

# Introduction à l'apprentissage automatique (machine learning)

Dany Anderson & Ariste Yougbaré  
Pole Data-IA

- ▶ **Introduction**
- ▶ Apprentissage Supervisé
- ▶ Apprentissage Supervisé : Introduction Mathématique
- ▶ Apprentissage Supervisé : Régression Logistique
- ▶ Apprentissage Supervisé : Régression Linéaire
- ▶ Pratique
- ▶ Résumé

L'apprentissage automatique (Machine Learning, ou ML) est une branche de l'intelligence artificielle (IA) qui permet aux ordinateurs d'apprendre et de prendre des décisions ou faire des prédictions sans être explicitement programmés pour chaque tâche. Au lieu de suivre des instructions figées, les machines utilisent des données pour s'entraîner : elles identifient des modèles, effectuent des prédictions et améliorent leurs performances au fil du temps. En termes plus simples, le ML permet aux ordinateurs d'améliorer automatiquement leurs performances dans une tâche grâce à l'expérience. Tout comme les humains apprennent de leurs expériences, les machines apprennent à partir des données qu'on leur fournit.

## **Années 1950 : Les Premiers Débuts**

- “Computing Machinery and Intelligence” d’Alan Turing propose le **Test de Turing**, un concept fondateur pour l’IA.

## **Années 1980 : Systèmes Experts**

- Les systèmes basés sur des règles gagnent en importance, tentant d’imiter la prise de décision humaine dans des domaines spécifiques.

## **Années 2000 : Explosion des Données**

- Une puissance de calcul accrue et de vastes ensembles de données relancent l’intérêt et entraînent des avancées significatives en apprentissage automatique.

## **Aujourd’hui : L’IA Partout**

- L’apprentissage automatique est intégré dans d’innombrables applications, des systèmes de recommandation aux véhicules autonomes.

# Les Principaux Types d'Apprentissage Automatique

- **Supervised Learning (Apprentissage supervisé)** Utilise des données étiquetées pour entraîner un modèle à prédire ou classifier. Le modèle généralise à partir d'exemples connus. **Exemple** : Prédiction des prix immobiliers. **Algorithmes** : Régression linéaire, arbres de décision, forêts aléatoires, SVM.
- **Unsupervised Learning (Apprentissage non supervisé)** Découvre des structures ou regroupements dans des données non étiquetées, sans réponses prédéfinies. **Exemple** : Segmentation de clients pour le marketing. **Algorithmes** : K-Means, PCA, DBSCAN, autoencodeurs.
- **Reinforcement Learning (Apprentissage par renforcement)** Apprend par essais et récompenses pour maximiser une récompense cumulative dans un environnement. **Exemple** : Apprentissage de la marche pour un robot. **Algorithmes** : Q-Learning, DQN, PPO.

- ▶ Introduction
- ▶ **Apprentissage Supervisé**
- ▶ Apprentissage Supervisé : Introduction Mathématique
- ▶ Apprentissage Supervisé : Régression Logistique
- ▶ Apprentissage Supervisé : Régression Linéaire
- ▶ Pratique
- ▶ Résumé

- **Classification** La classification consiste à prédire une catégorie ou une étiquette de classe. La sortie est discrète : le modèle classe les données dans des groupes prédéfinis.

**Exemples :**

- Déterminer si un email est “spam” ou “non spam” (deux classes distinctes).
- Classer une fleur parmi trois espèces en fonction de ses caractéristiques.

- **Régression** La régression consiste à prédire une valeur continue ou une quantité. La sortie est numérique : le modèle prédit une valeur sur une échelle continue.

**Exemples :**

- Prédire le prix d’une maison en fonction de ses caractéristiques (taille, nombre de pièces, localisation).
- Prévoir la température de demain à partir de données historiques.

- ▶ Introduction
- ▶ Apprentissage Supervisé
- ▶ **Apprentissage Supervisé : Introduction Mathématique**
- ▶ Apprentissage Supervisé : Régression Logistique
- ▶ Apprentissage Supervisé : Régression Linéaire
- ▶ Pratique
- ▶ Résumé



L'apprentissage supervisé consiste à apprendre une fonction :

$$f : \mathbb{R}^d \rightarrow \mathbb{R}$$

ou

$$f : \mathbb{R}^d \rightarrow \{0, 1\}$$

à partir d'un ensemble de données étiquetées.

# Définition Formelle

On dispose d'un dataset :

$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^m$$

- $x^{(i)}$  : vecteur de caractéristiques (features)
- $y^{(i)}$  : variable cible (label)
- $m$  : nombre d'exemples

Nous cherchons un modèle paramétré par  $\theta$  (souvent  $w$  et  $b$ ) tel que :

$$\hat{y} = f(x; \theta)$$

L'objectif est de minimiser une fonction de coût :

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)})$$

où :

- $L$  : fonction de perte (loss)
- $J$  : coût global (moyenne des pertes)

La majorité des modèles supervisés simples reposent sur :

$$z = w^T x + b$$

- $w$  : poids
- $b$  : biais
- $z$  : combinaison linéaire

Puis éventuellement une fonction d'activation :

$$\hat{y} = g(z)$$

Un modèle supervisé suit toujours le même cycle :

1. Forward pass : calcul de la prédiction  $\hat{y}$
2. Calcul de l'erreur via une fonction de coût
3. Backpropagation : calcul des gradients
4. Mise à jour des paramètres

Règle générale de mise à jour :

$$\theta := \theta - \alpha \nabla J(\theta)$$

- $\alpha$  : taux d'apprentissage
- $\nabla J(\theta)$  : gradient

# Pourquoi le Gradient ?

Le gradient indique la direction de plus forte augmentation de la fonction de coût.

Pour minimiser la fonction :

$$\theta := \theta - \alpha \nabla J(\theta)$$

On avance donc dans la direction opposée au gradient.

Si la fonction est convexe :

- Minimum global garanti
- Convergence stable

# Classification vs Régression

|                       | <b>Classification</b> | <b>Régression</b> |
|-----------------------|-----------------------|-------------------|
| Sortie                | Discrète              | Continue          |
| Exemple               | Spam / Non spam       | Prix immobilier   |
| Fonction d'activation | Souvent non-linéaire  | Aucune (linéaire) |
| Fonction de coût      | BCE, CCE              | MSE, RMSE         |

- ▶ Introduction
- ▶ Apprentissage Supervisé
- ▶ Apprentissage Supervisé : Introduction Mathématique
- ▶ **Apprentissage Supervisé : Régression Logistique**
- ▶ Apprentissage Supervisé : Régression Linéaire
- ▶ Pratique
- ▶ Résumé



La régression logistique est utilisée pour la classification binaire. Elle modélise la probabilité qu'une observation appartienne à la classe 1.

## Forward Pass

$$z = w^T x + b$$

- $x \in \mathbb{R}^d$  : vecteur d'entrée
- $w \in \mathbb{R}^d$  : poids
- $b \in \mathbb{R}$  : biais

Fonction d'activation (Sigmoid) :

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

# Fonction de Coût : Binary Cross Entropy

Pour une observation :

$$L = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Pour  $m$  observations :

$$J(w, b) = -\frac{1}{m} \sum_{i=1}^m \left[ y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

Étape 1 : dérivée par rapport à  $z$

$$\frac{\partial L}{\partial z} = \hat{y} - y$$

Gradient par rapport aux paramètres

$$\frac{\partial J}{\partial w} = \frac{1}{m} X^T (\hat{y} - y)$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})$$

Algorithme d'optimisation : Descente de Gradient

$$w := w - \alpha \frac{\partial J}{\partial w}$$

$$b := b - \alpha \frac{\partial J}{\partial b}$$

- $\alpha$  : learning rate
- Objectif : minimiser  $J(w, b)$

- ▶ Introduction
- ▶ Apprentissage Supervisé
- ▶ Apprentissage Supervisé : Introduction Mathématique
- ▶ Apprentissage Supervisé : Régression Logistique
- ▶ Apprentissage Supervisé : Régression Linéaire**
- ▶ Pratique
- ▶ Résumé

# Régression Linéaire : Principe

La régression linéaire est utilisée pour prédire une valeur continue.

## Forward Pass

$$\hat{y} = w^T x + b$$

Il n'y a pas de fonction d'activation.

# Fonction de Coût : RMSE

On définit d'abord le MSE :

$$MSE = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2$$

Puis la racine :

$$RMSE = \sqrt{MSE}$$

# Backpropagation : Gradients

On dérive le MSE pour simplifier les calculs.

$$\frac{\partial J}{\partial w} = \frac{2}{m} X^T (\hat{y} - y)$$

$$\frac{\partial J}{\partial b} = \frac{2}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})$$



$$w := w - \alpha \frac{\partial J}{\partial w}$$

$$b := b - \alpha \frac{\partial J}{\partial b}$$

- $\alpha$  : learning rate
- Objectif : minimiser l'erreur quadratique

# Résumé Comparatif

|                  | Régression Logistique | Régression Linéaire |
|------------------|-----------------------|---------------------|
| Type de sortie   | Probabilité $[0, 1]$  | Valeur continue     |
| Activation       | Sigmoid               | Aucune              |
| Fonction de coût | BCE                   | RMSE / MSE          |
| Gradient clé     | $\hat{y} - y$         | $2(\hat{y} - y)$    |

- ▶ Introduction
- ▶ Apprentissage Supervisé
- ▶ Apprentissage Supervisé : Introduction Mathématique
- ▶ Apprentissage Supervisé : Régression Logistique
- ▶ Apprentissage Supervisé : Régression Linéaire
- ▶ **Pratique**
- ▶ Résumé

# Practice time



Vous trouverez les ressources nécessaire aux liens suivants.

## Ressources

- Régression Linéaire Simple
- Régression Linéaire pour la Détection de Chat

- ▶ Introduction
- ▶ Apprentissage Supervisé
- ▶ Apprentissage Supervisé : Introduction Mathématique
- ▶ Apprentissage Supervisé : Régression Logistique
- ▶ Apprentissage Supervisé : Régression Linéaire
- ▶ Pratique
- ▶ Résumé

- Le Machine Learning permet aux machines d'apprendre à partir des données sans programmation explicite.
- Trois types principaux : supervisé, non supervisé et par renforcement.
- En apprentissage supervisé, on apprend une fonction à partir de données étiquetées.
- Deux grandes tâches : classification (sortie discrète) et régression (sortie continue).
- Les modèles linéaires reposent sur une combinaison  $w^T x + b$ .
- L'apprentissage consiste à minimiser une fonction de coût via la descente de gradient.
- Régression logistique (classification) et linéaire (régression) utilisent des fonctions de coût différentes.

Merci pour votre attention