

Intelligence Artificielle (IA) - Généralités -

Groupe des étudiants : CIPA4N

Khadidja OULD AMER

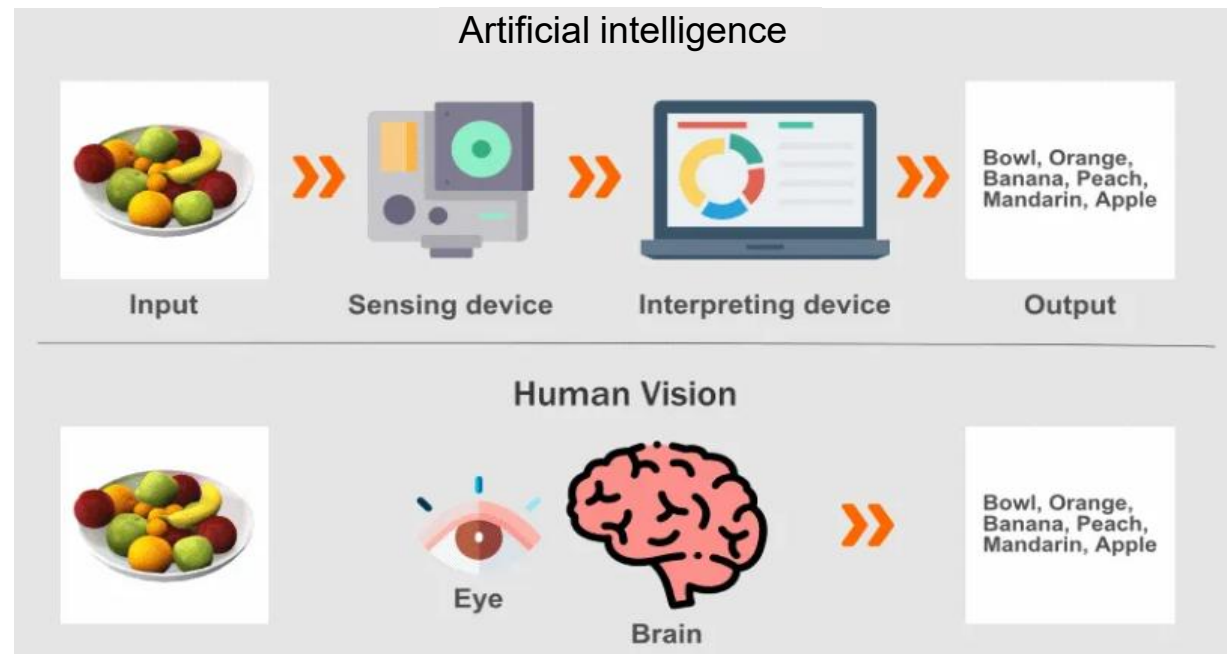
E-Mail : khadidja.ouldamer@isen-ouest.yncrea.fr

Informations pratiques

- Volume horaire : 15h
- Organisation des séances :
 - Séance 1 : Généralités (CM1 + TP1)
 - Séance 2 : CM2 + TP2
 - Séance 3 : TP3 + TP4
 - Séance 4 : TP4 + évaluation (1h)

Qu'est ce que l'IA ?

L'intelligence artificielle (IA) désigne l'ensemble des techniques, programmes et algorithmes qui permettent à une machine d'imiter certaines fonctions cognitives humaines, telles que l'apprentissage, le raisonnement, la résolution de problèmes, la perception ou encore la prise de décision.

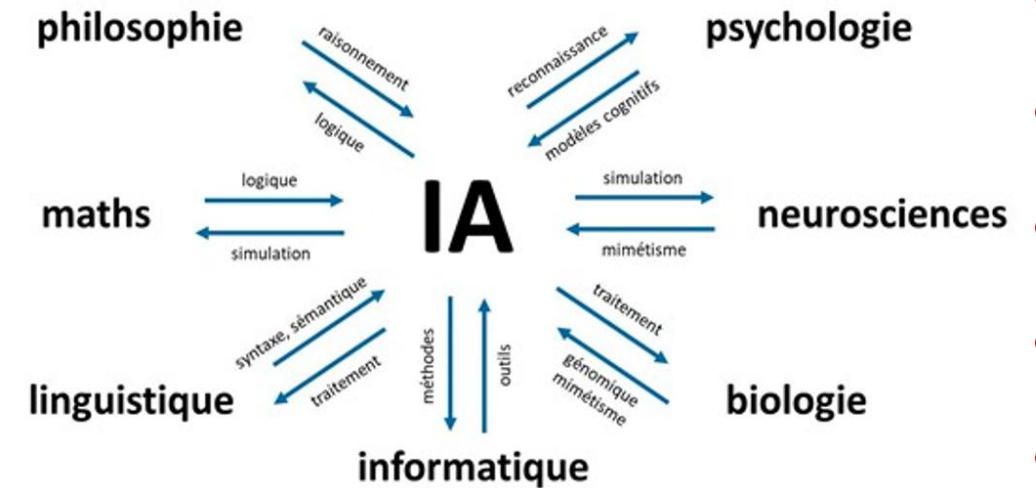


[1] https://fr.wikipedia.org/wiki/Intelligence_artificielle

Qu'est ce que l'IA ?

Un domaine interdisciplinaire

- **Neurosciences** : s'inspirer du fonctionnement du cerveau
- **Mathématiques** : modéliser et simuler des systèmes
- **Linguistique** : comprendre et traiter le langage naturel
- **Informatique** : concevoir des algorithmes performants



[1] https://fr.wikipedia.org/wiki/Intelligence_artificielle

Applications



Chatbots



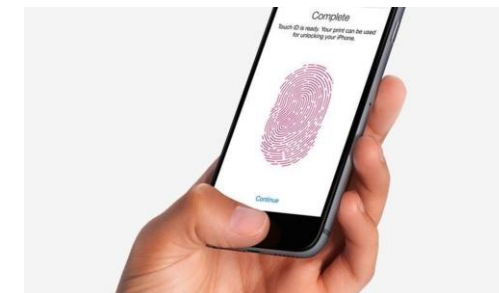
Systèmes de recommandation



Assistants virtuels



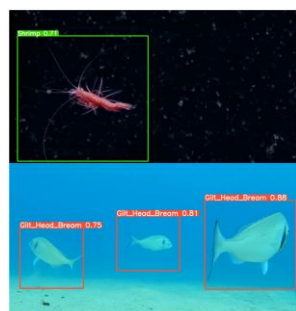
Traduction automatique



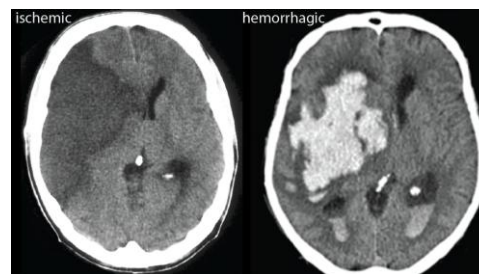
Reconnaissance des empreintes



Reconnaissance faciale



Environnement



Aide au diagnostic



Robotique



Voiture autonome



A little girl is sitting in front of a large painted rainbow .

Description automatique des images 2D

Intelligence artificielle

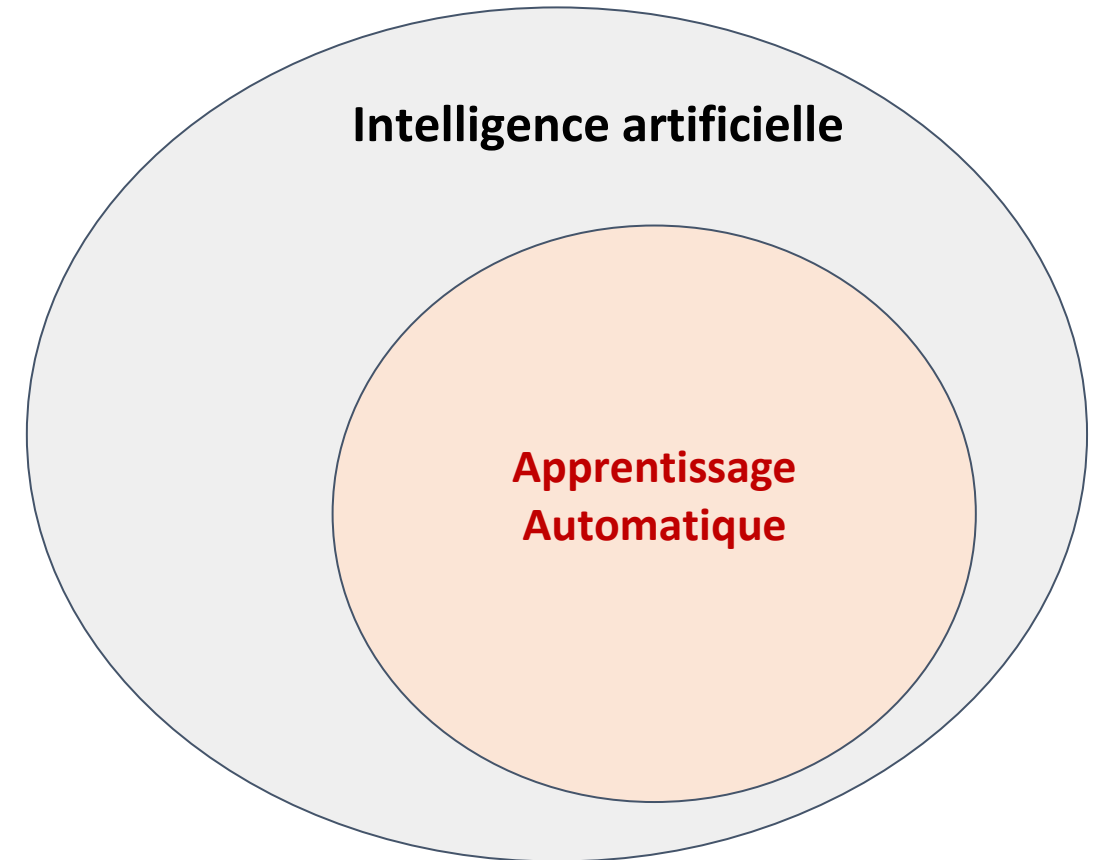
Domaine de l'informatique qui regroupe l'ensemble des technologies visant à créer des systèmes capables de reproduire ou d'imiter certaines formes d'intelligence humaine.

https://fr.wikipedia.org/wiki/Intelligence_artificielle

Disciplines de l'intelligence artificielle

Apprentissage Automatique (Machine Learning)

Son objectif est de permettre à un système **d'apprendre à partir de données** à l'aide d'algorithmes mathématiques et statistiques. Plutôt que d'être programmé explicitement pour exécuter une tâche spécifique, le système est entraîné sur un jeu de données. Il identifie alors des relations pour construire un modèle, lui permettant d'améliorer ses performances de manière autonome et de réaliser des prédictions sur de nouvelles données [1].

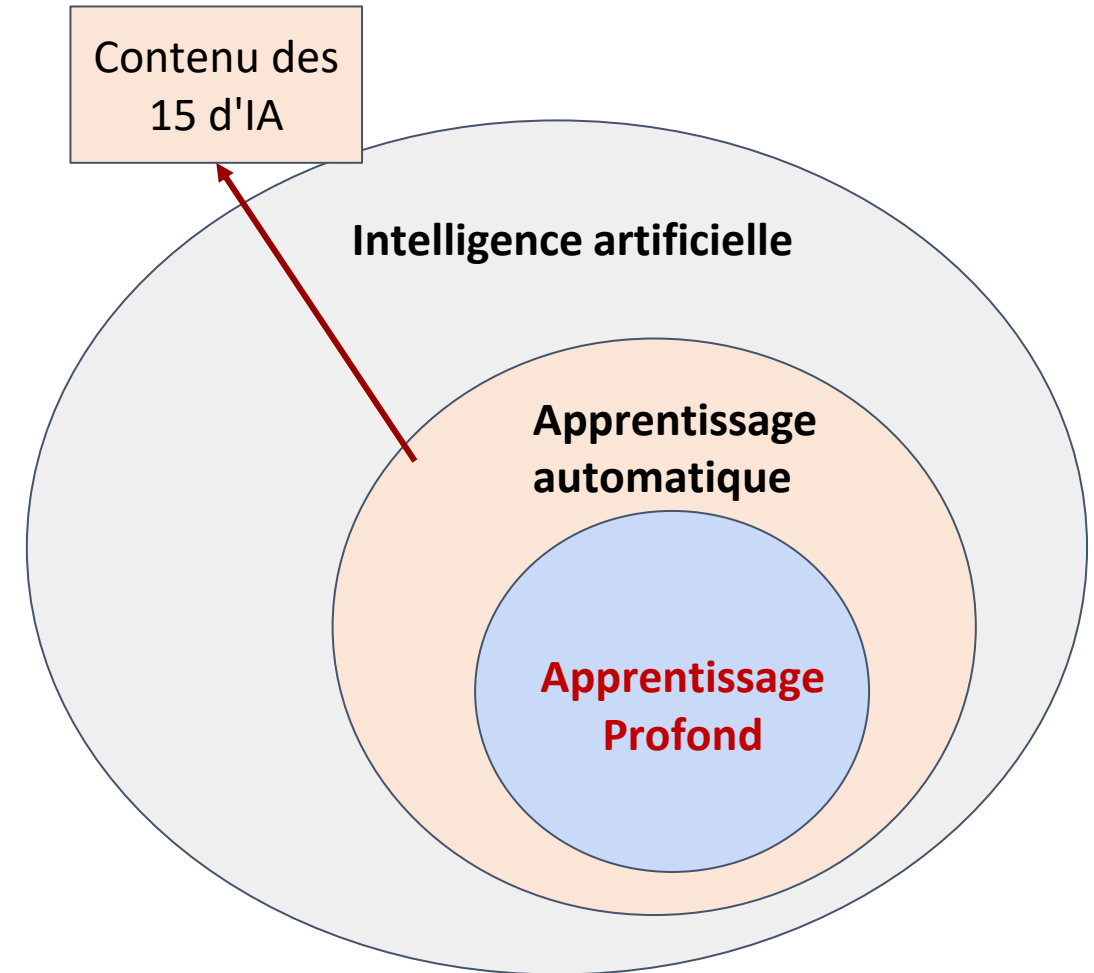


https://fr.wikipedia.org/wiki/Apprentissage_automatique

Disciplines de l'intelligence artificielle

Apprentissage Profond (Deep Learning)

Sous-ensemble spécifique de l'apprentissage automatique. Le principe reste le même (entraîner des modèles à partir de données). La différence essentielle réside dans les modèles utilisés : l'apprentissage profond s'appuie sur des **réseaux de neurones artificiels** qui permettent de modéliser les données avec un très haut niveau d'abstraction et d'en extraire des caractéristiques complexes.



https://fr.wikipedia.org/wiki/Apprentissage_profond

Qu'est-ce que l'Apprentissage Automatique?

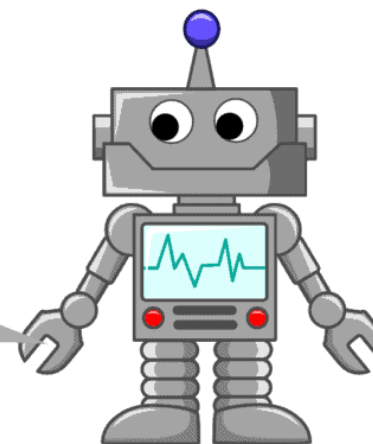
Un domaine de l'IA qui donne aux systèmes la capacité **d'apprendre automatiquement à partir de données** pour prédire des résultats **sans être explicitement programmés pour chaque tâche.**

Humain



Je peux tout apprendre automatiquement à partir de mes expériences.
Est-ce ton cas ?.

Ordinateur

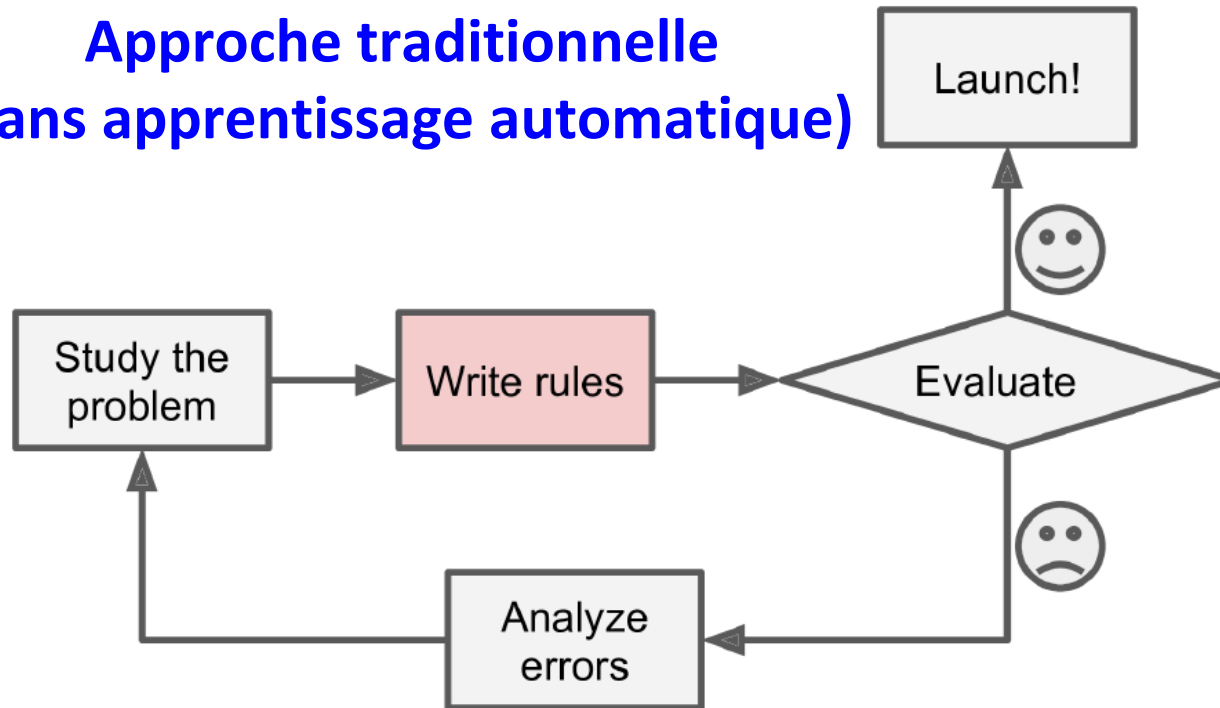


Oui, je peux aussi apprendre à partir de données passées avec l'aide de l'apprentissage automatique.

Apprentissage automatique

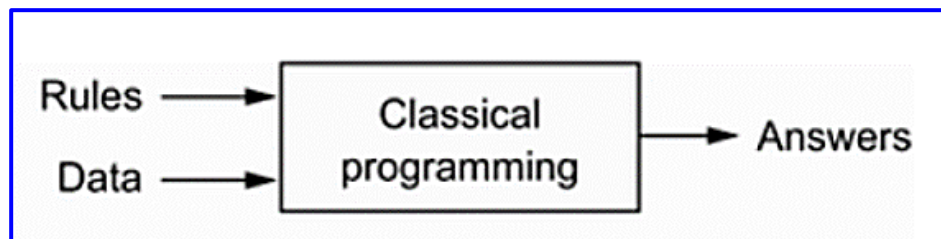
- **Exemple** : Détection de SPAM dans les emails (spam ou non-spam)

Approche traditionnelle (sans apprentissage automatique)



Limites :

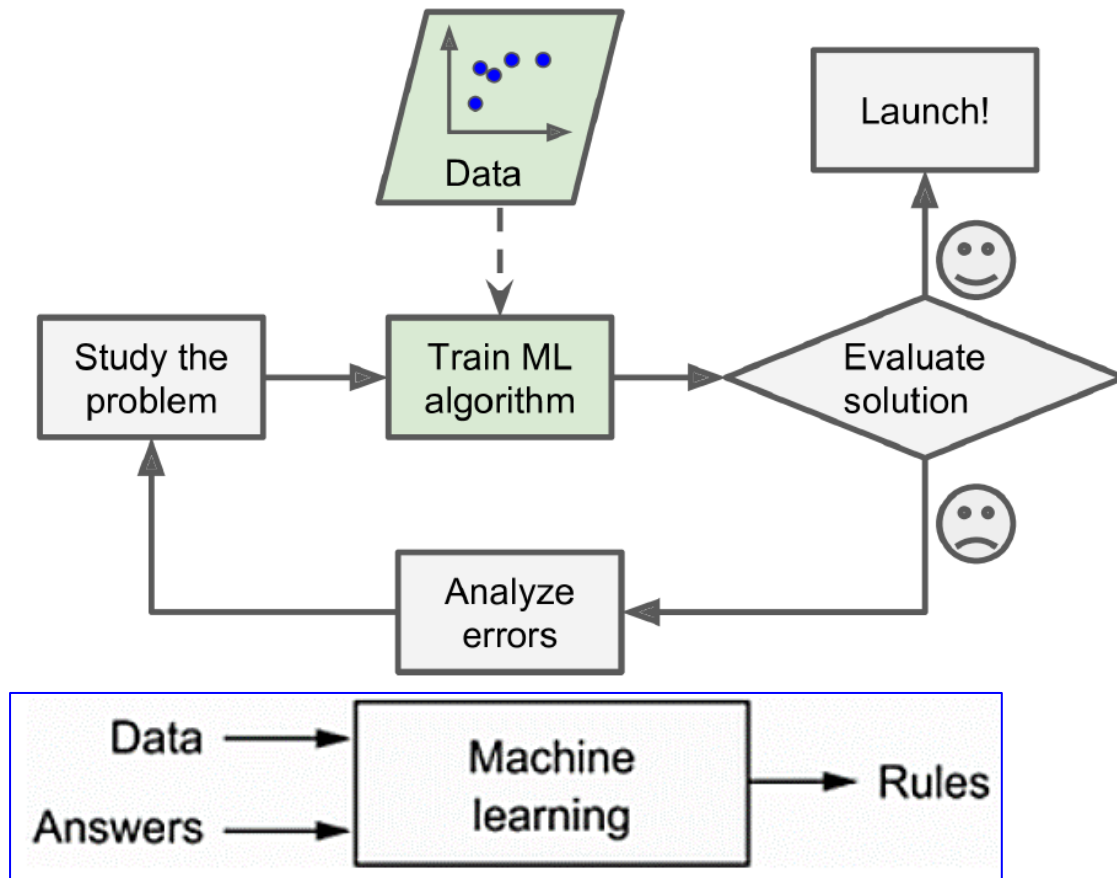
- Approche rigide basée uniquement sur des règles manuelles.
- Inefficace face à l'évolution rapide des techniques des spam.



Apprentissage automatique

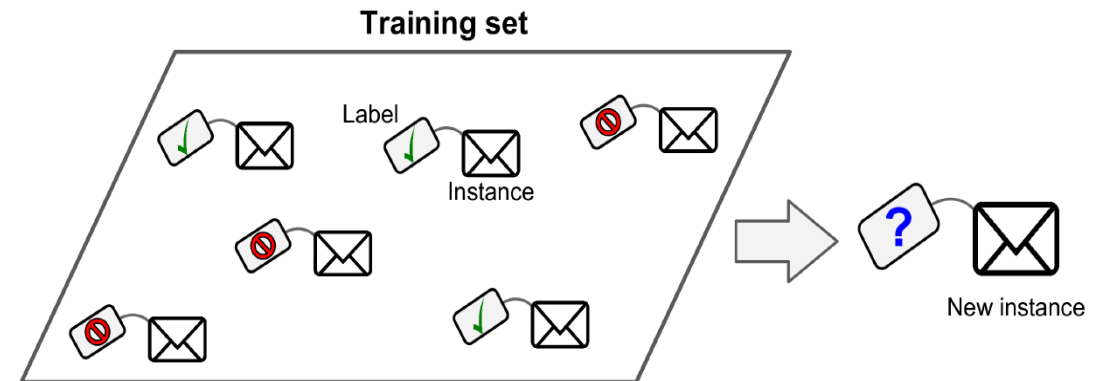
- **Exemple** : Détection de SPAM dans les emails (spam ou non-spam)

Approche basée sur l'apprentissage automatique



Solution :

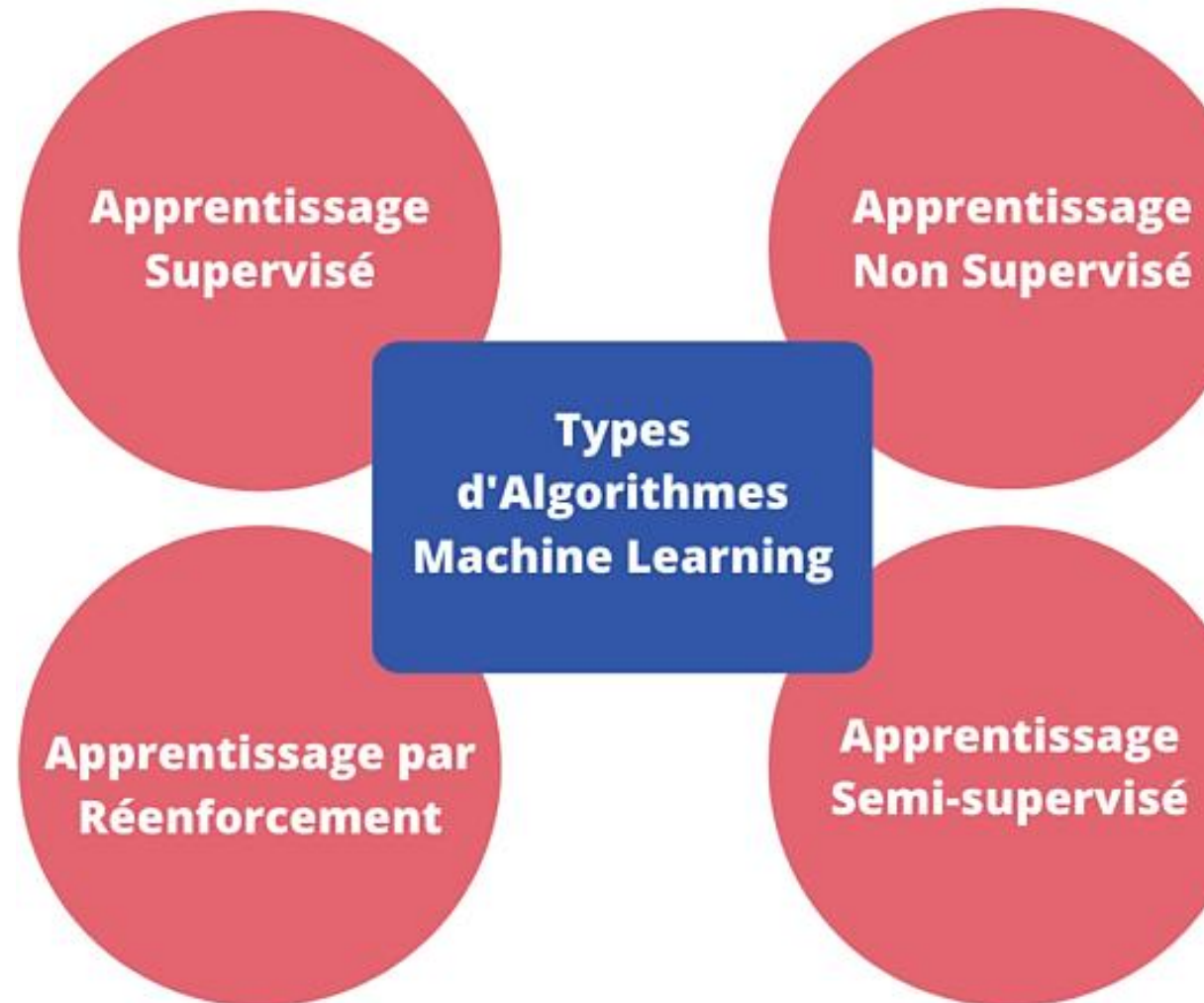
- Au lieu d'écrire des règles, on entraîne un modèle sur des emails étiquetés (spam/non-spam) afin qu'il puisse classer automatiquement de nouveaux emails.
- Plus flexible et efficace.



Les variables peuvent être : fréquence de certains mots, présence de pièces jointes, longueur du message, etc.
La sortie est discrète (classe : spam / non-spam).

Types des systèmes d'apprentissage automatique

Méthodes d'apprentissage automatique



Apprentissage supervisé

Apprentissage supervisé : un mode d'apprentissage où un modèle apprend à établir une relation entre des **données d'entrée (les *features*)** et une **sortie connue (la variable cible)**. L'entraînement se fait à partir d'un ensemble d'exemples annotés, dans lequel chaque entrée est associée à la bonne réponse. L'objectif est que le modèle puisse ensuite prédire correctement la sortie pour de nouvelles données qu'il n'a jamais vues.

Selon la variable cible, on distingue deux types de problèmes :

La classification

Prédire une **classe** ou **catégorie** discrète.

Exemples : détection de spam (spam / pas spam),
résultat d'un test médical (positif / négatif),
reconnaissance d'un animal (chat, chien, etc).

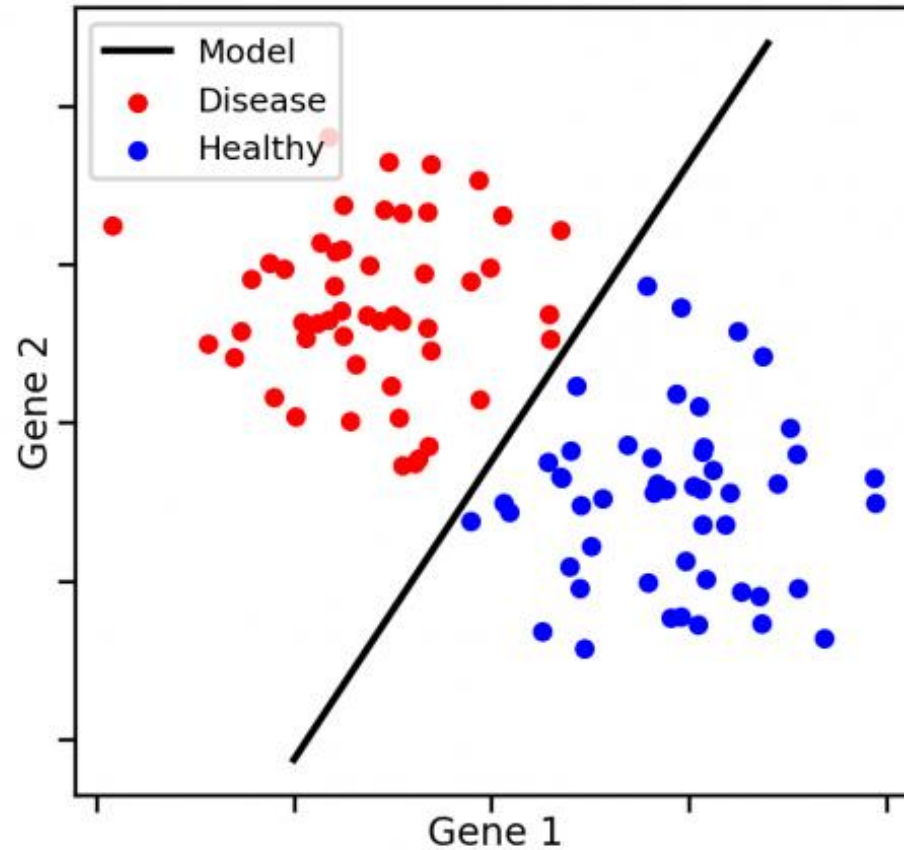
La régression

Prédire une **valeur numérique** continue.

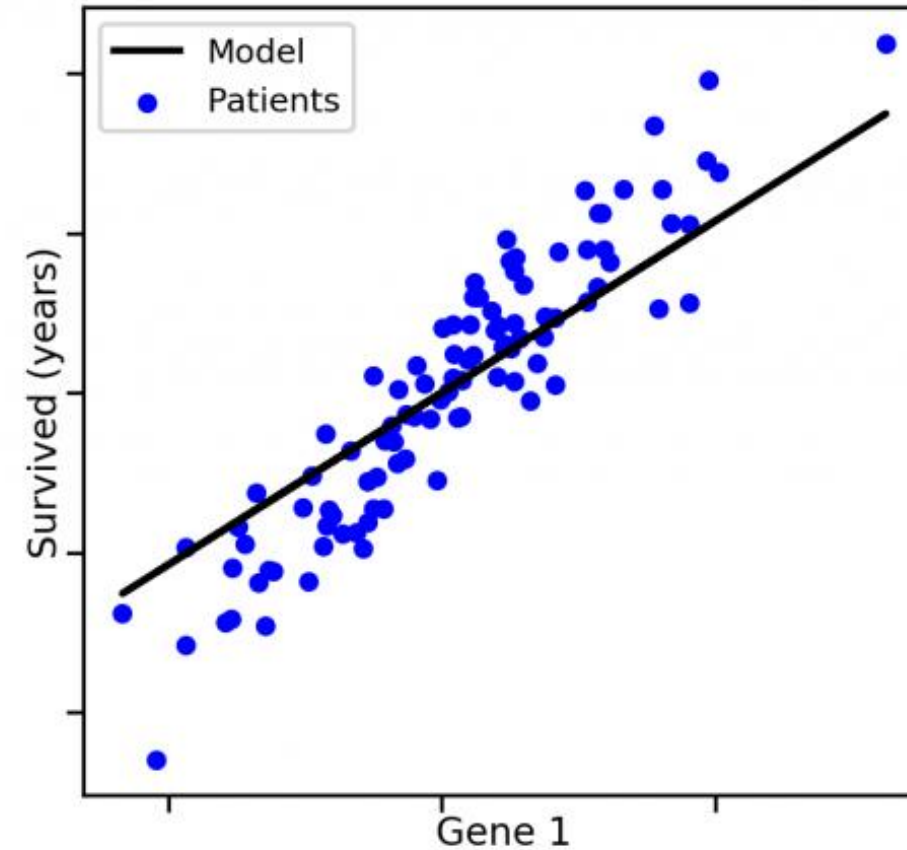
Exemples : prix d'un logement, température,
chiffre d'affaires.

Apprentissage supervisé

Classification



Regression



Apprentissage supervisé

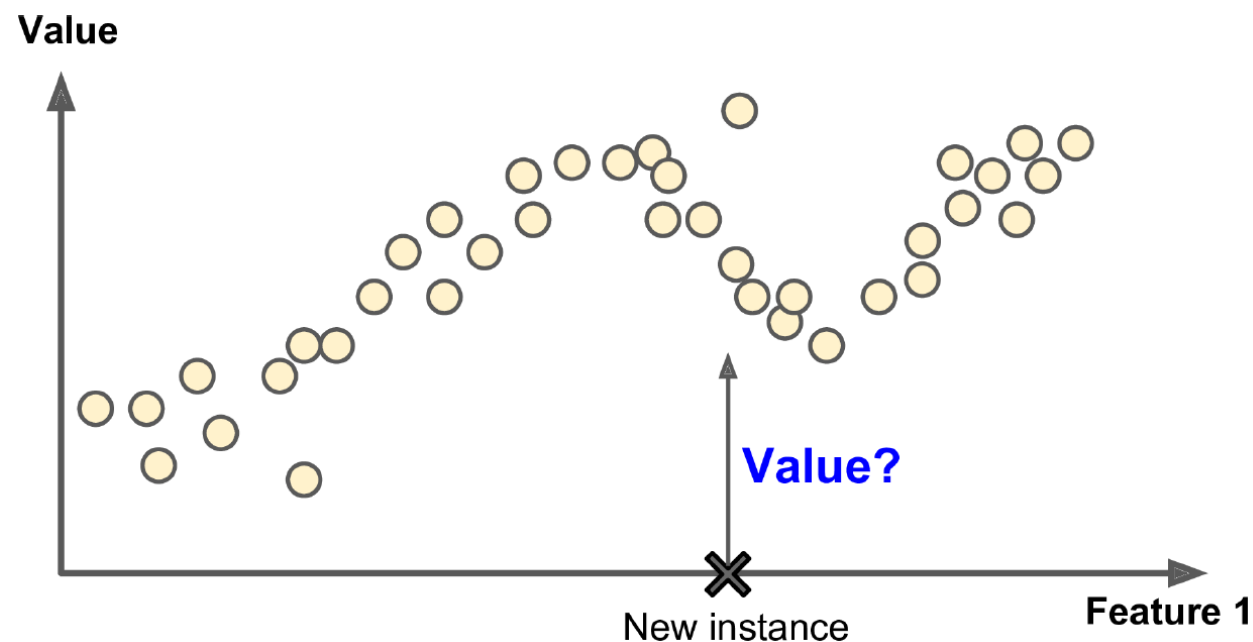
Exemples de régression :

Prédire le prix d'une voiture à

partir de ses caractéristiques :
kilométrage, âge, marque, etc.

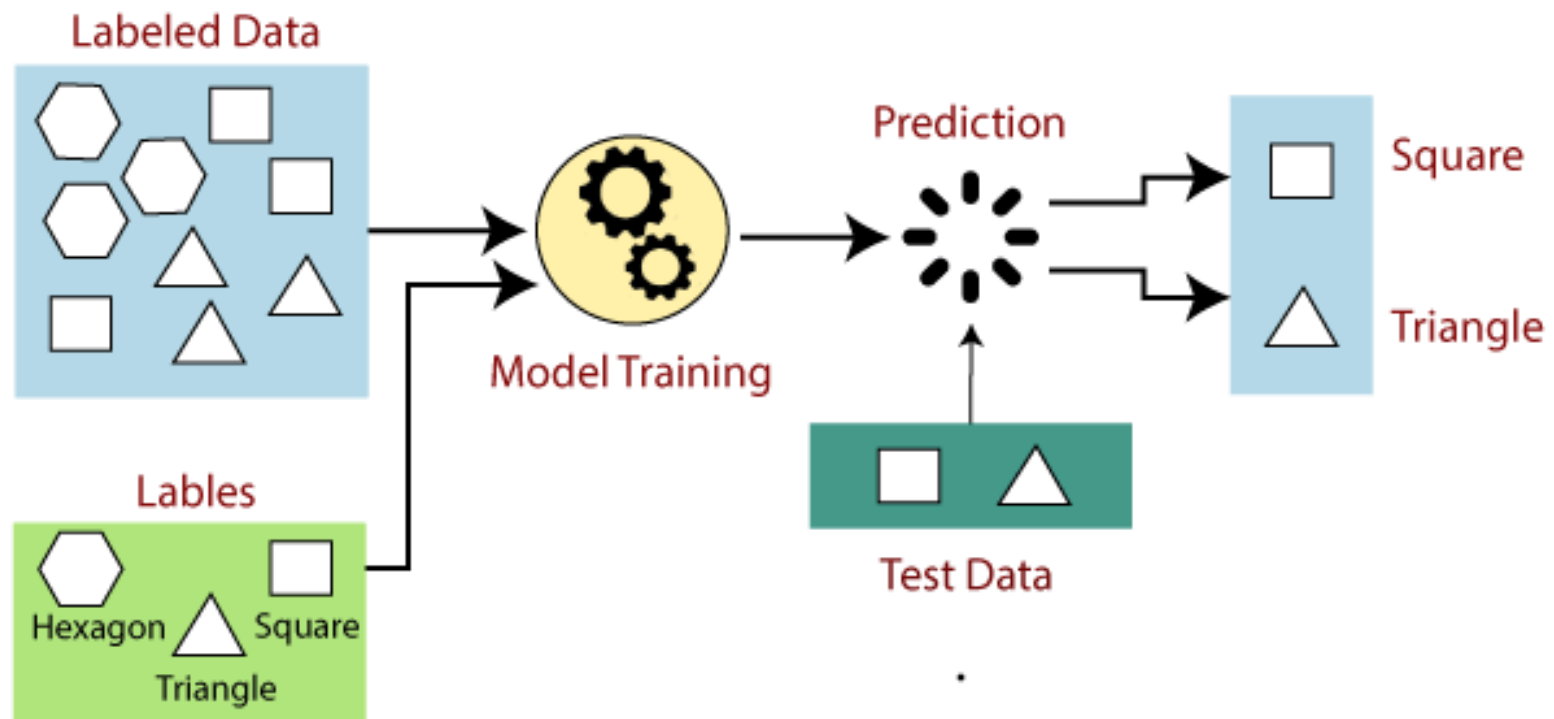
Estimer le prix d'une maison

à partir de variables telles que :
superficie (m²), nombre de chambres,
localisation, année de construction, etc.



Apprentissage supervisé

Exemple de classification : Détection des formes



Apprentissage supervisé

Quelques méthodes d'apprentissage supervisé :

- o K-plus proche voisin
- o Régression linéaire
- o Régression logistique
- o Support Vector Machines
- o Arbre de décision
- o

https://scikit-learn.org/stable/supervised_learning.html

Apprentissage non-supervisé

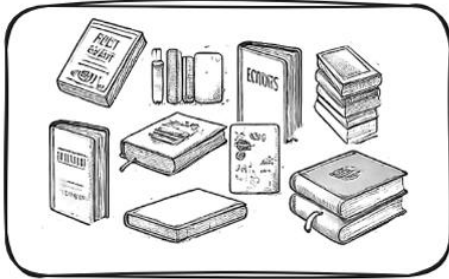
Apprentissage non supervisé

Le modèle est entraîné sur un jeu de données **sans valeurs cibles associées**.

L'objectif n'est pas de prédire une sortie connue, mais d'**explorer la structure interne des données** afin de mettre en trouver :

- des **regroupements naturels** (clustering),
- des **relations ou corrélations cachées** entre les variables,
- une **représentation simplifiée** des données (réduction de dimensionnalité).

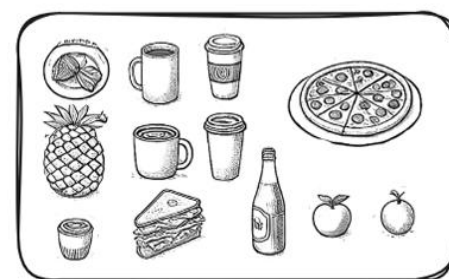
Apprentissage non-supervisé



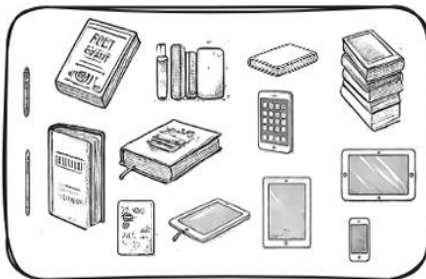
Livres



Appareils électroniques



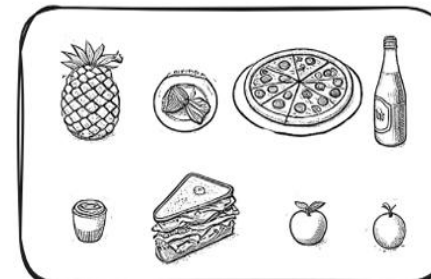
Nourriture



Livres et tablettes



Accessoires Bureau

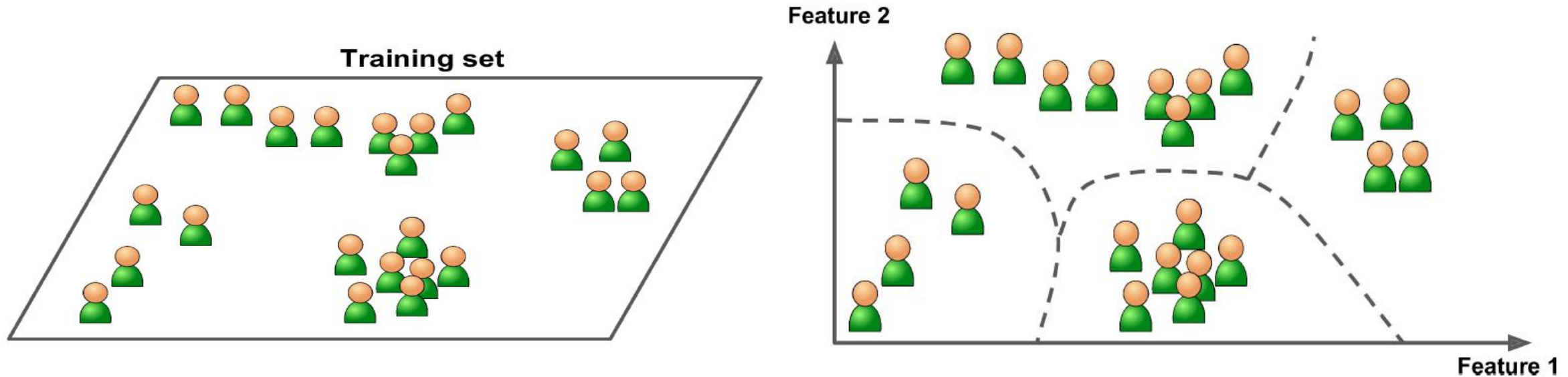


Nourriture

Catégorisation intuitive (en haut) vs. Catégorisation machine (en bas)

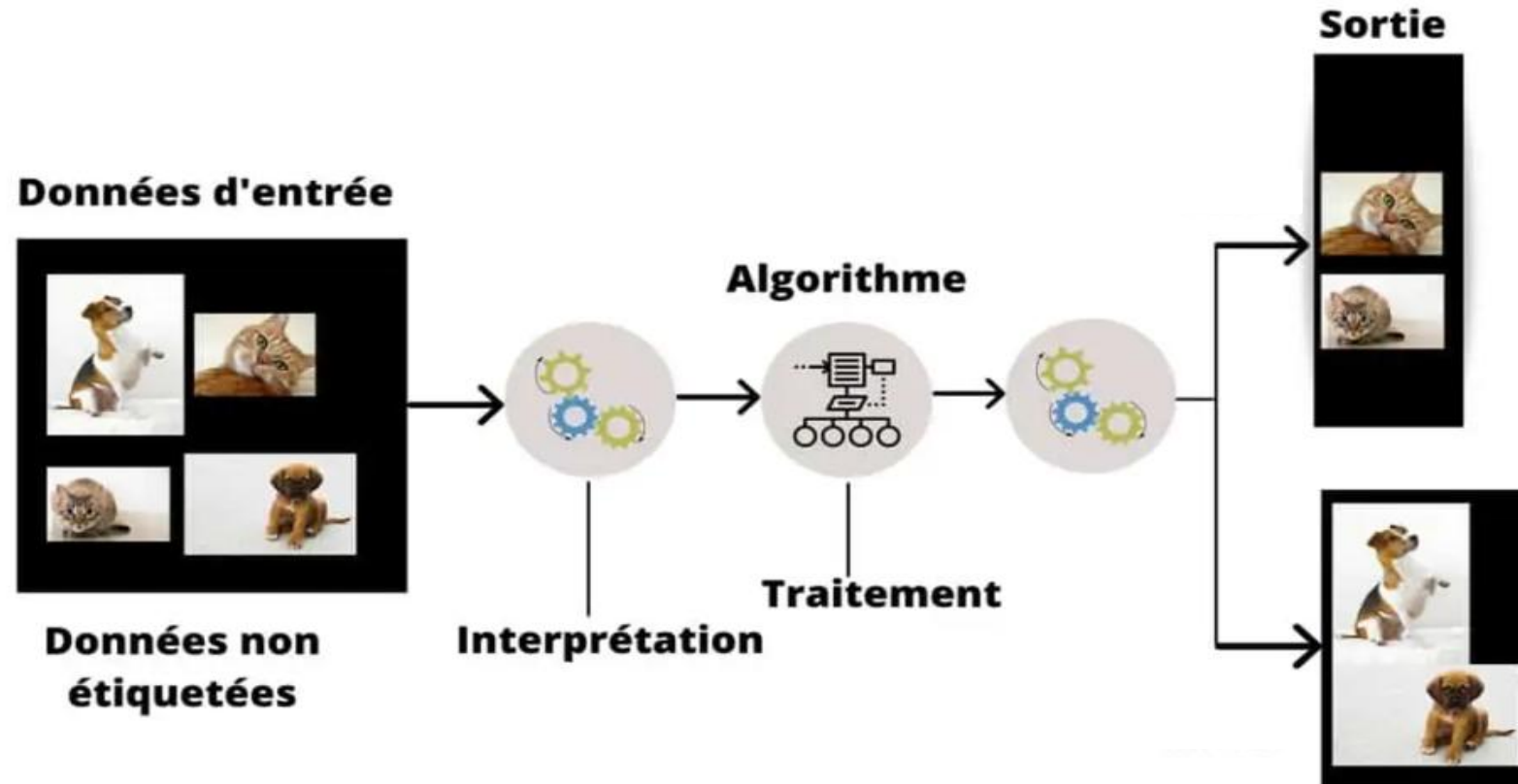
<https://www.machinelearnia.com/>

Apprentissage non-supervisé



Exemple: Segmentation des visiteurs d'un blog selon leur comportement de navigation (durée de visite, pages vues, fréquence, etc.). Les résultats doivent être interprétés par un humain pour en tirer du sens. Par exemple, les visiteurs peuvent être segmentés en clusters tels que : lecteurs réguliers, nouveaux visiteurs, utilisateurs inactifs, etc.

Apprentissage non-supervisé



Le modèle regroupe les données par similarité sans qu'on lui ait explicitement fourni les étiquettes "chien" et "chat".

Apprentissage non-supervisé

- Quelques méthodes :
 - **Le clustering** : une technique essentielle dans l'apprentissage non-supervisé qui permet de regrouper des points de données **non étiquetés** en catégories/clusters (groupes). Les points de données sont attribués à un groupe en fonction de leur similarité.
 - K-Means
 - DBSCAN
 - Hierarchical Cluster Analysis



<https://scikit-learn.org/stable/modules/clustering.html>

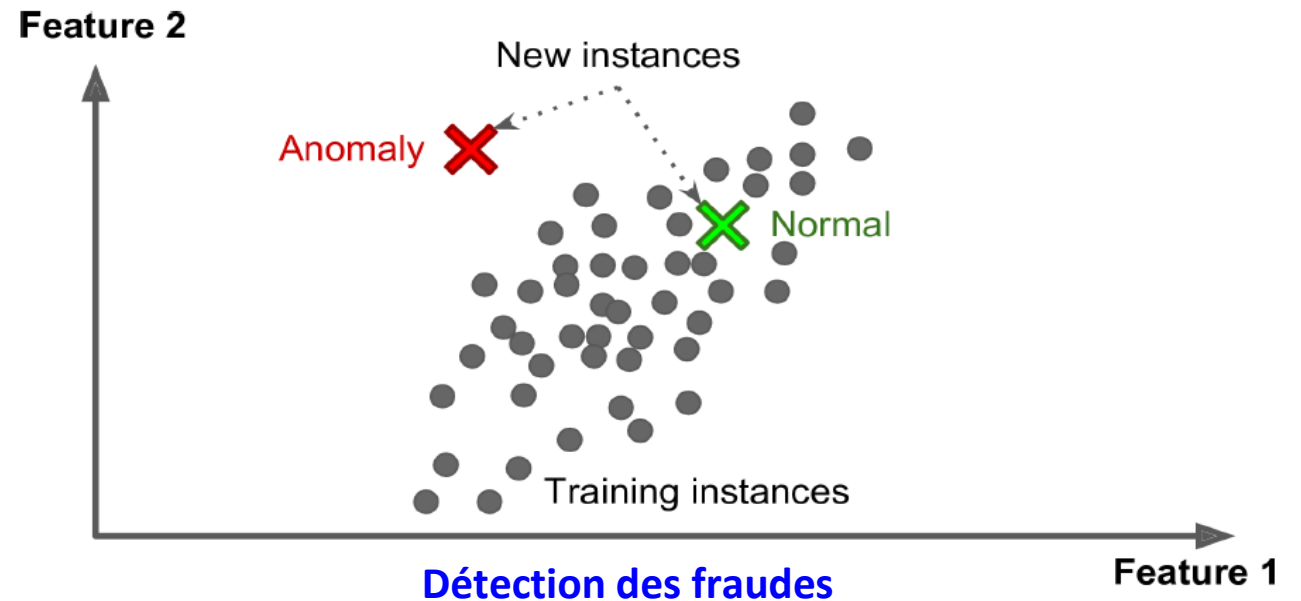
Apprentissage non-supervisé

- Quelques méthodes :
 - **Détection d'anomalies** : détection des valeurs aberrantes dans un ensemble de données.

➤ **One-class SVM** : apprend ce qui est "**normal**" à partir de données d'entraînement et utilise cette connaissance pour identifier les anomalies.

➤ **Applications :**

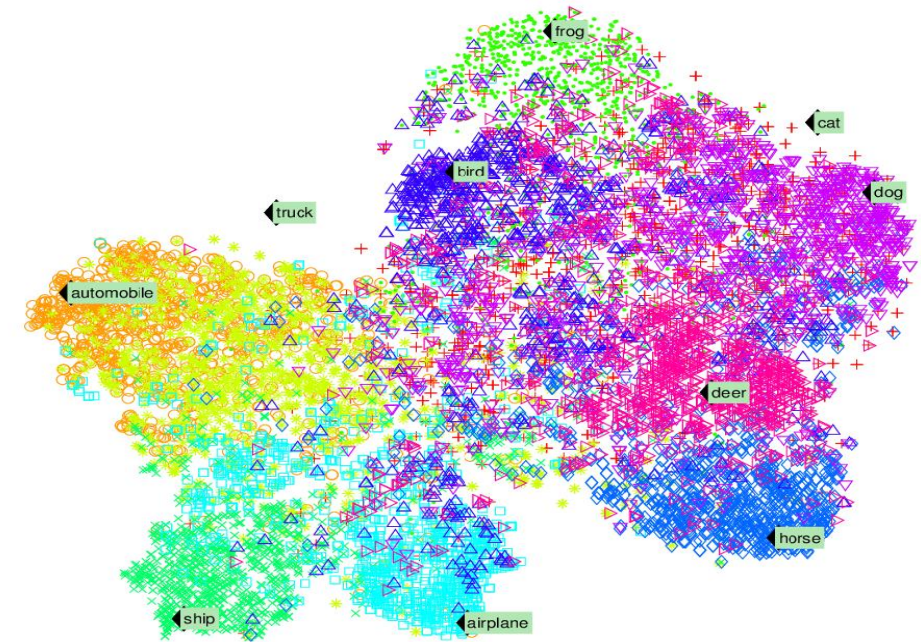
- Détection de fraudes,
- Surveillance des pannes,
- Cybersécurité



Exp. Si une transaction avec un montant très élevé ou effectuée dans un pays inhabituel apparaît, elle sera détectée comme une anomalie.

Apprentissage non-supervisé

- Quelques méthodes :
 - **Visualisation et réduction de dimensionnalité** : pour simplifier des données complexes tout en préservant leurs principales caractéristiques.
 - Principal Component Analysis (PCA)
 - Kernel PCA
 - t-distributed Stochastic Neighbor Embedding (t-SNE)



Apprentissage non-supervisé

- Quelques méthodes :
 - **Apprentissage des règles d'association** : consiste à découvrir des relations significatives (appelées *associations*) entre différents éléments dans un jeu de données.
 - Exemple : lorsqu'un client achète du pain et des œufs, il est probable qu'il achète également du lait. Ces relations permettent de créer des promotions ciblées, comme : "*Pain + Œuf*" = *Réduction sur le Lait*.

{Pain, Oeuf} \longrightarrow {Lait}

Itemset = {Pain, Oeuf, Lait}

Apprentissage semi-supervisé

Apprentissage semi-supervisé : approche hybride combinant :

- **Apprentissage supervisé** → basé sur des données **étiquetées**
- **Apprentissage non supervisé** → basé sur des données **non étiquetées**

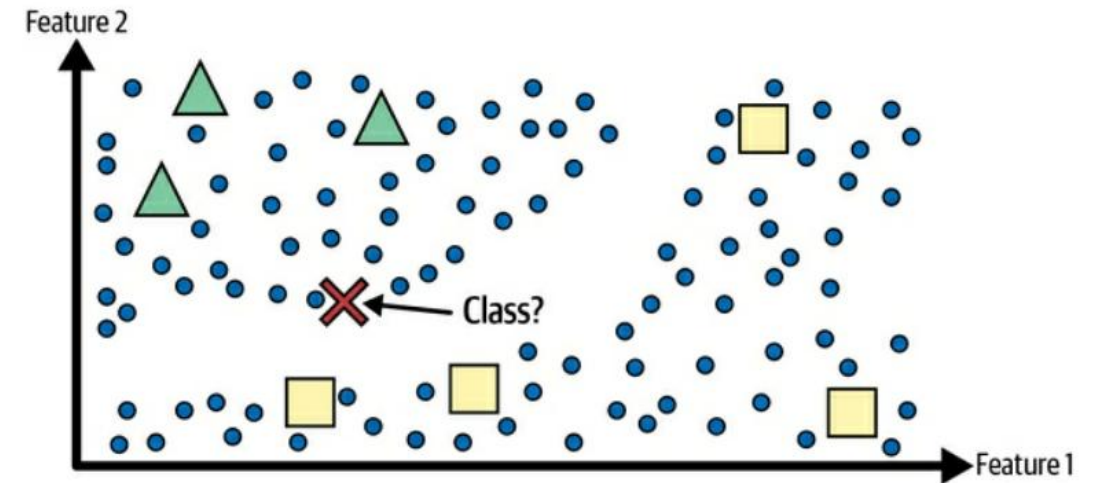
Avantages :

- Exploiter efficacement un grand volume de données non étiquetées, avec une petite quantité de données étiquetées
- Favorise la généralisation et réduit le risque de surapprentissage (overfitting)
- Réduit le coût et le temps liés à l'annotation manuelle des données

Apprentissage semi-supervisé

Principe de fonctionnement

- Le modèle apprend sur les données étiquetées.
- Il attribue des pseudo-étiquettes aux données non étiquetées en fonction de leur proximité avec les données réellement annotées.
- L'entraînement se fait simultanément sur vraies étiquettes + pseudo-étiquettes.
- Le modèle généralisé peut alors prédire correctement de nouvelles données.



Semi supervisé avec 2 classes étiquetées (traingles et carrés)

Apprentissage semi-supervisé

- **Exemple** : chargement des photos dans Google Photos
 - o **Partie non-supervisée** : Google analyse les photos et regroupe les visages similaires (sans savoir qui ils sont):
 - la personne A apparaît dans les photos 1, 5 et 11
 - la personne B apparaît dans les photos 2, 5 et 7
 - o **Partie supervisée** : l'utilisateur ajoute les noms aux visages.



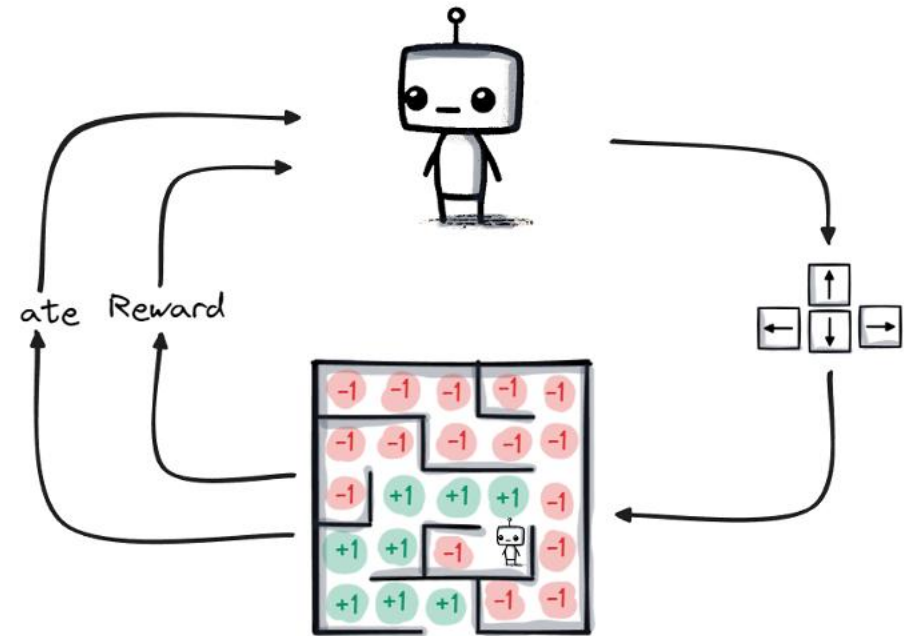
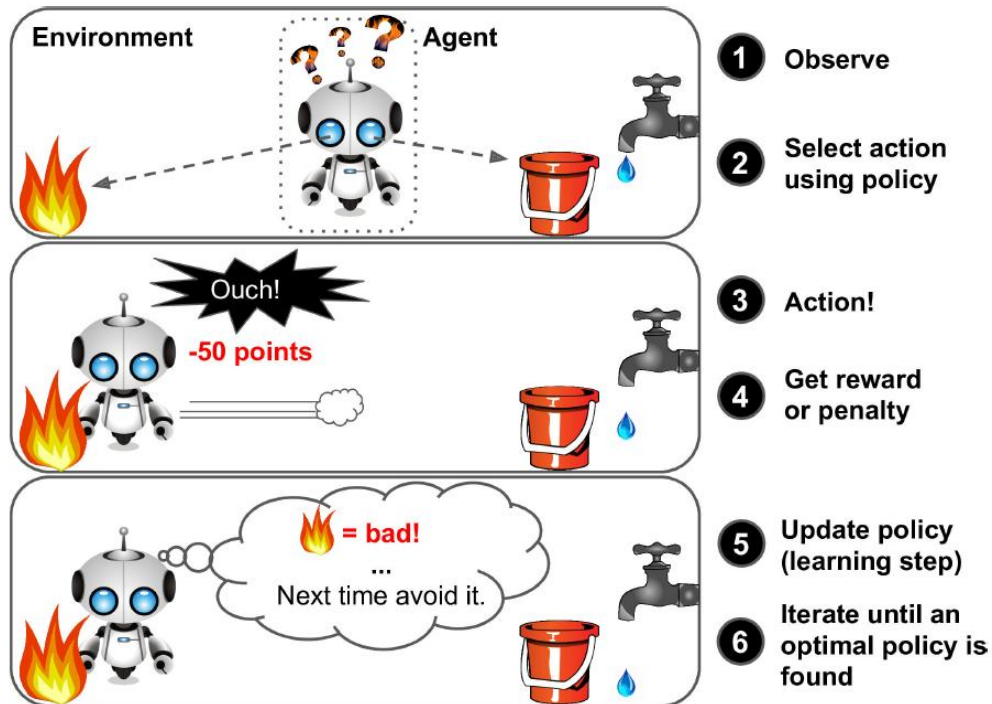
Ajouter un nom

Retrouvez rapidement vos photos en faisant une recherche par nom

- o **Résultat** : le modèle est entraîné avec ces informations pour reconnaître automatiquement les personnes dans toutes les autres photos.

Apprentissage par renforcement

L'**apprentissage par renforcement** est une méthode d'IA où un **agent** (par exemple, un robot, un programme, un joueur virtuel) apprend à prendre des décisions optimales en interagissant avec un environnement et en recevant des **récompenses** (positives ou négatives) en retour de ses actions.



Apprentissage par renforcement

Apprentissage par renforcement :

- **Exemples :**
 - i. Voiture autonome
 - ii. Robotiques
 - iii. Jeux vidéo
 - iv. Jeux de société : DeepMind's AlphaGo
 - v.



Types de méthodes d'apprentissage automatique

1. Apprentissage via la supervision humaine ?

- apprentissage supervisé (supervised learning)
- apprentissage non-supervisé (unsupervised learning)
- apprentissage semi-supervisé (semisupervised learning)
- apprentissage par renforcement (reinforcement learning)

2. Mode d'apprentissage :

- **Apprentissage hors ligne (batch learning)**

Le modèle est entraîné sur un ensemble de données complet et fixe.
Aucune mise à jour après la phase d'entraînement initiale.

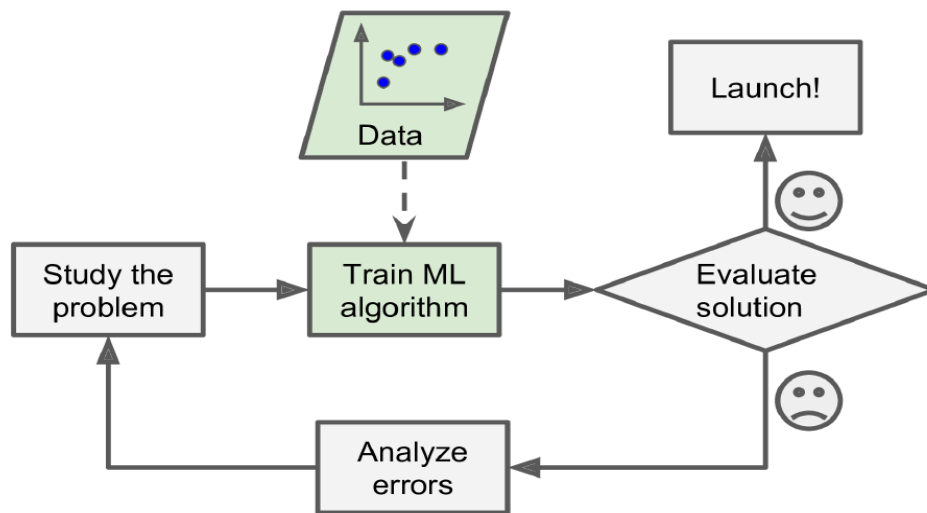
- **Apprentissage en ligne (online learning)**

Le modèle apprend de façon continue et incrémentale.
Mise à jour permanente au fur et à mesure de l'arrivée des nouvelles données.

Apprentissage en ligne VS Apprentissage hors ligne

Apprentissage hors ligne (batch learning) :

- Le modèle est entraîné sur un ensemble de données complet et fixe.
- Temps d'apprentissage long, surtout avec de gros volumes.
- Modèle figé après déploiement → réentraînement complet nécessaire pour intégrer de nouvelles données.



Exemple :

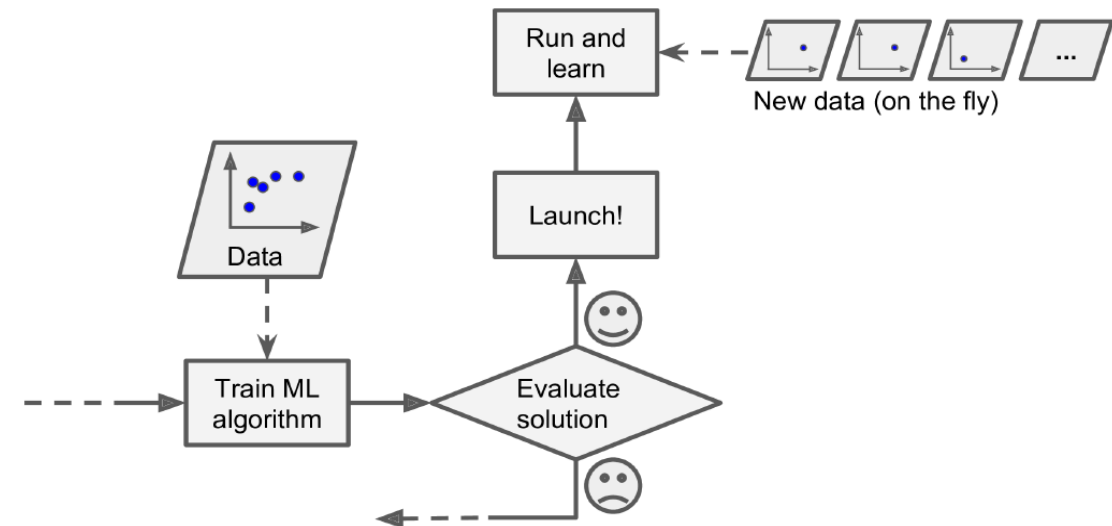
Un modèle de reconnaissance d'images entraîné sur un large dataset comme ImageNet. Une fois entraîné, le modèle reste figé sauf si un nouvel apprentissage complet est relancé avec des données supplémentaires.



<https://www.image-net.org/>

Apprentissage en ligne VS Apprentissage hors ligne

- **Apprentissage en ligne (online learning) :**
Le modèle apprend de façon continue et incrémentale.
Mise à jour permanente au fur et à mesure de l'arrivée des nouvelles données.
- **Adapté aux flux continus :** pour les données en temps réel (streaming, surveillance, etc.).
- **Exemple :** Lorsqu'un nouvel e-mail est étiqueté comme « spam », le modèle met à jour ses paramètres instantanément, sans nécessiter de réentraînement sur l'historique complet.
- **Efficacité des ressources :** Consommation mémoire réduite, car les données sont traitées séquentiellement par lots.



Etapes d'un projet d'apprentissage automatique

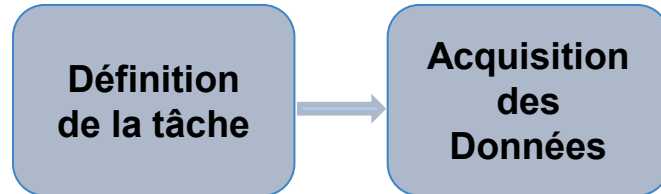
Etapes d'un projet d'apprentissage automatique

Définition de la tâche

- **Cadrer le problème :**
 - Input et output
 - Type d'apprentissage automatique
 - Etude des solutions existantes



Etapes d'un projet d'apprentissage automatique



➤ Collection et compréhension des données :

- Téléchargement
- Informations sur les données
- Découverte et visualisation des données
- Stockage
- Annotation

Exemple de la régression

$$Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(2000)} \end{bmatrix} = \begin{bmatrix} 5.67 \\ \vdots \end{bmatrix}$$

$$X = \begin{bmatrix} (\mathbf{x}^{(1)})^T \\ (\mathbf{x}^{(2)})^T \\ \vdots \\ (\mathbf{x}^{(1999)})^T \\ (\mathbf{x}^{(2000)})^T \end{bmatrix} = \begin{pmatrix} -118.29 & 33.91 & 1,416 & 38,372 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

	Classes = Labels	Features		
Instance 1		attribut 1	attribut m
⋮				
Instance n				

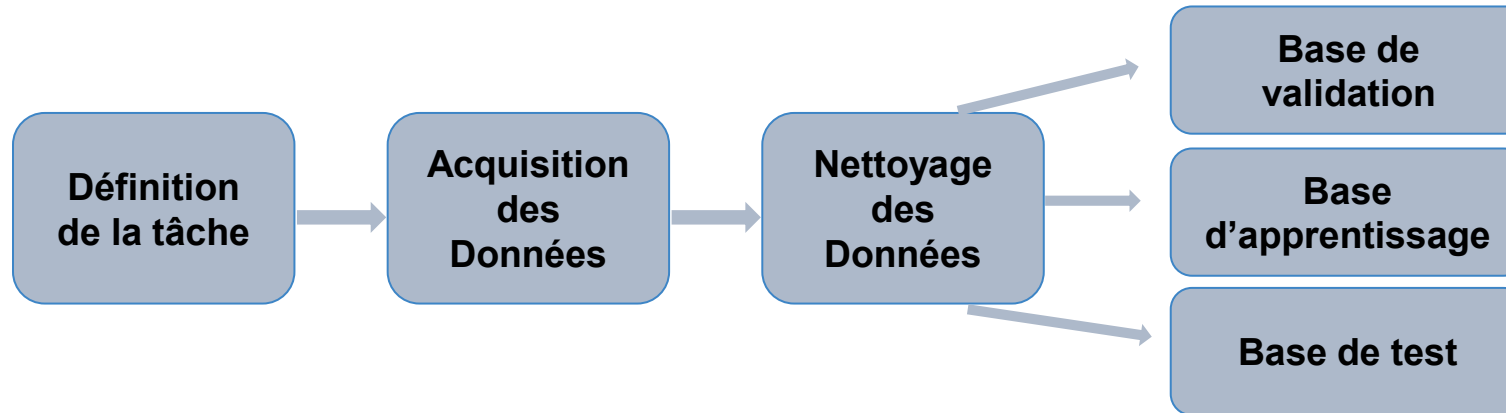
Etapes d'un projet d'apprentissage automatique



➤ Préparation des données :

- Vectorisation : données \rightarrow valeurs numériques
- Normalisation des données dans un interval
- Nettoyage : Gestion des valeurs manquantes, ...

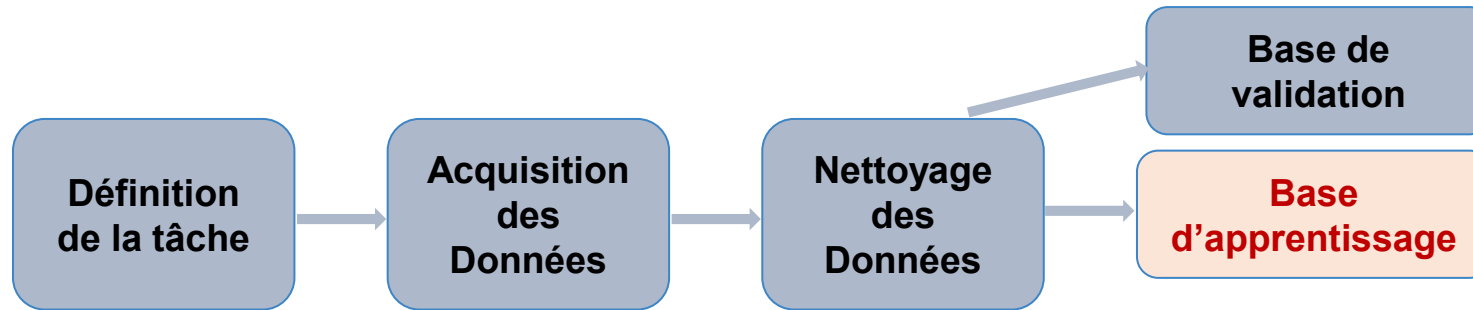
Etapes d'un projet d'apprentissage automatique



➤ **Répartition des données en 3 parties:**

- Base d'apprentissage
- Base de validation
- Base de test

Etapes d'un projet d'apprentissage automatique

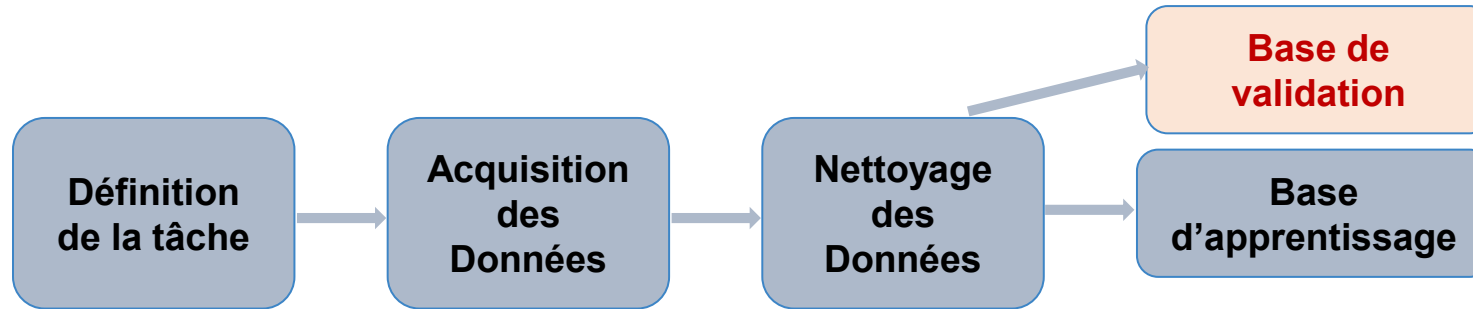


➤ Répartition des données en 3 parties:

Base d'apprentissage

- Sert à **entraîner le modèle**.
- Ajustement des **paramètres internes** (ex. coefficients en régression linéaire).

Etapes d'un projet d'apprentissage automatique

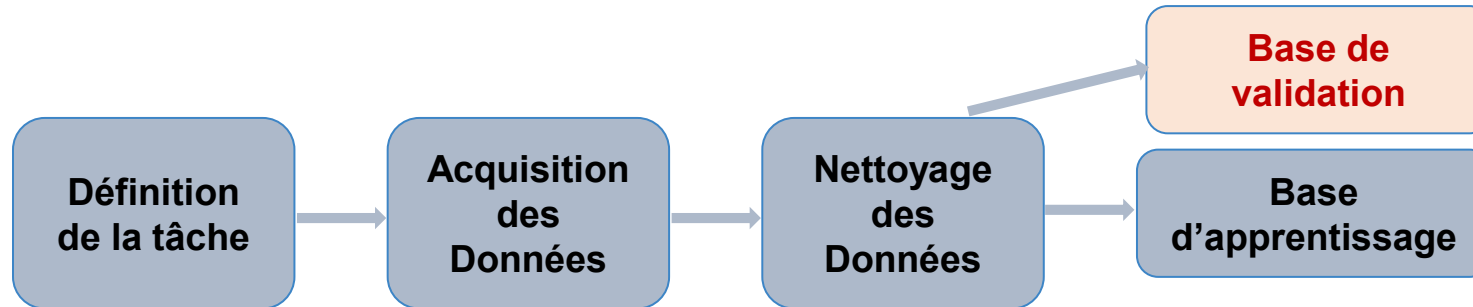


4- Répartition des données en 3 parties:

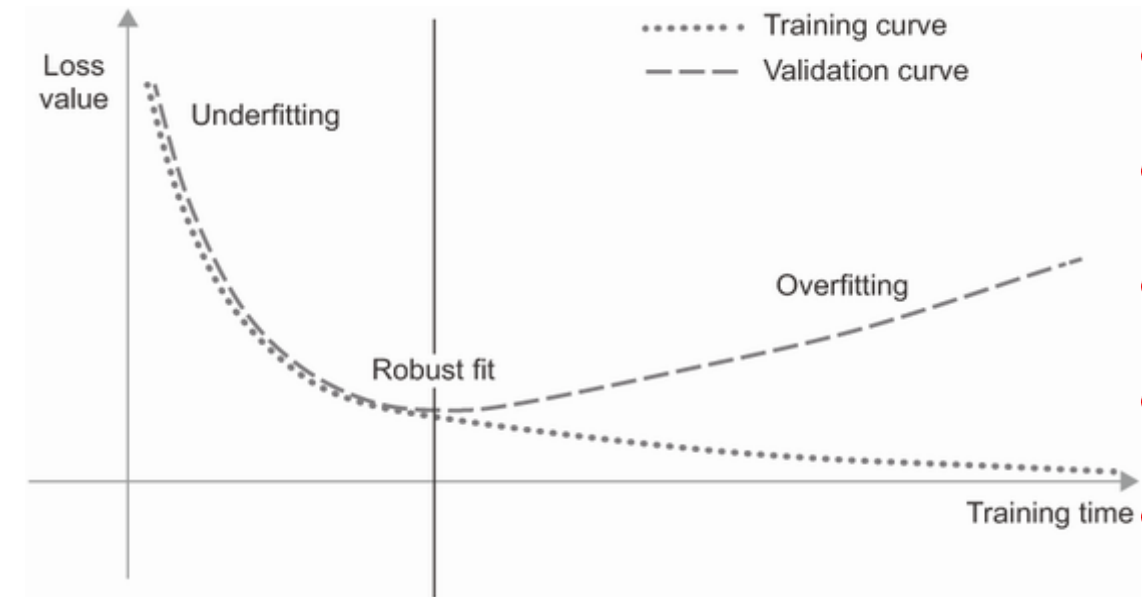
Base de validation :

- Elle sert à évaluer les performances du modèle sur des données jamais vues, afin d'ajuster les hyperparamètres (ex. le taux d'apprentissage, la profondeur d'un arbre, etc.).
- Elle permet également de contrôler le surapprentissage en fournissant une estimation de la performance du modèle sur des données non utilisées lors de l'entraînement.
- Lorsque plusieurs modèles sont entraînés, c'est sur l'ensemble de validation que l'on se base pour choisir le meilleur.

Etapes d'un projet d'apprentissage automatique

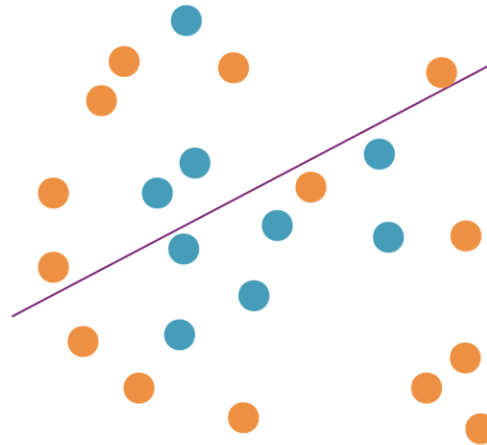


⚠ **La validation permet de contrôler le sur-apprentissage (overfitting) :**
si le modèle obtient de très bonnes performances sur l'ensemble d'entraînement mais de faibles résultats sur l'ensemble de validation, c'est qu'il ne généralise pas correctement.



Sous apprentissage - Underfitting

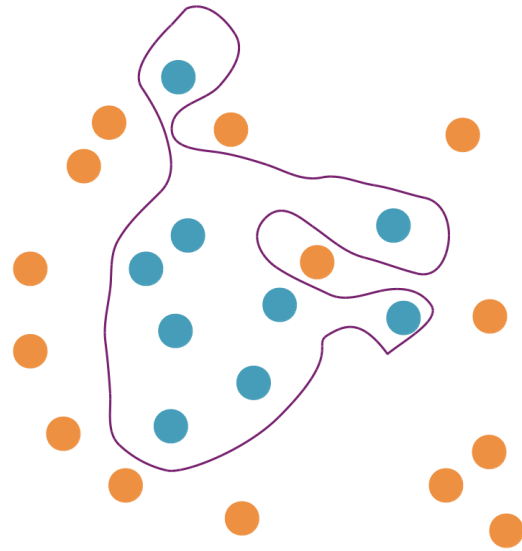
Se produit lorsqu'un modèle est trop simple ou insuffisamment entraîné pour capturer les relations présentes dans les données. Il n'arrive donc pas à bien représenter les données d'entraînement et obtient de **faibles performances**, aussi bien sur les données d'entraînement que sur les nouvelles données.



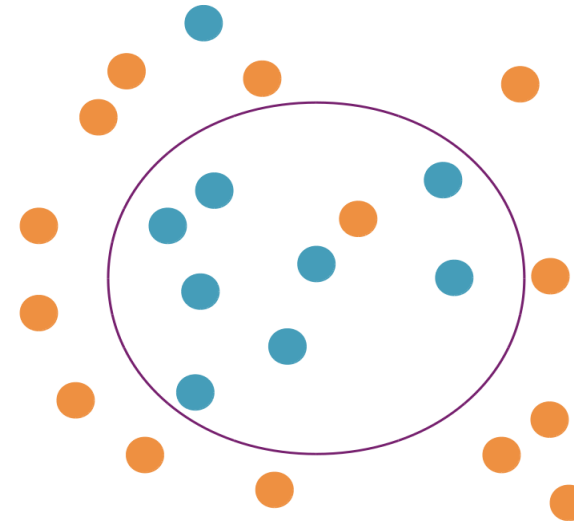
Sous apprentissage (Underfitting) :
Le modèle est trop simple pour apprendre
la structure des données

Sur-apprentissage - Overfitting

Se produit lorsqu'un modèle **apprend trop** spécifiquement les données d'entraînement, y compris leurs particularités et leurs irrégularités. Il obtient alors d'excellentes performances sur ces données, mais perd sa capacité à généraliser et donne de mauvais résultats sur de nouvelles données.



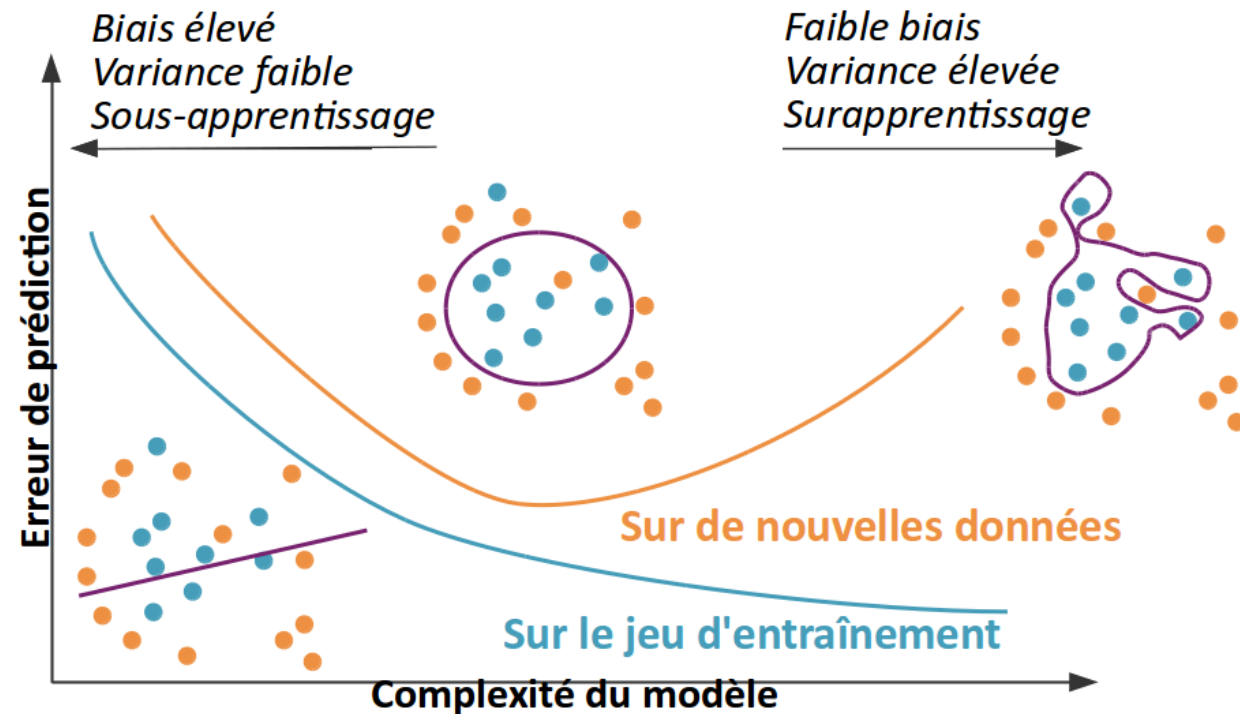
Sur-apprentissage (Overfitting)
Modèle trop complexe : s'adapte au bruit



Meilleure généralisation
Modèle plus simple : apprend les tendances

Compromis Biais-variance

Trouver une **complexité intermédiaire** où le biais et la variance sont équilibrés
erreur minimale sur des **nouvelles données**.

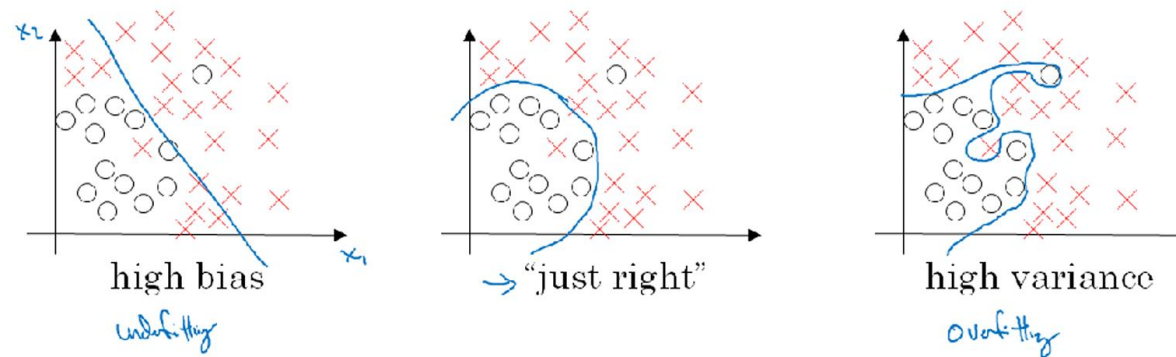


Biais = erreur due à la simplification excessive du modèle.

Variance = erreur due à la sensibilité excessive du modèle aux données d'entraînement.

Compromis Biais-variance

Bias and Variance



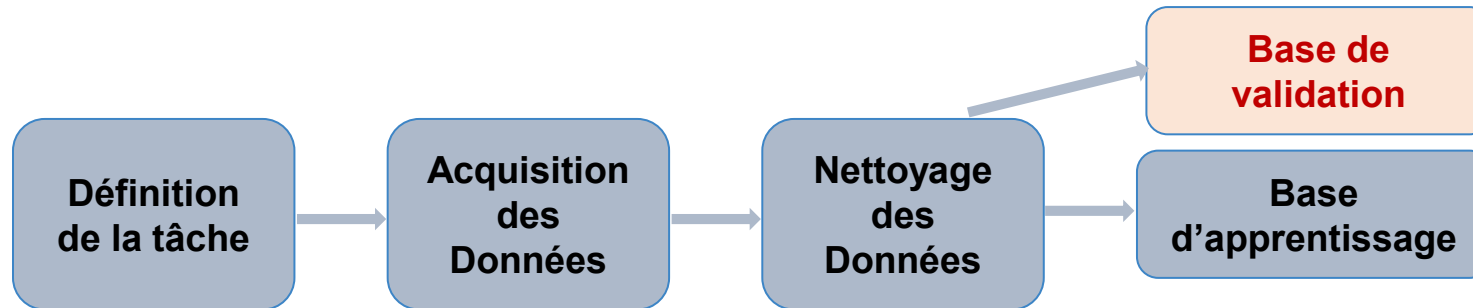
Cat classification



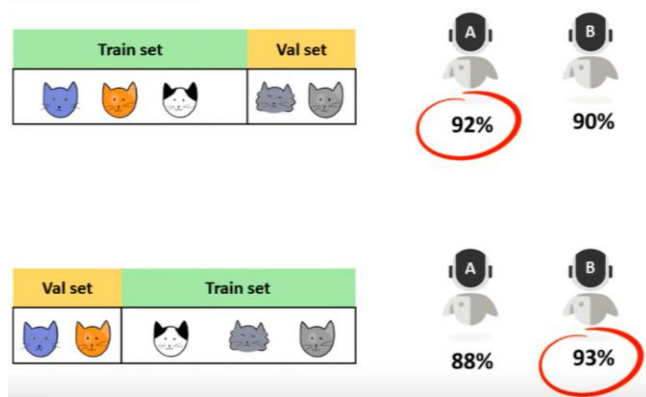
Train set error:	1%	15%	15%	0.5%
Dev set error:	11%	16%	30%	1%
	high variance ↑	high bias ↑	high bias & high variance	low bias low variance ↑
Human: 100%				
Optimal (Bayes) error: 15%				
			Blurry images	

<https://www.coursera.org/learn/deep-neural-network>

Etapes d'un projet d'apprentissage automatique



Problème



Comparaison Modèle A vs Modèle B :

le choix du meilleur modèle dépend du découpage.
Un seul découpage du dataset peut donner une vision biaisée.

Solution

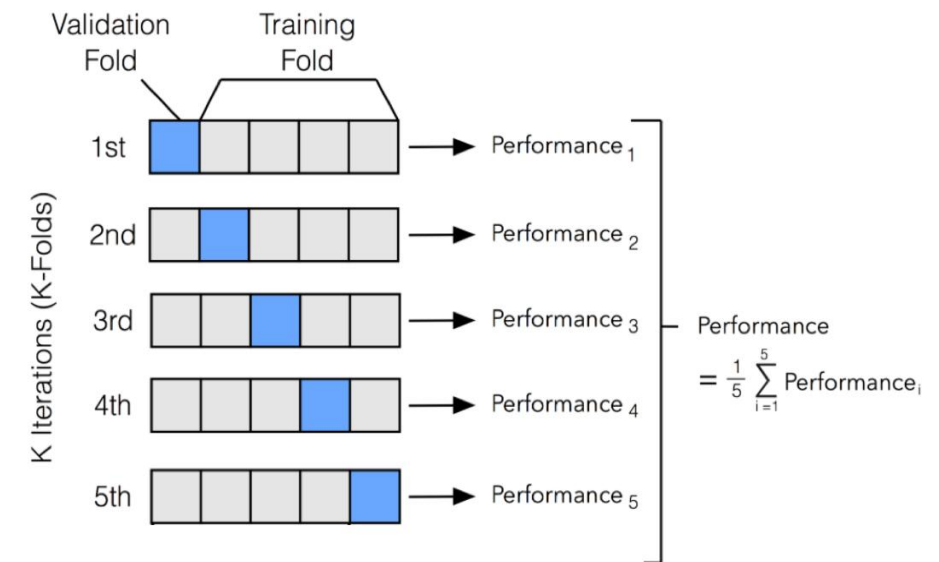
	Train set					A	B
Split 1	Val	Train	Train	Train	Train	0.92	0.91
Split 2	Train	Val	Train	Train	Train	0.88	0.90
Split 3	Train	Train	Val	Train	Train	0.89	0.91
Split 4	Train	Train	Train	Val	Train	0.93	0.92
Split 5	Train	Train	Train	Train	Val	0.86	0.90
						0.89	0.92

Validation croisée : plusieurs découpages (train/val),
Entraînement + validation sur chaque découpe,
Évaluer la robustesse du modèle.

La validation croisée

Validation croisée (K-Fold Cross Validation)

- Les données sont divisées en K sous-ensembles (folds).
- À chaque itération :
 - Un fold est utilisé comme ensemble de validation.
 - Les K-1 folds restants servent à l'entraînement.
- Ce processus est répété K fois, en changeant à chaque fois le fold de validation.
- La performance finale est calculée comme la moyenne des K performances obtenues.



La validation croisée

Évaluer la robustesse du modèle :

Le modèle est testé sur plusieurs sous-ensembles de données.
On obtient ainsi une estimation plus fiable et plus stable de ses performances globales.

Réduire le risque de surapprentissage (overfitting)

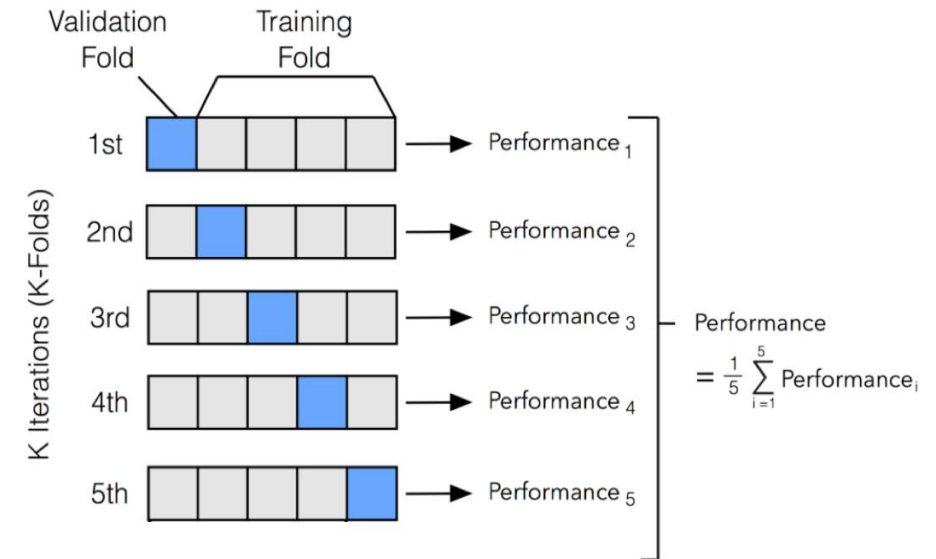
À chaque itération, le modèle est réentraîné sur un échantillon différent. Il est ensuite évalué sur des données jamais vues.
Cela donne une meilleure estimation de sa capacité de généralisation.

Mieux exploiter l'ensemble des données

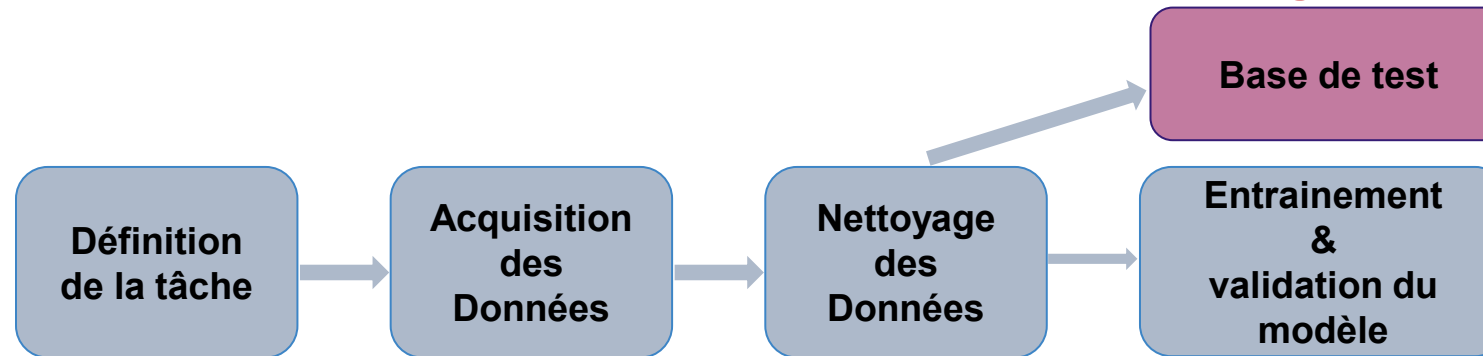
Chaque observation sert à la fois pour l'entraînement et pour la validation (mais jamais en même temps).
On maximise ainsi l'utilisation du jeu de données disponible.

Comparer équitablement plusieurs modèles

Les performances obtenues sont donc comparables, sans biais lié au découpage des données.

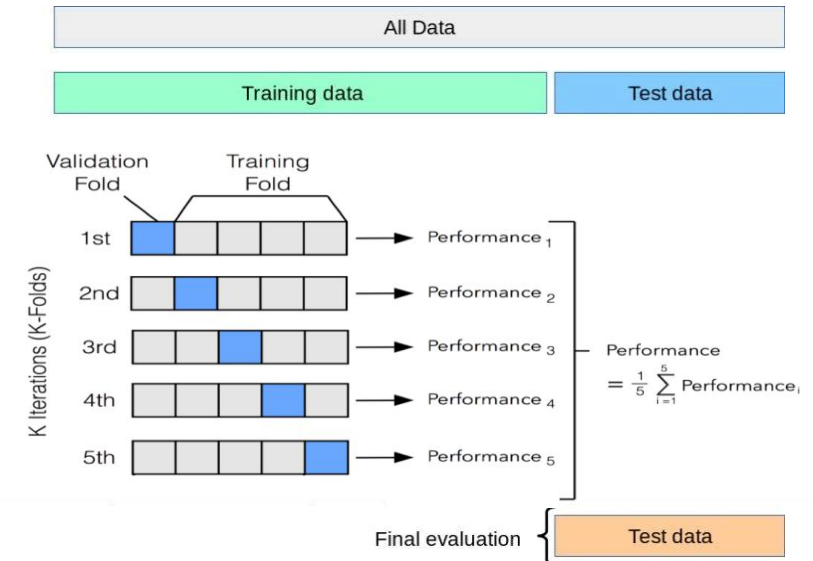


Etapes d'un projet d'apprentissage automatique



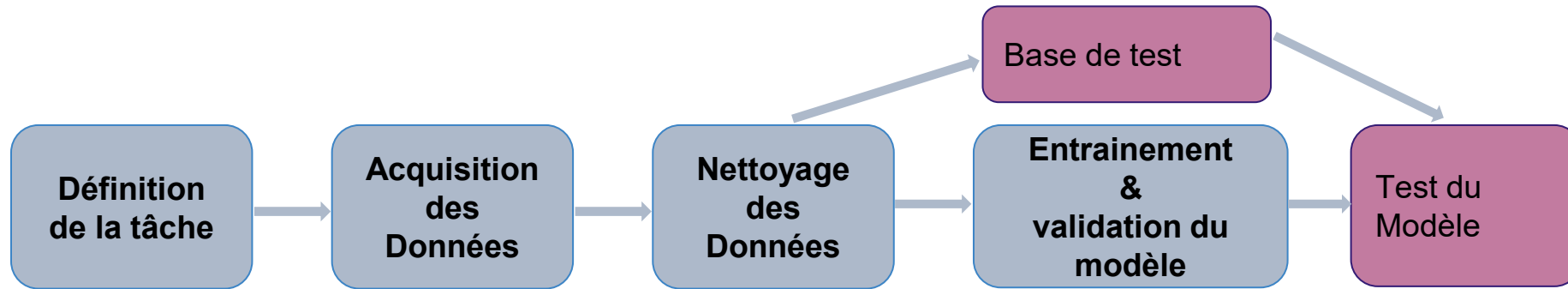
➤ Répartition des données en 3 parties:

- base d'apprentissage** : pour apprendre le modèle
- base de validation** :
 - pour trouver les paramètres optimaux du modèle = fine-tuning
 - pour contrôler le sur-apprentissage (over-fitting) = généralisation du modèle
- base de test** : pour tester le modèle et obtenir une mesure finale de la performance



<https://vitalflux.com/k-fold-cross-validation-python-example/>

Etapes d'un projet d'apprentissage automatique

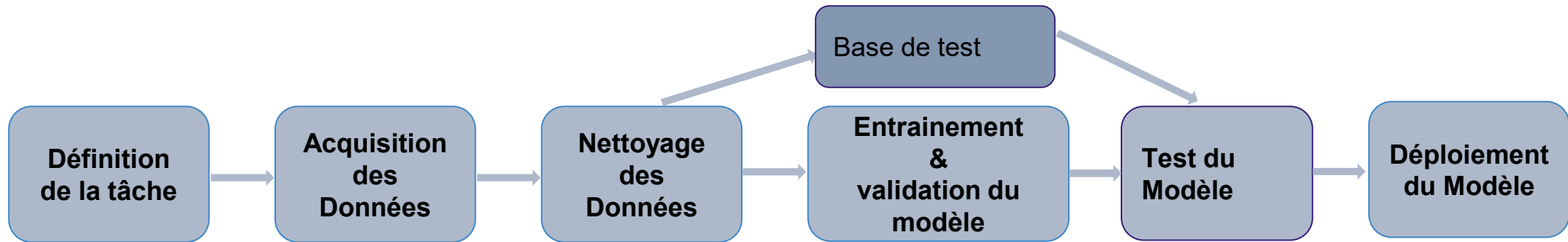


➤ Définition d'un protocole d'évaluation :

Utilisation des mesures de performance (ou métriques) pour évaluer la qualité du modèle :

- Exactitude (Accuracy)
 - Précision (Precision)
 - F1-Score
 - Erreur quadratique moyenne (Mean Squared Error - MSE)
 - Racine carrée de l'erreur quadratique moyenne (Root Mean Square Error - RMSE)
- Évaluation de la classification
- Évaluation de la régression

Etapes d'un projet d'apprentissage automatique



Outils pour les travaux pratiques



Notebook

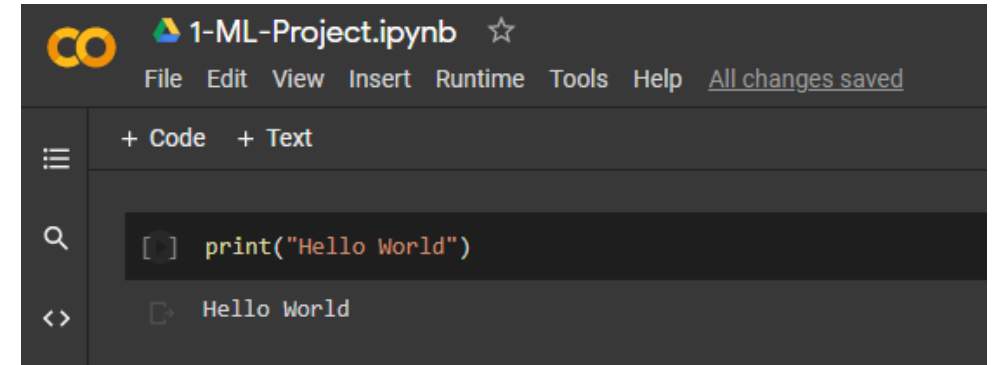
- Un notebook est une interface de programmation interactive permettant de combiner des sections en langage naturel et des sections en langage informatique [1]
- Un notebook vous permet d'écrire et d'exécuter du code Python dans votre navigateur. Réellement, les codes sont interprétés par un serveur cloud



[1] [https://fr.wikipedia.org/wiki/Notebook_\(programmation\)](https://fr.wikipedia.org/wiki/Notebook_(programmation))

Pourquoi utiliser Google Colab ?

- **Aucune configuration requise** : Vous pouvez exécuter des scripts Python sans avoir besoin d'installer quoi que ce soit sur votre ordinateur personnel.
- **Accès gratuit aux GPUs** : Colab offre l'accès à des GPUs et TPUs pour accélérer les calculs sans coût supplémentaire.
- **Facilité de partage** : Comme Google Docs, vous pouvez facilement partager vos notebooks avec d'autres, collaborer en temps réel, ou les publier en ligne.



Bibliothèques



Premiers pas sur Colab

- 1.Ouvrir Colab** : Visitez colab.research.google.com et connectez-vous avec votre compte Google.
- 2.Créer un nouveau notebook** : Cliquez sur **Nouveau notebook** où un nouveau document sera créé.
- 3.Renommer votre notebook** : Cliquez sur le titre (par défaut, "Sans titre") en haut de la page pour le renommer.
- 4.Écrire du code** : Vous pouvez commencer à taper du code Python dans les cellules. Pour exécuter le code, cliquez sur le bouton "play" à gauche de la cellule ou utilisez le raccourci clavier **Shift + Enter**.

Premiers pas sur Colab

5.Ajouter des cellules : Utilisez les boutons **+ Code** ou **+ Texte** dans la barre d'outils pour ajouter de nouvelles cellules de code ou de texte.

6.Importer des données : Utilisez la bibliothèque **google.colab** pour importer facilement des fichiers depuis votre Google Drive ou télécharger des fichiers depuis votre ordinateur.

7.Sauvegarder et partager : Votre notebook est automatiquement sauvegardé dans votre Google Drive. Vous pouvez partager votre notebook en cliquant sur le bouton **Partager** en haut à droite.

Étape 1: Importer les bibliothèques nécessaires

Pour charger des données dans un notebook Google Colab, vous pouvez utiliser différentes méthodes selon la source de vos données.

Si on veut charger les fichiers stockés sur notre Google Drive.

```
import pandas as pd  
from google.colab import drive
```

Pour accéder aux fichiers stockés sur votre Google Drive, vous devez monter votre drive dans le runtime de Colab:

```
drive.mount('/content/drive')
```



Lorsque vous exécutez cette commande, vous serez invité à suivre un lien pour vous authentifier avec votre compte Google. Après l'authentification, copiez le code fourni et collez-le dans la boîte de dialogue de Colab.

Étape 2: Charger les données

Après avoir monté le drive, vous pouvez accéder aux fichiers stockés dessus. Supposons que vous avez un fichier nommé data.csv dans un dossier Datasets à la racine de votre Drive. Vous pouvez le charger comme suit:

```
file_path = '/content/drive/My Drive/Datasets/data.csv'  
data = pd.read_csv(file_path)
```

Il est toujours une bonne pratique de vérifier les premières lignes de votre dataframe pour vous assurer que les données sont chargées correctement:

```
print(data.head())
```



Conseils supplémentaires

- Si votre fichier CSV contient des paramètres spécifiques comme un séparateur différent (par exemple des points-virgules au lieu de virgules), vous pouvez spécifier cela avec l'argument `sep` dans la fonction **read_csv**, par exemple : **pd.read_csv(file_path, sep=';')**.
- Vous pouvez également charger des fichiers directement depuis un URL ou utiliser d'autres formats de fichiers comme Excel avec **pd.read_excel()**.

Références principales

