Machine Leraning

Enseignante: Bènène Fradi Boumiza

Niveau : GL4

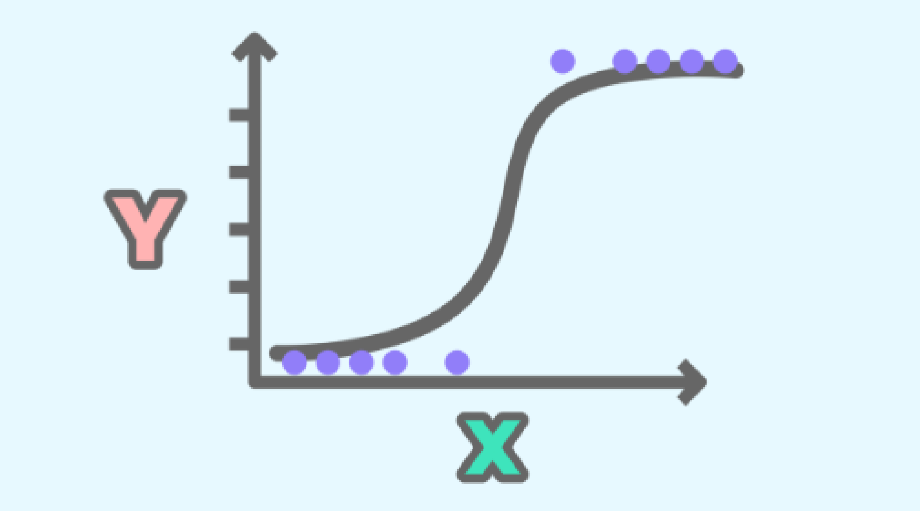
Année universitaire: 2025/2026

La Régression

❑ Principe de la Régression

La **régression** est une méthode qui permet de prédire une **valeur continue** à partir de variables explicatives.

- Contrairement à la classification (qui attribue une étiquette ou une classe), la régression cherche une **quantité numérique**.
- Exemple intuitif : prédire la **taille d'une personne** en fonction de son âge.



Caractéristiques principales :

- La sortie (Y) est une variable quantitative continue.
- On cherche à trouver une relation entre les variables d'entrée (X) et la sortie (Y).
- On mesure la qualité du modèle à l'aide d'une **fonction coût** (par ex. l'erreur quadratique moyenne, MSE).

La Régression

- La régression sert à trouver la relation d'une variable par rapport à une ou plusieurs autres. $\hat{y}(x) = f(x_1, x_2, \ldots, x_m)$
- Dans l'apprentissage automatique, le but de la régression est d'estimer une valeur (numérique) de sortie à partir des valeurs d'un ensemble de caractéristiques en entrée.
- Par exemple, estimer le prix d'une maison en se basant sur sa surface, nombre des étages, son emplacement, etc. Donc, le problème revient à estimer une fonction de calcul en se basant sur des données d'entrainement.

La Régression

❑ Étapes principales d'un algorithme de régression

1. **Formuler une hypothèse (modèle)**
   - Exemple linéaire :

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- $\hat{y}$ = prédiction
- x_j = variables explicatives
- θ_j = paramètres/poids à apprendre

2. **Définir une fonction de coût**
   - Mesure l'écart entre les valeurs prédites et réelles .
   - Exemple : MSE (Erreur Quadratique Moyenne).
3. **Optimiser les paramètres**
   - Utiliser un algorithme comme la **descente de gradient** pour ajuster les θj et **minimiser la fonction de coût**.
4. **Évaluer le modèle**
   - Vérifier la qualité des prédictions avec des métriques (MSE, RMSE, R^2, etc.).

La Régression

❑ Formuler une hypothèse : Modèles de Régression

❑ Régression linéaire :

But : établir un lien entre une variable dépendante Y et une variable indépendante X pour pouvoir ensuite faire des prévisions sur Y lorsque X est mesurée.

Exemple 1

L'analyse de la température de fonctionnement d'un procédé chimique sur le rendement du produit a donné les valeurs suivantes pour la température X_i et le rendement correspondant Y_i :

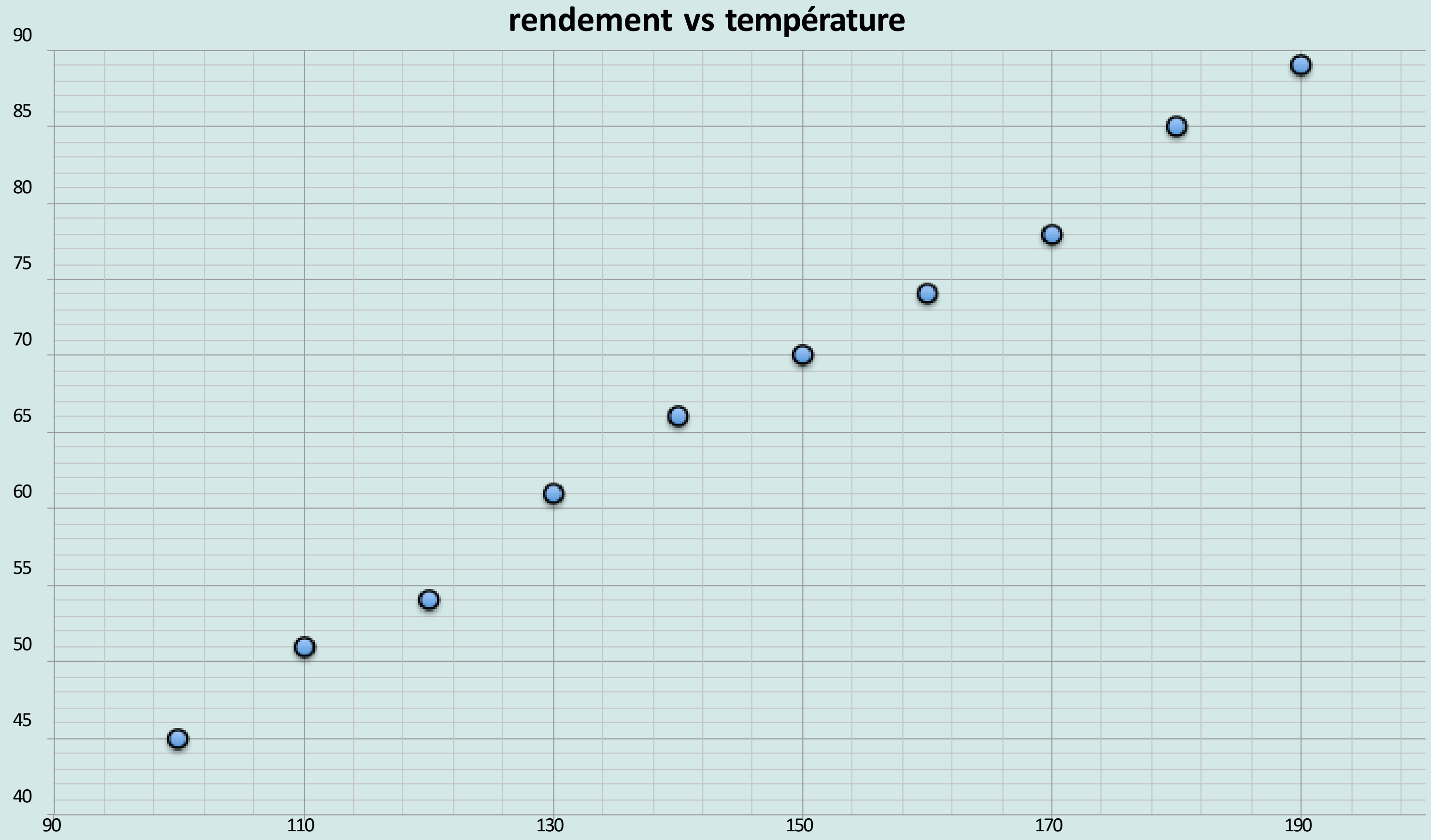
Température °C	Rendement %	Température °C	Rendement %
100	45	150	70
110	51	160	74
120	54	170	78
130	61	180	85
140	66	190	89

La Régression

❑ Formuler une hypothèse : Modèles de Régression

❑ Régression linéaire : Exemple 1 (suite)

Le graphe ci-dessous représente les points $(X_i,\ Y_i)$ pour ces données et suggère une relation linéaire entre X et Y .

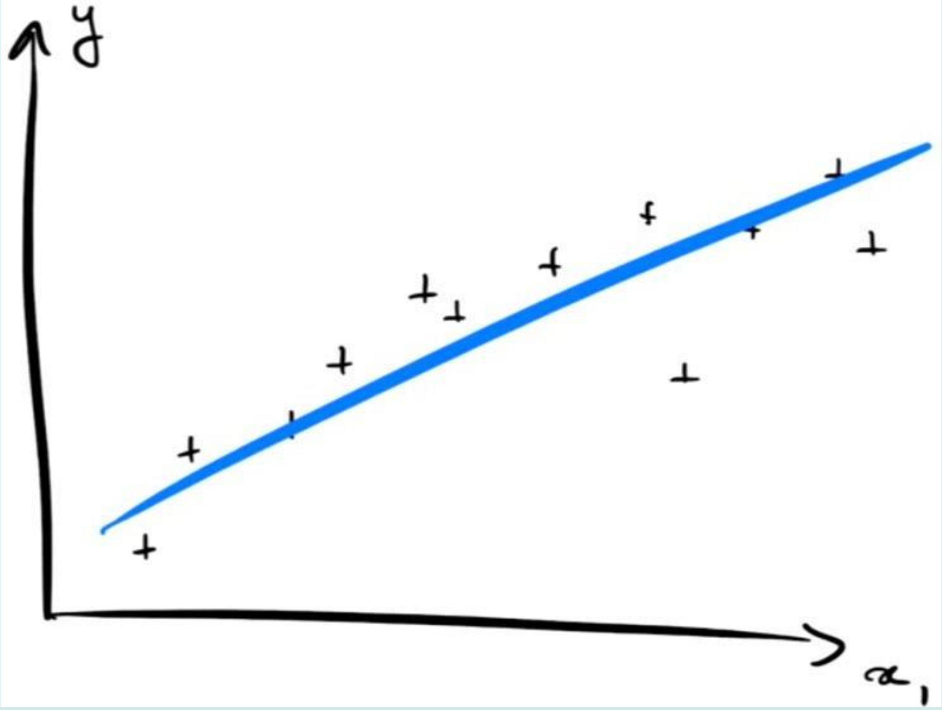


La Régression

❑ Formuler une hypothèse : Modèles de Régression

❑ Régression linéaire : Régression linéaire simple

- **Principe :** on cherche la **meilleure droite** qui passe au plus près des points.



Modèle linaire

$f(x) = ax + b$

on laisse la machine trouver la valeur de a et b qui donne les meilleurs résultats.

Exemple : prédire le **prix d’un appartement** en fonction de sa **surface**.

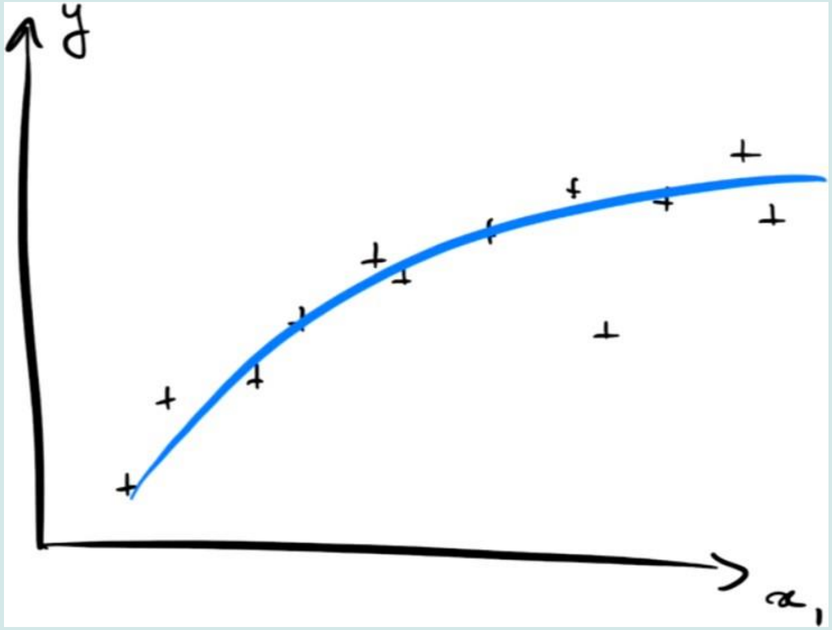
La Régression

❑ Formuler une hypothèse : Modèles de Régression

❑ Régression linéaire : Régression linéaire multiple

- **Principe :** on ajoute plusieurs variables explicatives.

$$y = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_p x_p$$



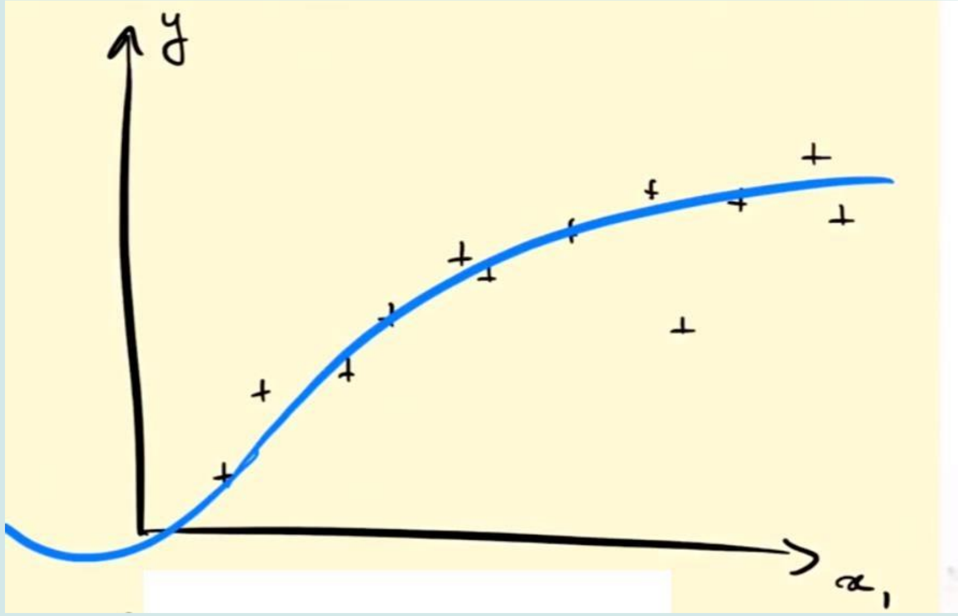
Exemple : prédire le **prix d'un appartement** en fonction de la **surface, localisation, nombre de chambres**.

La Régression

❑ Formuler une hypothèse : Modèles de Régression

❑Régression linéaire : Régression polynomiale (non linéaire)

- **Principe :** au lieu d'une droite, on ajuste une **courbe** (polynôme).



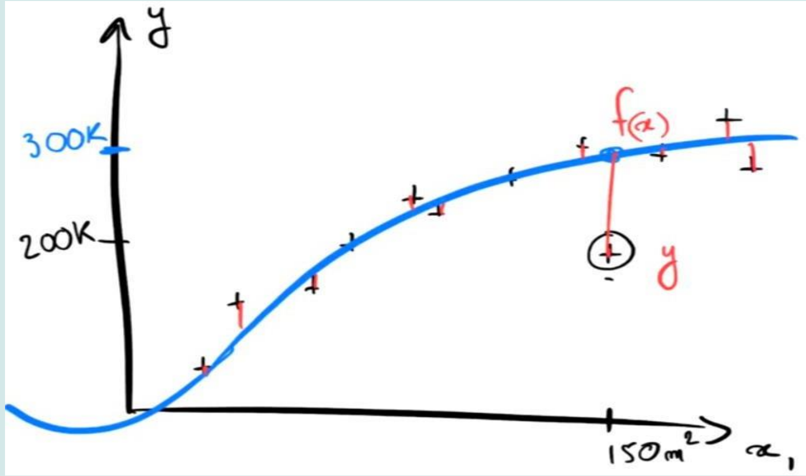
$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_p x^p$$

La Régression

❑ FONCTION COÛT

- ❑ La **fonction de perte (ou fonction de coût)** est une fonction mathématique qui mesure **l’écart entre la prédiction du modèle** et la **valeur réelle attendue.**

$$Loss_{function} = f(f\ \ x\ \ , y)$$



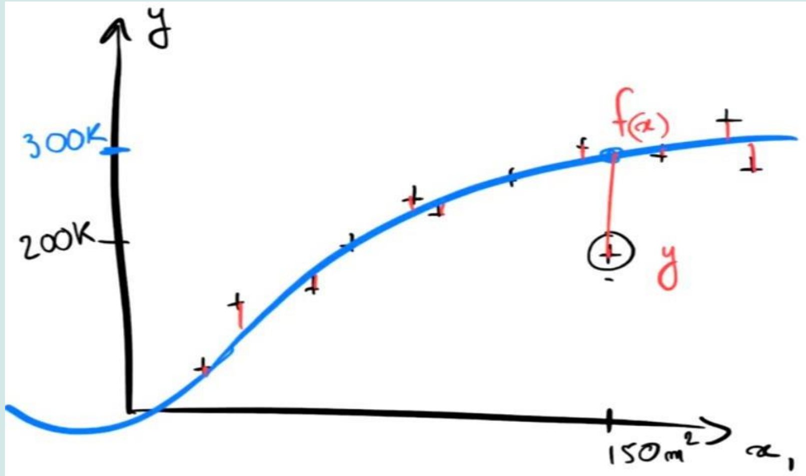
- La **loss function** calcule l’erreur → plus l’écart est grand, plus la perte est élevée.
- Objectif de l’apprentissage : **minimiser la fonction de perte**.

La Régression

❑ FONCTION COÛT

❑ Les fonctions coût possibles en régression :

- Erreur Quadratique Moyenne (Mean Squared Error, MSE)
- MAE (Mean Absolute Error)
- RMSE (Root Mean Squared Error)
- Huber Loss – Combinaison de MSE et MAE (utile quand il y a des outliers)



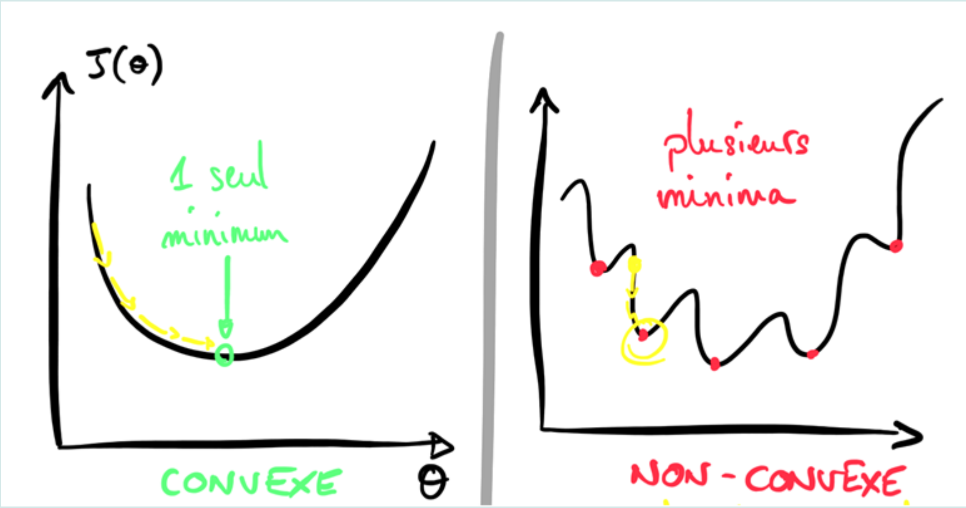
La Régression

- **Algorithme d'optimisation**

  - Quels sont les paramètres (les coefficients de la fonction de notre modèle) qui **minimise la fonction coût** (l'erreur entre la valeur estimée et la valeur réelle)?

  - on utilise un algorithme d'optimisation

    - Descente de gradient : algorithme de minmisation qui considérer la Fonction Coût comme une fonction **convexe**.

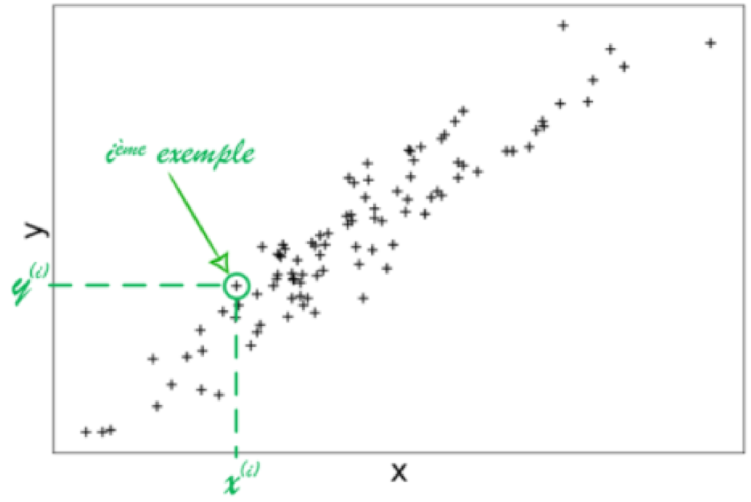


La Régression :Régression Linéaire simple

❑ Le Dataset

- ❑On veut développer un modèle de Machine Learning à partir d'un Dataset à une seule variable x_1
- ❑ On aura donc un Dataset avec m exemples et n=1 « *feature* » variable.
- ❑ Ce Dataset représente le prix d'un appartement y en fonction de sa surface habitable x dans une ville donnée. Ce Dataset pourrait nous donner le nuage de point suivant :

Prix y en €	Surface x m^2
350,000	100
160,000	50
280,000	80
…	…
235,000	75



y	x
$y^{(1)}$	$x^{(1)}$
$y^{(2)}$	$x^{(2)}$
$y^{(3)}$	$x^{(3)}$
…	…
$y^{(m)}$	$x^{(m)}$

La Régression :Régression Linéaire simple

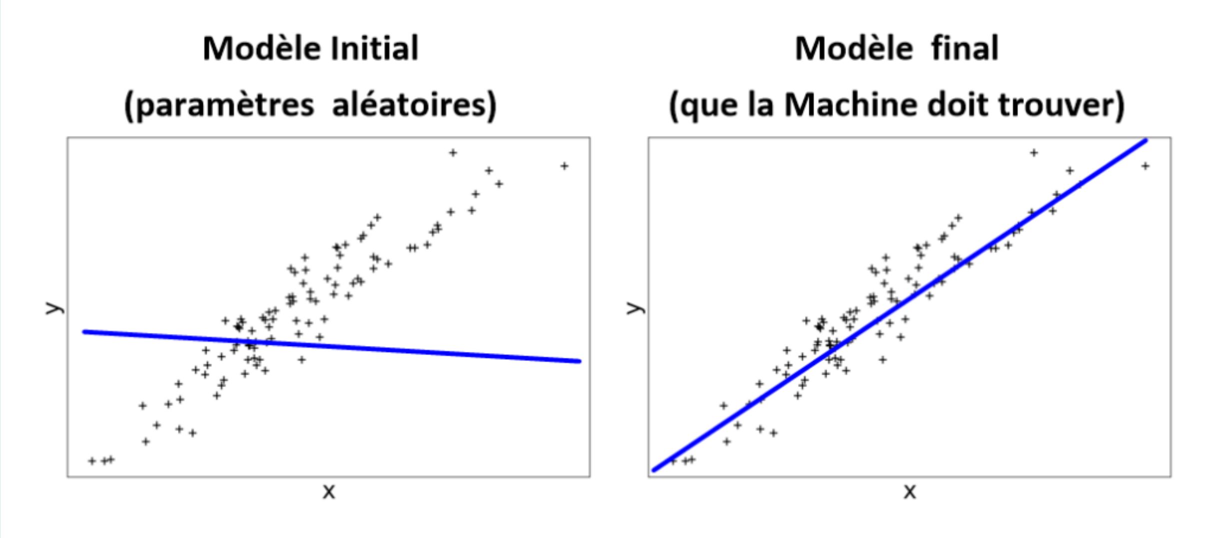
❑ Le Modèle

- ❑ D’après le nuage de point, on remarquait clairement qu’il suit une **tendance linéaire**, Notre modèle : $f(x) = ax + b$

- ❑ Pour le moment, nous ne connaissons pas la valeur des **paramètres** a et b

- ❑ Il est donc impossible de tracer une bonne droite sur le nuage de point, à moins de choisir des paramètres au hasard.

- ❑ Ce sera le rôle de la machine **d’apprendre** ces valeurs en minimisant la fonction coût.



La Régression :Régression Linéaire simple

❑ La Fonction Coût : L'Erreur Quadratique Moyenne

- ❑ La fonction coût nous permet d'évaluer la performance de notre modèle en mesurant les erreurs entre $f(x^{(i)})$ et $y^{(i)}$

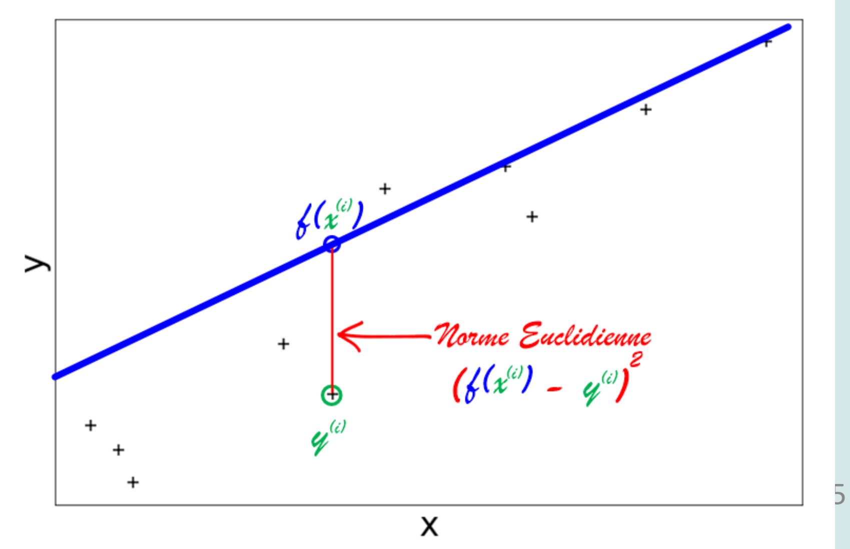
➔Comment mesurer ces erreurs ?

- ❑ Imaginez visiter un appartement à 200000 DT. Votre modèle de Machine Learning prédit que cet appartement vaut 300000 DT. Vous pourrez conclure que votre modèle fait donc une erreur de 300000 – 200000 = 100000 DT.

➔Ainsi, on pourrait se dire que pour mesurer nos erreurs, il faut faire calculer la différence $f(x) - y$

Cependant, si votre prédiction f(x) est inférieure à y , alors cette erreur est négative, ce qui n'est pas pratique

- ❑ Pour mesurer les erreurs entre les prédictions f(x) et les valeurs y du Dataset on calcule le carré de la différence $(f(x) - y)^2$: la **norme euclidienne**, qui représente la distance directe entre f(x) et y



La Régression :Régression Linéaire simple

❑ La Fonction Coût : L’Erreur Quadratique Moyenne

- ❑ Pour la régression linéaire, la fonction Coût J va être la moyenne de toutes nos erreurs, c’est-à-dire :

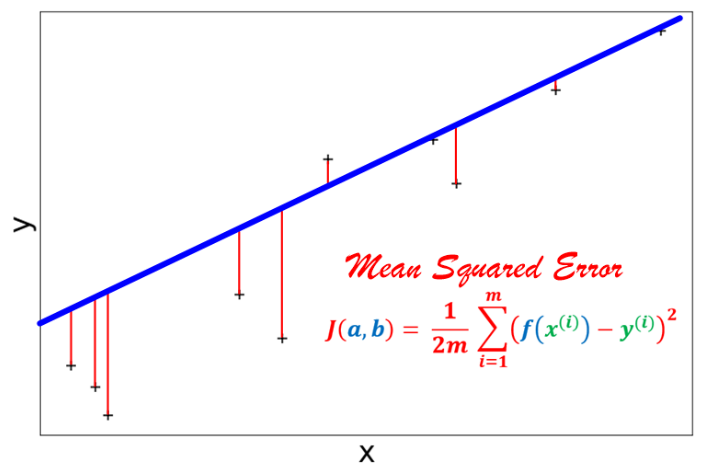
$$J = \frac{[f(x^{(1)}) - y^{(1)}]^2 + \ldots + [f(x^{(m)}) - y^{(m)}]^2}{m}$$

- ❑ Par convention on écrit cette fonction de la manière suivante, en rajoutant un coefficient $\frac{1}{2}$

→pour simplifier un calcul de dérivée qui viendra par la suite

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

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



- ❑ Cette fonction porte un nom : en français on l’appelle **l’Erreur Quadratique Moyenne**, et en anglais elle s’appelle **Mean Squared Error.**

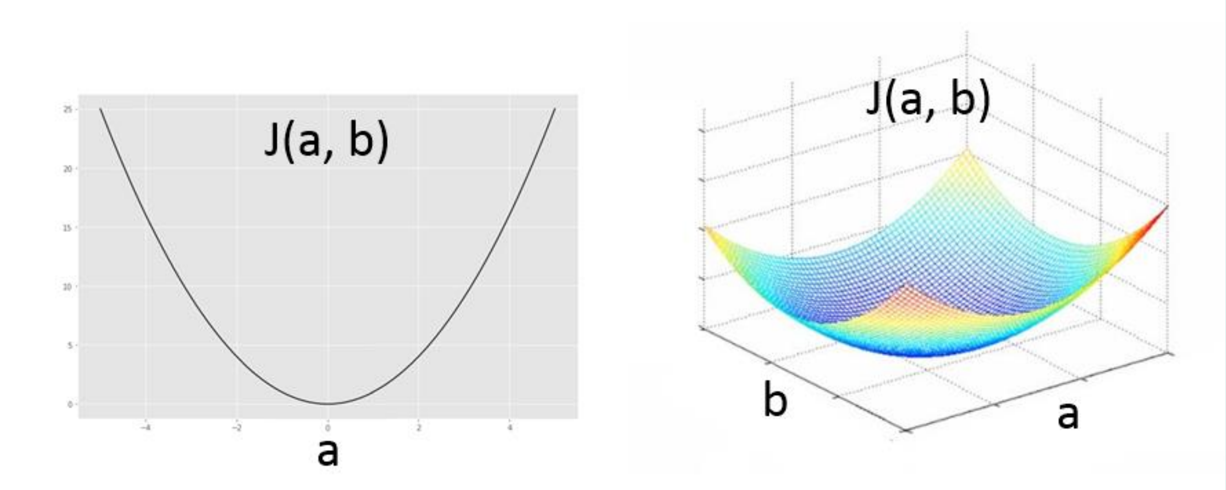
La Régression :Régression Linéaire simple

- **Mean Squared Error**

  - Fonction dépend de deux paramètres : a et b

  - **Convexe par rapport à a et b :** permet de trouver un **minimum global** →Cela garantit que la **descente de gradient** converge vers le **minimum global**

  - Cette propriété est très importante pour s'assurer de converger vers le minimum avec l'algorithme de la **descente de gradient** .



La Régression :Régression Linéaire simple

❑ R² : coefficient de détermination

- ❑ Il mesure **la proportion de la variance des données expliquée par le modèle**.
- ❑ Indique **dans quelle mesure le modèle suit la tendance des données réelles**.

$$R^2 = 1 - \frac{SS_{\text{résiduelle}}}{SS_{\text{totale}}} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

y_i : valeur réelle

$\hat{y}_i$: valeur prédite par le modèle

$\bar{y}$: moyenne des valeurs réelles

$SS_{\text{résiduelle}}$: somme des carrés des erreurs

SS_{totale} : somme des carrés totaux

- ❑ Indique la **proportion de la variation totale des données expliquée par le modèle**.
  - $R^2 = 1 \rightarrow$ modèle parfait
  - $R^2 = 0 \rightarrow$ le modèle ne fait pas mieux que la moyenne
  - $0 < R^2 < 1 \rightarrow$ proportion de variance expliquée.

La Régression :Régression Linéaire simple

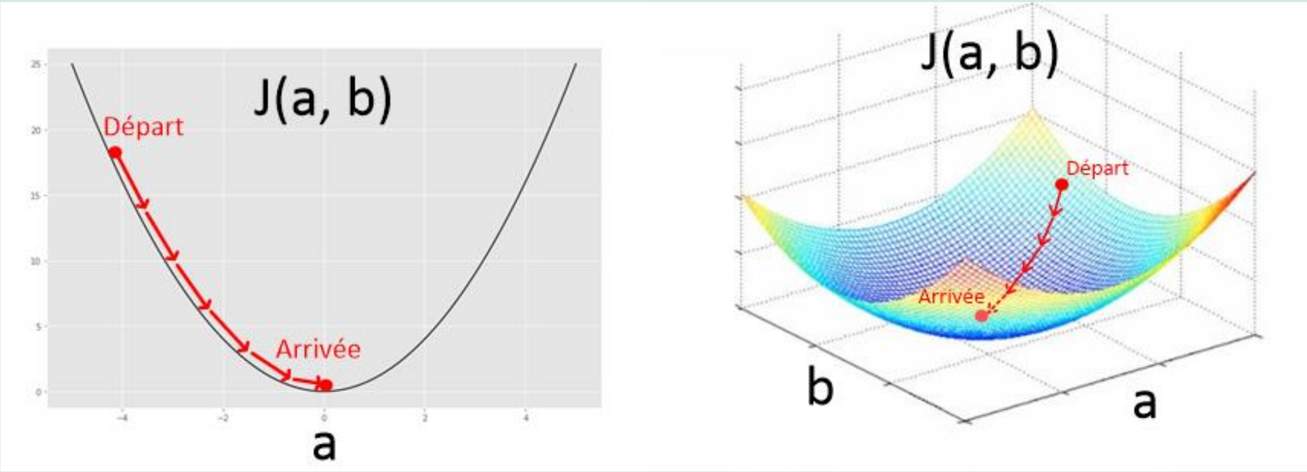
❑ L’algorithme d’apprentissage : Méthodes d’optimisation

- ❑ On dit que la machine apprend quand elle trouve quels sont les paramètres du modèle qui **minimisent** la fonction Coût.
- ❑ On cherche donc à développer un **algorithme de minimisation**.
- ❑ Il existe un paquet de méthodes de minimisation (méthode des moindres carrés, méthode de Newton, Gradient Descent, Simplex, etc.)
- ❑ Pour un modèle de régression linéaire, on utilise le plus souvent **la méthode des moindres carrés** (quand le problème est simple) et l’algorithme de **Gradient Descent** (pour les régressions plus compliquée).
- ❑ Le Gradient Descent, permet de converger progressivement vers le minimum de n’importe quelle fonction **convexe** (comme la Fonction Coût) en suivant la direction de la pente (le gradient) qui descend, d’où son nom de Gradient Descent.

La Régression :Régression Linéaire simple

❑ Méthodes d'optimisation

- ❑ Pour un modèle de régression linéaire, on utilise le plus souvent :
  - **la méthode des moindres carrés** →quand le problème est simple
  - l'algorithme de **Gradient Descent** → pour les régressions plus compliquée
- ❑ Le Gradient Descent, permet de converger progressivement vers le minimum de n'importe quelle fonction **convexe** (comme la Fonction Coût) en suivant la direction de la pente (le gradient) qui descend, d'où son nom de Gradient Descent.



La Régression :Régression Linéaire simple

❑ L'algorithme d'apprentissage: Méthode des moindres carrés (Ordinary Least Squares – OLS)

- la méthode **classique et exacte** utilisée quand le modèle de régression est **simple**.
- L'idée est de trouver la **droite la plus proche possible des points** en minimisant la **somme des carrés des erreurs**.

$$\text{Minimiser } \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

où

- y_i : valeur réelle
- $\hat{y}_i = ax_i + b$: valeur prédite

- Il existe une **formule directe** pour calculer les paramètres :

$$a = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

$$b = \bar{y} - a\bar{x}$$

La Régression :Régression Linéaire simple

❑ L'algorithme d'apprentissage:Descente de gradient (Gradient Descent)

- Méthode **numérique et itérative** utilisée quand le modèle est **complexe** ou contient **beaucoup de variables**.
- L'idée est de **partir d'une estimation aléatoire** des paramètres (poids), puis de les **ajuster progressivement** pour **minimiser l'erreur (coût)**.

- **Étapes de l'algorithme**

Régression Linéaire: Résumé des équations

- **Dataset :** (x, y) avec m exemples
- **Modèle :** $f(x) = ax + b$
- **Fonction Coût :** $J(a,b) = \frac{1}{2m}\sum_{i=1}^{m}\left(ax^{(i)} + b - y^{(i)}\right)^2$
- **Gradients :**

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

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

- **Algorithme de Gradient Descent :**

$$\begin{cases} a := a - \alpha \dfrac{\partial J(a,b)}{\partial a} \\ b := b - \alpha \dfrac{\partial J(a,b)}{\partial b} \end{cases}$$

La Régression :Régression Linéaire simple

- **Moindres carrés**
  - **Avantages :**
    - Donne **la solution exacte** pour $a$et $b$qui minimisent la MSE.
    - Très **rapide à calculer** avec une petite formule analytique.
    - Convient parfaitement aux petits datasets simples.
  - **Inconvénients :**
    - Si le problème devient très complexe ou multivarié, la formule peut devenir lourde.
- **Descente de gradient**
  - **Avantages :**
    - Peut résoudre des problèmes très complexes ou avec beaucoup de variables.
    - Nécessaire pour les grands datasets ou des modèles non linéaires.
  - **Inconvénients :**
    - Solution approximative, nécessite **plusieurs itérations** pour converger.
    - Le choix du **taux d'apprentissage** αest crucial.
    - Pour un seul pas, les résultats sont **très approximatifs**.

La Régression :Régression Linéaire simple

❑ Exercice d'application

On étudie la relation entre la vitesse d'une voiture (en km/h) et sa consommation de carburant (en L/100 km). On dispose du dataset suivant :

Essai	Vitesse (km/h)	Consommation (L/100 km)
1	40	7.5
2	60	6.8
3	80	6.5
4	100	6.7
5	120	7.2

Questions

1. Définir la fonction de coût MSE pour une droite $y = a\,x + b$.
2. Méthode analytique (moindres carrés) : calculer a et b qui minimisent la MSE.
3. Descente de gradient : effectuer manuellement un pas avec $\alpha = 0.01$, $a0 = 0$, $b0 = 0$.
4. Estimer la consommation pour une voiture roulant à 90 km/h.

La Régression :Régression Linéaire simple

❑ Exercice d'application

1. la fonction de coût MSE pour : y=f(x)

$$f(x) = ax + b$$

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

La Régression :Régression Linéaire simple

- **Exercice d'application**

Méthode analytique (moindres carrés)

Formules :

$$a = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

$$b = \bar{y} - a\bar{x}$$

Données :

$$\bar{x} = 80, \quad \bar{y} = 6.94$$

Calcul :

$$a = \frac{-36}{4000} = -0.009$$

$$b = 6.94 - (-0.009)(80) = 6.94 + 0.72 = 7.66$$

Équation de la droite :

$$\boxed{y = -0.009x + 7.66}$$

La Régression :Régression Linéaire simple

- **Exercice d'application : 2.descente de gradient iteration1:**

Formules

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

$$\frac{\partial J}{\partial a} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}(ax^{(i)} + b - y^{(i)}) \qquad \frac{\partial J}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(ax^{(i)} + b - y^{(i)})$$

Calcul des résidus pour $a_0 = 0, b_0 = 0$

Pour chaque i, $a_0 x_i + b_0 - y_i = 0 \cdot x_i + 0 - y_i = -y_i$.

Donc :

$$\sum_{i=1}^{5} x_i(a_0 x_i + b_0 - y_i) = \sum_{i=1}^{5} x_i(-y_i) = -\sum_{i=1}^{5} x_i y_i = -2189.0.$$

$$\sum_{i=1}^{5}(a_0 x_i + b_0 - y_i) = \sum_{i=1}^{5}(-y_i) = -\sum_{i=1}^{5} y_i = -34.7.$$

La Régression :Régression Linéaire simple

- Exercice d'application : 2.descente de gradient iteration1

Gradients (valeurs numériques)

$$\frac{\partial J}{\partial a}\Big|_{(0,0)} = \frac{1}{m}\sum_i x_i(a_0x_i + b_0 - y_i) = \frac{1}{5}\cdot(-2189.0) = -\frac{2189.0}{5}$$

$$\Rightarrow \quad \frac{\partial J}{\partial a} = -437.8.$$

$$\frac{\partial J}{\partial b}\Big|_{(0,0)} = \frac{1}{5}\cdot(-34.7) = -\frac{34.7}{5}.$$

$$\Rightarrow \quad \frac{\partial J}{\partial b} = -6.94$$

La Régression :Régression Linéaire simple

- **Exercice d'application : 2.descente de gradient iteration1**

Mise à jour des paramètres

Règle : $a := a - \alpha \frac{\partial J}{\partial a}$, $b := b - \alpha \frac{\partial J}{\partial b}$ avec $\alpha = 0.01$.

Calculs :

- Pour a :

$$a_1 = a_0 - 0.01 \cdot (-437.8) = 0 - 0.01 \cdot (-437.8).$$

$$a_1 = 0 + 4.378 = \boxed{4.378}.$$

- Pour b :

$$b_1 = b_0 - 0.01 \cdot (-6.94) = 0 - 0.01 \cdot (-6.94).$$

$$b_1 = 0 + 0.0694 = \boxed{0.0694}.$$

$$\boxed{y = 4.378\,x + 0.0694.}$$

La Régression :Régression Linéaire simple

- Exercice d'application:3 prediction

Prédiction — Moindres carrés

$$y(90) = -0.009 \times 90 + 7.66$$

$$\boxed{y(90) = 6.85 \text{ L}/100 \text{ km}}$$

Prédiction — Descente de gradient (1 itération)

$$y(90) = 4.378 \times 90 + 0.0694$$

Valeur très approximative, car un seul pas n'a pas encore convergé.

$$\boxed{y(90) = 394.09 \text{ L}/100 \text{ km}}$$