# PROSIT 2

Intelligence Artificielle

LUNDI 6 MARS 2023

CESI
Mathilde Ballouhey

# Table des matières

# 1. Contexte

On cherche à prédire le prix de l'immobilier et à trouver les paramètres nécessaires.

# 2. Mots clés

## 2.1. Descente de gradient

Un algorithme d'optimisation utilisé pour minimiser la fonction de coût d'un modèle en ajustant les poids et les biais du modèle de manière itérative.

## 2.2. Block group

Un ensemble de données de recensement qui correspond à une petite zone géographique (par exemple, un bloc de rue), généralement utilisé pour agréger les données démographiques et socio-économiques.

## 2.3. Sur-apprentissage

Un phénomène qui se produit lorsqu'un modèle est entraîné sur des données d'entraînement à un point tel qu'il s'adapte trop étroitement à ces données, ce qui peut réduire sa capacité à généraliser correctement aux données inconnues.

## 2.4. Indicateur de qualité

Une mesure utilisée pour évaluer la qualité ou la performance d'un modèle d'IA, telle que la précision, le rappel, la courbe ROC, etc.

## 2.5. Pipeline

Un processus de développement d'IA qui divise le processus en plusieurs étapes distinctes, chacune ayant une fonction spécifique, afin d'optimiser le processus dans son ensemble.

## 2.6. Régression linéaire

Une technique d'apprentissage automatique qui modélise la relation entre une variable dépendante et une ou plusieurs variables indépendantes en utilisant une fonction linéaire.

## 2.7. Prix médian

Le prix qui divise la distribution des données en deux parties égales, où 50% des données sont en dessous du prix médian et 50% sont au-dessus.

## 2.8. Fonction objectif

Une fonction qui définit l'objectif d'un modèle d'IA, telle que la minimisation de l'erreur de prédiction ou la maximisation de la précision.

## 2.9. Fonction de coût

Une fonction qui mesure l'écart entre les prévisions d'un modèle et les valeurs réelles, et qui est utilisée pour ajuster les poids et les biais du modèle pendant l'apprentissage.

# 3. Besoins / Problématiques

Comment déterminer des paramètres pour que la modélisation prédise le prix de l'immobilier ?

# 4. Contraintes

- Python
- Stratégie fixée (Mathématiques)

## 5. Livrables

- Workshop
- Jupyter Notebook :
    - Déterminer les fonctions coût et objectif
    - Déterminer les paramètres
    - Modéliser le prix de l'immobilier

## 6. Généralisation

Outil de prédiction de données.

## 7. Hypothèses

- Une descente de gradient permet la prédiction
- La régression linéaire permet la prédiction
- Il existe une montée de gradient

## 8. Plan d'action

- Mots-Clés
- Descente de gradient
- Régression linéaire
- Pipeline
- Python ♥ sklearn

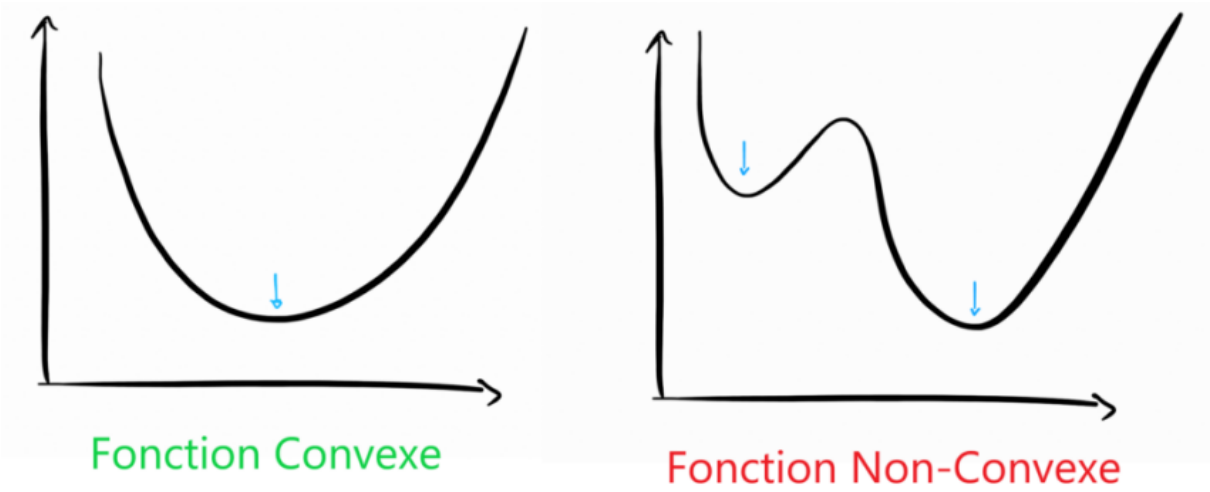
# Cours

## 1. Descente de gradient

### 1.1. Définition

La Descente de Gradient est un algorithme d'optimisation qui permet de trouver le minimum de n'importe quelle fonction convexe en convergeant progressivement vers celui-ci.

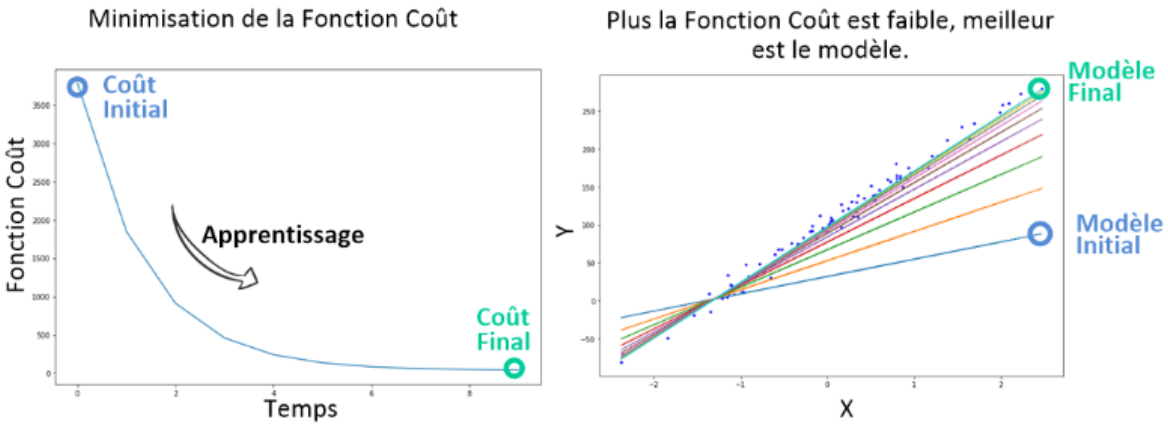
Une fonction convexe est une fonction dont l'allure ressemble à celle d'une belle vallée avec au centre un minimum global. A l'inverse, une fonction non-convexe est une fonction qui présente plusieurs minimums locaux et l'algorithme de descente de gradient ne doit pas être utilisé sur ces fonctions, au risque de se bloquer au premier minima rencontré.




### 1.2. Importance en Machine Learning

En Machine Learning, on va utiliser l'algorithme de la Descente de Gradient dans les problèmes d'apprentissage supervisé pour minimiser la fonction coût, qui justement est une fonction convexe.

C'est grâce à cet algorithme que la machine apprend, c'est-à-dire trouve le meilleur modèle. En effet, rappelez-vous que minimiser la fonction coût revient à trouver les paramètres $a$, $b$, $c$, etc. qui donnent les plus petites erreurs entre notre modèle et les points $y$ du Dataset.



## 1.3. Exemple

Voici un exemple qui permet d'expliquer la stratégie utilisée dans la descente de gradient :

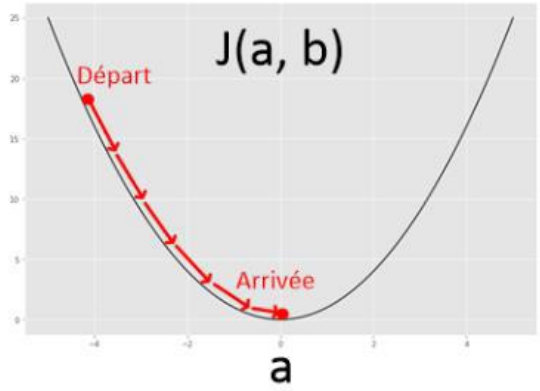
- *Imaginez que vous soyez perdu en pleine montagne. Votre but est de rejoindre un refuge situé au point le plus bas de la vallée dans laquelle vous vous situé. Le problème, c'est que vous n'avez pas pris de carte avec vous et vous ignorez donc complètement les coordonnées de ce refuge.*
- **Pour résoudre ce problème, on peut passer par une stratégie en deux étapes :**
    - Depuis votre position actuelle, vous cherchez tout autour de vous la direction de là où la pente descend le plus fort.
    - Une fois que vous avez trouvé cette direction, vous la suivez sur une certaine distance (disons que vous marchez 300 mètres) puis vous répétez l'opération de l'étape 1.
- En répétant ainsi les étapes 1 et 2 en boucle, vous êtes sûr de converger vers le minimum de la vallée.

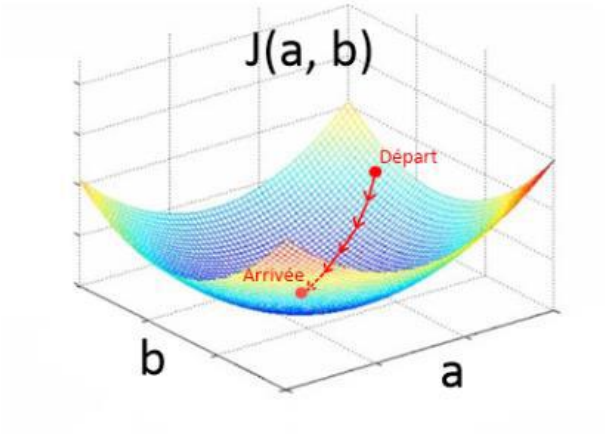
En passant de l'analogie au Machine Learning, voici ce que nous avons :

La vallée dans laquelle nous sommes est la fonction coût et nous allons répéter nos deux étapes pour pouvoir minimiser cette fonction :

- Tout d'abord, de notre point de départ, on mesure la valeur de la pente en ce point en calculant donc la dérivée de notre fonction coût.
- Puis, on progresse d'une certaine distance notée $\alpha$ dans la direction de la pente. On appelle cette distance Learning Rate ou vitesse d'apprentissage. Le fait de faire cela a pour résultat de modifier la valeur des paramètres de notre modèle (dans notre cas, ce sont les coordonnées de notre position dans la vallée.)

On se rend compte qu'avec la répétition en boucle de ces deux étapes, notre algorithme devient itératif. Si on représente la fonction coût $J(a,b)$ que l'on a développé pour une régression linéaire, on obtient cela :




En résumé, voici comment il faut percevoir les choses entre l'exemple et l'application en ML :

| **Analogie de la montagne** | **Machine Learning** |
|---|---|
| Vallée convexe | Fonction Coût |
| Pente de la vallée | Dérivée de la Fonction Coût |
| Distance parcourue (300 mètres) | Learning Rate multiplié par la dérivée |

| | |
|---|---|
| Coordonnées initiales (nous sommes perdus) | Paramètres initiaux (choisis au hasard) |
| Coordonnées du refuge | Les paramètres qui donnent le meilleur modèle |

Pour savoir comment implémenter l'algorithme réellement : Article sur la descente de gradient sur Machine Learnia par Guillaume Saint-Cirgue

# 2. Régression linéaire

## 2.1. Définition

La régression linéaire est une méthode statistique utilisée pour prédire une variable continue en fonction d'une ou plusieurs variables d'entrée. Le but est de trouver la relation linéaire entre ces variables d'entrée et la variable de sortie, de sorte que nous puissions utiliser cette relation pour prédire la valeur de la variable de sortie pour de nouvelles données.

## 2.2. Exemple

Un exemple simple de régression linéaire serait de prédire le prix d'une maison en fonction de sa surface habitable. Dans ce cas, la surface habitable serait notre variable d'entrée et le prix de la maison serait notre variable de sortie. Nous pourrions collecter des données sur le prix et la surface habitable de plusieurs maisons et utiliser ces données pour créer un modèle de régression linéaire.

Voici comment cela fonctionnerait. Supposons que nous ayons les données suivantes :

- Surface habitable (en mètres carrés) : 50, 70, 90, 110, 130
- Prix (en milliers d'euros) : 100, 130, 160, 190, 220

Nous pouvons représenter ces données sur un graphique en plaçant la surface habitable sur l'axe horizontal et le prix sur l'axe vertical. Ensuite, nous pouvons dessiner une ligne qui représente la meilleure relation linéaire possible entre les deux variables. Cette ligne sera notre modèle de régression linéaire.

Pour trouver cette ligne, nous utilisons une formule mathématique appelée "équation de régression linéaire". Cette équation a la forme suivante :

$$y = mx + b$$

Où :

- $y$ est la variable de sortie (prix dans notre exemple),
- x est la variable d'entrée (surface habitable dans notre exemple),
- m est le coefficient de pente (qui représente l'augmentation ou la diminution du prix pour chaque unité de surface habitable supplémentaire),
- b est l'ordonnée à l'origine (qui représente le prix de base de la maison sans tenir compte de la surface habitable).

Pour trouver les valeurs de m et b qui donnent la meilleure ligne de régression linéaire possible, nous utilisons une méthode appelée "méthode des moindres carrés". Cette méthode consiste à minimiser la somme des carrés des écarts entre les valeurs réelles de la variable de sortie et les valeurs prédites par notre modèle de régression linéaire.

*Exemple plus claire pour la méthode des moindres carrées : La méthode des Moindres carrés par Romain GILLET*

Une fois que nous avons trouvé les valeurs de m et b qui donnent la meilleure ligne de régression linéaire possible, nous pouvons utiliser cette ligne pour prédire le prix de maisons pour lesquelles nous avons la surface habitable mais pas le prix.

Par exemple, si nous possédons une maison avec une surface habitable de 80 mètres carrés, nous pouvons utiliser notre modèle de régression linéaire pour prédire son prix :

$$y = mx + b$$

$$y = 2.09x + 20$$

$$y = 2.09 * 80 + 20$$

$$y = 178.20\, milliers\ d'euros$$

Notre modèle de régression linéaire prédit donc que cette maison coûterait environ 178 200 euros.

## 3. Pipeline

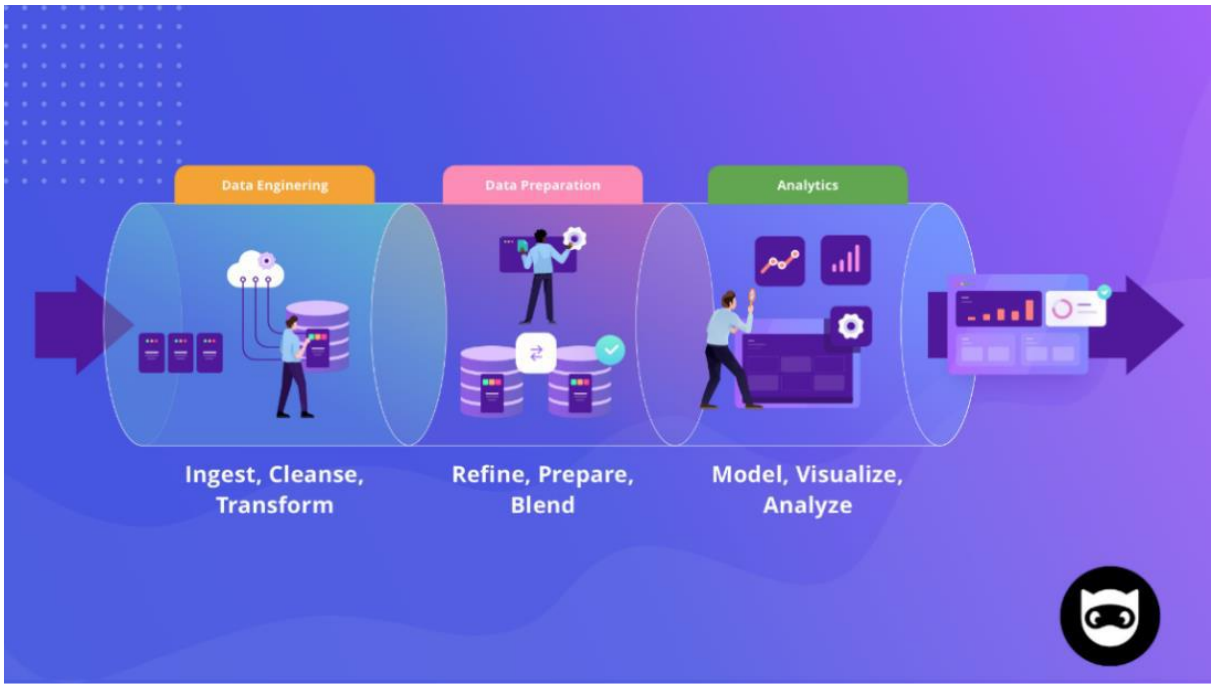
Le principe de pipeline en IA consiste à diviser le processus de développement d'un système d'IA en plusieurs étapes distinctes, appelées "étapes de traitement". Chaque étape est conçue pour accomplir une tâche spécifique et elle est généralement reliée à l'étape suivante dans un flux séquentiel ou "pipeline".

Le pipeline d'IA typique comprend les étapes de prétraitement, de modélisation et d'évaluation. La première étape, le prétraitement, implique la collecte, la sélection et la préparation des données à utiliser pour entraîner le modèle d'IA. Cette étape comprend souvent des tâches telles que la normalisation des données, la suppression des valeurs aberrantes et la transformation des données brutes en un format utilisable par le modèle.

La deuxième étape, la modélisation, implique la création et l'entraînement d'un modèle d'IA à l'aide des données préparées lors de la première étape. Cette étape comprend le choix de l'algorithme de modélisation, le réglage des paramètres de l'algorithme, l'entraînement du modèle et la vérification de sa précision.

La dernière étape, l'évaluation, consiste à tester le modèle sur des données qui n'ont pas été utilisées lors de la phase de formation. Cette étape permet de déterminer la précision du modèle et sa capacité à généraliser au-delà des données d'entraînement.

Le pipeline en IA est un processus itératif et continu qui peut être amélioré en ajoutant des étapes supplémentaires ou en ajustant les paramètres existants pour améliorer la performance globale du modèle. Le principe de pipeline permet de diviser le processus complexe de développement d'IA en tâches plus petites et plus gérables, ce qui facilite la gestion du projet et l'amélioration continue du modèle d'IA.




## 4. Python ♥ sklearn

## 5. Hypothèses

- Une descente de gradient permet la prédiction
- La régression linéaire permet la prédiction
- Il existe une montée de gradient

## 6. Livrable

## *7.* Validation des Hypothèses

# Bibliographie

**Aucune source spécifiée dans le document actif.**