

# Machine Learning

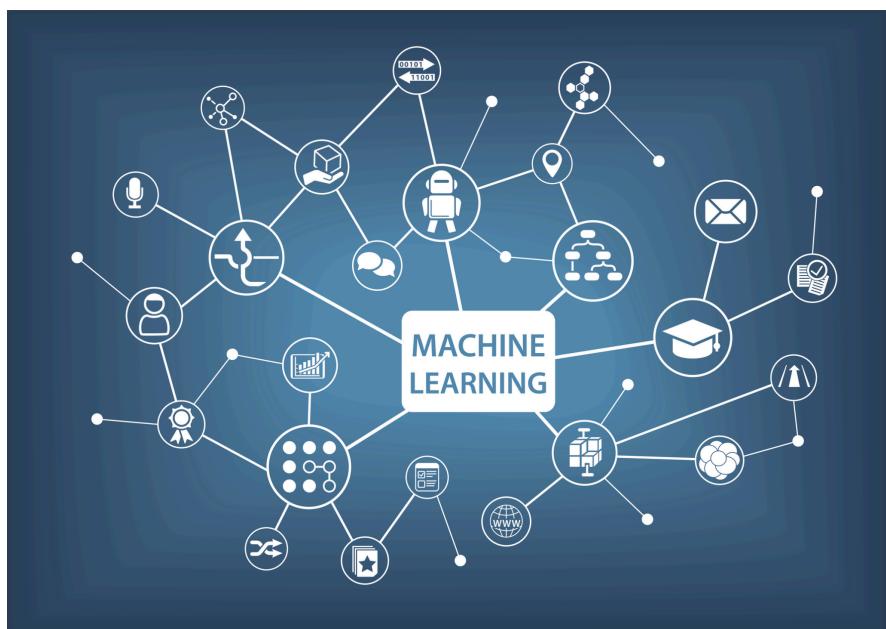
<b>1. Introduction</b>	<b>6</b>
1.1. Contexte et importance du Machine Learning	6
1.1.1. Contexte du Machine Learning	6
1.1.1.1. Statistiques et probabilités	6
1.1.1.2. Informatique	6
1.1.1.3. Neurosciences et Cognition	6
1.1.2. Importance du Machine Learning	7
1.1.2.1. Analyse de Données Massives	7
1.1.2.2. Automatisation des Tâches	7
1.1.2.3. Amélioration Continue	7
1.1.2.4. Application Innovantes	7
1.1.3. Exemples	8
1.2. Evolution historique	9
1.2.1. Les débuts	9
1.2.2. L'essor des algorithmes de ML	9
1.2.3. La révolution des données	10
1.2.4. Aujourd'hui et demain	10
<b>2. Fondamentaux du Machine Learning</b>	<b>11</b>
2.1. La Science des Données	11
2.1.1. Collecte de Données	11
2.1.1.1. Méthodes et défis	11
2.1.1.2. Outils de collecte	11
2.1.2. Prétraitement des Données	11
2.1.2.1. Techniques courantes (nettoyage, normalisation, réduction de dimensionnalité)	11
2.1.2.2. Cas d'utilisation pratiques	11
2.1.3. Analyse Exploratoire des Données	11
2.1.3.1. Outils et visualisations	11
2.1.3.2. Études de cas	11
2.2. L'Apprentissage Automatique et Profond	11
2.2.1. L'apprentissage Automatique (Machine Learning)	11
2.2.1.1. Définition et Principe	11
2.2.1.2. Algorithmes courants	11
2.2.1.3. Applications courantes	11
2.2.2. L'apprentissage Profond (Deep Learning)	11
2.2.2.1. Définition et Principe	11
2.2.2.2. Réseaux de Neurones Profonds	11
2.2.2.3. Applications courantes	11
2.2.3. Comparaison	11
2.3. Corrélation Linéaire (de Pearson) entre Deux Variables	11
2.3.1. Définition et calcul	11
2.3.1.1. Définition	11
2.3.1.2. Formule de Calcul	11
2.3.2. Applications pratiques	11
2.3.2.1. Analyse de la Relation entre Variables	11
2.3.2.2. Modélisation et Prédiction	11
2.3.2.3. Recherche Scientifique et Études Empiriques	11
2.3.3. Exemples concrets	11
<b>3. Méthodes d'Apprentissage</b>	<b>11</b>
3.1. L'apprentissage Supervisé	11
3.1.1. Principe et fonctionnement	11
3.1.1.1. Définition	11

3.1.1.2. Processus	11
3.1.1.3. Types de problèmes	12
3.1.2. Études de cas	12
3.2. L'apprentissage Non Supervisé	12
3.2.1. Principe et fonctionnement	12
3.2.1.1. Définition	12
3.2.1.2. Processus	12
3.2.1.3. Types de problèmes	12
3.2.2. Études de cas	12
3.3. Résumé	12
<b>4. Techniques de Classification</b>	<b>12</b>
4.1. La Classification Supervisée	12
4.1.1. Techniques et algorithmes	12
4.1.1.1. Arbres de décision	12
4.1.1.2. Forêt aléatoire	12
4.1.1.3. Machine à vecteurs de support (SVM)	12
4.1.1.4. Réseaux de neurones	12
4.1.2. Métriques de performance	12
4.1.2.1. Précision	12
4.1.2.2. Rappel	12
4.1.2.3. F1-score	12
4.1.2.4. Courbe ROC-AUC	12
4.1.3. Cas d'utilisation	12
4.1.3.1. Diagnostic médical	12
4.1.3.2. Filtrage de courriers indésirables	12
4.1.3.3. Reconnaissance de la parole	12
4.2. La Classification Non Supervisée	12
4.2.1. Techniques et algorithmes	12
4.2.1.1. K-means	12
4.2.1.2. Clustering hiérarchique	12
4.2.1.3. Algorithme DBSCAN	12
4.2.2. Métriques de performance	12
4.2.2.1. Indice de Silhouette	12
4.2.2.2. Davies-Boudin Index	12
4.2.2.3. Matrice de confusion (utilisée avec étiquetage manuel)	12
4.2.3. Cas d'utilisation	12
4.2.3.1. Segmentation de clients	12
4.2.3.2. Détection de fraudes	12
4.2.3.3. Classification de documents	12
4.3. Résumé	13
<b>5. Techniques de Régression</b>	<b>13</b>
5.1. La Régression	13
5.1.1. Concepts de base	13
5.1.1.1. Définition	13
5.1.1.2. Objectif	13
5.1.2. Types de Régression	13
5.1.2.1. Régression Linéaire	13
5.1.2.2. Régression Polynomiale	13
5.1.2.3. Régression Logistique	13
5.1.2.4. Régression Ridge et Lasso	13
5.1.3. Exemples pratiques	13
5.2. Résumé	13

<b>6. Validation et Évaluation</b>	<b>13</b>
6.1. La Validation Croisée	13
6.1.1. Importance de la validation croisée	13
6.1.1.1. Objectif	13
6.1.1.2. Avantages	13
6.1.2. Méthodes de Validation Croisée	13
6.1.2.1. K-fold Cross-Validation	13
6.1.2.2. Leave-One-Out Cross Validation	13
6.1.2.3. Stratified K-fold Cross Validation	13
6.1.3. Techniques Avancées	13
6.1.3.1. Nested Cross-Validation	13
6.1.3.2. Monte Carlo Cross-Validation	13
6.2. Les Données d'Entraînement, de Test et de Validation	13
6.2.1. Rôle de chaque type de données	13
6.2.1.1. Données d'Entraînement	13
6.2.1.2. Données de Test	13
6.2.1.3. Données de Validation	13
6.2.2. Stratégies de partitionnement	13
6.2.2.1. Partitionnement Simple	13
6.2.2.2. Partitionnement Stratifié	13
6.2.3. Impact sur la performance des modèles	13
6.2.3.1. Ensemble d'Entraînement de Taille Adéquate	13
6.2.3.2. Données de Test Représentatives	14
6.2.3.3. Validation Croisée	14
6.3. Résumé	14
<b>7. Optimisation et Fonctionnement des Modèles</b>	<b>14</b>
7.1. Fiabilité, Biais et Bruit	14
7.1.1. Fiabilité	14
7.1.1.1. Définition et importance de la fiabilité	14
7.1.1.2. Méthodes d'Évaluation	14
7.1.1.3. Cas d'utilisation	14
7.1.2. Biais	14
7.1.2.1. Définition et Sources de Biais	14
7.1.2.2. Impact des Biais	14
7.1.2.3. Techniques de Correction	14
7.1.3. Bruit	14
7.1.3.1. Définition et Types de Bruit	14
7.1.3.2. Impact du Bruit	14
7.1.3.3. Techniques de Réduction du Bruit	14
7.2. Une Fonction de Coût	14
7.2.1. Définition et importance dans le Machine Learning	14
7.2.1.1. Définition détaillée de la fonction de coût	14
7.2.1.2. Importance de la fonction de coût dans l'entraînement des modèles	14
7.2.2. Types courants de fonctions de coût	14
7.2.2.1. Fonction de coût quadratique (ou de l'erreur quadratique moyenne / MSE)	14
7.2.2.2. Fonction de coût logarithmique (log-loss)	14
7.2.2.3. Fonction de coût Hinge	14
7.2.3. Rôle dans les modèles de Machine Learning	14
7.2.3.1. Comment les fonctions de coût influencent l'entraînement du modèle	14
7.2.3.2. Impact sur les décisions de mise à jour des poids	14
7.2.4. Utilisation pour évaluer la performance du modèle	14
7.2.4.1. Mesurer la qualité de la prédiction	14

7.2.4.2. Comparaison entre différentes fonctions de coût pour un même modèle	14
7.2.5. Relation avec la précision et le sur-ajustement	14
7.2.5.1. Comment la fonction de coût aide à trouver le bon compromis entre précision et sur-ajustement	14
7.3. Fonctions de coût spécifiques	14
7.3.1. Fonctions de coût pour les problèmes de régression	15
7.3.1.1. Erreur quadratique moyenne (MSE)	15
7.3.1.2. Erreur absolue moyenne (MAE)	15
7.3.2. Fonctions de coût pour les problèmes de classification	15
7.3.2.1. Entropie croisée	15
7.3.2.2. Hinge Loss	15
7.4. La Descente de Gradient	15
7.4.1. Principe et algorithme de base	15
7.4.1.1. Introduction à la descente de gradient	15
7.4.1.2. Étapes de l'algorithme classique (calcul de gradient, mise à jour des poids ...)	15
7.4.2. Variantes et optimisations avancées	15
7.4.2.1. Descente de gradient stochastique et mini-batch	15
7.4.2.2. Technique d'optimisation (momentum, RMSprop, Adam ...)	15
7.5. Résumé	15
<b>8. Conclusion</b>	<b>15</b>
8.1. Synthèse des points principaux	15
8.2. Défis actuels et limites du Machine Learning	15
8.3. Perspectives et évolutions futures	15

# 1. Introduction



## 1.1. Contexte et importance du Machine Learning

Le Machine Learning, ou apprentissage automatique, est une branche de l'intelligence artificielle (IA) se concentrant sur la création de systèmes capables d'apprendre à partir de données et de s'améliorer avec l'expérience.

### 1.1.1. Contexte du Machine Learning

Le concept même de Machine Learning trouve ses racines dans plusieurs disciplines telles que :

#### 1.1.1.1. Statistiques et probabilités

Les modèles statistiques sont à la base de nombreux algorithmes de Machine Learning.



#### 1.1.1.2. Informatique

La capacité de traiter des quantités massives de données et de les analyser rapidement est rendue possible par les avancées en informatique.



### 1.1.1.3. Neurosciences et Cognition

Le fonctionnement des réseaux de neurones artificiels s'inspire du cerveau humain.



## 1.1.2. Importance du Machine Learning

Le Machine Learning est de plus en plus essentiel pour plusieurs raisons :



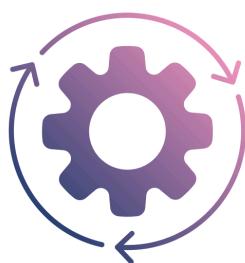
### 1.1.2.1. Analyse de Données Massives

Avec l'explosion des données (Big Data), les algorithmes de Machine Learning sont capables d'extraire des informations précieuses et de repérer des tendances cachées.



### 1.1.2.2. Automatisation des Tâches

Des tâches complexes peuvent être automatisées, comme la reconnaissance faciale, la traduction automatique et la recommandation de produits.



### 1.1.2.3. Amélioration Continue

Les systèmes de Machine Learning s'améliorent avec le temps, devenant de plus en plus précis et efficaces à mesure qu'ils traitent de nouvelles données.



### 1.1.2.4. Application Innovantes

Ils sont utilisés dans une variété d'industries, allant de la médecine (diagnostics basés sur les images) à la finance (détection des fraudes) et à l'agriculture (optimisation des cultures).

### 1.1.3. Exemples

Santé :

Diagnostic de maladies à partir d'images médicales comme les radiographies ou les IRM, afin d'identifier des anomalies plus rapidement et avec une précision parfois supérieure à celle des radiologues humains.

Prévision des épidémies en analysant des données provenant de diverses sources, permettant ainsi de prendre des mesures préventives plus efficaces.



Transport :



Véhicules autonomes grâce à l'interprétation des signaux provenant de capteurs en temps réel assurant de surcroît la sécurité et l'efficacité de la conduite.

Optimisation des itinéraires via des systèmes de gestion du trafic analysant les conditions de circulation en temps réel et proposant les itinéraires les plus rapides ou économiques, réduisant ainsi embouteillage et consommation de carburant.

Marketing :

Personnalisation des publicités via l'analyse des historiques d'achats et les préférences clients, offrant des recommandations de produits personnalisées et pertinentes.

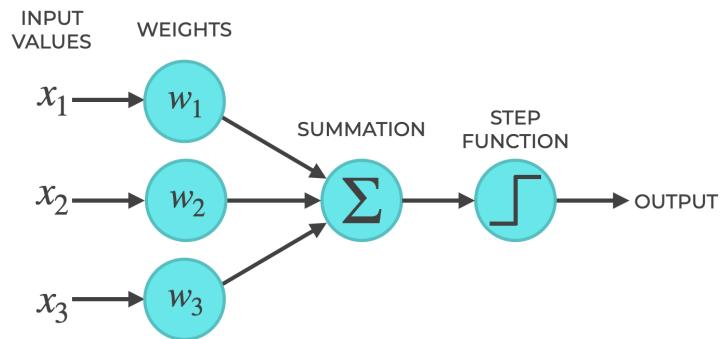


Le machine Learning joue un rôle crucial dans l'ère numérique actuelle, en transformant la manière dont les données sont analysées et utilisées pour prendre des décisions éclairées.

## 1.2. Evolution historique

### 1.2.1. Les débuts

#### THE STRUCTURE OF A PERCEPTRON



L'histoire du Machine Learning (ML) remonte au milieu du 20e siècle grâce au travail pionnier du mathématicien britannique Alan Turing qui permit de jeter les bases de l'Intelligence Artificielle, suscitant une curiosité quant à la possibilité de créer des machines capables de penser et d'apprendre. En 1950, Turing publie un article intitulé “*Computing Machinery and Intelligence*”, dans lequel il propose le célèbre test de Turing comme critère d'intelligence d'une machine. C'est en 1956, lors de la Conférence de Dartmouth, qu'apparaît officiellement le terme

Intelligence Artificielle ouvrant ainsi la voie au développement du ML.

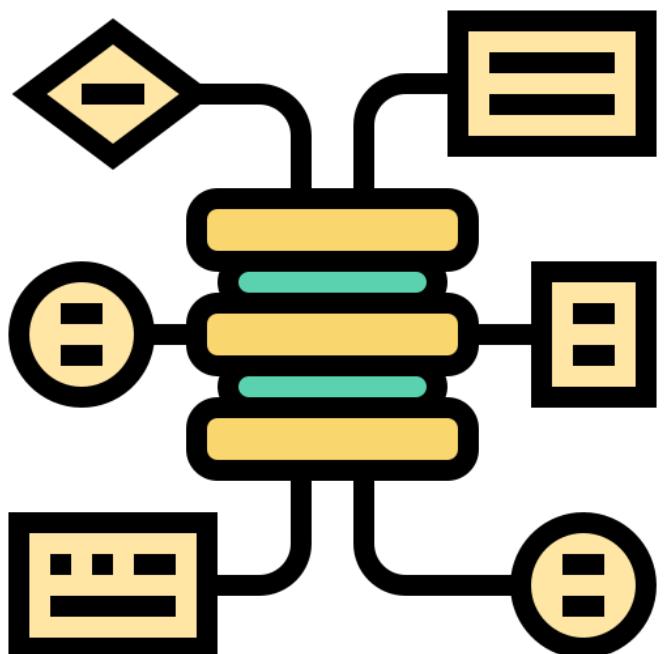
Un des premiers programmes d'apprentissage automatique fut le “Perceptron”, développé par Frank Rosenblatt en 1957, qui était capable d'apprendre à reconnaître des formes simples.

### 1.2.2. L'essor des algorithmes de ML

Au cours des décennies suivantes, les chercheurs ont développé une large gamme d'algorithmes de ML, chacun ayant ses forces et faiblesses.

Durant les années 1960, les arbres de décision sont devenus une méthode populaire pour résoudre les problèmes de classification, grâce à leur structure intuitive et leur facilité d'interprétation.

Les années 1980 ont vu l'introduction de techniques plus avancées, telles que les machines à vecteur de supports (SVM) et les K-Nearest Neighbours (K-NN), offrant performances et polyvalence améliorées.

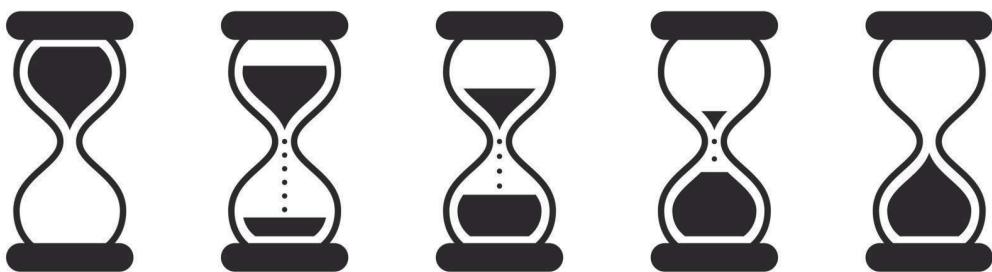


Continuant d'évoluer, le domaine du ML s'enrichit de nouvelles méthodes d'ensemble telles que Random Forests et Gradient Boosting (1990) repoussant encore et toujours les limites de ce que le ML peut réaliser.



### 1.2.3. La révolution des données

L'histoire du ML est indéniablement liée à la croissance des données numériques. Au fur et à mesure de la numérisation du monde, de grandes quantités de données sont devenues disponibles, permettant d'explorer un terrain fertile pour le développement des algorithmes de ML. Avec l'émergence d'Internet, des médias sociaux et du Internet des Objets (IoT), une quantité toujours plus grande de données voient le jour, donnant aux algorithmes de ML de la matière première essentielle à leur développement.



### 1.2.4. Aujourd'hui et demain

Aujourd'hui, le ML est à l'avant-garde de la recherche et du développement en IA, avec l'apprentissage profond et l'apprentissage par renforcement.

Ces techniques avancées ont ouvert de nouvelles perspectives dans des domaines tels que la robotique, les véhicules autonomes et l'IA du jeu, mettant ainsi en valeur l'incommensurable potentiel du ML au 21e siècle et à l'avenir.

Au fur et à mesure de son évolution, le ML permet aux chercheurs et praticiens d'explorer de nouveaux moyens d'exploiter ses capacités, repoussant les limites du possible et ouvrant des opportunités incalculables de croissance et de progrès.

## 2. Fondamentaux du Machine Learning

Le Machine Learning, ou apprentissage automatique, repose sur la capacité des systèmes informatiques à apprendre à partir de données pour effectuer des prédictions ou prendre des décisions sans être explicitement programmés. Les fondements de cette discipline s'appuient sur la science des données, qui englobe la collecte, le prétraitement et l'analyse exploratoire des données.

### 2.1. La Science des Données

La science des données est un domaine interdisciplinaire qui combine des techniques statistiques, informatiques et analytiques pour extraire des connaissances et des insights à partir de données structurées et non structurées. Elle est essentielle pour transformer des données brutes en informations exploitables, servant de base au Machine Learning.

#### 2.1.1. Collecte de Données

La collecte de données est le processus de rassemblement d'informations pertinentes pour une analyse ultérieure. Elle peut être effectuée via diverses méthodes, chacune présentant ses propres défis et nécessitant des outils spécifiques.

##### 2.1.1.1. Méthodes et défis

Les principales méthodes de collecte de données incluent :

- **Méthodes quantitatives** : Elles impliquent la collecte de données numériques pouvant être mesurées et analysées statistiquement. Les enquêtes et les questionnaires sont des exemples courants. Ces méthodes permettent de recueillir des informations sur les tendances, les préférences et les comportements des individus.
- **Méthodes qualitatives** : Ces techniques visent à obtenir une compréhension approfondie des motivations, des opinions et des raisons sous-jacentes. Elles incluent des entretiens, des groupes de discussion et des observations. Ces méthodes sont particulièrement utiles pour explorer des sujets complexes où les réponses ne peuvent pas être facilement quantifiées.

Les défis associés à la collecte de données comprennent :

- **Qualité des données** : Assurer l'exactitude, la cohérence et la fiabilité des données collectées est crucial pour éviter des analyses biaisées ou erronées.

- **Volume de données** : Gérer de grandes quantités de données peut être complexe et nécessite des infrastructures adéquates pour le stockage et le traitement.
- **Respect de la vie privée** : Il est essentiel de protéger les informations personnelles des individus et de se conformer aux réglementations en vigueur lors de la collecte de données.

#### 2.1.1.2. Outils de collecte

Divers outils sont disponibles pour faciliter la collecte de données :

- **Web scraping** : Cette technique permet d'extraire des informations à partir de sites web en utilisant des outils tels que BeautifulSoup ou Scrapy. Elle est particulièrement utile pour collecter des données publiques disponibles en ligne.
- **API (Interfaces de Programmation d'Applications)** : Les API permettent d'accéder à des données structurées fournies par des services en ligne, comme les réseaux sociaux ou les plateformes de commerce électronique. Par exemple, l'API Twitter offre un accès aux tweets publics pour des analyses diverses.
- **Plateformes d'enquête en ligne** : Des outils tels que SurveyMonkey ou Google Forms facilitent la création et la distribution de questionnaires pour collecter des données directement auprès des participants.

### 2.1.2. Prétraitement des Données

Le prétraitement des données est une étape cruciale qui consiste à transformer des données brutes en un format approprié pour l'analyse. Il améliore la qualité des données et augmente la précision des modèles analytiques.

#### 2.1.2.1. Techniques courantes (nettoyage, normalisation, réduction de dimensionnalité)

- **Nettoyage des données** : Cette technique vise à identifier et corriger les erreurs, telles que les valeurs manquantes, les doublons ou les incohérences, afin d'améliorer la qualité des données.
- **Normalisation** : Elle consiste à transformer les variables pour les placer sur une échelle commune, facilitant ainsi la comparaison et l'analyse des données. Par exemple, la mise à l'échelle des valeurs pour qu'elles se situent entre 0 et 1.

- **Réduction de dimensionnalité** : Cette technique vise à diminuer le nombre de variables dans un ensemble de données tout en conservant les informations essentielles, ce qui peut améliorer l'efficacité des modèles et réduire le risque de surapprentissage.

#### 2.1.2.2. Cas d'utilisation pratiques

Le prétraitement des données est appliqué dans divers domaines :

- **Finance** : Nettoyer et normaliser les données financières pour détecter des anomalies ou prédire des tendances du marché.
- **Santé** : Prétraiter les données médicales pour identifier des facteurs de risque ou personnaliser des traitements.
- **Marketing** : Analyser les comportements des consommateurs en nettoyant et en structurant les données issues des interactions clients.

#### 2.1.3. Analyse Exploratoire des Données

L'analyse exploratoire des données (AED) est une étape fondamentale du processus de data science. Elle consiste à examiner, comprendre et visualiser les données avant d'appliquer des modèles de Machine Learning. L'objectif est d'identifier des tendances, des relations, des anomalies et d'obtenir des insights exploitables.

Une AED bien réalisée permet de :

- Déetecter les erreurs et incohérences dans les données.
- Comprendre la distribution des variables et leurs relations.
- Identifier les valeurs aberrantes (outliers) pouvant fausser les analyses.
- Guider le choix des modèles de Machine Learning les plus appropriés.

##### 2.1.3.1. Outils et visualisations

L'analyse exploratoire repose sur des outils statistiques et des techniques de visualisation qui facilitent l'interprétation des données.

###### **Outils d'analyse exploratoire**

Plusieurs outils permettent d'effectuer une AED de manière efficace :

- **Python** : avec des bibliothèques comme Pandas, NumPy, Seaborn et Matplotlib.

- **R** : un langage puissant pour l'analyse statistique, avec ggplot2 pour la visualisation.
- **Tableau** : un logiciel interactif pour explorer et visualiser les données sans coder.
- **Power BI** : une alternative à Tableau, intégrée aux solutions Microsoft.
- **Excel** : bien que limité pour de gros volumes de données, il reste utile pour les statistiques descriptives et graphiques de base.

### Techniques de visualisation

La visualisation des données est essentielle pour comprendre leur distribution et leurs relations. Voici quelques graphiques fréquemment utilisés :

- **Histogrammes** : Ils permettent d'analyser la distribution des données numériques. Exemple : la répartition des âges dans un dataset de clients.
- **Diagrammes en boîte (boxplots)** : Ils aident à repérer les valeurs aberrantes et comprendre la dispersion des données.
- **Nuages de points (scatter plots)** : Ils illustrent la relation entre deux variables, comme la corrélation entre la taille et le poids d'un individu.
- **Matrices de corrélation** : Elles montrent la force et la direction des relations entre plusieurs variables.
- **Courbes de densité** : Elles permettent d'observer la distribution d'une variable continue.
- **Heatmaps (cartes de chaleur)** : Utilisées pour visualiser des matrices de données et identifier des tendances cachées

### 2.1.3.2. Études de cas

#### Cas 1 : Analyse des ventes d'une entreprise e-commerce

**Contexte :** Une entreprise de vente en ligne souhaite comprendre les tendances de ses ventes pour optimiser ses campagnes marketing.

**Approche :**

- Chargement et nettoyage des données (suppression des valeurs manquantes).
- Visualisation de la répartition des ventes par catégorie de produits via un histogramme.
- Analyse des pics de ventes au cours du mois avec un graphique en courbes.
- Identification des produits générant le plus de chiffre d'affaires à l'aide d'une heatmap.

**Résultat :** L'analyse révèle que les ventes augmentent considérablement lors des promotions et pendant les fêtes de fin d'année. L'entreprise décide d'intensifier ses campagnes marketing en novembre et décembre.

---

#### Cas 2 : Détection de fraudes bancaires

**Contexte :** Une banque cherche à identifier les transactions suspectes dans les paiements par carte bancaire.

**Approche :**

- Utilisation de boxplots pour détecter les transactions présentant des montants anormalement élevés.
- Visualisation des transactions sur une carte pour repérer des anomalies géographiques.
- Création d'une matrice de corrélation pour analyser les liens entre variables (ex. nombre de transactions effectuées et montants dépensés).

**Résultat :** L'analyse met en évidence des comportements inhabituels (achats de montants élevés en peu de temps à l'étranger), permettant à la banque de renforcer ses contrôles et d'améliorer la détection de fraudes.

---

### **Cas 3 : Prédiction de la satisfaction des clients dans un centre d'appels**

**Contexte :** Une entreprise veut analyser les facteurs influençant la satisfaction de ses clients après une interaction avec le service client.

**Approche :**

- Regroupement des avis clients et analyse de la fréquence des mots-clés les plus utilisés dans les feedbacks.
- Utilisation de diagrammes circulaires pour observer la répartition des niveaux de satisfaction.
- Analyse des temps d'attente moyens et de leur impact sur la satisfaction avec un scatter plot.

**Résultat :** L'étude montre que les clients les plus insatisfaits sont ceux qui ont attendu plus de 10 minutes avant d'être pris en charge. L'entreprise décide alors d'optimiser la gestion des appels pour réduire le temps d'attente.

### **Conclusion**

L'analyse exploratoire des données est une étape cruciale dans tout projet de data science et de machine learning. Elle permet de mieux comprendre les données avant de les utiliser dans un modèle prédictif. Grâce à des outils et des visualisations adaptés, les entreprises et chercheurs peuvent détecter des tendances, identifier des anomalies et prendre des décisions éclairées basées sur les données.

## 2.2. L'Apprentissage Automatique et Profond

L'intelligence artificielle repose sur deux sous-domaines essentiels : l'apprentissage automatique (Machine Learning) et l'apprentissage profond (Deep Learning). Ces techniques permettent aux machines d'analyser des données, d'apprendre des schémas et d'effectuer des prédictions ou classifications avec un minimum d'intervention humaine.

### 2.2.1. L'apprentissage Automatique (Machine Learning)

#### 2.2.1.1. Définition et Principe

L'apprentissage automatique est une branche de l'intelligence artificielle qui consiste à développer des algorithmes capables d'apprendre à partir de données et d'effectuer des tâches précises sans être explicitement programmés pour chacune d'elles. Le Machine Learning repose sur des modèles statistiques et mathématiques qui ajustent leurs paramètres en fonction des observations fournies.

#### 2.2.1.2. Algorithmes courants

Les algorithmes de Machine Learning sont classés en trois grandes catégories :

- **Apprentissage supervisé** : Utilisé lorsque les données d'entraînement sont étiquetées. Exemples d'algorithmes :
  - Régression linéaire et logistique
  - Machines à vecteurs de support (SVM)
  - Arbres de décision et forêts aléatoires
  - Réseaux de neurones artificiels simples
- **Apprentissage non supervisé** : Utilisé pour analyser des données non étiquetées et découvrir des structures sous-jacentes. Exemples :
  - Algorithmes de clustering (K-means, DBSCAN)
  - Réduction de dimensionnalité (ACP, t-SNE)
- **Apprentissage par renforcement** : Basé sur l'interaction avec un environnement et l'apprentissage par essai-erreur. Exemples :
  - Q-Learning
  - Deep Q-Networks (DQN)

### 2.2.1.3. Applications courantes

L'apprentissage automatique est appliqué dans de nombreux domaines :

- **Reconnaissance d'images et de texte** : OCR, détection d'objets
- **Systèmes de recommandation** : Netflix, Amazon, Spotify
- **Analyse des sentiments et NLP** : Traitement du langage naturel
- **Prédiction et analyse financière** : Détection des fraudes bancaires
- **Médecine** : Diagnostic assisté par intelligence artificielle

## 2.2.2. L'apprentissage Profond (Deep Learning)

### 2.2.2.1. Définition et Principe

L'apprentissage profond est une sous-catégorie du Machine Learning qui utilise des réseaux de neurones artificiels composés de plusieurs couches cachées pour modéliser des relations complexes dans les données. Ces modèles sont particulièrement efficaces pour le traitement des images, du texte et de la parole.

### 2.2.2.2. Réseaux de Neurones Profonds

Les réseaux de neurones profonds sont des structures inspirées du cerveau humain, où chaque couche apprend une représentation abstraite des données d'entrée. Exemples de réseaux :

- **Réseaux de neurones convolutifs (CNN)** : Utilisés pour la vision par ordinateur
- **Réseaux de neurones récurrents (RNN, LSTM, GRU)** : Spécialisés dans les séries temporelles et le NLP
- **Transformers** : Modèles avancés utilisés dans GPT, BERT

### 2.2.2.3. Applications courantes

- **Voitures autonomes** : Analyse en temps réel des images
- **Assistants vocaux** : Siri, Alexa, Google Assistant
- **Traduction automatique** : Google Translate
- **Génération de texte** : ChatGPT, BERT

### 2.2.3. Comparaison

Critère	Machine Learning	Deep Learning
Données requises	Moins de données	Grandes quantité de données
Temps d'entraînement	Rapide à modéré	Long (GPU/TPU nécessaire)
Interprétabilité	Bonne	Difficile (boîte noire)
Exemple d'application	Prédictions, recommandations	Reconnaissance vocale, vision

## 2.3. Corrélation Linéaire (de Pearson) entre Deux Variables

La corrélation linéaire est un outil statistique permettant de mesurer la relation entre deux variables quantitatives. L'un des indices les plus utilisés est le coefficient de corrélation de Pearson.

### 2.3.1. Définition et calcul

#### 2.3.1.1. Définition

Le coefficient de Pearson mesure la force et la direction d'une relation linéaire entre deux variables. Il varie entre -1 (corrélation négative parfaite) et +1 (corrélation positive parfaite), avec 0 indiquant une absence de corrélation.

#### 2.3.1.2. Formule de Calcul

Le coefficient de Pearson est donné par :

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2} \times \sqrt{\sum(Y_i - \bar{Y})^2}}$$

où :

- $X_i$  et  $Y_i$  sont les valeurs des variables,
- $X$  et  $Y$  sont leurs moyennes respectives.

### **2.3.2. Applications pratiques**

#### **2.3.2.1. Analyse de la Relation entre Variables**

La corrélation de Pearson est utilisée pour vérifier si deux variables sont liées, comme :

- La température et la consommation d'énergie
- Le revenu et la consommation de produits de luxe

#### **2.3.2.2. Modélisation et Prédiction**

Une forte corrélation peut indiquer une relation causale potentielle, ce qui est utile pour la construction de modèles de Machine Learning.

#### **2.3.2.3. Recherche Scientifique et Études Empiriques**

Dans les sciences sociales et la médecine, la corrélation de Pearson permet d'évaluer des relations entre divers phénomènes (ex : niveau d'éducation et revenu).

### **2.3.3. Exemples concrets**

- Étude de la corrélation entre l'exercice physique et l'espérance de vie.
- Analyse du lien entre le nombre d'heures de sommeil et la productivité au travail.
- Impact du prix du pétrole sur les coûts de transport.

### 3. Méthodes d'Apprentissage

L'apprentissage automatique se divise en plusieurs catégories selon la nature des données et des objectifs d'apprentissage. Parmi celles-ci, on distingue principalement l'apprentissage supervisé et l'apprentissage non supervisé.

#### 3.1. L'apprentissage Supervisé

L'apprentissage supervisé est une méthode dans laquelle un algorithme apprend à partir d'un ensemble de données d'entraînement étiquetées. L'objectif est de généraliser un modèle capable de prédire la sortie correcte pour de nouvelles données.

##### 3.1.1. Principe et fonctionnement

###### 3.1.1.1. Définition

L'apprentissage supervisé consiste à entraîner un modèle en lui fournissant des exemples associant des entrées (features) à des sorties attendues (labels). L'algorithme ajuste ses paramètres afin de minimiser l'erreur entre ses prédictions et les valeurs réelles.

###### 3.1.1.2. Processus

Le processus d'apprentissage supervisé suit plusieurs étapes :

- **Collecte et préparation des données** : Sélection et nettoyage des données.
- **Séparation des données** : Répartition en ensembles d'entraînement et de test.
- **Sélection d'un modèle** : Choix d'un algorithme adapté (régression, classification).
- **Entraînement du modèle** : Ajustement des paramètres à partir des données d'entraînement.
- **Évaluation** : Vérification de la performance sur l'ensemble de test.
- **Optimisation** : Réglage des hyperparamètres pour améliorer les performances.

###### 3.1.1.3. Types de problèmes

Les principales applications de l'apprentissage supervisé sont :

- **Classification** : Attribution d'une catégorie à une donnée (ex : reconnaissance d'images).
- **Régression** : Prédiction d'une valeur continue (ex : estimation des prix immobiliers).

### 3.1.2. Études de cas

- **Détection de spams** : Identification automatique des e-mails indésirables.
- **Prédiction de maladies** : Utilisation d'algorithmes pour diagnostiquer des pathologies à partir de symptômes.
- **Reconnaissance faciale** : Utilisation de modèles entraînés pour identifier des visages dans des images.

## 3.2. L'apprentissage Non Supervisé

Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé fonctionne sans étiquettes explicites. L'algorithme analyse les structures sous-jacentes des données et regroupe les éléments similaires.

### 3.2.1. Principe et fonctionnement

#### 3.2.1.1. Définition

L'apprentissage non supervisé cherche à découvrir des modèles cachés dans des données brutes en regroupant ou en réduisant la dimensionnalité des données sans indication préalable.

#### 3.2.1.2. Processus

Les principales étapes de l'apprentissage non supervisé sont :

- **Prétraitement des données** : Normalisation et nettoyage.
- **Choix d'un algorithme** : Sélection de la méthode d'analyse (clustering, réduction de dimensionnalité).
- **Analyse des structures** : Recherche des motifs récurrents dans les données.
- **Validation et interprétation** : Évaluation des regroupements obtenus.

#### 3.2.1.3. Types de problèmes

- **Clustering** : Segmentation des clients en marketing.
- **Réduction de dimensionnalité** : Compression d'images tout en conservant l'essentiel des informations.
- **Détection d'anomalies** : Identification de fraudes bancaires.

### 3.2.2. Études de cas

- **Segmentation des clients** : Analyse des habitudes d'achat pour adapter les stratégies marketing.
- **Détection de fraudes** : Identification de transactions suspectes sans étiquetage préalable.
- **Systèmes de recommandation** : Groupement des préférences utilisateur pour proposer du contenu pertinent.

## 3.3. Résumé

L'apprentissage supervisé et non supervisé sont deux piliers essentiels du Machine Learning, chacun étant adapté à des problématiques spécifiques. Le choix entre ces approches dépend du type de données et de l'objectif recherché.

Critère	Apprentissage Supervisé	Apprentissage Non Supervisé
Données étiquetée	Oui	Non
Objectif	Prédictions	Regroupement, analyse
Algorithmes courants	Régression, SVM, Réseaux de neurones	K-means, PCA, autoencodeurs
Applications	Prédictions de ventes, reconnaissance faciale	Détection d'anomalies, segmentation des clients

# 4. Techniques de Classification

La classification est un problème central en apprentissage supervisé. Elle consiste à attribuer des catégories à des données d'entrée, souvent utilisées dans des domaines tels que la reconnaissance d'images, la prédiction des comportements d'achats, ou encore la détection de fraudes.

## 4.1. La Classification Supervisée

### 4.1.1. Techniques et algorithmes

Les algorithmes de classification supervisée sont utilisés lorsque les données d'apprentissage sont étiquetées. Cela signifie que chaque donnée d'entrée est associée à une étiquette ou catégorie. Voici quelques-uns des algorithmes les plus utilisés :

#### 4.1.1.1. Arbres de décision

Un arbre de décision est un modèle prédictif qui divise les données en sous-ensembles basés sur des critères de décision. L'objectif est de créer une structure en forme d'arbre où chaque nœud représente une décision à prendre, et les feuilles contiennent les étiquettes de classe.

- **Avantages** : Facile à comprendre et à interpréter, rapide à entraîner.
- **Inconvénients** : Sensible à l'overfitting (surapprentissage)

#### 4.1.1.2. Forêt aléatoires

Une forêt aléatoire est un ensemble d'arbres de décision. Elle utilise plusieurs arbres de décision indépendants pour faire des prédictions. Les résultats des arbres sont combinés pour améliorer la précision et réduire la variance par rapport à un seul arbre de décision.

- **Avantages** : Moins susceptible au surapprentissage, meilleure performance sur des ensembles de données complexes.
- **Inconvénients** : Moins interprétable qu'un arbre de décision unique.

#### 4.1.1.3. Machine à vecteurs de support (SVM)

La machine à vecteurs de support est un algorithme de classification qui cherche à trouver une hyperplan qui sépare au mieux les classes de données. SVM est particulièrement efficace dans des espaces de grande dimension et pour des marges de séparation nettes.

- **Avantages** : Efficace dans des espaces de haute dimension, performe bien avec des marges larges.
- **Inconvénients** : Moins performant avec de grandes quantités de données.

#### 4.1.1.4. Réseaux de neurones

Les réseaux de neurones sont inspirés du fonctionnement du cerveau humain. Ces modèles sont capables de classer des données complexes en apprenant des représentations à partir des données d'entrée.

- **Avantages** : Très puissant pour des données non linéaires et de grande dimension.
- **Inconvénients** : Nécessite un grand volume de données et de puissance de calcul.

### 4.1.2. Métriques de performance

Pour évaluer la performance des algorithmes de classification, plusieurs métriques sont utilisées. Les principales sont :

#### 4.1.2.1. Précision

La précision est le rapport entre le nombre de prédictions correctes et le nombre total de prédictions. Elle mesure la capacité d'un modèle à prédire correctement les instances positives.

- **Formule** :  $\text{Précision} = (\text{Vrais positifs}) / (\text{Vrais positifs} + \text{Faux positifs})$ .

#### 4.1.2.2. Rappel

Le rappel, ou sensibilité, mesure la capacité d'un modèle à identifier toutes les instances positives.

- **Formule** :  $\text{Rappel} = (\text{Vrais positifs}) / (\text{Vrais positifs} + \text{Faux négatifs})$ .

#### 4.1.2.3. F1-score

Le F1-score est la moyenne harmonique entre la précision et le rappel. Il est utilisé lorsqu'il y a un déséquilibre entre les classes positives et négatives.

- **Formule :**  $F1\text{-score} = 2 * (\text{Précision} * \text{Rappel}) / (\text{Précision} + \text{Rappel})$ .

#### 4.1.2.4. Courbe ROC-AUC

La courbe ROC (Receiver Operating Characteristic) et l'AUC (Area Under Curve) mesurent la performance du modèle en termes de taux de vrais positifs (sensibilité) contre le taux de faux positifs. L'AUC donne une évaluation du modèle, avec une valeur proche de 1 indiquant un modèle très performant.

### 4.1.3. Cas d'utilisation

#### 4.1.3.1. Diagnostic médical

Les modèles de classification supervisée peuvent aider à prédire des maladies à partir de données médicales, comme la classification des images médicales pour détecter des anomalies telles que les tumeurs.

#### 4.1.3.2. Filtrage de courriers indésirables

Les systèmes de classification supervisée sont utilisés pour identifier et filtrer les courriers indésirables (spams) dans les boîtes de réception des utilisateurs.

#### 4.1.3.3. Reconnaissance de la parole

Les modèles de classification sont également appliqués dans la reconnaissance vocale, où des algorithmes sont entraînés à associer des sons à des mots ou des commandes spécifiques.

## 4.2. La Classification Non Supervisée

L'apprentissage non supervisé est utilisé lorsque les données d'apprentissage ne contiennent pas d'étiquettes. Ce type d'apprentissage cherche à découvrir des structures cachées ou des relations dans les données.

### 4.2.1. Techniques et algorithmes

#### 4.2.1.1. K-means

L'algorithme K-means est utilisé pour regrouper les données en k clusters distincts, en minimisant la variance intra-cluster. Il est simple et efficace pour des tâches de segmentation.

- **Avantages** : Rapide et simple à comprendre.
- **Inconvénients** : Nécessite de spécifier le nombre de clusters à l'avance.

#### 4.2.1.2. Clustering hiérarchique

Le clustering hiérarchique crée une hiérarchie de clusters en fusionnant ou divisant les clusters de manière itérative. Il peut être utilisé pour explorer les données sans avoir à définir un nombre de clusters.

- **Avantages** : Aucune hypothèse préalable sur le nombre de clusters.
- **Inconvénients** : Plus coûteux en termes de calcul.

#### 4.2.1.3. Algorithme DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est un algorithme de clustering basé sur la densité. Il peut identifier des clusters de forme arbitraire et est robuste aux anomalies.

- **Avantages** : Identifie bien les outliers.
- **Inconvénients** : Sensible aux paramètres choisis.

## 4.2.2. Métriques de performance

Les métriques courantes en classification non supervisée incluent :

### 4.2.2.1. Indice de Silhouette

L'indice de silhouette mesure la qualité d'un clustering en prenant en compte la cohésion interne des clusters et leur séparation.

### 4.2.2.2. Davies-Boudin Index

L'indice de Davies-Bouldin est une mesure qui évalue la compacité et la séparation des clusters.

### 4.2.2.3. Matrice de confusion (utilisée avec étiquetage manuel)

La matrice de confusion peut être utilisée pour évaluer la performance des clusters si un étiquetage manuel est effectué, en comparant les prédictions avec les véritables catégories.

## 4.2.3. Cas d'utilisation

### 4.2.3.1. Segmentation de clients

La segmentation de clients utilise des algorithmes de clustering pour regrouper les consommateurs en fonction de leurs comportements d'achat.

### 4.2.3.2. Détection de fraudes

Le clustering non supervisé peut être utilisé pour détecter des comportements inhabituels ou suspects, comme les transactions frauduleuses dans les systèmes bancaires.

### 4.2.3.3. Classification de documents

Le clustering est utilisé pour organiser automatiquement des documents en catégories similaires, par exemple pour la recherche d'informations.

### 4.3. Résumé

Critère	Classification Supervisée	Classification Non Supervisée
Types de données	Données étiquetées	Données non étiquetée
Objectif	Prédiction de classe	Découverte de structure cachées
Algorithmes	Arbres de décision, SVM, Réseaux de neurones	K-means, DBSCAN, CLusetring hiérarchique
Applications	Diagnostic médical, Filtrage de spams	Segmentation des clients, Détection de fraudes

Les deux types de classification ont des applications très différentes, et le choix entre une méthode supervisée ou non supervisée dépend du type de données disponibles et de l'objectif de l'analyse.

# 5. Techniques de Régression

La régression est une technique statistique fondamentale en apprentissage supervisé utilisée pour prédire une variable continue en fonction d'autres variables. Contrairement à la classification, où l'objectif est de prédire une étiquette de classe, la régression cherche à prédire une valeur numérique.

## 5.1. La Régression

### 5.1.1. Concepts de base

#### 5.1.1.1. Définition

La régression est un type d'apprentissage supervisé où le modèle apprend à prédire une variable continue à partir de données d'entrée. L'objectif est de trouver la relation entre la variable cible (la variable à prédire) et les variables d'entrée (ou caractéristiques).

#### 5.1.1.2. Objectif

L'objectif principal de la régression est de minimiser l'erreur entre les prédictions du modèle et les valeurs réelles de la variable cible. Ce processus est souvent accompli en ajustant un modèle mathématique à travers un ensemble de données d'apprentissage. Les algorithmes de régression cherchent à identifier la fonction qui prédit le mieux la variable cible, en se basant sur les variables explicatives.

### 5.1.2. Types de Régression

Il existe plusieurs types d'algorithmes de régression, chacun ayant des caractéristiques spécifiques et des domaines d'application particuliers.

#### 5.1.2.1. Régression Linéaire

La régression linéaire est l'un des modèles de régression les plus simples. Elle suppose qu'il existe une relation linéaire entre la variable cible et les variables explicatives. Le modèle est représenté par une équation de la forme :

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

où  $y$  est la variable cible,  $x_1, x_2, \dots, x_n$  sont les variables explicatives, et  $\beta_0, \beta_1, \dots, \beta_n$  sont les coefficients à estimer.

- **Avantages** : Facile à comprendre, rapide à entraîner, interprétable.
- **Inconvénients** : Limité aux relations linéaires, sensible aux outliers.

#### 5.1.2.2. Régression Polynomiale

La régression polynomiale est une extension de la régression linéaire. Elle permet de modéliser des relations non linéaires en ajoutant des termes polynomiaux aux variables explicatives.

L'équation devient :

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \cdots + \beta_n x_n^p$$

où  $p$  est le degré du polynôme.

- **Avantages** : Permet de modéliser des relations non linéaires entre les variables.
- **Inconvénients** : Risque de surajustement (overfitting) si le degré du polynôme est trop élevé.

#### 5.1.2.3. Régression Logistique

Bien que son nom contienne "régression", la régression logistique est en réalité utilisée pour des problèmes de classification binaire. Elle permet de prédire la probabilité qu'une observation appartienne à une certaine classe (0 ou 1). Le modèle repose sur la fonction logistique (sigmoïde) pour prédire des valeurs comprises entre 0 et 1 :

$$P(y = 1|X) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n))}$$

Elle est principalement utilisée dans des contextes de classification, bien qu'elle puisse également être utilisée pour modéliser la probabilité dans des situations de régression.

- **Avantages** : Simple à comprendre, efficace pour des problèmes de classification binaire.
- **Inconvénients** : Ne peut pas être utilisée pour prédire des valeurs continues.

#### 5.1.2.4. Régression Ridge et Lasso

La régression Ridge et la régression Lasso sont des variantes de la régression linéaire qui ajoutent des régularisations pour éviter l'overfitting.

- **Régression Ridge** : Elle ajoute une pénalité basée sur la somme des carrés des coefficients ( $\lambda \sum_i \beta_i^2$ ), ce qui permet de réduire l'ampleur des coefficients sans les annuler complètement.
- **Régression Lasso** : Lasso (Least Absolute Shrinkage and Selection Operator) ajoute une pénalité basée sur la somme des valeurs absolues des coefficients ( $\lambda \sum_i |\beta_i|$ ), ce qui peut entraîner la réduction de certains coefficients à zéro, effectuant ainsi une sélection de caractéristiques.
- **Avantages** : Aide à gérer les modèles complexes et réduit l'overfitting.
- **Inconvénients** : La sélection de la régularisation ( $\lambda$ ) nécessite une validation croisée.

### 5.1.3. Exemples pratiques

Voici quelques exemples d'utilisation des techniques de régression dans différents domaines :

- **Prédiction des prix immobiliers** : Utilisation de la régression linéaire ou polynomiale pour prédire le prix des maisons en fonction de diverses caractéristiques, comme la superficie, l'emplacement, et l'âge du bien.
- **Analyse des ventes** : Utilisation de la régression pour prédire les ventes futures d'un produit en fonction des tendances passées et des variables économiques comme le revenu des consommateurs.
- **Estimation des risques en finance** : La régression peut être utilisée pour prédire la probabilité de défaut d'un emprunteur en fonction de ses caractéristiques financières.
- **Prévisions météorologiques** : Utilisation de la régression pour prédire les températures futures ou les niveaux de précipitations sur la base des données passées et des conditions actuelles.

## 5.2. Résumé

Critère	Régression Linéaire	Régression Polynomiale	Régression Logistique	Régression Ridge/Lasso
Type de relation	Linéaire	Non linéaire	Fonction logistique	Linéaire avec régularisation
Utilisation principale	Prédiction de valeurs continues	Modélisation de relations non linéaires	Classification binaire	Réduction de l'overfitting
Sensibilité aux outliers	Sensible	Sensible	Moins sensible	Moins sensible
Risque d'overfitting	Moyen	Élevé si le degré est trop élevé	Non applicable	Faible avec régularisation

# 6. Validation et Évaluation

La validation et l'évaluation sont des étapes essentielles dans le processus de construction d'un modèle d'apprentissage automatique. Elles permettent de s'assurer que le modèle généré est performant et capable de généraliser sur des données qu'il n'a pas vues durant l'entraînement. Cette section aborde les techniques de validation, les types de données utilisés, et les stratégies permettant d'évaluer la performance d'un modèle.

## 6.1. La Validation Croisée

### 6.1.1. Importance de la validation croisée

La validation croisée est une méthode clé dans l'évaluation de modèles d'apprentissage automatique. Elle permet de tester un modèle sur différents sous-ensembles de données afin de s'assurer qu'il est robuste et qu'il ne s'adapte pas de manière excessive (overfitting) aux données d'entraînement.

#### 6.1.1.1. Objectif

L'objectif principal de la validation croisée est de fournir une estimation plus fiable de la performance d'un modèle. En divisant les données en plusieurs sous-ensembles, la validation croisée permet de tester le modèle sur différentes portions des données, ce qui permet de mieux comprendre sa capacité à généraliser.

#### 6.1.1.2. Avantages

Les principaux avantages de la validation croisée incluent :

- **Évaluation plus fiable** : Contrairement à une simple séparation entre données d'entraînement et de test, la validation croisée permet de s'assurer que les résultats du modèle ne sont pas dus à un simple hasard ou à une particularité des données de test.
- **Réduction du biais de validation** : Elle permet de minimiser le biais qui pourrait être introduit par un échantillon de test non représentatif.
- **Utilisation efficace des données** : Chaque point de donnée est utilisé à la fois pour l'entraînement et pour les tests, ce qui maximise l'utilisation des données disponibles.

### 6.1.2. Méthodes de Validation Croisée

Il existe plusieurs techniques de validation croisée, chacune ayant des caractéristiques spécifiques qui peuvent être adaptées à différents types de problèmes.

#### 6.1.2.1. K-fold Cross-Validation

La validation croisée K-fold consiste à diviser l'ensemble des données en K sous-ensembles de taille égale (ou presque égale). Le modèle est ensuite entraîné sur K-1 sous-ensembles et testé sur le dernier sous-ensemble. Ce processus est répété K fois, chaque sous-ensemble étant utilisé une fois comme ensemble de test.

- **Avantages** : Fournit une estimation plus précise de la performance du modèle en utilisant toutes les données pour l'entraînement et les tests.
- **Inconvénients** : Peut être plus coûteux en termes de temps de calcul, surtout pour de grands ensembles de données.

#### 6.1.2.2. Leave-One-Out Cross Validation

La validation croisée Leave-One-Out consiste à utiliser un seul échantillon comme ensemble de test, et tous les autres comme ensemble d'entraînement. Ce processus est répété pour chaque échantillon de l'ensemble de données.

- **Avantages** : Cette méthode utilise efficacement toutes les données disponibles, avec une évaluation très précise de la performance.
- **Inconvénients** : Très coûteuse en calcul pour les grands ensembles de données, car elle nécessite  $n \times n$  itérations, où  $n$  est le nombre d'échantillons.

#### 6.1.2.3. Stratified K-fold Cross Validation

La validation croisée stratifiée K-fold est une variante de la validation croisée K-fold où les sous-ensembles sont formés de manière à conserver la proportion des classes dans chaque sous-ensemble. Cette méthode est particulièrement utile pour les problèmes de classification où les classes sont déséquilibrées.

- **Avantages** : Garantit que chaque fold contient une distribution équilibrée des classes, ce qui est crucial pour les ensembles de données déséquilibrés.
- **Inconvénients** : Peut être plus complexe à mettre en œuvre que la validation K-fold classique.

### 6.1.3. Techniques Avancées

#### 6.1.3.1. Nested Cross-Validation

La validation croisée imbriquée (nested cross-validation) est une technique avancée qui est utilisée pour évaluer des modèles tout en effectuant une recherche d'hyperparamètres. Elle consiste à utiliser une validation croisée dans le cadre de la recherche des meilleurs hyperparamètres et une validation croisée distincte pour évaluer la performance du modèle final.

**Avantages** : Permet une évaluation complète du modèle et de ses hyperparamètres en minimisant le biais.

**Inconvénients** : Nécessite davantage de ressources computationnelles.

#### 6.1.3.2. Monte Carlo Cross-Validation

La validation croisée Monte Carlo (ou validation croisée aléatoire) consiste à effectuer plusieurs partitions aléatoires des données en ensembles d'entraînement et de test. Chaque partition est utilisée pour entraîner et tester le modèle plusieurs fois, ce qui permet de réduire le biais dû à la répartition des données.

- **Avantages** : Plus flexible que la K-fold, elle permet de tester le modèle sur des échantillons très variés.
- **Inconvénients** : Les résultats peuvent être sensibles à la manière dont les données sont échantillonées.

## 6.2. Les Données d'Entraînement, de Test et de Validation

Les données sont généralement divisées en trois catégories : données d'entraînement, de test et de validation. Chacune joue un rôle spécifique dans le processus d'entraînement et d'évaluation du modèle.

### 6.2.1. Rôle de chaque type de données

#### 6.2.1.1. Données d'Entraînement

Les données d'entraînement sont utilisées pour apprendre les paramètres du modèle. Le modèle est ajusté sur ces données pour minimiser l'erreur de prédiction.

#### 6.2.1.2. Données de Test

Les données de test sont utilisées pour évaluer la performance du modèle après l'entraînement. Elles permettent de simuler des données qu'un modèle pourrait rencontrer dans un contexte réel.

#### 6.2.1.3. Données de Validation

Les données de validation sont utilisées pour ajuster les hyperparamètres du modèle. Elles aident à déterminer les meilleurs paramètres avant de tester le modèle sur les données de test.

### 6.2.2. Stratégies de partitionnement

Le partitionnement des données en ensembles d'entraînement, de validation et de test est une étape cruciale pour l'évaluation du modèle.

#### 6.2.2.1. Partitionnement Simple

Le partitionnement simple consiste à diviser les données en deux ensembles : un pour l'entraînement et un autre pour les tests. Bien qu'efficace, cette méthode peut être biaisée si l'échantillon de test n'est pas représentatif.

#### 6.2.2.2. Partitionnement Stratifié

Le partitionnement stratifié assure que chaque partition maintient la même proportion de classes que l'ensemble original des données. Cette méthode est particulièrement utile pour les problèmes de classification déséquilibrée.

### 6.2.3. Impact sur la performance des modèles

Un bon partitionnement des données a un impact significatif sur la performance du modèle.

#### 6.2.3.1. Ensemble d'Entraînement de Taille Adéquate

Une taille d'ensemble d'entraînement appropriée est nécessaire pour assurer que le modèle peut apprendre des patterns significatifs sans surajustement.

#### 6.2.3.2. Données de Test Représentatives

Les données de test doivent être représentatives du domaine d'application réel afin que les évaluations du modèle soient fiables.

#### 6.2.3.3. Validation Croisée

La validation croisée permet d'obtenir une estimation robuste de la performance du modèle sur des données inédites et aide à déterminer sa capacité à généraliser.

### 6.3. Résumé

La validation et l'évaluation sont des étapes cruciales dans l'apprentissage automatique pour garantir la robustesse et la généralisation d'un modèle.

- **Validation croisée :** Technique permettant d'évaluer un modèle en utilisant différents sous-ensembles de données. On distingue plusieurs méthodes :
  - *K-fold cross-validation* : divise les données en  $K$  sous-ensembles pour des évaluations successives.
  - *Leave-One-Out (LOO-CV)* : utilise chaque échantillon comme test unique, mais est coûteux en calcul.
  - *Stratified K-fold* : préserve la répartition des classes, utile pour les données déséquilibrées.
  - *Nested Cross-Validation* : combine validation et optimisation des hyperparamètres.
  - *Monte Carlo Cross-Validation* : effectue des divisions aléatoires répétées pour réduire le biais.
- **Types de données :**
  - *Données d'entraînement* : utilisées pour apprendre les paramètres du modèle.
  - *Données de validation* : aident à ajuster les hyperparamètres.
  - *Données de test* : servent à évaluer la performance finale du modèle.
- **Partitionnement des données :**
  - *Simple* : séparation en ensembles d'entraînement et de test.
  - *Stratifié* : conserve la distribution des classes.

Une bonne gestion de ces techniques permet d'éviter le surajustement et d'assurer une évaluation fiable du modèle.

# 7. Optimisation et Fonctionnement des Modèles

L'optimisation et le bon fonctionnement des modèles d'apprentissage automatique sont essentiels pour garantir des performances élevées et une bonne généralisation. Cette section explore des concepts clés tels que la fiabilité, le biais et le bruit, l'importance des fonctions de coût, ainsi que les techniques d'optimisation comme la descente de gradient.

## 7.1. Fiabilité, Biais et Bruit

### 7.1.1. Fiabilité

#### 7.1.1.1. Définition et importance de la fiabilité

La fiabilité d'un modèle d'apprentissage automatique désigne sa capacité à fournir des prédictions cohérentes et précises sur de nouvelles données. Un modèle fiable doit être robuste aux variations des données et ne pas être excessivement influencé par des anomalies ou des changements mineurs.

#### 7.1.1.2. Méthodes d'Évaluation

Pour mesurer la fiabilité d'un modèle, plusieurs méthodes sont utilisées :

- **Validation croisée** : permet de tester le modèle sur plusieurs sous-ensembles de données.
- **Analyse des erreurs** : observation des erreurs sur différents ensembles de test pour identifier les éventuelles faiblesses du modèle.
- **Robustesse aux perturbations** : ajout de bruit ou modification des données d'entrée pour évaluer la stabilité du modèle.

#### 7.1.1.3. Cas d'utilisation

La fiabilité est essentielle dans des domaines critiques comme :

- **Médical** : diagnostic basé sur des images ou des données biologiques.
- **Finance** : détection de fraudes ou prévisions boursières.
- **Automobile** : conduite autonome nécessitant des décisions sûres en temps réel.

## 7.1.2. Biais

### 7.1.2.1. Définition et Sources de Biais

Le biais représente une erreur systématique dans un modèle qui entraîne des prédictions incorrectes. Les principales sources de biais incluent :

- **Biais dans les données** : échantillons non représentatifs ou déséquilibrés.
- **Biais algorithmique** : simplification excessive d'un modèle qui ne capture pas la complexité des données.
- **Biais humain** : introduction d'erreurs lors de l'annotation des données.

### 7.1.2.2. Impact des Biais

Un modèle biaisé peut avoir des conséquences négatives, telles que :

- **Inexactitude des prédictions** : erreurs fréquentes sur certains types de données.
- **Discrimination** : décisions inéquitables, notamment dans les systèmes de recrutement ou de prêt bancaire.
- **Faible généralisation** : sur-apprentissage sur certaines caractéristiques des données.

### 7.1.2.3. Techniques de Correction

Pour réduire le biais, plusieurs approches sont possibles :

- **Rééquilibrage des données** : collecte de données plus diversifiées ou augmentation artificielle des minorités.
- **Régularisation** : pénalisation des modèles trop simplistes.
- **Audits et tests éthiques** : analyse des résultats pour détecter des inégalités dans les prédictions.

### 7.1.3. Bruit

#### 7.1.3.1. Définition et Types de Bruit

Le bruit est constitué d'informations inutiles ou erronées qui peuvent fausser l'apprentissage d'un modèle. Il peut être :

- **Bruit aléatoire** : erreurs accidentnelles dans les données (ex. fautes de frappe).
- **Bruit systématique** : erreurs répétées dues à un problème dans la collecte des données.
- **Bruit d'étiquetage** : erreurs dans les annotations des classes.

#### 7.1.3.2. Impact du Bruit

Le bruit peut entraîner :

- **Sur-ajustement** : le modèle apprend des détails non pertinents.
- **Baisse de précision** : erreurs accrues sur des données réelles.
- **Mauvaise convergence** : difficulté à optimiser la fonction de coût.

#### 7.1.3.3. Techniques de Réduction du Bruit

- **Nettoyage des données** : suppression des valeurs aberrantes.
- **Utilisation de modèles robustes** : régularisation et algorithmes résistants aux anomalies.
- **Augmentation des données** : enrichissement du dataset avec des données fiables.

## 7.2. Une Fonction de Coût

### 7.2.1. Définition et importance dans le Machine Learning

#### 7.2.1.1. Définition détaillée de la fonction de coût

La fonction de coût mesure l'écart entre les prédictions du modèle et les valeurs réelles. Elle guide l'optimisation en ajustant les paramètres du modèle pour minimiser cette erreur.

#### 7.2.1.2. Importance de la fonction de coût dans l'entraînement des modèles

Une fonction de coût bien choisie permet :

- D'orienter l'apprentissage du modèle.
- D'éviter le sur-apprentissage ou l'apprentissage trop simpliste.
- D'améliorer la vitesse de convergence.

### 7.2.2. Types courants de fonctions de coût

La fonction de coût joue un rôle crucial dans l'entraînement des modèles de machine learning, car elle guide l'optimisation en mesurant l'écart entre les prédictions du modèle et les valeurs réelles. Voici les principales fonctions de coût utilisées en fonction du type de problème à résoudre.

#### 7.2.2.1. Fonction de coût quadratique (ou de l'erreur quadratique moyenne / MSE)

L'**erreur quadratique moyenne** (*Mean Squared Error*, MSE) est largement utilisée pour les problèmes de régression. Elle est définie comme suit :

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

où :

- $n$  est le nombre total d'échantillons,
- $y_i$  est la valeur réelle de l'échantillon  $i$ ,
- $\hat{y}_i$  est la valeur prédite par le modèle.

Le MSE pénalise fortement les grandes erreurs (en raison de l'élévation au carré), ce qui peut poser problème si des outliers sont présents dans les données. Cependant, il est simple à optimiser et très efficace pour les modèles de régression linéaire.

#### 7.2.2.2. Fonction de coût logarithmique (log-loss)

La **log-loss** (*logarithmic loss* ou *cross-entropy loss*) est utilisée pour les problèmes de classification binaire et multi-classes. Elle est définie par :

$$\text{Log-loss} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

où :

- $y_i$  est la classe réelle (0 ou 1),
- $\hat{y}_i$  est la probabilité prédictive pour la classe 1.

Cette fonction pénalise davantage les erreurs lorsque la confiance du modèle est élevée mais incorrecte. Elle est particulièrement adaptée aux modèles de classification probabilistes tels que la régression logistique et les réseaux de neurones.

#### 7.2.2.3. Fonction de coût Hinge

La **Hinge Loss** est utilisée pour les modèles de classification, notamment les **Machines à Vecteurs de Support (SVMs)**. Elle est définie comme suit pour un problème de classification binaire :

$$\text{Hinge Loss} = \sum_{i=1}^n \max(0, 1 - y_i \hat{y}_i)$$

où :

- $y_i$  est la vraie classe (+1 ou -1),
- $\hat{y}_i$  est la sortie du modèle (score non probabiliste).

La Hinge Loss encourage une marge maximale entre les classes et est donc idéale pour les SVMs. Cependant, elle ne fonctionne pas bien avec des modèles qui produisent des probabilités, comme la régression logistique.

### 7.2.3. Rôle dans les modèles de Machine Learning

#### 7.2.3.1. Comment les fonctions de coût influencent l'entraînement du modèle

La fonction de coût sert de guide pour l'algorithme d'optimisation. Un bon choix de fonction de coût permet :

- D'adapter l'apprentissage du modèle au type de problème (régression ou classification).
- De garantir une convergence stable vers un modèle performant.
- D'éviter les erreurs extrêmes en pénalisant différemment les écarts.

### 7.2.3.2. Impact sur les décisions de mise à jour des poids

Lors de l'entraînement, l'optimisation repose sur la **descente de gradient**, qui ajuste les poids du modèle en minimisant la fonction de coût. Une fonction de coût bien définie impacte directement :

- La vitesse de convergence du modèle.
- La stabilité de l'apprentissage.
- L'efficacité de l'ajustement aux données d'entraînement.

## 7.2.4. Utilisation pour évaluer la performance du modèle

### 7.2.4.1. Mesurer la qualité de la prédiction

La fonction de coût permet de quantifier l'écart entre les prédictions du modèle et la réalité. Un score faible signifie généralement que le modèle généralise bien aux données nouvelles, tandis qu'un score élevé indique un besoin d'optimisation.

### 7.2.4.2. Comparaison entre différentes fonctions de coût pour un même modèle

Différentes fonctions de coût peuvent être testées pour un même modèle afin d'identifier celle qui minimise au mieux l'erreur et améliore la précision globale. Par exemple :

- **MSE vs MAE** en régression : le MAE est plus robuste aux outliers que le MSE.
- **Log-loss vs Hinge Loss** en classification : la log-loss est plus adaptée aux modèles probabilistes, tandis que la Hinge Loss est utilisée pour les SVMs.

### 7.2.5. Relation avec la précision et le sur-ajustement

#### 7.2.5.1. Comment la fonction de coût aide à trouver le bon compromis entre précision et sur-ajustement

Une fonction de coût bien choisie doit équilibrer **précision** et **généralisation**.

- Un modèle qui minimise trop fortement la fonction de coût sur l'ensemble d'entraînement peut **sur-ajuster** (overfitting).
- À l'inverse, un modèle qui généralise trop mal peut **sous-ajuster** (underfitting).

L'ajout de techniques de **régularisation** (L1, L2) peut aider à atténuer ces problèmes.

## 7.3. Fonctions de coût spécifiques

### 7.3.1. Fonctions de coût pour les problèmes de régression

#### 7.3.1.1. Erreur quadratique moyenne (MSE)

Déjà détaillée précédemment, elle est couramment utilisée en régression linéaire.

#### 7.3.1.2. Erreur absolue moyenne (MAE)

L'**erreur absolue moyenne** (*Mean Absolute Error*, MAE) est définie comme suit :

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Elle est plus robuste aux outliers que le MSE, mais moins stable pour la descente de gradient.

### 7.3.2. Fonctions de coût pour les problèmes de classification

#### 7.3.2.1. Entropie croisée

Déjà expliquée dans la section **log-loss**, elle est la référence pour la classification probabiliste.

#### 7.3.2.2. Hinge Loss

Déjà expliquée dans la section dédiée aux SVMs.

## 7.4. La Descente de Gradient

### 7.4.1. Principe et algorithme de base

#### 7.4.1.1. Introduction à la descente de gradient

La descente de gradient est un algorithme d'optimisation utilisé pour minimiser la fonction de coût. Il repose sur le calcul du **gradient**, qui indique la direction à suivre pour réduire l'erreur.

#### 7.4.1.2. Étapes de l'algorithme classique (calcul de gradient, mise à jour des poids ...)

- Calcul du gradient de la fonction de coût par rapport aux paramètres du modèle.
- Mise à jour des poids dans la direction opposée du gradient.
- Itération jusqu'à convergence.

### 7.4.2. Variantes et optimisations avancées

#### 7.4.2.1. Descente de gradient stochastique et mini-batch

- **Batch Gradient Descent** : mise à jour des poids après avoir calculé le gradient sur tout l'ensemble d'entraînement.
- **Stochastic Gradient Descent (SGD)** : mise à jour après chaque échantillon, réduisant le temps de calcul mais augmentant la variance.
- **Mini-batch Gradient Descent** : compromis entre les deux, où les mises à jour se font sur des petits sous-ensembles de données.

#### 7.4.2.2. Technique d'optimisation (momentum, RMSprop, Adam ...)

Des optimisations comme **Momentum**, **RMSprop** et **Adam** permettent d'accélérer la convergence et d'éviter de rester bloqué dans des minima locaux.

## 7.5. Résumé

L'optimisation des modèles de machine learning repose sur des concepts clés comme la **fonction de coût** et la **descente de gradient**.

### Fiabilité, Biais et Bruit

- **Fiabilité** : Mesure la robustesse d'un modèle ; évaluée via des métriques adaptées.
- **Biais** : Résulte d'hypothèses trop simplificatrices ; impacte la généralisation.
- **Bruit** : Variabilité aléatoire dans les données ; peut être réduit par des techniques comme le lissage ou la régularisation.

### Fonctions de coût

- Définissent l'écart entre les prédictions et la réalité.
- **Régression** : MSE (Erreur Quadratique Moyenne) et MAE (Erreur Absolue Moyenne).
- **Classification** : Log-loss (Entropie Croisée) pour les modèles probabilistes, Hinge Loss pour les SVMs.

### Descente de Gradient

- Algorithme d'optimisation qui ajuste les poids du modèle pour minimiser la fonction de coût.
- Variantes : **SGD (Stochastique)** pour des mises à jour rapides, **Mini-batch** pour un compromis entre précision et efficacité.
- Optimisations avancées : **Momentum**, **RMSprop**, **Adam** pour une convergence plus rapide et stable.

### Équilibre entre précision et sur-ajustement

- Une bonne fonction de coût aide à éviter le sur-ajustement en trouvant un compromis entre **précision sur les données d'entraînement** et **capacité de généralisation**.
- Des techniques comme la **régularisation** ou l'**early stopping** peuvent être utilisées pour éviter un modèle trop complexe.

**En résumé**, l'optimisation d'un modèle repose sur le choix judicieux de la fonction de coût et la stratégie de mise à jour des poids. Une bonne compréhension de ces éléments permet de maximiser la performance et la fiabilité d'un modèle de machine learning.

## 8. Conclusion

9.

### 9.1. Synthèse des points principaux

Tout au long de ce document, nous avons exploré les fondements et les techniques du **Machine Learning**, depuis son contexte historique jusqu'aux méthodes modernes d'optimisation des modèles.

- Nous avons vu que le **Machine Learning** repose sur des principes issus des **statistiques, des probabilités et de l'informatique**, avec des applications variées dans des domaines aussi divers que la **santé, la finance, le marketing et la cybersécurité**.
- Les **méthodes d'apprentissage** se divisent principalement en **supervisé** et **non supervisé**, chacune étant adaptée à des problématiques spécifiques.
- Les **techniques de classification** et de **régression** sont essentielles pour modéliser et prédire des phénomènes à partir de données.
- L'**évaluation et la validation** des modèles sont cruciales pour garantir leur robustesse et leur capacité de généralisation.
- Enfin, nous avons approfondi les stratégies d'**optimisation**, notamment à travers les **fonctions de coût** et la **descente de gradient**, qui jouent un rôle clé dans l'amélioration des performances des modèles.

### 9.2. Défis actuels et limites du Machine Learning

Malgré ses avancées spectaculaires, le Machine Learning rencontre encore plusieurs défis :

- **Qualité et biais des données**
  - La performance des modèles dépend fortement des données utilisées. Les biais présents dans les jeux de données peuvent conduire à des décisions discriminatoires ou erronées.
- **Explicabilité des modèles**
  - Les modèles complexes, notamment ceux issus du **Deep Learning**, sont souvent considérés comme des "boîtes noires". Il est donc difficile d'interpréter leurs décisions, ce qui pose des problèmes en matière d'**éthique** et de **confiance**.

- **Coût computationnel et consommation énergétique**
  - L'entraînement des modèles de grande envergure requiert des ressources considérables, ce qui soulève des questions sur leur **impact environnemental** et leur **accessibilité** pour des entreprises aux ressources limitées.
- **Sécurité et robustesse**
  - Les modèles de Machine Learning sont vulnérables aux **attaques adversariales**, où de petites modifications des données d'entrée peuvent tromper le modèle et entraîner des résultats erronés.
- **Généralisation et transfert de connaissances**
  - Un modèle performant sur un jeu de données spécifique peut échouer dans un contexte légèrement différent. Le **transfert d'apprentissage** et **l'adaptation aux nouveaux environnements** restent des défis majeurs.

### 9.3. Perspectives et évolutions futures

L'avenir du Machine Learning est prometteur, avec plusieurs évolutions attendues :

- **Automatisation accrue (AutoML)** : Des outils facilitant l'optimisation automatique des modèles sans intervention humaine.
- **Machine Learning éthique** : Développement de techniques garantissant des décisions plus transparentes et équitables.
- **Apprentissage fédéré** : Permettant d'entraîner des modèles sur des **données décentralisées**, sans compromettre la confidentialité.
- **Modèles plus efficents** : Optimisation des algorithmes pour réduire leur empreinte carbone et rendre l'**IA plus durable**.
- **Intégration avec d'autres disciplines** : Le **quantum computing**, la **biologie computationnelle** et l'**IA hybride** ouvrent de nouvelles voies pour repousser les limites actuelles.

**En conclusion**, le Machine Learning est en constante évolution et continue de transformer de nombreux domaines. Les défis à relever sont nombreux, mais les avancées technologiques et méthodologiques offrent des perspectives fascinantes pour l'avenir.

# 10. Quelques Sources

[Une petite histoire du Machine Learning](#)

[7 Chapitres Cruciaux Sur L'apprentissage Automatique](#)

[Histoire et évolution du Machine Learning](#)

[Internet des Objets](#)

[Learn statistics easily](#)

[Collecte de données : Qu'est-ce que c'est, méthodes et outils](#)

[Méthodes de collecte de données](#)

[Qu'est ce que la collecte de données scientifiques](#)

[Data science: Tout savoir sur la collecte et la gestion des données](#)

[Qu'est-ce que la collecte de données : un guide complet](#)

[Prétraitement des données : concepts, importance et outils | Astera](#)

[Prétraitement des données : Un guide complet avec des exemples en Python | DataCamp](#)

[Le prétraitement des données en data science](#)

[Decision Trees — scikit-learn](#)

[RandomForestClassifier — scikit-learn](#)

[Support Vector Machines — scikit-learn](#)

[Réseaux de neurones avec TensorFlow.](#)

[Metrics and scoring: quantifying the quality of predictions — scikit-learn](#)

[KMeans — scikit-learn](#)

[Clustering — scikit-learn](#)

[DBSCAN — scikit-learn](#)

[LinearRegression — scikit-learn](#)

[PolynomialFeatures — scikit-learn](#)

[LogisticRegression — scikit-learn](#)

[Ridge — scikit-learn](#)

[Régression Ridge et Lasso: une illustration et une explication à l'aide de Sklearn en Python](#)