



Apprentissage Artificiel

par

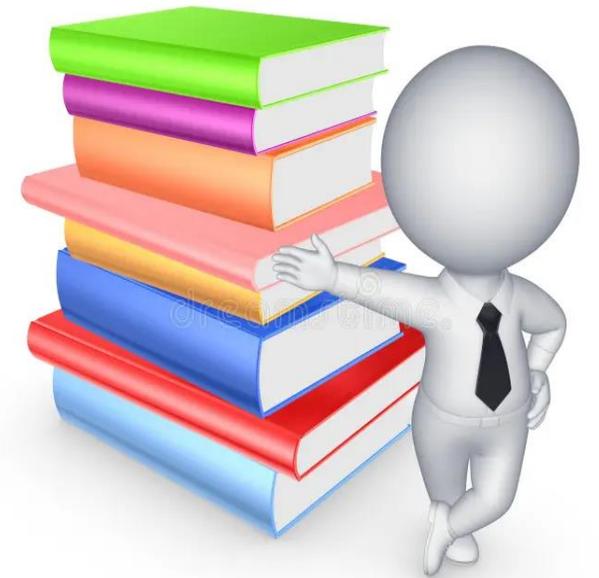
Dr. Samira LAGRINI



Année universitaire:2024/2025

Plan du cours

- ❑ Comprendre ce qu'est le Machine Learning (ML)
- ❑ Pourquoi il est important.
- ❑ Explorer des applications concrètes du ML.
- ❑ Découvrir les différents types de ML
- ❑ Identifier les étapes du développement d'un modèle ML.



Qu'est-ce que l'apprentissage automatique?

L'apprentissage automatique (ou Machine Learning ML) est un domaine de l'intelligence artificielle qui permet à un système d'apprendre à partir de données pour accomplir une tâche spécifique sans être explicitement programmé.

Contrairement à la programmation traditionnelle, où un développeur écrit des règles précises pour résoudre un problème, l'apprentissage artificiel utilise **des données** pour "apprendre" ces règles.



Qu'est-ce que l'apprentissage automatique?

Exemple:

un modèle de reconnaissance d'images apprend à identifier un chat en analysant des milliers de photos de chats étiquetées.

 CHIEN	 CHAT	 CHAT	 CHIEN	 CHIEN
 CHAT	 CHAT	 CHIEN	 CHAT	 CHIEN
 CHAT	 CHIEN	 CHIEN	 CHAT	 CHAT
 CHIEN	 CHAT	 CHIEN	 CHAT	 CHIEN
 CHIEN	 CHIEN	 CHAT	 CHAT	 CHAT

Qu'est-ce que l'apprentissage automatique?

Plus simplement:

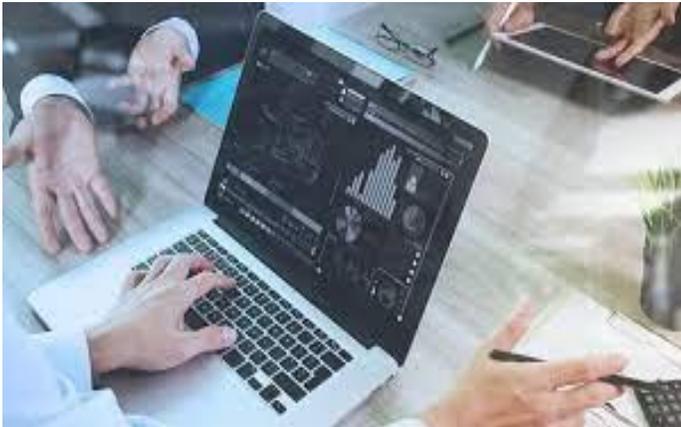
Le Machine Learning est une démarche de développement d'une fonction de prédiction à partir de données d'exemples, sans programmation explicite des règles



Pourquoi le ML est-il important aujourd'hui ?

Explosion des données

le ML offre la possibilité d'analyser d'énormes volumes de données pour identifier des tendances et des patterns qui seraient impossibles à détecter manuellement



Automatisation des tâches complexes

ce qui réduit les coûts et les erreurs humains

Prédictions précises

Dans des secteurs comme la **finance**, le commerce, la **santé**, et les **systèmes de recommandation**, le ML permet de prendre des décisions précise en temps réel.



Différence entre IA, ML et Deep Learning

Intelligence Artificielle (IA)

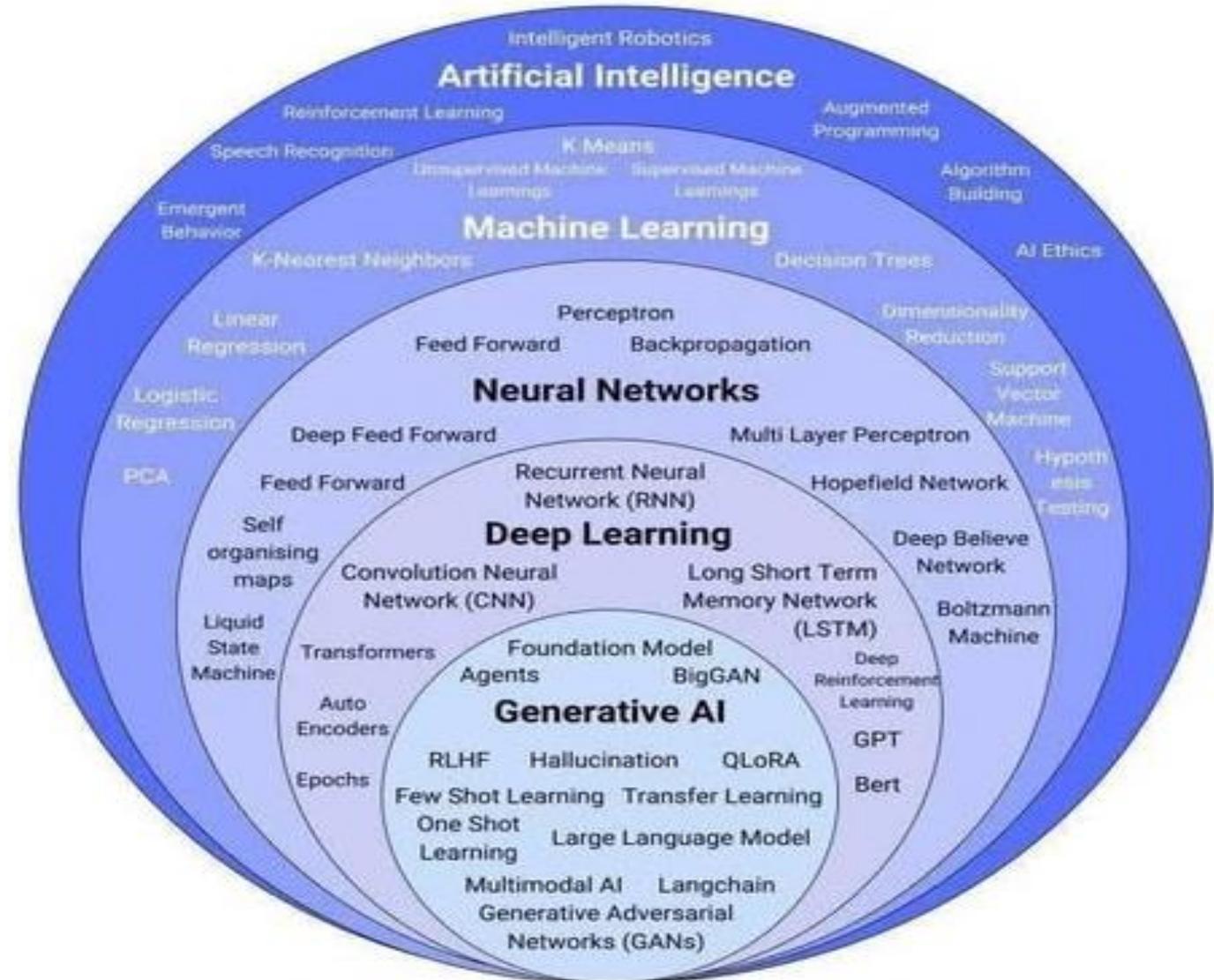
Concept global qui regroupe toutes les techniques visant à rendre une machine intelligente.

Machine Learning (ML)

Sous-domaine de l'IA où les modèles apprennent à partir de données.

Deep Learning (DL)

Sous-domaine du ML utilisant des réseaux de neurones profonds pour traiter des données complexes.



Machine Learning vs. Deep Learning : quelles différences ?

	Deep learning	Machine learning
Type Données d'entraînement	Utilise également des données non structurées (Textes, images, vidéos)	données structurées et semi-structurées
Volume de données d'entraînement	nécessite un grand volume de données d'entraînement	peut être utilisé avec des ensembles de données plus petits
Architecture	utilise des réseaux de neurones artificiels (RNN) multicouches pour extraire des caractéristiques à partir de données brutes et pour apprendre à partir de ces caractéristiques.	utilise des algorithmes d'apprentissage pour former des modèles
Capacité de généralisation	DL a une capacité de généralisation supérieure par rapport au ML	
Besoins en puissance de calcul	nécessite une puissance de calcul élevée pour entraîner des modèles complexes	peut fonctionner sur des machines moins puissantes
Applications	utilisé pour des tâches plus complexes (la reconnaissance d'images, et de la parole , la traduction automatique, le NLP...)	utilisé pour les tâches de prédiction, de classification et de clustering,

Applications pratiques de ML

Santé et diagnostic médical

L'apprentissage automatique aide à analyser les images médicales, à prédire les maladies, ou même à personnaliser les traitements pour les patients.



Reconnaissance vocale

Des systèmes comme Siri, Alexa, ou Google Assistant utilisent l'apprentissage automatique pour comprendre et répondre aux commandes vocales.



Détection de fraudes

Dans les banques et les systèmes financiers, le ML est utilisé pour détecter des transactions frauduleuses.



Applications pratiques de ML

Reconnaissance d'images

Des applications comme la reconnaissance faciale (Facebook, iPhone) , les systèmes de sécurité utilisant des caméras intelligentes, et la détection d'objets dans des images ou des vidéos (ex: véhicule autonome).



Systèmes de recommandation

Des plateformes comme Netflix, YouTube, ou Amazon utilisent des algorithmes pour recommander des films, vidéos, ou produits basés sur les préférences des utilisateurs.



Applications pratiques de ML

Traitement du langage naturel (NLP)

Des applications telles que la traduction automatique (Google Translate), la génération de texte (comme les chatbots), et l'analyse des sentiments sur les réseaux sociaux.

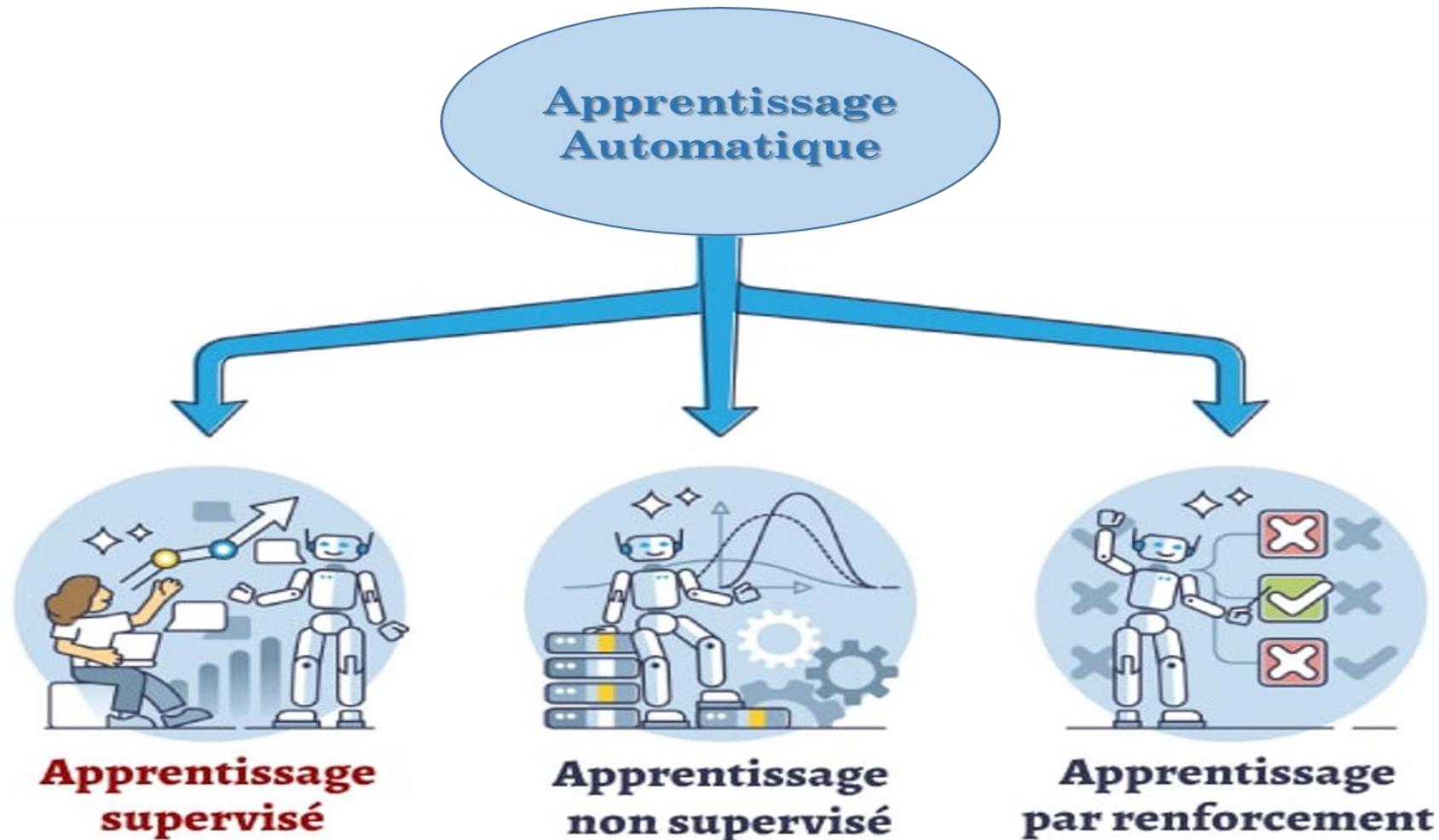


Analyse de données massives (Big Data)

Les entreprises utilisent le ML pour extraire des informations pertinentes de grandes quantités de données afin d'optimiser les décisions stratégiques.



Types d'Apprentissage



Apprentissage Supervisé

- ❑ Consiste à entraîner un modèle sur un ensemble de données étiquetées
- ❑ Chaque instance d'entraînement est associée à une étiquette (label).
- ❑ L'objectif est d'utiliser l'ensemble de données pour produire un **modèle** qui prend en entrée un vecteur de caractéristiques et génère une information permettant de déduire le label associé à ce vecteur de caractéristiques.

The diagram shows a table of training data with 14 columns and 15 rows. Annotations include: 'Vecteur de caractéristiques (Features vector)' pointing to columns 2-13; 'Instance (échantillon)' pointing to a row; 'Données d'entraînement' at the bottom; 'Caractéristiques (features)' pointing to columns 2-13; and 'Label (classe ou un une valeur numérique)' pointing to the 'Class' column.

1	2	3	4	5	6	7	8	9	10	11	12	13	Class
4	4	7300	140	135	3200	4.5	3.0	1.8	3.4	1190	1	1	A
4	4	4367	119	130	1101	4.2	2.9	1.2	2.6	1141	1	1	A
4	4	5006	95	124	1416	4.2	2.9	1.3	2.8	1134	1	1	A
4	4	6331	126	134	1595	4.3	2.7	1.0	2.8	1117	1	1	A
4	4	5433	121	138	725	4.2	2.9	0.8	3.4	1159	1	1	A
4	4	5452	135	116	1081	3.9	2.8	0.8	2.7	1113	1	1	A
4	4	6508	99	127	1308	4.4	2.7	1.1	2.8	1179	1	1	A
4	4	4678	118	129	629	4.1	2.8	0.8	2.9	1153	1	1	A
4	4	4642	92	126	1612	4.5	2.9	1.1	3.0	1128	1	1	A
4	4	6944	114	134	1251	4.2	2.9	0.8	2.8	1147	1	1	A
4	4	3944	130	128	821	3.8	2.7	1.0	3.0	1123	1	1	A
4	4	5551	124	132	1681	4.1	2.9	1.4	3.1	1103	1	1	A
4	4	5458	103	134	2137	4.4	2.6	1.3	2.9	1180	1	1	A
4	4	5907	125	133	365	4.3	2.7	1.4	2.7	1116	1	1	A
4	4	6470	101	121	1749	4.2	3.0	1.0	2.9	1166	1	1	A

Types de Tâches

▪ Classification

Attribuer une classe (label) à une entrée.

Exemples

- ✓ Détection d'e-mails spam/non-spam.
- ✓ Reconnaissance faciale.
- ✓ Diagnostic médical (malade/sain).

▪ Régression (Sortie continue)

Prédire une valeur numérique.

Exemples

- ✓ Prédiction du prix d'une maison en fonction de ses caractéristiques.
- ✓ Estimation des ventes futures d'un produit.
- ✓ Estimation de la consommation d'énergie.
- ✓ Prévission de la température.

Exemple

tumor size	texture	perimeter	...	outcome	time
18.02	27.6	117.5		N	31
17.99	10.38	122.8		N	61
20.29	14.34	135.1		R	27
...					

Prédire le **résultat** de la maladie est un problème de **classification**.

Prédire le **temps** est un problème de **régression**

Algorithmes courants

▪ Classification

- ✓ Support Vector Machines (SVM)
- ✓ Régression Logistique
- ✓ k-Nearest Neighbors (k-NN)
- ✓ Arbres de décision
- ✓ Forêts aléatoires (Random Forest)
- ✓ Réseaux de neurones

▪ Régression

- ✓ Régression linéaire
- ✓ Régression polynomiale
- ✓ Régression logistique
(sert à estimer la probabilité d'appartenir à une classe)

Apprentissage Non Supervisé (Unsupervised Learning)

- Le modèle est entraîné sur des données non étiquetées
- L'objectif est de créer un modèle qui prend en entrée un vecteur de caractéristiques et le transforme soit en un autre vecteur (réduction de dimensionnalité), soit en une valeur exploitable (clustering) pour résoudre un problème concret.

Types de Tâches

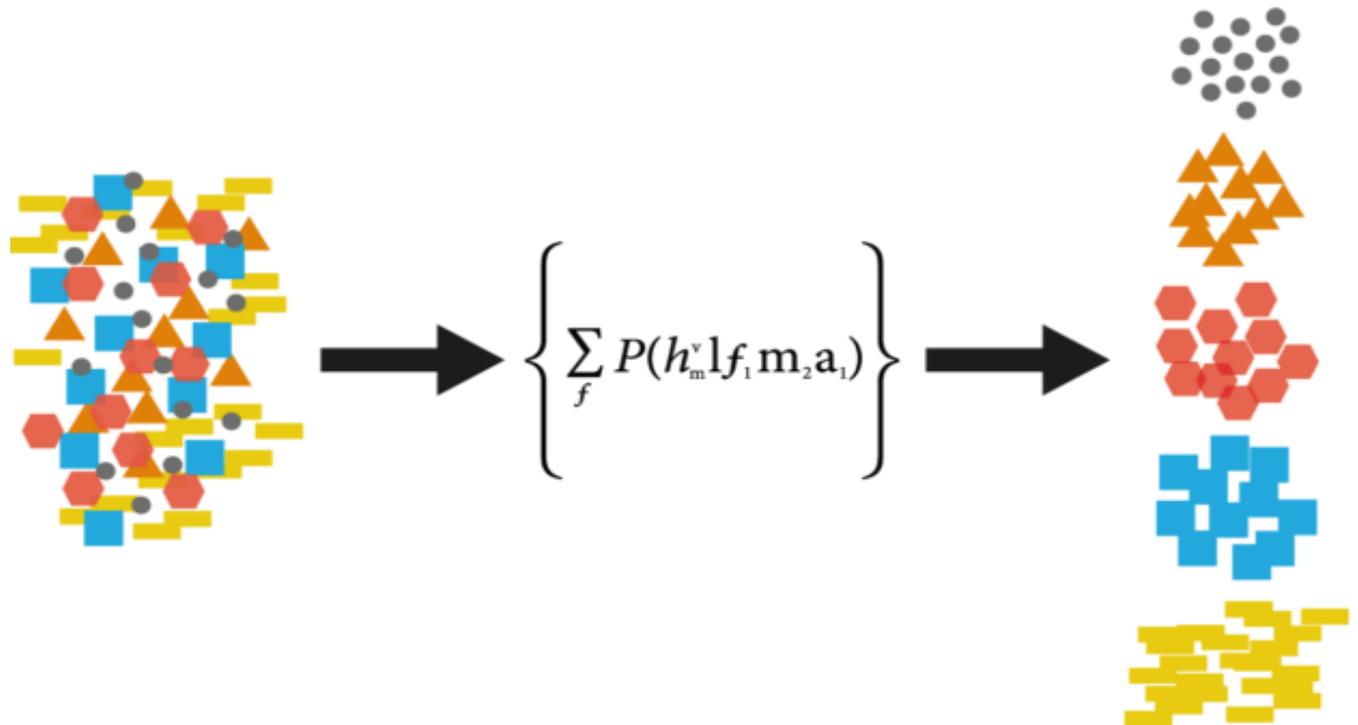
- Clustering (Regroupement)
- Réduction de dimensionnalité
- Extraction de Règles d'Association
- Détection d'anomalies
-

Le Clustering

- Consiste à regrouper des données similaires en clusters
- Le modèle attribue un identifiant de cluster (numero) à chaque vecteur de caractéristiques de l'ensemble de données.

Algorithmes courants :

- ✓ K-Means
- ✓ Hierarchical Clustering
- ✓ DBSCAN



Le Clustering

Exemple

Segmentation des Clients dans un Supermarché

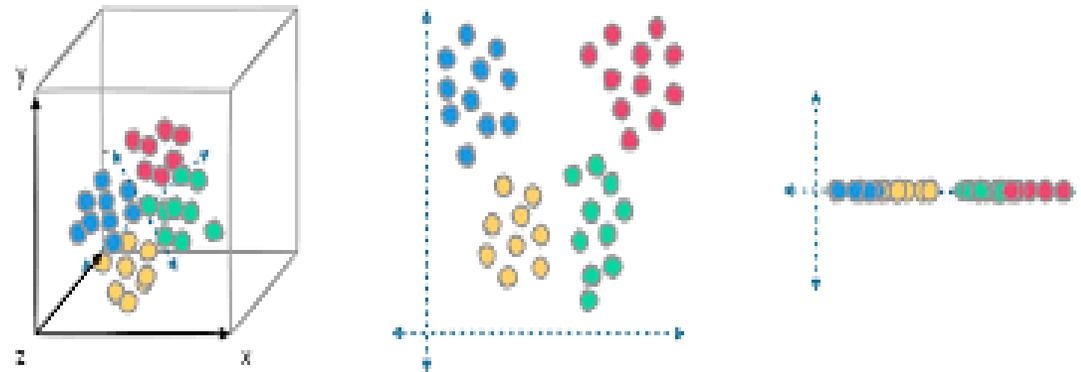
- **Objectif** : Segmenter les clients en groupes similaires selon leurs habitudes d'achat.
- **Données** : Historique des achats des clients.
- **Algorithme** : **K-Means** pour classer les clients en **3 groupes**
 - Clients occasionnels
 - Acheteurs réguliers
 - Clients premium

Réduction de dimensionnalité

- ❑ Consiste à projeter les données dans un espace de dimension inférieure afin de simplifier leur analyse et faciliter la visualisation des données.
- ❑ En réduction de dimensionnalité, le modèle produit un vecteur de caractéristiques contenant moins de dimensions que le vecteur d'entrée

Algorithmes courants :

- ✓ L'Analyse en Composantes Principales (ACP / PCA)
- ✓ t-SNE (t-Distributed Stochastic Neighbor Embedding)
- ✓ L'Auto-encodeur (Autoencoder)



Extraction de Règles d'Association

- Identifier des **relations cachées** entre les variables dans un ensemble de données.

Algorithmes Courants

- ✓ **Algorithme Apriori**
- ✓ **FP-Growth (Frequent Pattern Growth)**
Plus efficace qu'Apriori pour les grandes bases de données.

Apprentissage semi-supervisé

- ❑ Dans ce type d'apprentissage, l'ensemble d'entraînement contient à la fois des exemples étiquetés et non étiquetés.
- ❑ Généralement, le nombre d'exemples non étiquetés est supérieur à celui des exemples étiquetés.
- ❑ L'idée c'est que l'utilisation d'un grand nombre d'exemples non étiquetés puisse aider l'algorithme à produire un modèle plus performant.

Principe de fonctionnement

- On entraîne d'abord un modèle avec les données étiquetées.
- On utilise ce modèle pour faire des prédictions sur les données non étiquetées.
- On sélectionne les prédictions les plus fiables et on les ajoute aux données d'entraînement.
- On réentraîne le modèle avec ces nouvelles données.
- Le modèle devient plus performant en exploitant toute la structure des données, même avec peu d'étiquettes.

Pourquoi utiliser l'apprentissage semi-supervisé ?



➤ Réduire le coût d'annotation des données

Étiqueter un grand volume de données peut être coûteux.

➤ Tirer parti de données disponibles

De nombreux domaines disposent de vastes quantités de données non étiquetées (ex. : reconnaissance d'image, traitement du langage naturel).

➤ Améliorer la généralisation

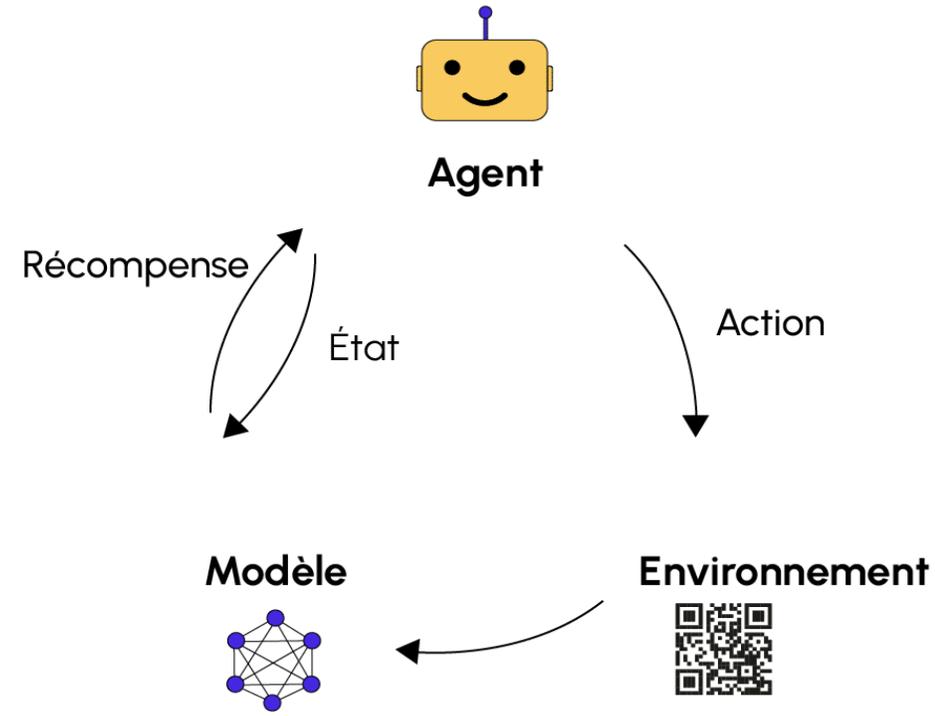
L'utilisation d'un grand nombre de données non étiquetées aide à mieux capturer la structure sous-jacente des données.

Apprentissage par Renforcement (Reinforcement Learning - RL)

- L'apprentissage par renforcement (Reinforcement Learning - RL) désigne l'ensemble des méthodes qui permettent à un agent (machine, programme, robot) d'apprendre à prendre des décisions en **interagissant** avec un **environnement** de manière autonome.
- L'agent apprend en recevant des récompenses ou des pénalités en fonction de ses actions dans son environnement.

Fonctionnement

- 1) l'agent observe l'état actuel de l'environnement.
- 2) Il choisit et exécute une action.
- 3) L'environnement réagit, change d'état et fournit une récompense.
- 4) L'agent met à jour sa stratégie pour maximiser les récompenses futures.



5) Ce cycle se répète jusqu'à ce que l'agent trouve une stratégie optimale (**policy**) qui lui permet de choisir la meilleure action à exécuter à long terme.

Applications de RL

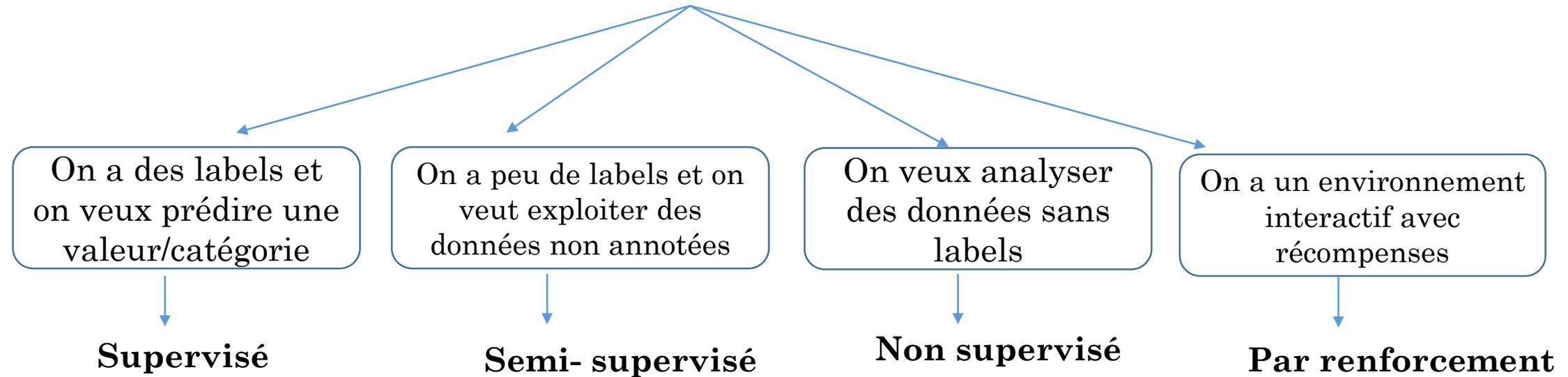
- ✓ **Jeux vidéo**
- ✓ **Robotique** (robots autonomes, drones)
- ✓ **Finance**
- ✓ **Conduite autonome** (véhicules intelligents)



Quel type choisir ?



Le choix dépend des données disponibles et de l'objectif de la tâche



Cycle de Développement d'un Modèle de Machine Learning

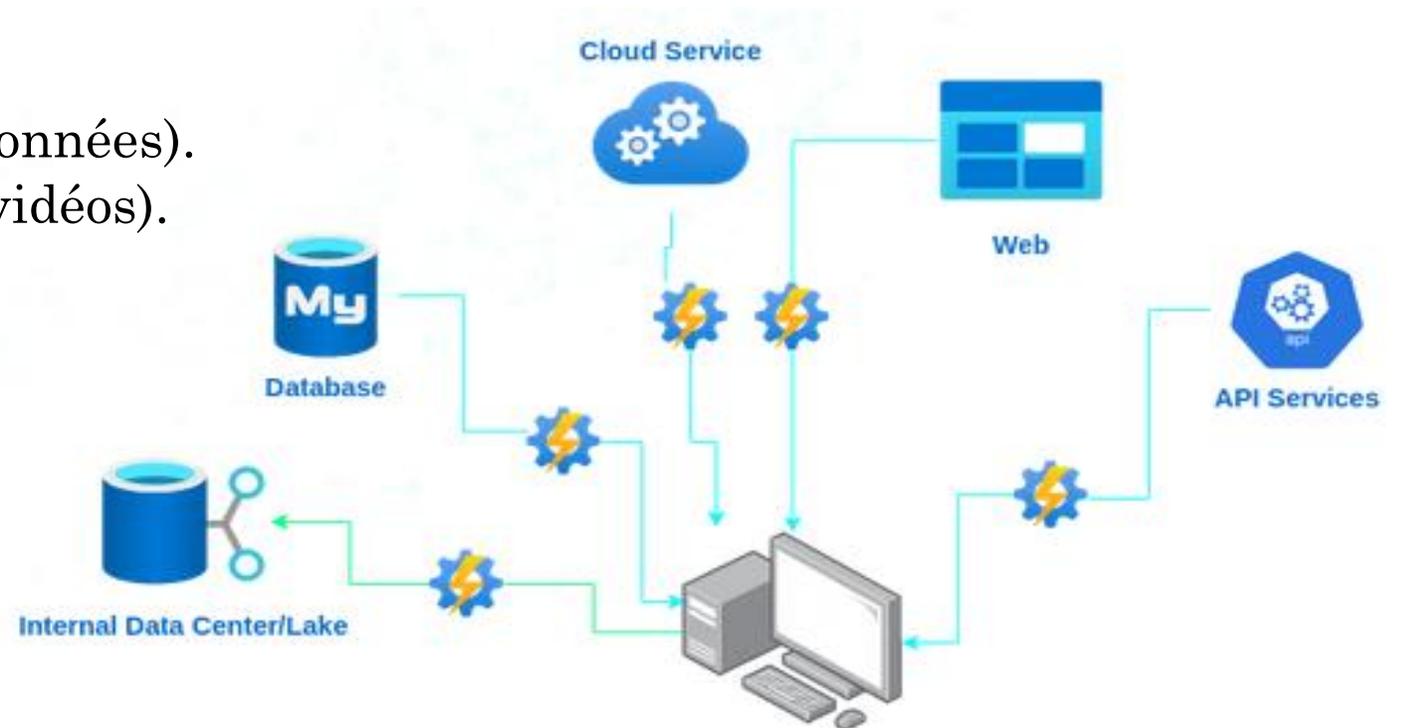
Définition du problème

- ✓ Identifier les objectifs visés.
- ✓ Quel type de données recueillir (numériques, images, vidéos...),
- ✓ Quels résultats attendre (données de sortie)
- ✓ Identifier le type de ML le plus adapté
(supervisé/non supervisé/semi supervisé, par renforcement...).
- ✓ Déterminer les métriques de performance



Collecte des données

- ❑ Données disponibles sur plusieurs sources
(bases de données, serveurs Big Data, sites web, API...)
- ❑ Divers type de données
 - ✓ Structurées (tableaux, bases de données).
 - ✓ Non structurées (textes, images, vidéos).



Préparation des données

❑ Nettoyage des données

- ✓ Gestion des valeurs manquantes,
- ✓ Suppression des doublons,
- ✓ Correction des incohérences.

❑ Transformation des données

- ✓ Normalisation,
- ✓ Encodage des variables catégoriques,
- ✓ Extraction de caractéristiques à partir de texte ou d'images.

❑ Séparation des ensembles d'entraînement, de validation et de test.

À quoi sert chaque ensemble



Préparation des données

- ❑ **Jeu d'entraînement** : Utilisé pour entraîner le modèle.
- ❑ **Jeu de validation** : Utilisé pendant l'entraînement pour ajuster les hyperparamètres (optimiser le modèle).
- ❑ **Jeu de test** : Utilisé après l'entraînement pour évaluer la performance finale du modèle.



Sélection et entraînement du modèle

❑ Choisir l'algorithme de Machine Learning adapté

- ✓ Si l'interprétabilité des résultats est primordiale, il est préférable d'opter pour un modèle simple avec peu de variables (ex : un arbre de décision).
- ✓ En revanche, si l'objectif principal est la performance prédictive, un modèle plus complexe sera privilégié.

❑ Implémenter le modèle et ajuster ses hyperparamètres

Une fois l'algorithme choisi, il est essentiel d'optimiser ses hyperparamètres afin d'améliorer ses performances.

❑ Entraîner le modèle sur l'ensemble d'entraînement prétraité

Améliorer progressivement et de manière itérative la capacité du modèle à résoudre une tâche, tout en minimisant une fonction d'erreur ou de coût...

Évaluation et validation du modèle

- ❑ Tester le modèle sur l'ensemble de validation.
- ❑ Utiliser des métriques de performance appropriées
par exemple: l'accuracy, la précision, le rappel, et la F1-score pour les problèmes de classification, et l'erreur quadratique moyenne (MSE) pour les problèmes de régression.
- ❑ Ajuster les hyperparamètres en fonction des résultats.

Optimisation et amélioration

□ **Feature Engineering (Ingénierie des caractéristiques):**

Améliorer la qualité des données d'entrée (features) du modèle en :

- ✓ **Sélectionnant les variables les plus pertinentes :**
- ✓ **Créant de nouvelles variables**
- ✓ **Transformation des données**



Pourquoi?

Un bon choix de variables d'entrée (caractéristiques) peut améliorer significativement la performance du modèle.

Optimisation et amélioration

❑ Amélioration de l'algorithme ou exploration d'autres modèles.

- ✓ Tester d'autres algorithmes (si le modèle ne donne pas de bons résultats).
- ✓ Modifier la **complexité du modèle** ou **Régulariser** le modèle pour éviter le sur-apprentissage



Pourquoi ?

Certains algorithmes sont plus adaptés à des types de données ou de problèmes spécifiques.

- ## ❑ Améliorer les performances en combinant plusieurs modèles (techniques de bagging, boosting).

Déploiement du modèle

❑ Exportation du modèle

Enregistrer le modèle pour pouvoir l'utiliser ultérieurement, sans avoir à le réentraîner. Plusieurs formats sont utilisés pour cela (Bibliothèque Python (Pickle, Joblib ONNX,..) TensorFlow SavedModel, etc.).

❑ Intégration dans une application ou API

le modèle exporté peut être intégré dans une application ou exposé via une API pour permettre son utilisation par d'autres systèmes. (Flask, FastAPI).

❑ Mise en production et surveillance des performances.

Le modèle doit être surveillé et mis à jour en cas de dégradation des performances.



Merci de
votre
attention