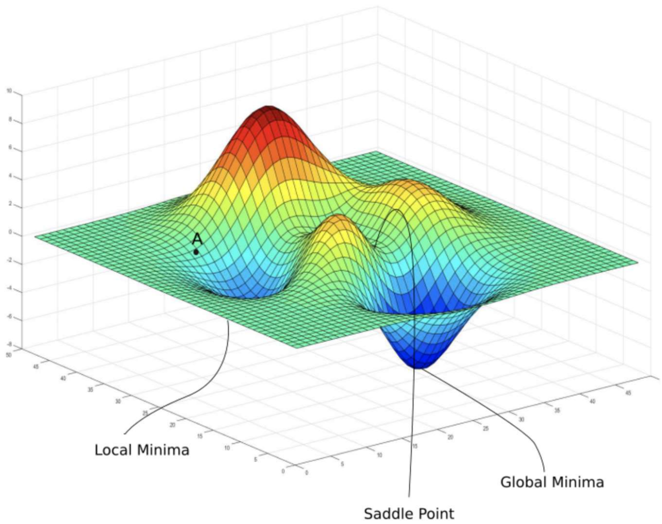
Régression linéaire

Descente du gradient:

Algorithme :

- Il est sensible aux minimums locaux.

https://www.machinelearningworks.com/tutorials/gradient-descent

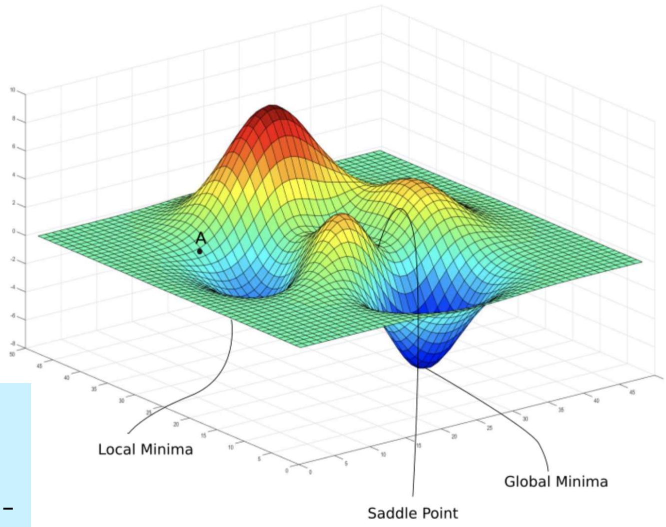


Régression linéaire

Descente du gradient:

Algorithme :

- Il est sensible aux minimums locaux.
- **Mais** la fonction coût associée à la régression linéaire a un seul minimum, minimum global.



Plus d'infos:

- https://machinelearnia.com/regression-lineaire-simple/
- https://medium.com/analytics-vidhya/journey-of-gradient-descent-from-local-to-global-c851eba3d367

Régression linéaire

Descente du gradient:

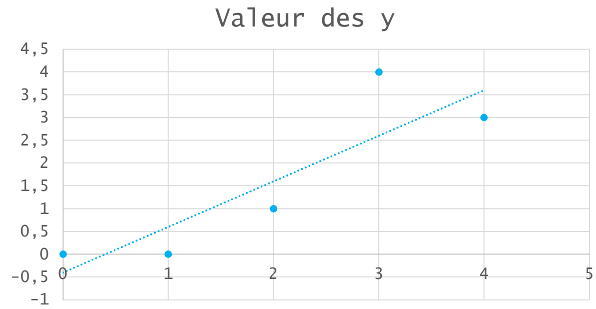
Exercice : Nous voulons prédire le nombre de modules non validés

en S2 (y) en fonction du nombre de modules non validés

en S1 (x)

1. Pour $\theta_0 = 2{,}5$ et $\theta_1 = 0.5$, quelle est la valeur de $h_\theta(4)$?
2. Calculer J(0,1) J(3,1) J(5,4)

	x(S1)	y(S2)
1	3	4
2	0	0
3	1	0
4	4	3
5	2	1



Régression linéaire

Descente du gradient:

<u>Exercice</u> : Nous voulons prédire le nombre de modules non validés

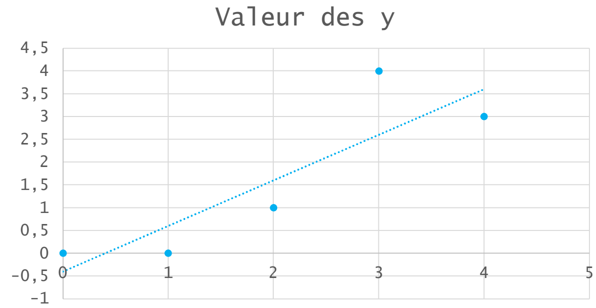
en S2 (y) en fonction du nombre de modules non validés

en S1 (x)

1. Pour $\boldsymbol{\theta_0} = 2.5$ et $\boldsymbol{\theta_1} = 0.5$, quelle est la valeur de $h_\theta(4)$?

$h_\theta(4) = \boldsymbol{\theta_0} + \boldsymbol{\theta_1}x$

	x(S1)	y(S2)
1	3	4
2	0	0
3	1	0
4	4	3
5	2	1



Régression linéaire

Descente du gradient:

<u>Exercice</u> : Nous voulons prédire le nombre de modules non validés

en S2 (y) en fonction du nombre de modules non validés

en S1 (x)

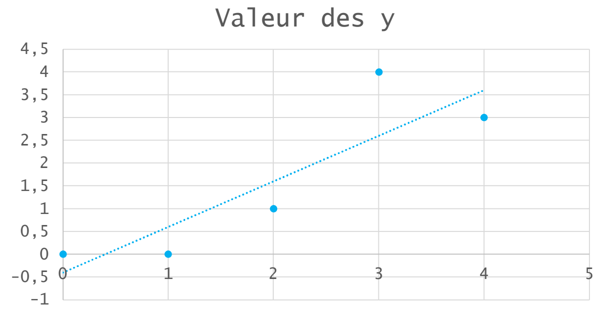
1. Pour $\boldsymbol{\theta_0} = 2.5$ et $\boldsymbol{\theta_1} = 0.5$, quelle est la valeur de $h_\theta(4)$?

$h_\theta(4) = \boldsymbol{\theta_0} + \boldsymbol{\theta_1}x$

$= 2.5 + 0.5*4$

$= 2.5 + 2 = 4.5$

	x(S1)	y(S2)
1	3	4
2	0	0
3	1	0
4	4	3
5	2	1



Régression linéaire

Descente du gradient:

Exercice : Nous voulons prédire le nombre de modules non validés

en S2 (y) en fonction du nombre de modules non validés

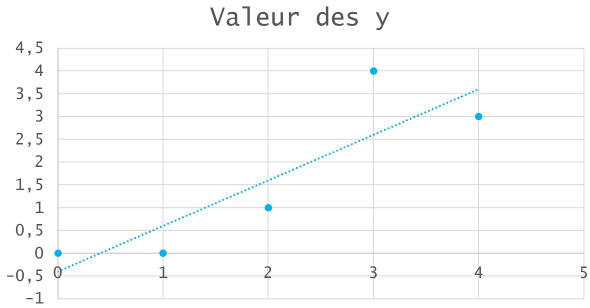
en S1 (x)

	x(S1)	y(S2)
1	3	4
2	0	0
3	1	0
4	4	3
5	2	1

Calculer J(0,1) J(3,1) J(5,4)

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

J(0,1) =



Régression linéaire

Descente du gradient:

Exercice : Nous voulons prédire le nombre de modules non validés

en S2 (y) en fonction du nombre de modules non validés

en S1 (x)

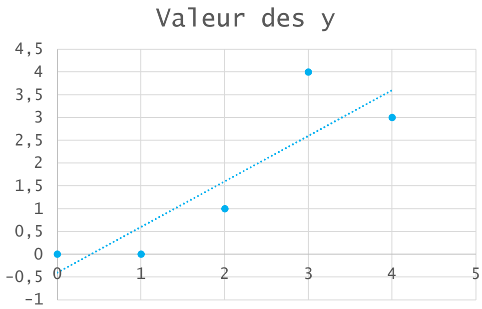
	x(S1)	y(S2)
1	3	4
2	0	0
3	1	0
4	4	3
5	2	1

Calculer J(0,1) J(3,1) J(5,4)

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

h(0,1) = x

J(0,1) = $\frac{1}{2*5}((3-4)^2+(1-0)^2+(4-3)^2+(2-1)^2)$



Régression linéaire

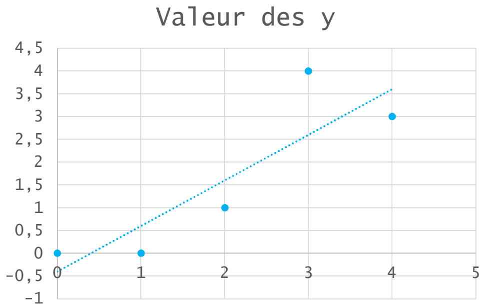
Descente du gradient:

Exercice : Nous voulons prédire le nombre de modules non validés

en S2 (y) en fonction du nombre de modules non validés

en S1 (x)

	x(S1)	y(S2)
1	3	4
2	0	0
3	1	0
4	4	3
5	2	1



Calculer J(0,1) J(3,1) J(5,4)

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

h(3,1) =

J(3,1) =

Régression linéaire

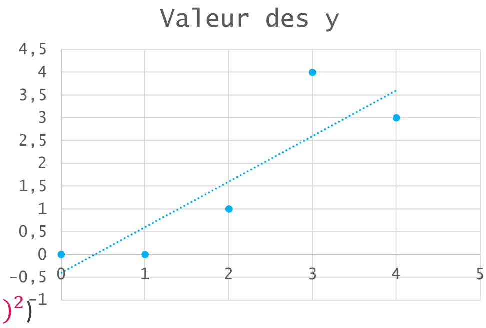
Descente du gradient:

Exercice : Nous voulons prédire le nombre de modules non validés

en S2 (y) en fonction du nombre de modules non validés

en S1 (x)

	x(S1)	y(S2)
1	3	4
2	0	0
3	1	0
4	4	3
5	2	1



Calculer J(0,1) J(3,1) J(5,4)

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

h(3,1) = 3+x

J(3,1) = $\frac{1}{2*5}((6-4)^2 + (3-0)^2 + (4-0)^2 + (7-3)^2 + (5-1)^2)$

Régression linéaire multiple

Problématique

En général, la variable à prédire y dépend de plusieurs variables x_1, x_2, …, x_n.

- $x_j^{(i)}$ valeur de la je variable pour la ie observation
- $y^{(i)}$ valeur de la variable y pour la ie observation
- $x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix}$ avec $x_0^{(i)} = 1\ pour\ i = 1 \ldots \mathrm{m}$

Régression linéaire multiple

Fonction hypothèse

La fonction hypothèse h devient :

$$\boldsymbol{h_\theta(x) = \ \theta_0 + \theta_1\, x_1 + \theta_2\, x_2 + \ldots + \theta_n\, x_n}$$

où $\boldsymbol{\theta_j}$ (j=1…n) sont les paramètres du modèle

<u>**Représentation matricielle :**</u>

$\theta \quad = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$ $\qquad X \quad = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$ avec $\boldsymbol{x_0 = 1}$

$$\boldsymbol{h_\theta(x) = \theta^T X}$$

Régression linéaire multiple

Fonction coût : Pour minimiser la fonction coût :

Descente du gradient	Équation normale
Itératif	Non itérative
Nécessité de centrer-réduire les variables	Pas nécessaire de centrer-réduire les variables
Nécessité de choisir le taux d'apprentissage α	Pas de paramètre à choisir
Performant quand le nombre de variables est grand	Lent quand le nombre de variables est grand
–	X^TX n'est pas toujours inversible (quand certaines variables explicatives sont linéairement corrélées entre elle ou quand on a trop de variables explicatives)

Régression logistique

La régression logistique est utilisée pour le classement et pas la régression. Mais, elle est considéré comme une méthode de régression puisqu'elle sert à estimer la probabilité d'appartenir à une classe.

Il y a trois types de régression logistique:

- **Régression logistique binaire:** ici, le but de la classification est d'identifier si un échantillon appartient à une classe ou non.
- **Régression logistique multinomiale:** ici, le but de la classification est d'identifier à quelle classe appartient-t-il un échantillon parmi plusieurs classes.
- **Régression logistique ordinale:** ici, le but de la classification est de chercher la classe d'un échantillon parmi des classes ordonnées. Un exemple de classes: non satisfait, satisfait, très satisfait.

Régression logistique binaire

Fonction hypothèse :

Appliquer une transformation sur la fonction hypothèse définie pour la régression linéaire pour ramener ses valeurs à l'intervalle [0,1]

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \qquad X = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \text{ avec } \boldsymbol{x_0 = 1}$$

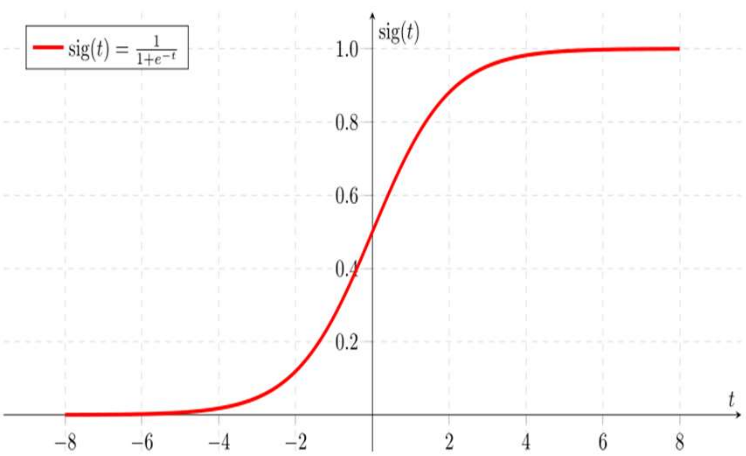
$$\boldsymbol{h_\theta(x) = g(\theta^T X)}$$

La fonction g est la fonction logistique :

$$\boldsymbol{g(u) = \frac{1}{1+e^{-u}}}$$

Régression logistique binaire

Décision de la classe



On dit qu'un échantillon x appartient à une classe donnée (classe positive) y=1 :

- On utilise un seuil (0.5 généralement)
- Si p(y=1 | x) >= 0.5 donc classe positive
- Sinon classe négative

Régression logistique multimodales

Décision de la classe

Pour n classes, définir n fonctions hypothèses

$$h_\theta^{(k)}(x) = P(y = k|x;\ \theta)\ avec\ k = 1 \dots n$$

→ Prédiction: **y** appartient à la classe **k** qui maximise :

$$h_\theta^{(k)}(x)$$

On utilise la méthode de « **un-contre-le-reste** » « **One-vs-all** ».

Régression

Avantages

La régression linéaire

- Simple à comprendre et à expliquer
- Utile pour l'analyse des données

La régression polynomiale

- Fournit une bonne approximation de la relation entre la variable dépendante y et la variable indépendante x.

La régression logistique

- Poutant elle est utilisée pour la classification, elle donne des probabilités pour les sorties.
- Le modèle logistique peut être mis à jour facilement.

Régression

Limites

La régression linéaire

- Elle donne des mauvaises performances s'il n'y a pas une relation linéaire.
- La plupart des phénomènes réelles ne correspondent pas la supposition du modèle linéaire.
- Sensible aux valeurs aberrantes

La régression polynomiale

- Les mêmes limites que la régression linéaire

La régression logistique

- Donne une mauvaise performance lorsqu'il existe plusieurs limites de décision ou des limites de décision non linéaires.

Travaux pratiques 1

- Workflow de Scikit-learn / Keras
- Régression linéaire/Régression polynomiale
- Régression logistique / kNN
- Naïve Bayes
- Arbres de décision
- SVM