

Programmation Python pour Intelligence Artificielle

Apprentissage Automatique

Année universitaire: 2024/2025

Qu'est-ce que l'intelligence artificielle ?



Définition:

L'intelligence artificielle (IA) est un domaine des sciences informatiques et mathématiques qui regroupe un ensemble de méthodes et de théories visant à créer des machines capables d'imiter certaines formes d'intelligence humaine.

L'objectif principal est de permettre à ces systèmes de résoudre des problèmes complexes, à l'image de la pensée humaine.

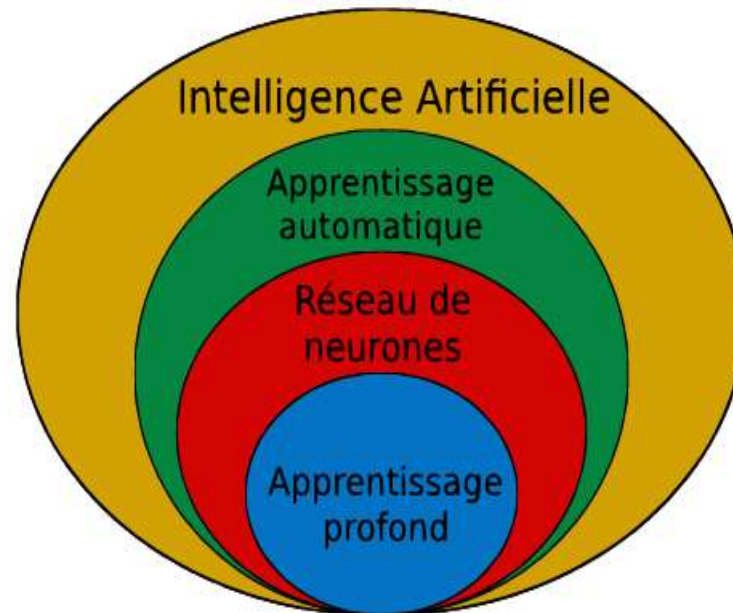
Pourquoi parle-t-on d'intelligence ?

Parce que l'IA permet à des ordinateurs ou à des robots de **prendre des décisions intelligentes** : par exemple, reconnaître une voix, comprendre un texte, ou conduire une voiture.



L'intelligence artificielle touche de nombreux champs : **probabilités, neurosciences, robotique, théorie des jeux, médecine, transports**, et bien d'autres encore.

Le domaine de l'intelligence artificielle se compose de plusieurs sous-domaines imbriqués.





L'apprentissage automatique (ou *Machine Learning*) est un sous-domaine de l'intelligence artificielle consistant à **extraire automatiquement des règles à partir de données ou d'expériences** pour résoudre un problème donné.

Les réseaux de neurones sont l'un des modèles les plus connus et les plus utilisés.

L'apprentissage profond (*Deep Learning*) est une branche du Machine Learning basée sur des réseaux de neurones complexes (profonds).

Introduction au Machine Learning:



Définition générale:

Le **Machine Learning (ML)**, ou **apprentissage automatique**, est une sous-discipline de l'intelligence artificielle (IA) qui se concentre sur la conception et le développement d'algorithmes capables d'apprendre à partir de données. Plutôt que d'être explicitement programmés pour effectuer une tâche, les systèmes de ML apprennent à partir de l'expérience (les données) pour faire des prédictions ou prendre des décisions.



Objectif

L'objectif principal du ML est de **modéliser automatiquement des comportements ou des relations complexes dans les données**, dans le but de :

- Prédire des résultats futurs (ex : prix d'un appartement),
- Classifier des objets ou événements (ex : reconnaître des visages ou des emails spam),
- Trouver des structures cachées (ex : regroupement de clients en marketing),
- Automatiser des tâches complexes (ex : conduite autonome, traduction automatique).

Mais comment apprendre ?



Pour donner à un ordinateur la capacité d'**apprendre**, on utilise des **méthodes d'apprentissage** qui sont fortement inspirées de la façon dont nous, les êtres humains, apprenons à faire des choses.

Parmi ces méthodes, on compte :

- L'apprentissage supervisé (Supervised Learning)
- L'apprentissage non supervisé (Unsupervised Learning)
- L'apprentissage par renforcement (Reinforcement Learning)

L'Apprentissage Supervisé



Pour maîtriser **l'apprentissage supervisé**, il faut absolument comprendre et connaître les 4 notions suivantes :

- Le Dataset
- Le Modèle et ses paramètres
- La Fonction Coût
- L'Algorithme d'apprentissage

Notion 1 : Le Dataset



On parle d'**apprentissage supervisé** lorsque l'on fournit à une machine beaucoup d'**exemples (x, y)** dans le but de lui faire apprendre la **relation** qui relie x à y .

En Machine Learning, on compile ces exemples (x, y) dans un tableau que l'on appelle Dataset :

- La variable y porte le nom de **target** (la cible). C'est la valeur que l'on cherche à prédire.
- La variable x porte le nom de **feature** (facteur). Un facteur influence la valeur de y , et on a en général beaucoup de **features (x_1, x_2, \dots)** dans notre Dataset que l'on regroupe dans une matrice X .



Exemple: Le Dataset

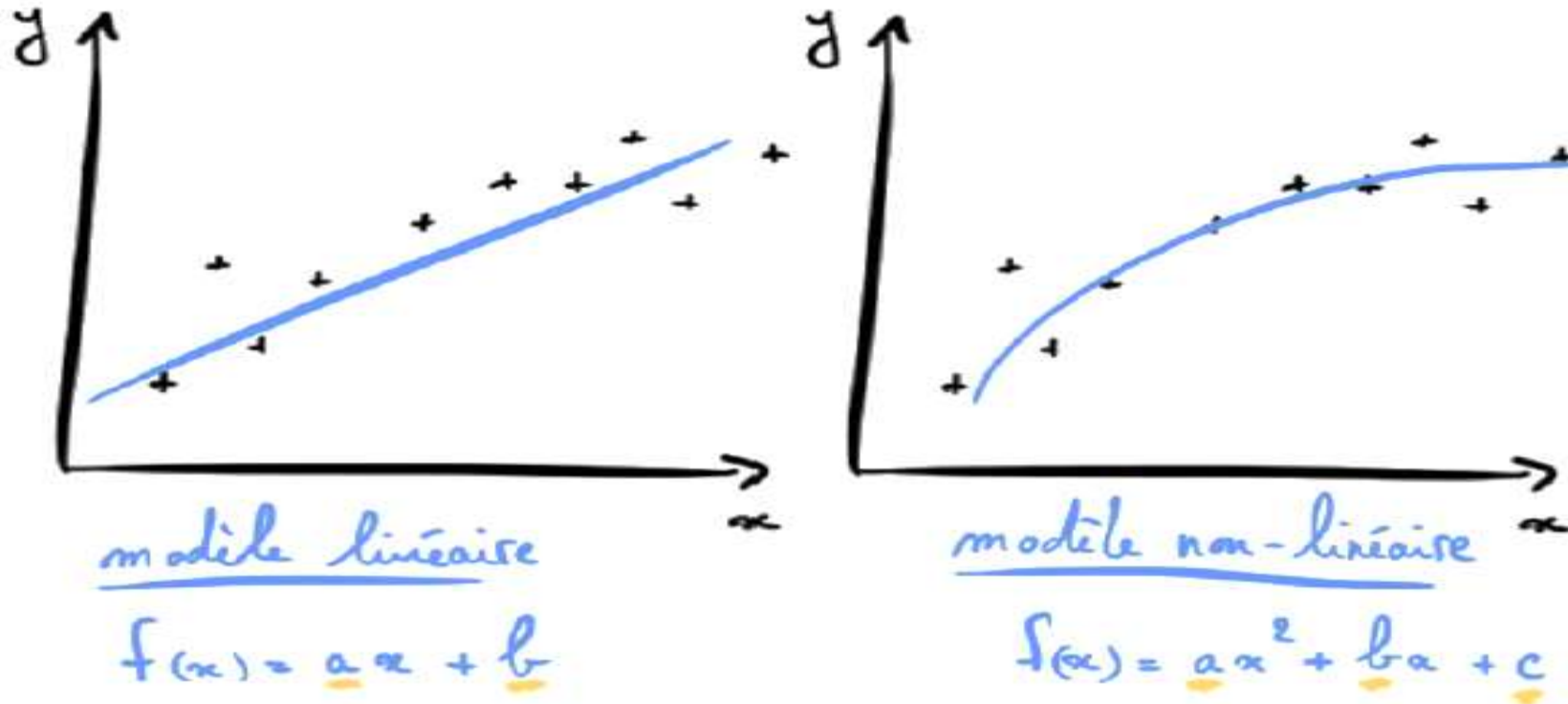
Ce Dataset regroupe des exemples d'appartements avec leur prix y ainsi que certaines de leurs caractéristiques (features).

Target y	x_1	x_2	x_3
Prix	Surface m2	N chambres	Qualité
€ 313,000.00	124	3	1.5
€ 2,384,000.00	339	5	2.5
€ 342,000.00	179	3	2
€ 420,000.00	186	3	2.25
€ 550,000.00	180	4	2.5
€ 490,000.00	82	2	1
€ 335,000.00	125	2	2



Notion 2 : Le Modèle et ses paramètres

En Machine Learning, on développe un modèle à partir du Dataset. Il peut s'agir d'un modèle linéaire, ou bien un modèle non-linéaire.

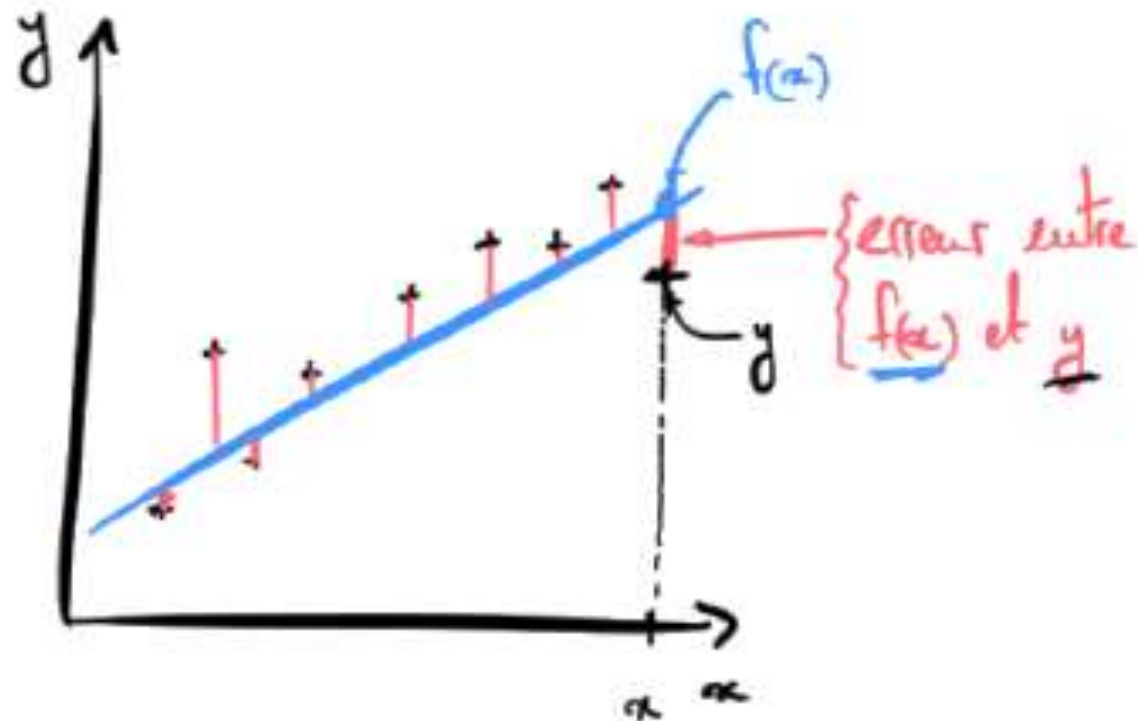


On définit a, b, c , etc. comme étant les paramètres d'un modèle.



Notion 3 : La Fonction Coût

Lorsqu'on teste notre modèle sur le Dataset, celui-ci nous donne des erreurs. L'ensemble de ces erreurs, c'est Fonction Coût.

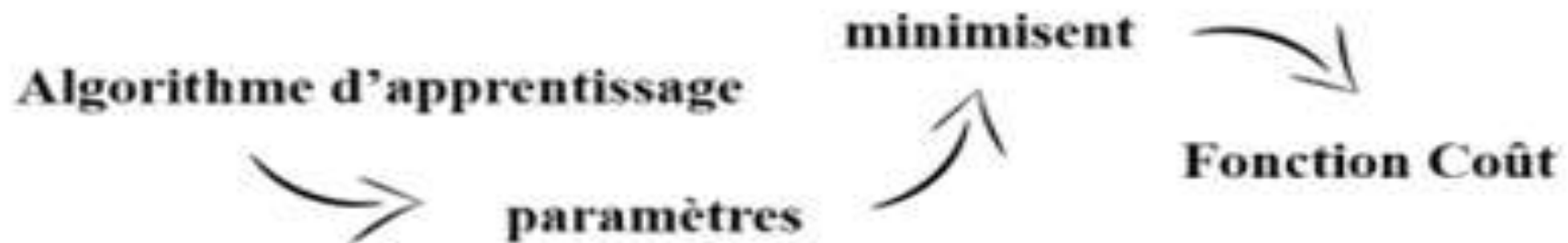


Fonction Coût = l'ensemble des erreurs.

Notion 4 : L'Algorithme d'apprentissage



L'objectif central du Machine Learning, c'est de trouver les paramètres du modèle qui minimisent la Fonction Coût. Pour cela, on utilise un algorithme d'apprentissage.



L'exemple le plus courant étant l'algorithme de **Gradient Descent**.

Les applications du Machine Learning



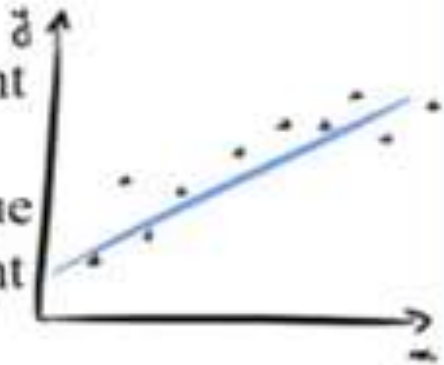
Avec le Machine Learning on peut développer des modèles pour résoudre 2 types de problèmes :

- Les problèmes de **Régression**
- Les problèmes de **Classification**

Régression

Les problèmes de **Régression** correspondent aux situations dans lesquelles la machine doit prédire la valeur d'une **variable quantitative** (variable continue)

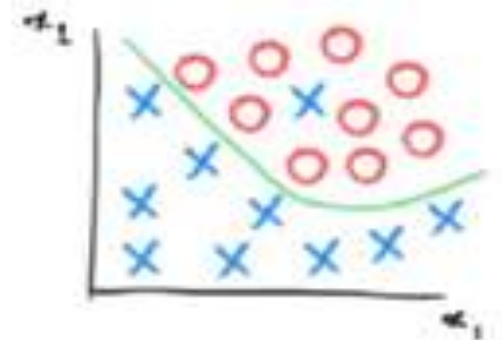
- Le prix d'un appartement
- L'évolution du climat
- Consommation électrique
- Durée de vie d'un patient



Classification

La classification correspond aux situations dans lesquelles la machine doit prédire la valeur d'une **variable qualitative** (variable discrète). Autrement dit, la machine doit classer ce qu'on lui donne dans des classes.

- Email Spam / non Spam
- Cancer / non Cancer
- Photo de Chat / Chien





La Régression Linéaire

Apprenez votre premier modèle linéaire



Voici la recette à suivre pour réaliser votre premier modèle de Machine Learning.

1. Récolter vos données

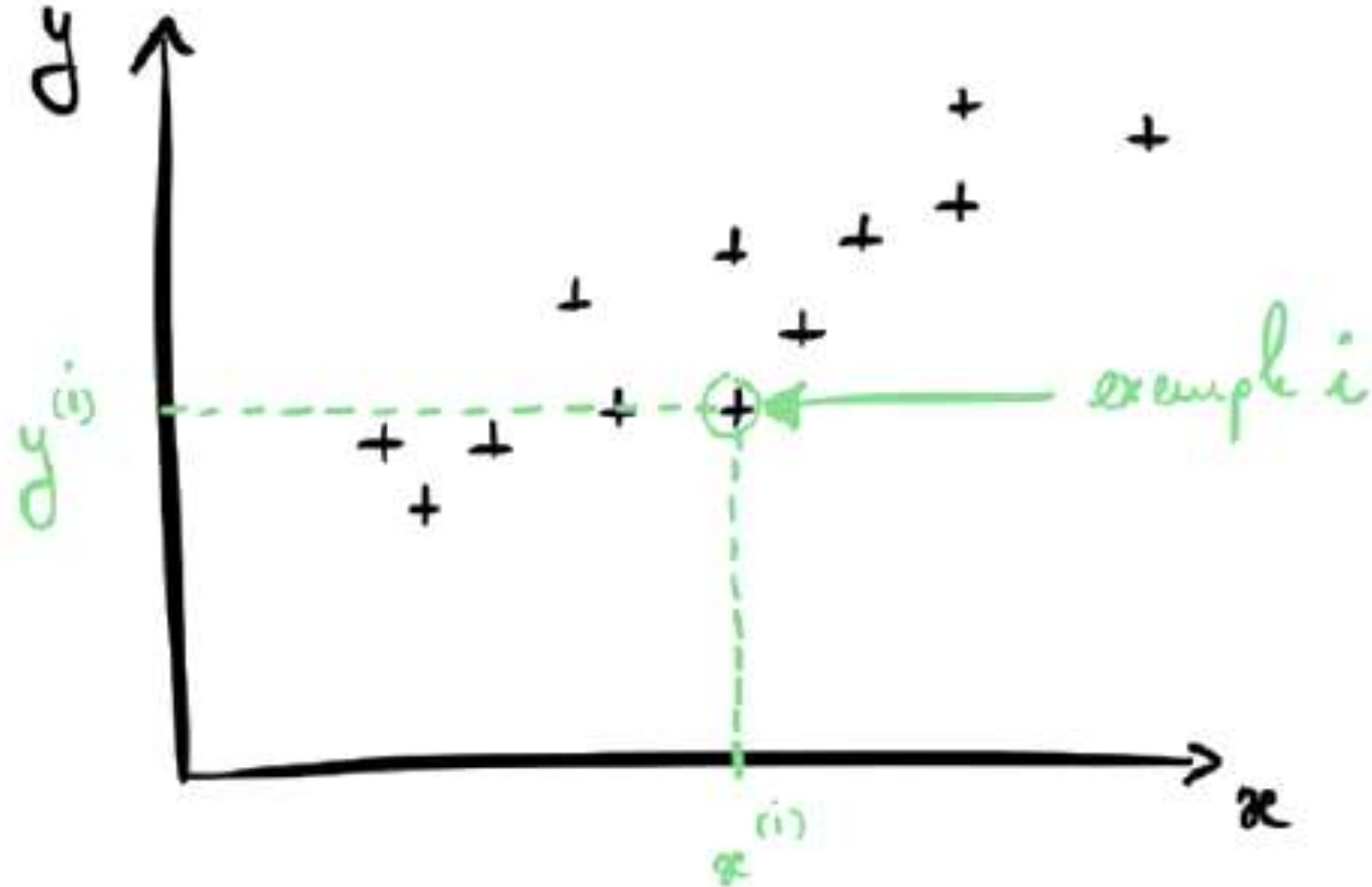
Imaginez que plusieurs agences immobilières vous aient fourni des données sur des appartements à vendre, notamment le prix de l'appartement (y) et la surface habitable (x). En Machine Learning, on dit que vous disposez de m exemples d'appartements.

On désigne :

$x^{(i)}$ la surface habitable de l'exemple i

$y^{(i)}$ le prix de l'exemple i

En visualisant votre **Dataset**, vous obtenez le nuage de points suivant :

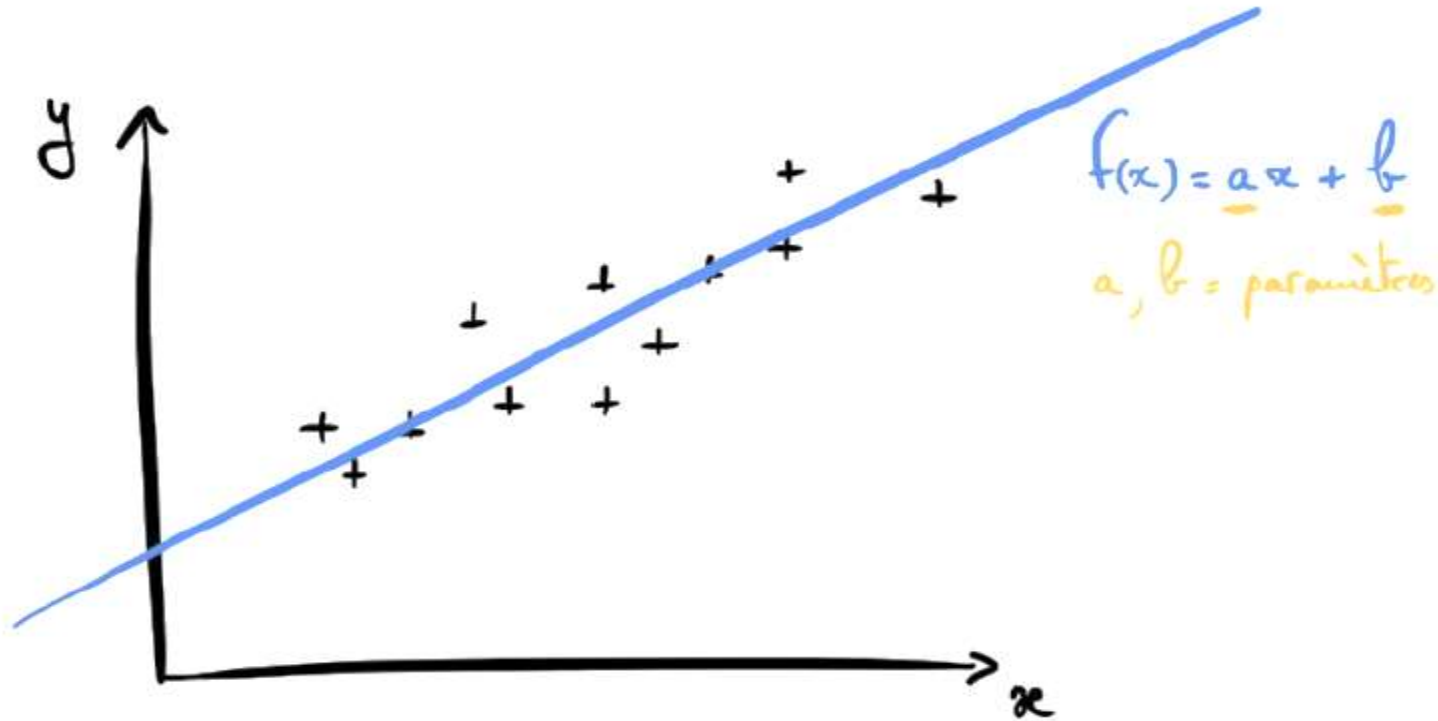




2. Créer un modèle linéaire

A partir de ces données, on développe un **modèle linéaire** $f(x)=ax+b$ où a et b sont les **paramètres** du modèle.

Un bon modèle donne de petites erreurs entre ses prédictions $f(x)$ et les exemples (y) du Dataset.



Nous ne connaissons pas les valeurs des paramètres a et b , ce sera le rôle de la machine de les trouver



3. Définir La Fonction Coût

Pour la régression linéaire, on utilise la **norme euclidienne** pour mesurer les **erreurs** entre $f(x)$ et (y) .

Concrètement, la formule pour exprimer l'erreur i entre le prix $y^{(i)}$ et la prédiction faites en utilisant la surface $x^{(i)}$: $erreur^{(i)} = (f(x^{(i)}) - y^{(i)})^2$

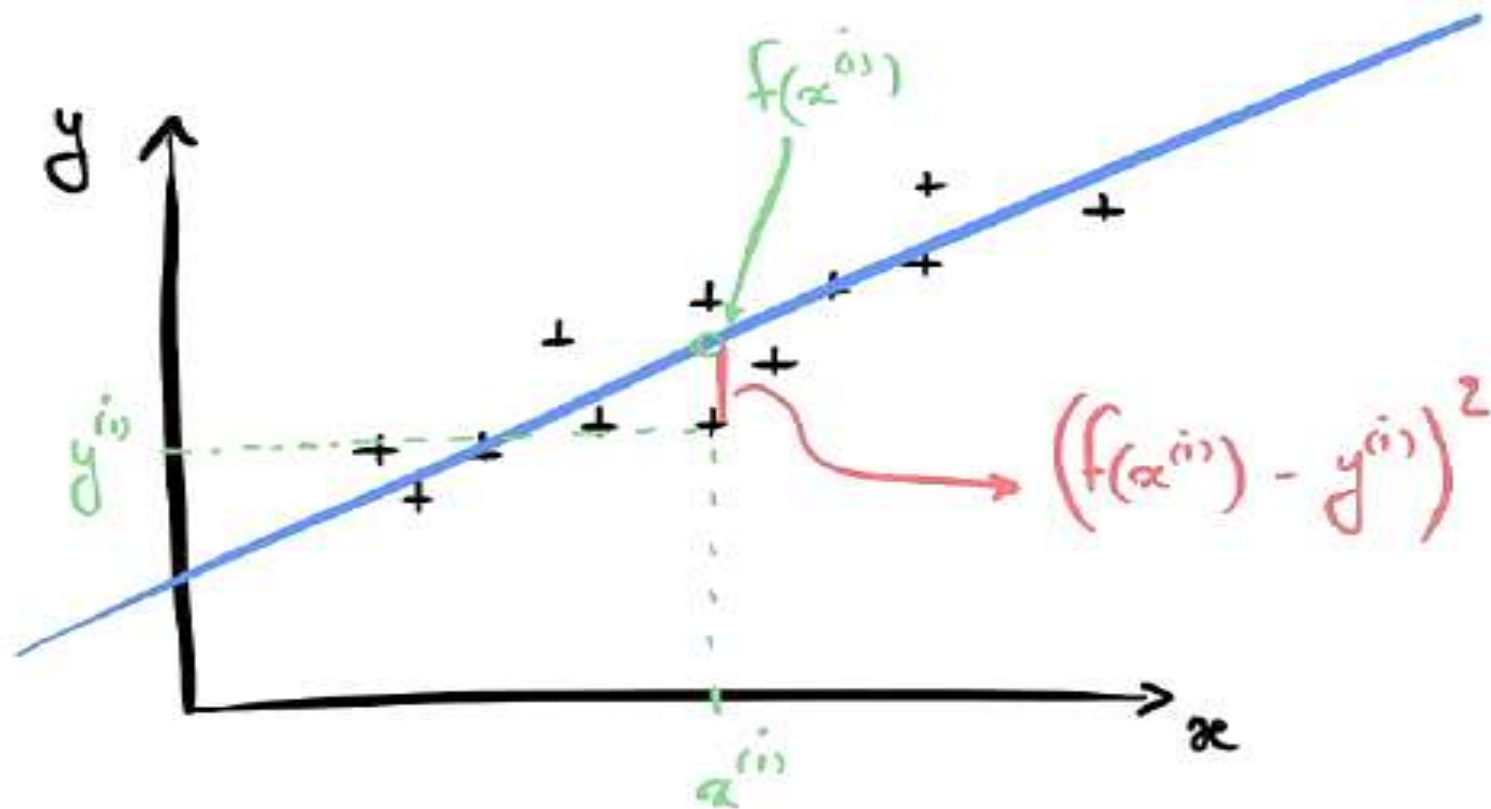
Chaque prédiction s'accompagne d'une erreur, on a donc **m erreurs**.

On définit la **Fonction Coût $J(a,b)$** comme étant la **moyenne** de toutes les erreurs :

$$J(a, b) = \frac{1}{2m} \sum_{i=1}^m erreur^i$$

$$J(a, b) = \frac{1}{2m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)})^2$$

Note : En français, cette fonction a un nom : c'est **l'erreur quadratique moyenne** (Mean Squared Error)



$$J(a, b) = \frac{1}{2m} \sum_{i=1}^m \left(\underline{a} x^{(i)} + \underline{b} - y^{(i)} \right)^2$$

Fonction Coût.



4. Trouver les paramètres qui minimisent la Fonction Coût

La prochaine étape est l'étape la plus excitante, il s'agit de laisser la machine apprendre quels sont les paramètres qui minimisent la Fonction Coût, c'est-à-dire les paramètres qui nous donnent le meilleur modèle.

Pour trouver le minimum, on utilise un algorithme d'optimisation qui s'appelle **Gradient Descent** (la descente de gradient).

Cet algorithme vous permet de trouver le **minimum** de la Fonction Coût $J(a, b)$ (le point le plus bas de la courbe) en partant de coordonnées a et b aléatoires :

1. Calculer la **pente** de la Fonction Coût, c'est-à-dire la **dérivée** de $J(a, b)$.
2. **Evoluer** d'une certaine **distance** \propto dans la direction de la pente la plus forte. Cela a pour résultat de modifier les paramètres a et b
3. Recommencer les étapes 1 et 2 jusqu'à atteindre le minimum de $J(a, b)$.



Répéter en boucle:

$$a = a - \alpha \frac{\partial J(a, b)}{\partial a}$$

$$b = b - \alpha \frac{\partial J(a, b)}{\partial b}$$

Je vous explique : à chaque itération de cette boucle, les paramètres a et b sont mis à jour en **soustrayant** leur propre valeur à la valeur de la **pente** $\frac{\partial J(a, b)}{\partial \dots}$ multipliée par la distance à parcourir α . On appelle α la vitesse d'apprentissage (**Learning rate**).

