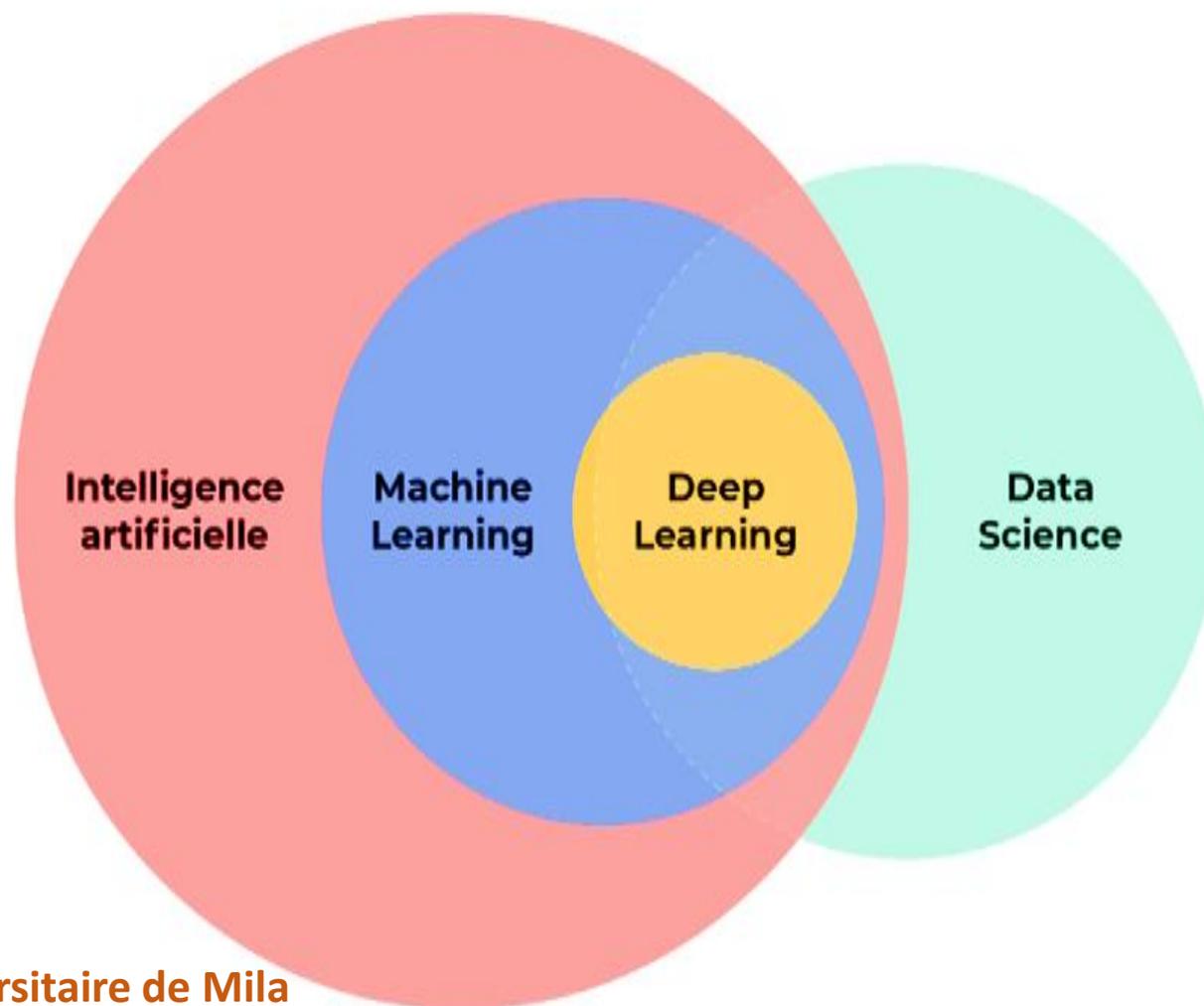




# Chapitre 03: Apprentissage automatique

Présente par DR:  
Hadjadj Abdelhalim

# Apprentissage automatique



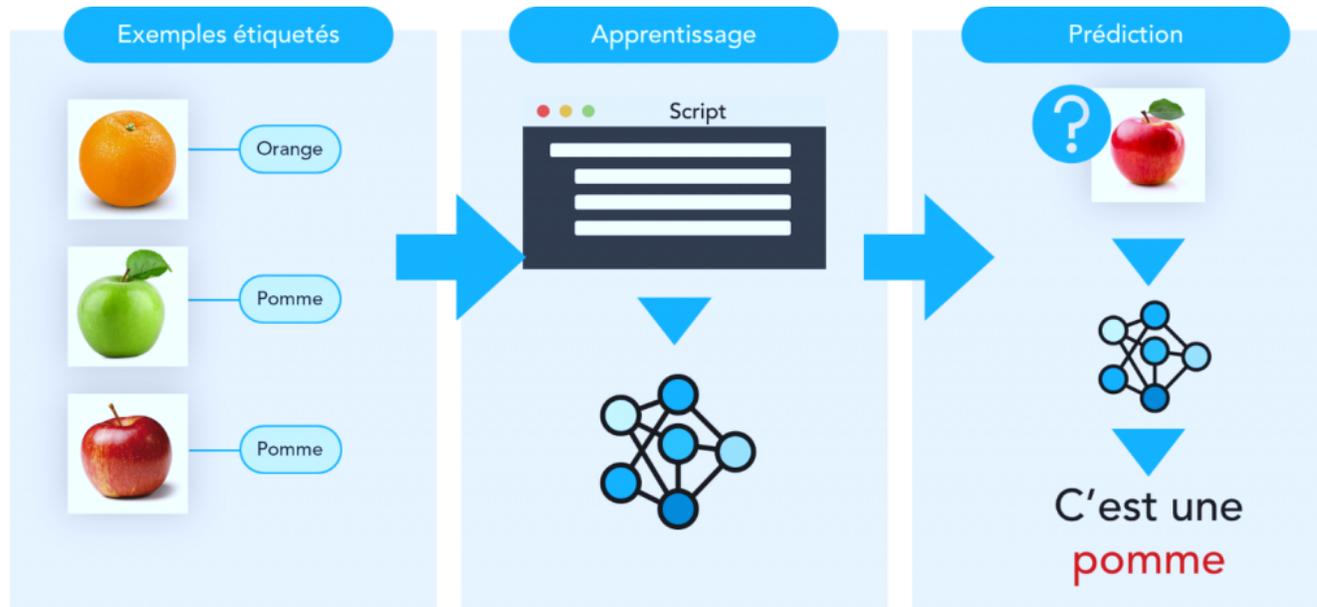
**Centre universitaire de Mila**

**Master I: Matière Intelligence artificielle : Principes et Applications**

# Apprentissage automatique

- **Définition**
- Est une branche de l'intelligence artificielle
- Il utilise des méthodes mathématiques et statistiques pour permettre aux ordinateurs de "découvrir" des **modèles** à partir de **données**.
- Il permet d'améliorer leurs performances dans la réalisation de tâches sans avoir été **explicitement programmés** pour chaque cas, en construisant de nouvelles connaissances à partir de l'expérience et en élaborant des modèles prédictifs pour des données futures.

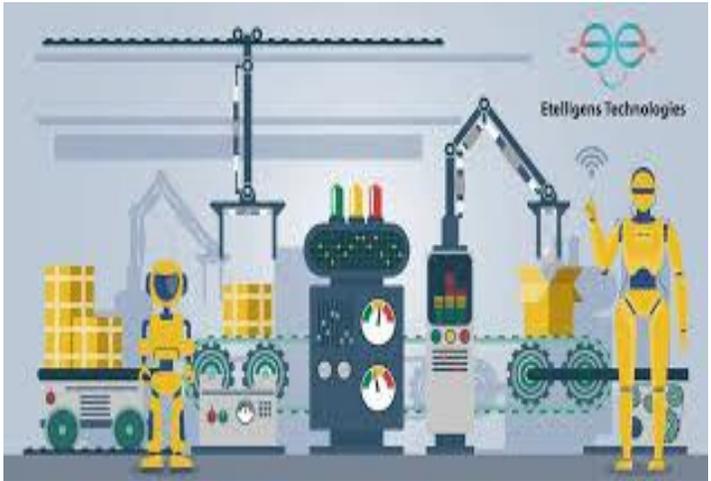
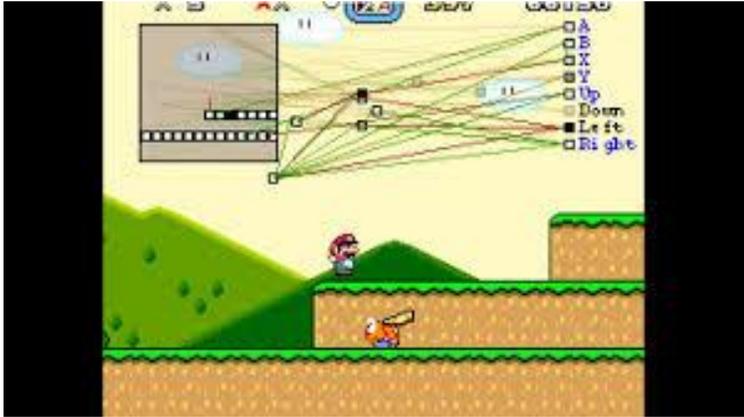
# Apprentissage automatique



Il vise à permettre aux machines d'analyser les données afin de fournir une sortie précise. Il est préoccupé par les modèles et la précision



# Applications

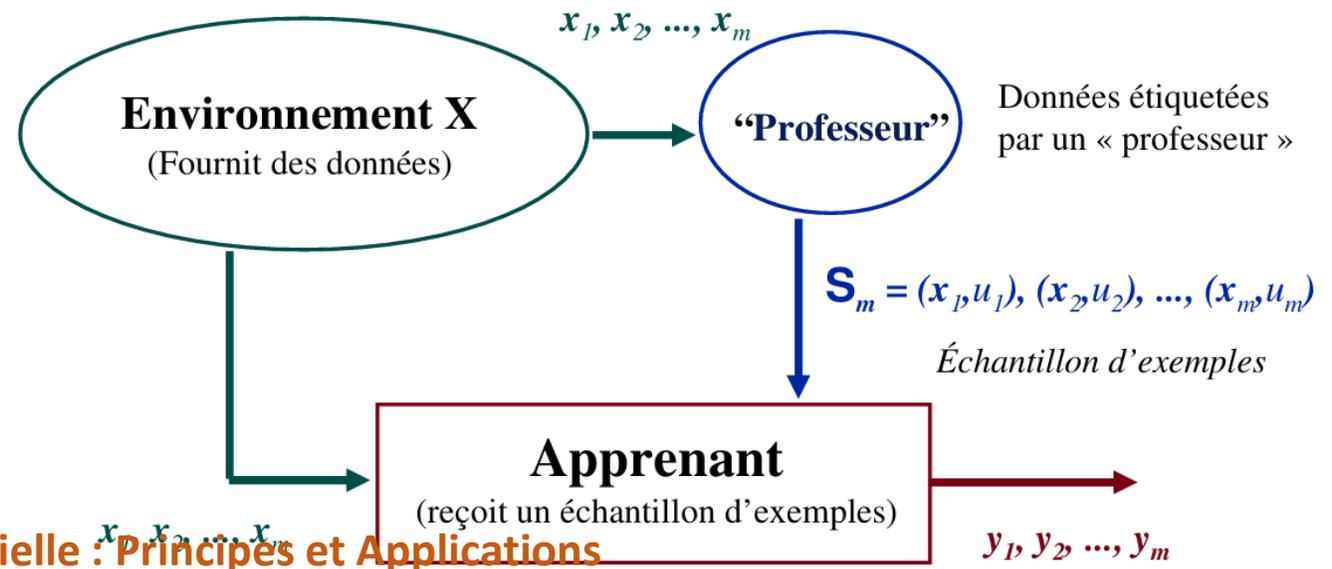


# Type d'apprentissage



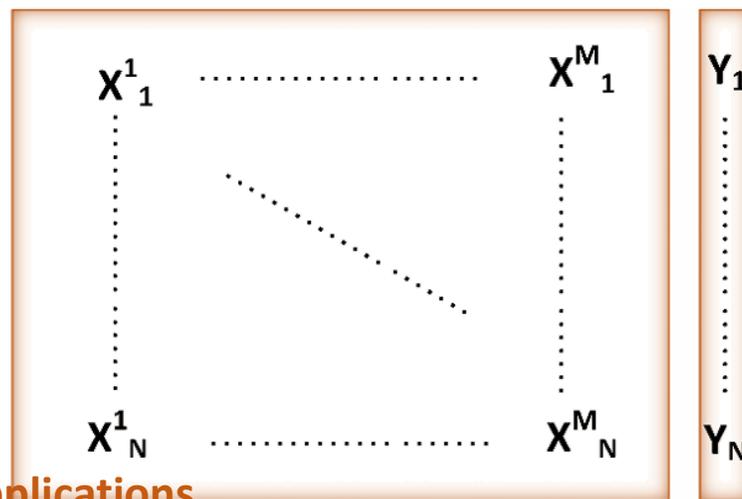
# Apprentissage supervise

- L'objectif de l'apprentissage supervise est d'apprendre à classer des instances à partir d'un ensemble d'exemples qui sont étiquetés par leurs classes,
- L'apprentissage est dit supervisé si les différentes étiquettes, ou classes, sont connues et assignées préalablement à chaque élément de la base de données.

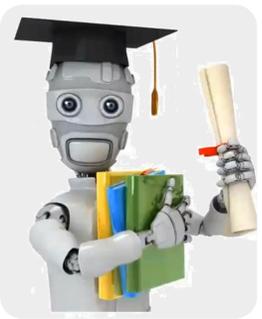
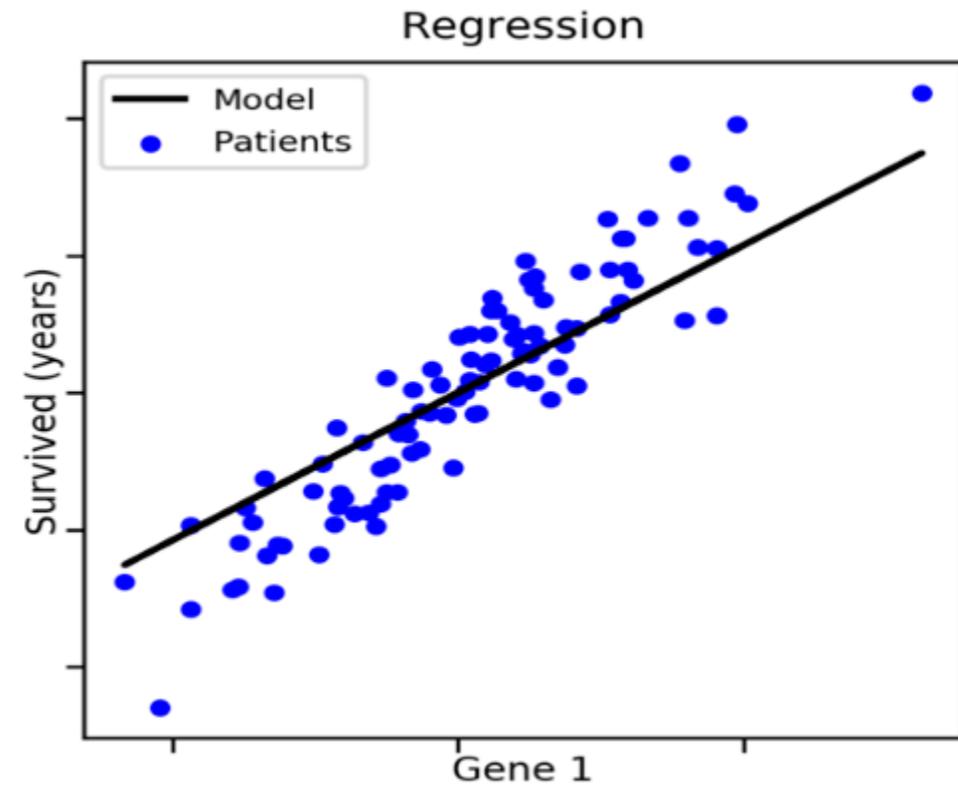
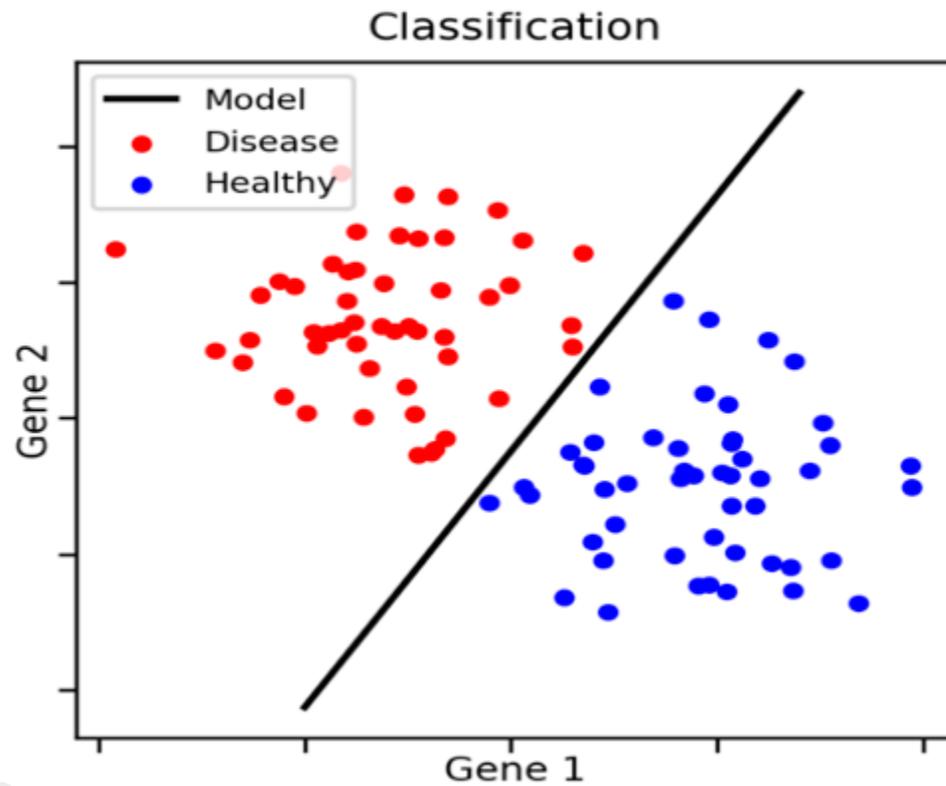


# Apprentissage supervise

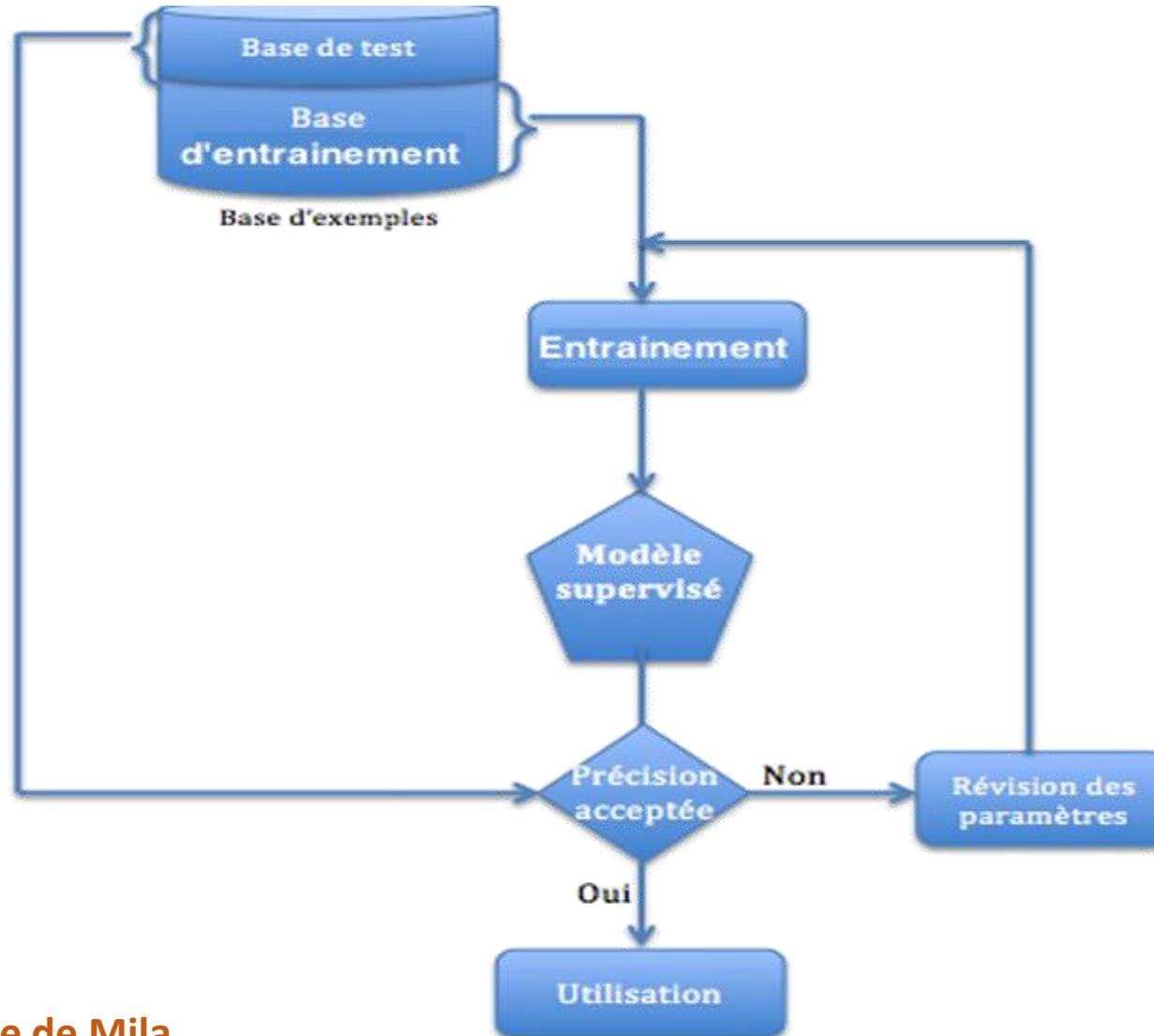
- les données sont composées d'une collection de  $N$  objets (variables d'entrée) décrits par  $M$  caractéristiques (attributs).
- Ces caractéristiques fournissent des informations sur l'étiquette de chaque objet (variable de sortie ou cible).
- Lorsque **la sortie est un nombre réel**, la tâche d'apprentissage automatique est appelée **régression** alors que si la sortie est **discrète (binaire ou catégorique)**, nous parlons de **classification**.



# Apprentissage supervise



# Apprentissage supervise



# Apprentissage non-supervise

- L'apprentissage supervise utilise des données doivent être étiquetées.
- dans le monde réel, les données ne le sont généralement pas, donc l'étiquetage doit se faire a la main, ce qui requiert un **temps considerable** de la part d un **expert humain**.

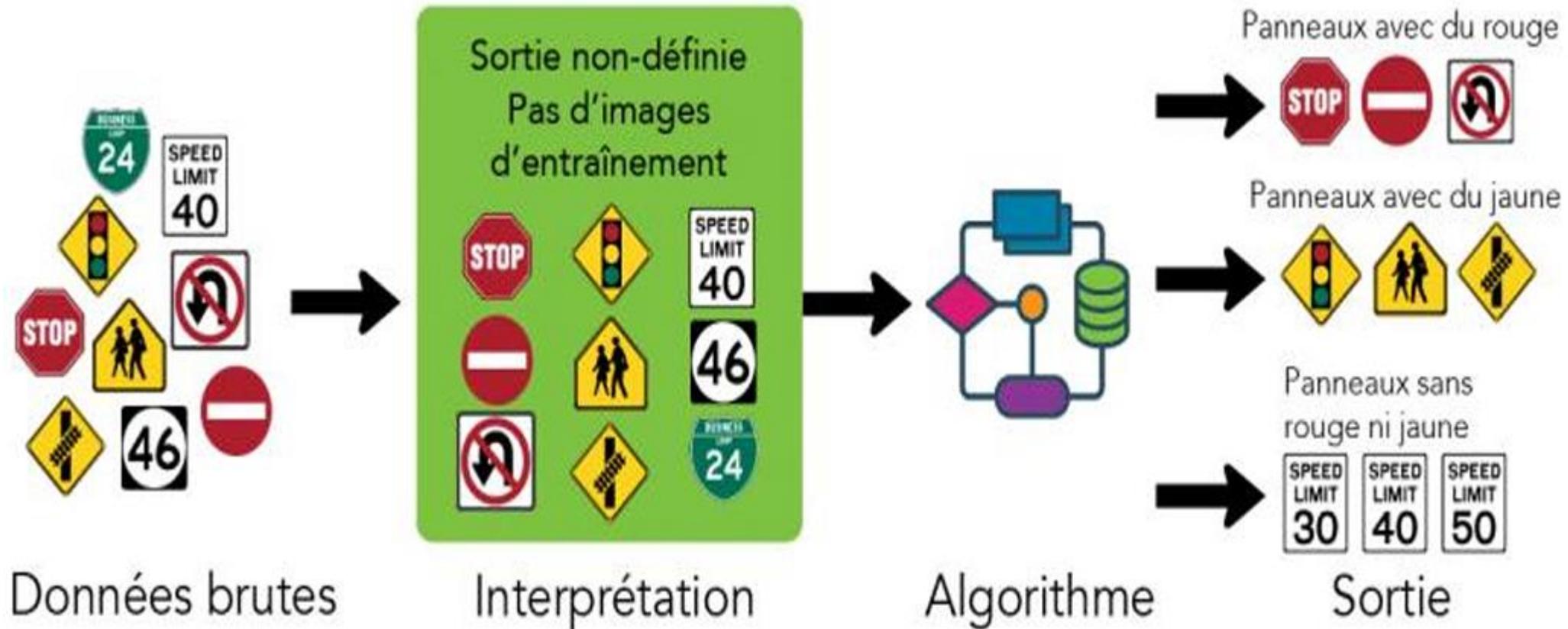


# Apprentissage non supervisé

- Apprentissage non supervisé est utilisé lorsque vous avez des données non étiquetées.
- En apprentissage non-supervisé, nous ne recevons que les données brutes, et l'objectif est de trouver des points communs de ces données.
- Ces algorithmes essaient de regrouper/cluster des objets ensemble (algorithme de clustering) et de trouver des liens entre différents objets (règles d'associations), etc.



# Apprentissage non-supervise

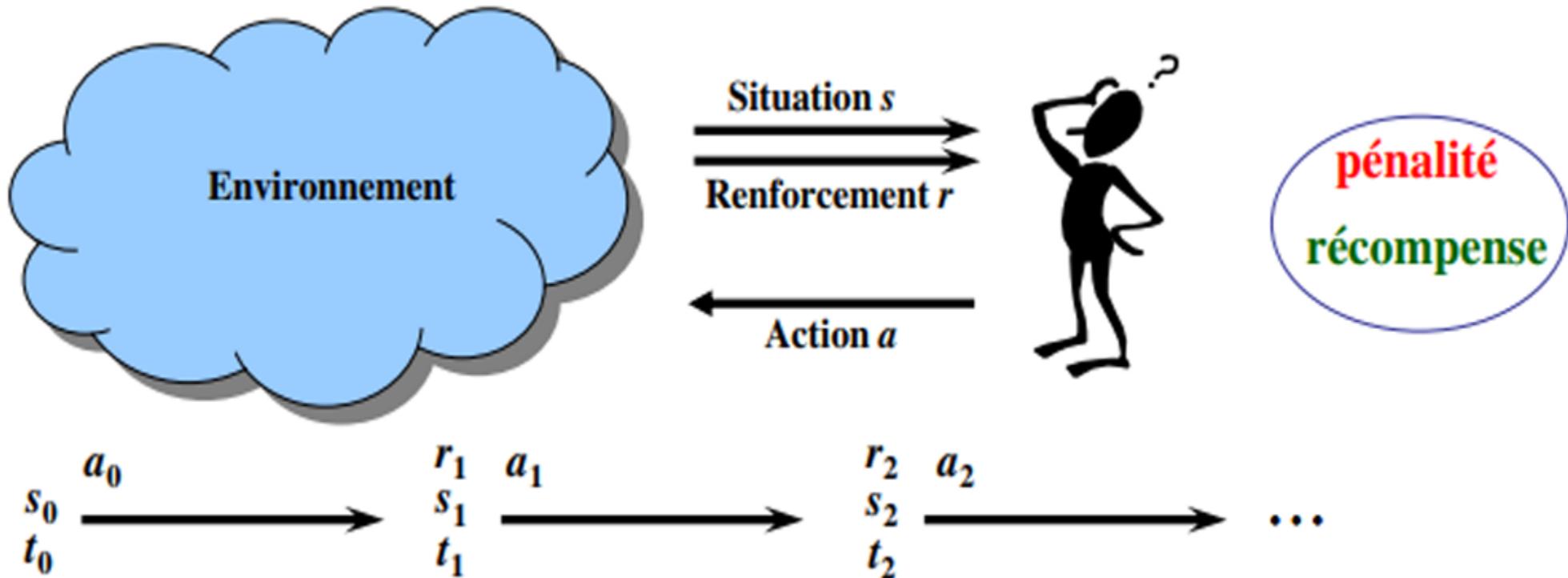


# Apprentissage par renforcement

- En apprentissage par renforcement, l'agent doit apprendre à bien agir dans son environnement afin de maximiser sa récompense.
- A chaque instant, l'agent peut percevoir (complètement ou partiellement) son environnement et choisit ensuite une action à effectuer. Après chaque action, il reçoit une récompense (qui peut être positif, négatif, ou zéro).
- A partir de ce feedback et de ses perceptions, l'agent doit apprendre un comportement qui l'amène à obtenir le plus de récompense possible.
- Théorie des jeux, de la robotique et des véhicules automatisés (véhicules autonomes)..



# Apprentissage par renforcement



# Choix de la méthode

- Ces trois types d'apprentissage sont complémentaires, et le choix de la bonne méthode d'apprentissage dépend du problème en question.
- Si nous voulons faire apprendre à un agent une **tache spécifique** dont nous pouvons fournir des exemples, l'apprentissage supervisé serait un bon choix.
- Si s'agit d'apprendre un **comportement compliqué**, l'apprentissage par renforcement serait probablement plus approprié.
- Si nous **n'avons pas d'objectif clair**, nous choisirons l'apprentissage non-supervisé pour découvrir des régularités dans un ensemble de données.

# Concepts Fondamentaux Les données

- Les données **d'apprentissage** serviront à entraîner le ou les algorithmes choisis ;
- Les données **de test** seront utilisées pour vérifier la performance de généralisation de classification/régression ;



# Concepts Fondamentaux Les données

Exemple : Un dataset est divisé en deux parties :  
80 % pour l'entraînement et 20 % pour le test.

❖ Importance : Cela permet de vérifier si le modèle a bien généralisé et non uniquement mémorisé les données d'entraînement.

# Concepts Fondamentaux Les données

- Les données de **validation** sont généralement utilisées à la fin du processus, et sont rarement examinées avant afin d'éviter d'introduire un quelconque **biais** dans le résultat final.
- Exemple : Dans la validation croisée k-fold, un jeu de données est divisé en 5 parties ( $k=5$ ). Chaque partie est utilisée une fois comme ensemble de test, tandis que les autres servent à l'entraînement.
- Importance : Cela permet de choisir le meilleur modèle et d'éviter les biais liés à une partition spécifique des données.

# Modèle

- **Définition** : Un modèle est une représentation mathématique qui **mappe les entrées** (caractéristiques du dataset) **vers une sortie** ou prédiction.
- **Structure** : Un modèle a généralement une **structure fixe** mais des **paramètres variables** qui peuvent être ajustés pour s'adapter aux données d'entraînement.
- **Exemples de modèles** :
  - **Régression linéaire** : Une ligne droite ajustée aux données.
  - **Arbre de décision** : Structure arborescente pour prendre des décisions basées sur les caractéristiques.
  - **Réseau de neurones** ,....

**Exemple concret** : Un modèle de régression linéaire peut être représenté par une équation du type  $y = ax^2 + bx + c$ ,  
où **a** et **b** sont des **paramètres** du modèle,  
**X** la caractéristique.

# Fonction de coût

- La fonction de coût (ou fonction de perte) mesure l'écart entre la **sortie prédite** par le modèle et la **valeur réelle** attendue. Elle indique l'**erreur** du modèle.
- **But** : Minimiser cette fonction de coût pour que le modèle fasse des prédictions aussi précises que possible.

- **Exemples** :

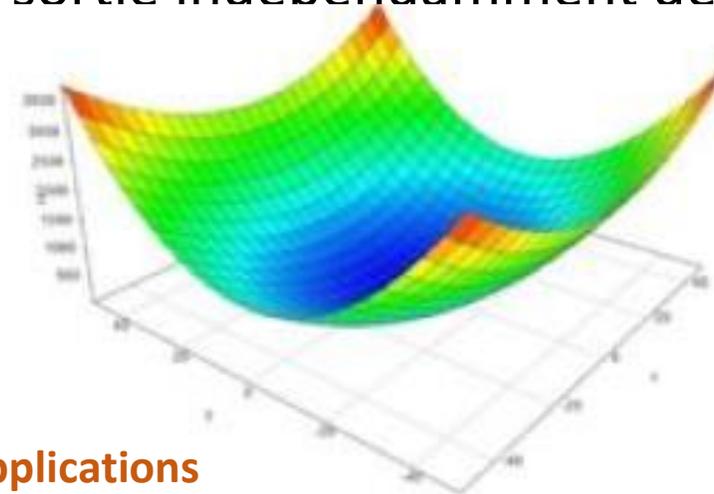
- **Erreur quadratique moyenne (MSE)** : Utilisée pour des problèmes de régression.

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

- **Logarithmic Loss (Log Loss)** : Utilisée pour la classification, mesurant l'incertitude des prédictions probabilistes.

# Paramètres

- Les paramètres sont **les valeurs internes** d'un modèle qui sont ajustées pendant l'entraînement pour **minimiser l'erreur**.
- Ils déterminent le comportement du modèle et la manière dont il fait des prédictions.
- **Types de paramètres :**
  - **Poids (weights)** : Dans les modèles linéaires ou les réseaux de neurones, les poids déterminent l'importance de chaque caractéristique.
  - **Biais (bias)** : Valeur ajoutée pour ajuster la sortie indépendamment des caractéristiques.



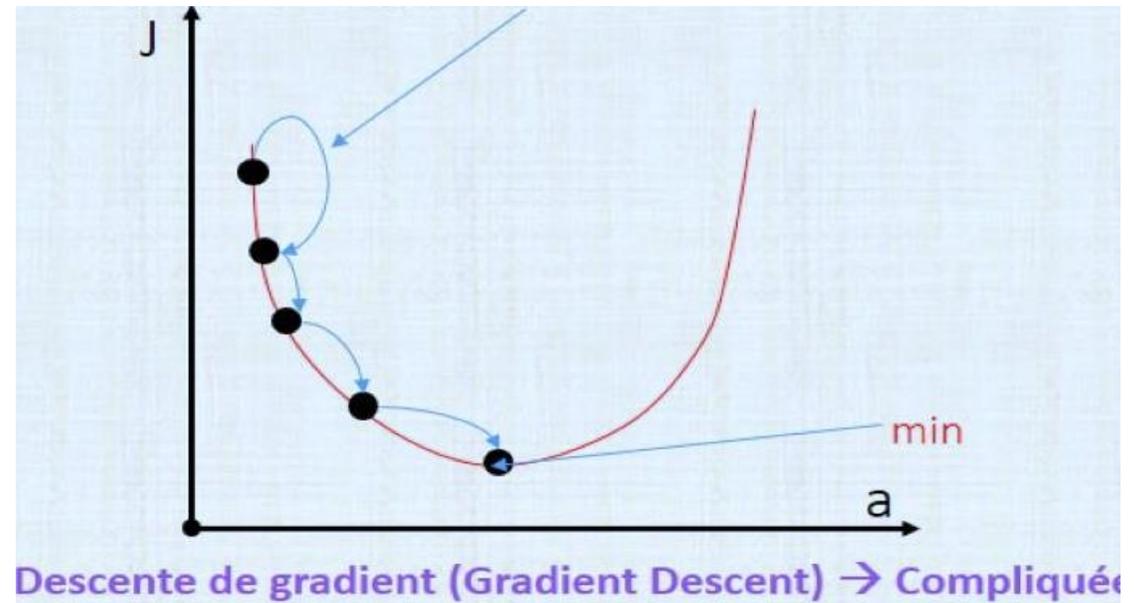
# Algorithme d'apprentissage

- L'algorithme d'apprentissage est la procédure ou méthode utilisée pour **ajuster les paramètres** du modèle en **minimisant la fonction de coût**.
- L'algorithme ajuste les paramètres du modèle en fonction des erreurs mesurées par la fonction de coût, afin **d'améliorer les prédictions**.
- **Exemples :**
  - **Méthode des Moindres Carrés** : Technique analytique qui trouve directement les valeurs optimales des paramètres en minimisant la somme des carrés des écarts entre les valeurs prédites et les valeurs observées. 
$$J(a, b) = \sum_{i=1}^n (y_i - (ax_i + b))^2$$
  - **Descente de gradient** : Méthode itérative qui ajuste les paramètres dans la direction qui diminue la fonction de coût.

# Descente de gradient

$$a_{i+1} = a_i - \alpha \frac{\partial J(a_i)}{\partial a}$$

$$\frac{\partial J(a_0)}{\partial a} = \frac{1}{m} \sum_{i=1}^m x^{(i)} (ax^{(i)} + b - y^{(i)})$$



# Métriques d'évaluation

- **Définition** : Les métriques permettent de **mesurer la qualité** d'un modèle et **d'évaluer ses performances** sur des données de test.
  - et se divise en deux types, l'un pour **la régression** et l'autre pour la **classification**

## **pour la regression**

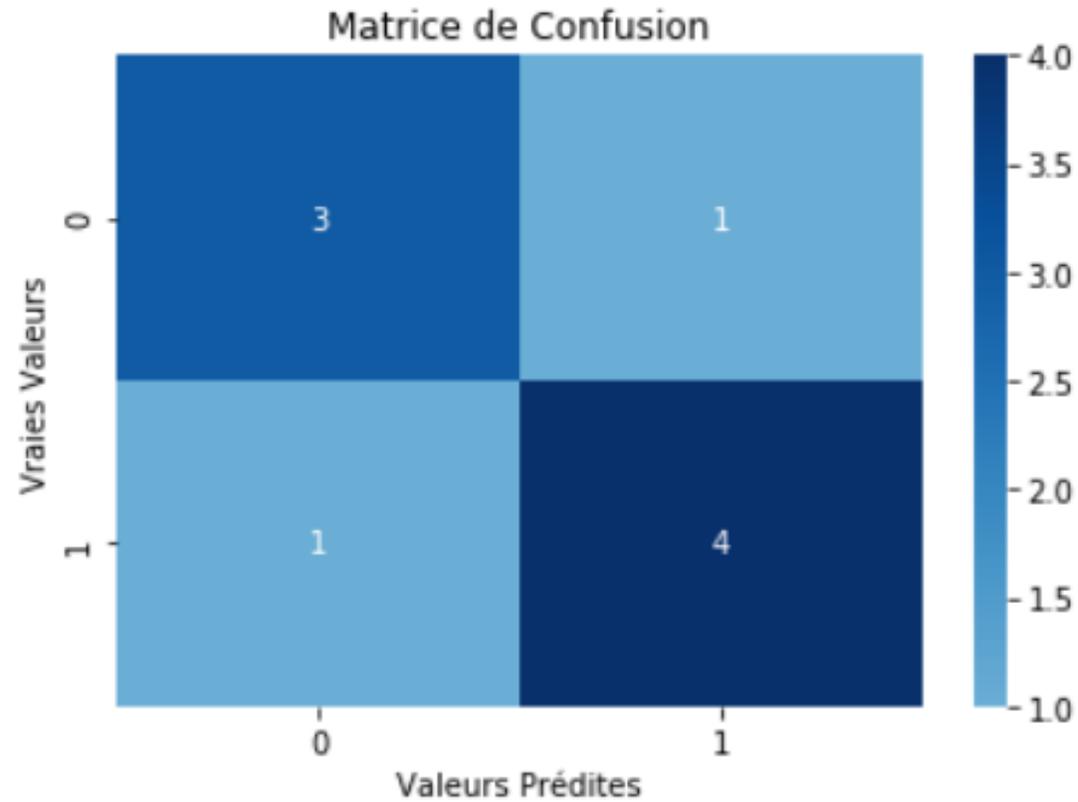
- ✓ `metrics.mean_absolute_error`
- ✓ `metrics.mean_squared_error`
- ✓ `metrics.median_absolute_error`

# Métriques d'évaluation

- **Exemples pour la classification :**
  - **Précision (Accuracy)** : Proportion de prédictions correctes.
  - **Rappel (Recall)** : Capacité du modèle à détecter toutes les occurrences positives.
  - **F-mesure (F1-score)** : Moyenne harmonique entre précision et rappel.
  - **Matrice de confusion** : Tableau qui résume les performances d'un modèle de classification.
- Les métriques permettent de comparer différents modèles et d'identifier leurs forces et faiblesses.

Confusion Matrix is :

```
[[3 1]
 [1 4]]
```



# Surapprentissage (Overfitting) et Sous-apprentissage (Underfitting)

- **Définition :**

- **Surapprentissage** : Le modèle est **trop adapté** aux données d'entraînement et ne généralise pas bien aux nouvelles données.
- **Sous-apprentissage** : Le modèle est **trop simple** pour capturer les tendances des données.

- **Exemple :**

- Un modèle de surapprentissage aura une **très bonne** performance sur les données **d'entraînement** mais **échouera** sur les données de **test**.
  - Un modèle de sous-apprentissage aura une **mauvaise performance** sur les **deux**.
- Ces concepts sont cruciaux pour ajuster la complexité du modèle afin d'obtenir une bonne généralisation.

# Questions