Université Sultan Moulay Slimane
École Nationale des Sciences Appliquées
- Khouribga -

Filière : Génie Informatique et Ingénierie des Données

Rapport de Projet ML

Réalisé par :
BOULIDAM ABDELLAH
BOUMLIK YOUNESS
HAFSI GHIZLANE

Encadré par :
Mr. GHAZDALI

Année universitaire :
2024/2025

Remerciements

Nous tenons à remercier notre professeur, **Monsieur Abdelghani Ghazdali**, pour ses explications claires et pédagogiques lors des cours de Machine Learning.

Nous lui exprimons également notre gratitude pour l'opportunité de travailler sur ce projet, qui nous a permis d'approfondir nos connaissances et de mettre en pratique des concepts essentiels, tels que les arbres de décision.

Ce projet a été une expérience enrichissante, contribuant à notre compréhension des techniques fondamentales en Machine Learning.

Table des matières

Liste des tableaux

Table des figures

1 Introduction

1.1 Definition du ML

Le Machine Learning, ou apprentissage automatique, est une sous-discipline de l'intelligence artificielle (IA) qui consiste à développer des algorithmes et des modèles capables d'apprendre à partir de données, d'identifier des schémas, et de prendre des décisions ou des prédictions sans être explicitement programmés pour effectuer ces tâches spécifiques. Contrairement aux systèmes classiques basés sur des règles préprogrammées, le Machine Learning s'appuie sur des techniques statistiques et des méthodes mathématiques pour analyser les données, détecter des relations cachées et généraliser à partir d'exemples.

1.2 Principes fondamentaux

1. Entraînement sur des données : Les algorithmes de ML utilisent un ensemble de données d'entraînement pour apprendre les caractéristiques ou les relations entre les variables (entrée et sortie).

2. Généralisation : Une fois entraîné, le modèle peut être utilisé pour effectuer des prédictions ou des classifications sur des données nouvelles et inconnues.

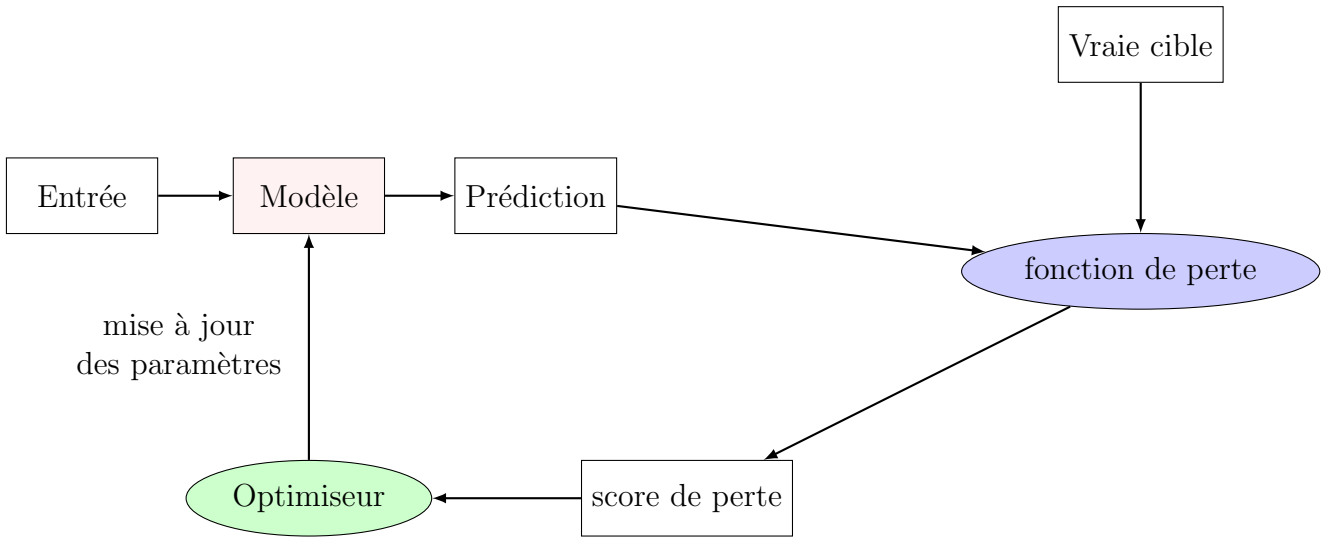
3. Apprentissage itératif : Les modèles ajustent leurs paramètres internes à travers des itérations successives pour minimiser les erreurs et améliorer les performances.

1.3 Types d'apprentissage automatique

1. **Apprentissage supervisé** :
   — Le modèle est formé sur des données étiquetées (chaque donnée d'entrée est associée à une sortie connue).
   — Exemple : Prédire le prix d'une maison à partir de ses caractéristiques (taille, localisation, etc.).
   — Algorithmes courants : régression linéaire, arbres de décision, SVM, réseaux neuronaux.
2. **Apprentissage non supervisé** :
   — Les données n'ont pas de sortie étiquetée, et le modèle doit découvrir seul des schémas ou des structures dans les données.
   — Exemple : Regrouper des clients en segments selon leur comportement d'achat.
   — Algorithmes courants : clustering (K-Means, DBSCAN), analyse en composantes principales (PCA).
3. **Apprentissage semi-supervisé** :
   — Mélange de données étiquetées et non étiquetées pour entraîner le modèle.
   — Utilisé lorsque l'étiquetage des données est coûteux ou difficile.
4. **Apprentissage par renforcement** :
   — Le modèle apprend par essais et erreurs en interagissant avec un environnement et en recevant des récompenses ou pénalités en fonction de ses actions.
   — Exemple : Formation de robots ou intelligence artificielle pour jouer à des jeux.
   — Algorithmes courants : Q-learning, Deep Q-Networks (DQN).

1.4 Composantes clés

1. **Données** :
   — Le Machine Learning repose sur des données de qualité. Plus les données sont variées, représentatives et nombreuses, plus le modèle est précis.
2. **Caractéristiques (features)** :
   — Les variables ou attributs issus des données sur lesquels le modèle s'appuie pour effectuer des prédictions.
3. **Modèle** :
   — La structure mathématique ou algorithmique qui effectue l'apprentissage.
4. **Fonction de coût** :
   — Mesure l'erreur entre les prédictions du modèle et les résultats attendus, utilisée pour optimiser les performances.
5. **Algorithme d'optimisation** :
   — Méthode pour ajuster les paramètres du modèle afin de minimiser la fonction de coût (exemple : descente de gradient).



2 Méthode de Machine Learning : Arbre de Décision

Un **arbre de décision** est un **modèle prédictif** qui organise les **décisions** sous forme d'une structure arborescente. Chaque **nœud interne** représente un **test** sur une **caractéristique** (ou **attribut**) d'un jeu de données, chaque **branche** correspond au **résultat** de ce test, et chaque **feuille** (ou **nœud terminal**) indique une **décision** ou un **résultat**.

Dans un contexte de **classification**, par exemple, ces feuilles correspondent à des **classes prédictives**. Ce type de modèle est particulièrement apprécié pour sa **simplicité d'interprétation** et sa capacité à **gérer des données catégoriques ou numériques**.

2.1 Objectif d'un arbre de décision

L'objectif principal d'un arbre de décision est de fournir un modèle de prédiction clair et interprétable pour résoudre des problèmes de classification ou de régression. Il apprend à partir d'un ensemble de données d'entraînement et généralise pour faire des prédictions sur de nouvelles données non étiquetées.

- **Prédiction** : Un arbre de décision est utilisé pour prédire la classe (classification) ou une valeur continue (régression).
  Exemple : Prédire si un client remboursera son crédit en fonction de ses antécédents financiers.
- **Séparation optimale des données** : Il sélectionne les attributs les plus pertinents à chaque étape pour maximiser l'information ou minimiser l'impureté.
  Exemple : Séparer les patients en groupes à faible ou haut risque de maladie selon leurs symptômes.
- **Interprétabilité** : Un arbre de décision est facile à comprendre et à visualiser.
  Exemple : Un schéma expliquant pourquoi une demande de prêt est acceptée ou refusée.
- **Généralisation des données** : Il doit bien fonctionner sur des données non vues, en évitant le sur-apprentissage.
  Exemple : Déterminer si un email est du spam en se basant sur des critères appris sur d'autres emails.
- **Simplicité et efficacité** : Il est rapide à entraîner et à exécuter, même sur de grands ensembles de données.
  Exemple : Un chatbot simple classant les requêtes des utilisateurs en différentes catégories.
- **Adaptabilité** : Il traite aussi bien les données numériques que catégorielles et peut être combiné avec d'autres modèles (ex. forêt aléatoire).
  Exemple : Un modèle prédisant la satisfaction client à partir d'avis textuels et de notes numériques.
- **Gestion des données manquantes et valeurs aberrantes** : Il peut gérer des données incomplètes et être robuste aux valeurs extrêmes.
  Exemple : Une analyse médicale où certaines informations patient peuvent être manquantes.

2.2 Structure d'un arbre de décision

1. **Racine** : Le nœud supérieur de l'arbre, représentant l'attribut ou la caractéristique qui offre la meilleure séparation des données. C'est le premier critère utilisé pour

diviser les données initiales.

2. **Nœuds internes** : Chaque nœud interne correspond à un test sur un attribut spécifique des données. Ce test permet de diviser les données en sous-ensembles distincts selon les caractéristiques définies.
3. **Branches** : Les branches relient les nœuds entre eux, représentant les résultats du test effectué à chaque nœud. Chaque branche correspond à un attribut particulier ou à une valeur spécifique issue du test effectué.
4. **Feuilles** : Les feuilles de l'arbre correspondent à des décisions finales ou des prédictions. Dans le cadre de la classification, une feuille contient une classe prédite pour les instances arrivant à ce nœud terminal, tandis qu'en régression, elle contient une valeur prédite.

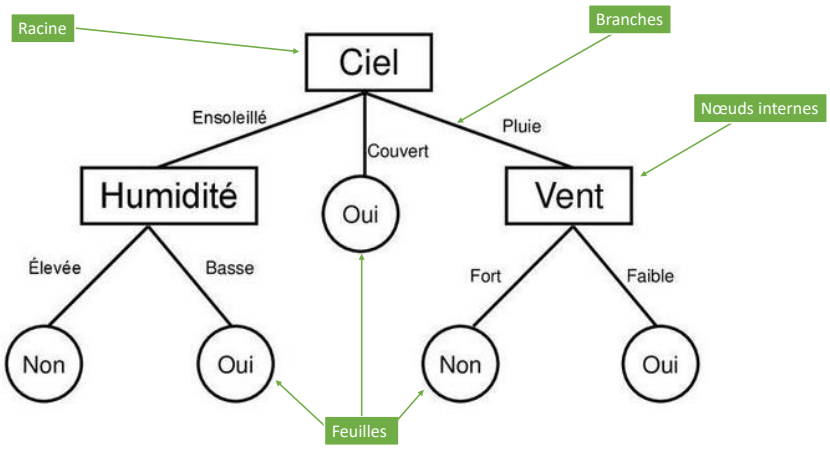


FIGURE 1 – Exemple d'un arbre de décision illustrant les concepts clés : la **racine** (point de départ), les **nœuds internes** (tests sur les attributs), les **branches** (liens entre les décisions) et les **feuilles** (résultats finaux).

2.3 Construction d'un arbre de décision

Un arbre de décision est construit de manière itérative en divisant les données en sous-groupes homogènes à l'aide d'un critère de sélection d'attributs. Cette construction repose sur un processus d'apprentissage supervisé qui suit plusieurs étapes fondamentales.

2.3.1 Principe général d'un algorithme d'arbre de décision

La construction d'un arbre de décision suit un processus récursif qui permet de segmenter les données en fonction des caractéristiques les plus pertinentes. Le principe général repose sur les étapes suivantes :

— Sélectionner le meilleur attribut pour diviser les données en sous-groupes.
— Créer un nœud pour cet attribut et établir des branches correspondant aux différentes valeurs possibles.
— Répéter récursivement le processus pour chaque sous-groupe jusqu'à atteindre un critère d'arrêt.

La condition d'arrêt peut être déterminée par plusieurs facteurs, notamment :
— La profondeur maximale de l'arbre.
— L'absence d'amélioration significative dans la séparation des données.
— L'atteinte d'un niveau de pureté suffisant dans les nœuds terminaux.

2.3.2 Problème du choix du critère de division

L'une des étapes clés de la construction d'un arbre de décision est le choix du critère qui permet de mesurer la qualité d'une séparation des données. Différents critères existent pour optimiser ce choix :
— **L'entropie et le gain d'information** : utilisés par l'algorithme ID3, ils permettent de quantifier la réduction d'incertitude après une division.
— **L'indice de Gini** : employé par l'algorithme CART, il mesure la pureté d'un sous-ensemble de données.
— **La variance** : utilisée pour les problèmes de régression, elle évalue la dispersion des valeurs prédites.

Le choix du critère influence directement la structure et la performance de l'arbre de décision. **Nous allons détailler ces critères dans la prochaine section.**

2.3.3 Les principaux algorithmes d'apprentissage supervisé pour les arbres de décision

Plusieurs algorithmes permettent de construire un arbre de décision en utilisant des critères de sélection spécifiques :
— **ID3 (Iterative Dichotomiser 3)** : utilise l'entropie et le gain d'information pour choisir l'attribut optimal à chaque nœud.
— **C4.5** : amélioration d'ID3, il prend en charge les valeurs continues et gère mieux les données manquantes.
— **CART (Classification and Regression Trees)** : utilise l'impureté de Gini pour les problèmes de classification et la variance pour les problèmes de régression.

Ces algorithmes reposent sur les critères de sélection évoqués précédemment et permettent d'obtenir des modèles adaptés aux différentes problématiques d'apprentissage supervisé.

3 Modèle mathématique des arbres de décision

Dans cette section, nous allons explorer les bases mathématiques qui sous-tendent les arbres de décision. Ces concepts jouent un rôle fondamental dans la construction et l'optimisation des arbres, permettant de segmenter efficacement les données en sous-ensembles homogènes. Nous allons tout d'abord introduire l'entropie, une mesure clé de la théorie de l'information, et voir comment elle peut être utilisée pour guider le choix des divisions au sein des données. À travers des exemples concrets, nous illustrerons également ces concepts pour une meilleure compréhension.

Les données utilisées dans notre rapport seront illustrées par un tableau comme celui-ci, où chaque ligne représente un exemple avec des caractéristiques (ou attributs) spécifiques, ainsi qu'une classe cible associée :

Ces données serviront d'illustration tout au long du rapport pour montrer comment les concepts mathématiques sont appliqués en pratique.

Forme des oreilles (x_1)	**Forme du visage (x_2)**	**Moustaches (x_3)**	**Classe (y)**
Pointues	Rond	Présentes	1
Ovales	Non rond	Présentes	1
Ovales	Rond	Absentes	0
Pointues	Non rond	Présentes	0
Ovales	Rond	Présentes	1
Flasques	Non rond	Absentes	0
Ovales	Rond	Absentes	1
Flasques	Rond	Absentes	0
Flasques	Rond	Absentes	0

TABLE 1 – Exemple de données utilisées dans le rapport

3.1 Entropie (H)

L'entropie est une mesure de l'incertitude ou de l'impureté d'un ensemble de données. Dans le contexte des arbres de décision, elle est utilisée pour évaluer à quel point un ensemble de données est homogène par rapport à la classe cible. Une entropie élevée indique une grande diversité des classes, tandis qu'une entropie nulle indique que l'ensemble des données est parfaitement homogène.

L'entropie d'un ensemble S est définie par la formule suivante :

$$H(S) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

où :
— n est le nombre total de classes dans l'ensemble S,
— p_i est la proportion d'exemples appartenant à la classe i dans S.

Exemple : Calcul de l'entropie Considérons un ensemble S de 9 exemples basé sur les données du Tableau 1. Les classes possibles (y) sont 1 (chat) et 0 (non-chat). Le nombre d'exemples dans chaque classe est donné par :
— 4 exemples dans la classe 1,
— 5 exemples dans la classe 0.

Les probabilités associées à chaque classe sont donc :

$$p_1 = \frac{4}{9}, \quad p_0 = \frac{5}{9}.$$

L'entropie de l'ensemble S est alors calculée comme suit :

$$H(S) = -\left(\frac{4}{9}\log_2\frac{4}{9} + \frac{5}{9}\log_2\frac{5}{9}\right).$$

En utilisant les logarithmes :

$$H(S) \approx -\left(0.444 \cdot (-1.169) + 0.556 \cdot (-0.847)\right).$$

$$H(S) \approx 0.991 bits.$$

Ainsi, l'entropie de cet ensemble est de 0.991, ce qui indique une distribution relativement équilibrée des classes.

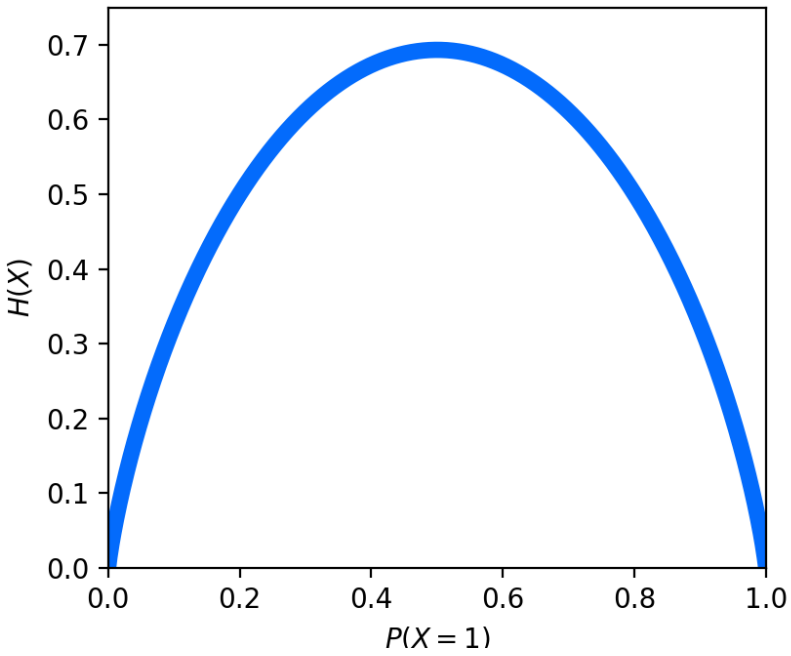


FIGURE 2 – Graphe de l'entropie en fonction de la probabilité

3.2 Gain d'information

Le gain d'information mesure la réduction de l'incertitude ou de l'entropie après avoir effectué une division des données selon un attribut donné. Lors de la construction d'un arbre de décision, le gain d'information est utilisé pour déterminer quel attribut (ou critère) doit être choisi comme racine ou à chaque nœud de l'arbre.

Le gain d'information associé à un attribut A pour un ensemble S est calculé comme suit :

$$Gain(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v),$$

où :
— $H(S)$ est l'entropie de l'ensemble S,
— $Values(A)$ est l'ensemble des valeurs possibles de l'attribut A,
— S_v est le sous-ensemble des données où l'attribut A prend la valeur v,
— $|S|$ et $|S_v|$ sont respectivement les tailles de S et S_v,
— $H(S_v)$ est l'entropie de S_v, l'ensemble des données filtré par la valeur v.

Le critère convenable est celui qui maximise le gain d'information, car il permet de réduire l'incertitude de manière optimale.

Exemple : Détermination de la racine convenable Utilisons les données du tableau donné pour déterminer quel attribut doit être choisi comme racine de l'arbre de décision.

Entropie initiale de l'ensemble S : Nous avons 5 exemples de classe 0 et 4 exemples de classe 1. L'entropie initiale $H(S)$ est donc donnée par :

$$H(S) = -\left(\frac{5}{9}\log_2\left(\frac{5}{9}\right) + \frac{4}{9}\log_2\left(\frac{4}{9}\right)\right) \approx 0.991\, bits.$$

Cas 1 : Division selon l'attribut x_1 (Forme des oreilles)

Les valeurs possibles pour x_1 sont : {Pointues, Ovales, Flasques}. Calculons l'entropie pour chaque sous-ensemble S_v :

— Pour $x_1 = Pointues$ (2 exemples : 1 dans la classe 0, 1 dans la classe 1) :

$$H(S_{Pointues}) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1.$$

— Pour $x_1 = Ovales$ (4 exemples : 1 dans la classe 0, 3 dans la classe 1) :

$$H(S_{Ovales}) = -\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right) \approx 0.811.$$

— Pour $x_1 = Flasques$ (3 exemples : 3 dans la classe 0, 0 dans la classe 1) :

$$H(S_{Flasques}) = 0(ensemblehomogène).$$

Le gain d'information pour x_1 est :

$$Gain(S, x_1) = 0.991 - \left(\frac{2}{9}\cdot 1 + \frac{4}{9}\cdot 0.811 + \frac{3}{9}\cdot 0\right) \approx 0.408.$$

Cas 2 : Division selon l'attribut x_2 (Forme du visage)

Les valeurs possibles pour x_2 sont : {Rond, Non rond}. Calculons l'entropie pour chaque sous-ensemble S_v :

— Pour $x_2 = Rond$ (6 exemples : 3 dans la classe 0, 3 dans la classe 1) :

$$H(S_{Rond}) = -\left(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right) = 1.$$

— Pour $x_2 = Nonrond$ (3 exemples : 2 dans la classe 0, 1 dans la classe 1) :

$$H(S_{Nonrond}) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) \approx 0.918.$$

Le gain d'information pour x_2 est :

$$Gain(S, x_2) = 0.991 - \left(\frac{6}{9}\cdot 1 + \frac{3}{9}\cdot 0.918\right) \approx 0.073.$$

Cas 3 : Division selon l'attribut x_3 (Présence des moustaches)

Les valeurs possibles pour x_3 sont : {Présentes, Absentes}. Calculons l'entropie pour chaque sous-ensemble S_v :

— Pour $x_3 = Présentes$ (4 exemples : 1 dans la classe 0, 3 dans la classe 1) :

$$H(S_{Présentes}) = -\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right) \approx 0.811.$$

— Pour $x_3 = Absentes$ (5 exemples : 4 dans la classe 0, 1 dans la classe 1) :

$$H(S_{Absentes}) = -\left(\frac{4}{5}\log_2\frac{4}{5} + \frac{1}{5}\log_2\frac{1}{5}\right) \approx 0.721.$$

Le gain d'information pour x_3 est alors :

$$Gain(S, x_3) = 0.991 - \left(\frac{4}{9} \cdot 0.811 + \frac{5}{9} \cdot 0.721\right) \approx 0.230.$$

Conclusion :
Les gains d'information pour chaque attribut sont les suivants :

$$Gain(S, x_1) \approx 0.408, \quad Gain(S, x_2) \approx 0.073, \quad Gain(S, x_3) \approx 0.230.$$

L'attribut x_1 (Forme des oreilles) présente le gain d'information le plus élevé (0.408), ce qui signifie qu'il réduit le plus l'entropie de l'ensemble S. Par conséquent, x_1 est le meilleur choix pour la racine de l'arbre de décision.

Le choix d'un attribut comme racine repose sur la capacité de cet attribut à maximiser le gain d'information, ce qui permet de séparer au mieux les exemples dans les sous-ensembles correspondants. Dans ce cas, la Forme des oreilles (x_1) est l'attribut qui divise les données de manière la plus efficace, réduisant ainsi l'incertitude (entropie) de l'ensemble initial.

3.3 Indice de Gini

L'indice de Gini est une mesure de l'impureté d'un ensemble de données. Il est utilisé dans les arbres de décision pour évaluer la qualité d'une séparation et est défini par la formule suivante :

$$Gini(S) = 1 - \sum_{i=1}^{n} p_i^2$$

où :
— n est le nombre total de classes dans l'ensemble S,
— p_i est la proportion d'exemples appartenant à la classe i dans S.

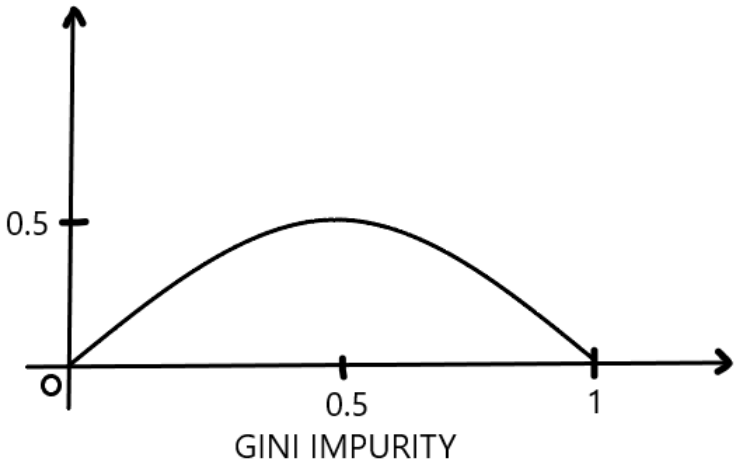


FIGURE 3 – Graphe de l'indice de Gini en fonction de la probabilité

Exemple : Calcul de l'indice de Gini Considérons un ensemble S de 9 exemples basé sur les données du tableau. Les classes possibles (y) sont 1 (chat) et 0 (non-chat). Le nombre d'exemples dans chaque classe est :

— 4 exemples dans la classe 1,

— 5 exemples dans la classe 0.

Les probabilités associées à chaque classe sont donc :

$$p_1 = \frac{4}{9}, \quad p_0 = \frac{5}{9}$$

L'indice de Gini de l'ensemble S est calculé comme suit :

$$Gini(S) = 1 - \left(\left(\frac{4}{9}\right)^2 + \left(\frac{5}{9}\right)^2\right)$$

$$Gini(S) = 1 - (0.1975 + 0.3086) = 0.4939$$

1. Attribut x_1 (Forme des Oreilles) Distribution :

— Pointues : 2 exemples (1 classe 1, 1 classe 0)

— Ovales : 4 exemples (3 classe 1, 1 classe 0)

— Flasques : 3 exemples (0 classe 1, 3 classe 0)

$$Gini(D_{x_1}) = \frac{2}{9}\times\left(1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2\right) + \frac{4}{9}\times\left(1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2\right) + \frac{3}{9}\times\left(1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2\right)$$

$$= \frac{2}{9} \times 0.5 + \frac{4}{9} \times 0.375 + \frac{3}{9} \times 0$$

$$= 0.1111 + 0.1667 + 0 = 0.2778$$

$$Gain_{Gini}(x_1) = Gini(S) - Gini(D_{x_1}) = 0.4939 - 0.2778 = 0.2161$$

2. Attribut x_2 (Forme du visage) Distribution :

— Rond : 6 exemples (3 classe 1, 3 classe 0)

— Non rond : 3 exemples (1 classe 1, 2 classe 0)

$$Gini(D_{x_2}) = \frac{6}{9} \times \left(1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2\right) + \frac{3}{9} \times \left(1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2\right)$$

$$= \frac{6}{9} \times 0.5 + \frac{3}{9} \times 0.4444$$

$$= 0.3333 + 0.1481 = 0.4814$$

$$Gain_{Gini}(x_2) = Gini(S) - Gini(D_{x_2}) = 0.4939 - 0.4814 = 0.0125$$

3. Attribut x_3 (Moustaches) Distribution :

— Présentes : 4 exemples (3 classe 1, 1 classe 0)

— Absentes : 5 exemples (1 classe 1, 4 classe 0)

$$Gini(D_{x_3}) = \frac{4}{9} \times \left(1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2\right) + \frac{5}{9} \times \left(1 - \left(\frac{1}{5}\right)^2 - \left(\frac{4}{5}\right)^2\right)$$

$$= \frac{4}{9} \times 0.375 + \frac{5}{9} \times 0.32$$

$$= 0.1667 + 0.1778 = 0.3445$$

$$Gain_{Gini}(x_3) = Gini(S) - Gini(D_{x_3}) = 0.4939 - 0.3445 = 0.1494$$

Conclusion En comparant les gains de Gini :

-
$$Gain_{Gini}(x_1) = 0.2161$$

-
$$Gain_{Gini}(x_2) = 0.0125$$

-
$$Gain_{Gini}(x_3) = 0.1494$$

L'attribut ayant le plus grand gain de Gini est x_1 (Forme des Oreilles) avec un gain de 0.2161. C'est donc cet attribut qui devrait être choisi comme premier critère de division dans l'arbre de décision.

4 Algorithmes d'arbres de décision

4.1 ID3 (Iterative Dichotomiser 3)

4.1.1 Origine

ID3 est l'un des premiers algorithmes d'arbres de décision, introduit par Ross Quinlan en 1986.

4.1.2 Critère de division

Utilise le **gain d'information** (basé sur l'entropie) pour choisir l'attribut de division.

4.1.3 Types de problèmes

Uniquement pour la **classification**.

4.1.4 Caractéristiques

— Ne supporte pas les attributs numériques (uniquement catégoriels).
— Ne gère pas les valeurs manquantes.
— Génère des arbres **non binaires** (un nœud peut avoir plusieurs branches).

4.1.5 Fonctionnement

1. Calcule l'entropie de l'ensemble de données.
2. Pour chaque attribut, calcule le gain d'information.
3. Choisit l'attribut avec le gain d'information maximal pour diviser le nœud.
4. Répète le processus récursivement pour chaque sous-ensemble.

4.1.6 Limitations

— Tendance au surajustement (overfitting) si l'arbre devient trop profond.
— Ne supporte pas la régression.

4.1.7 Exemple

Si vous avez un ensemble de données avec des attributs catégoriels comme "Temps" (Ensoleillé, Pluvieux, Nuageux), ID3 choisira l'attribut qui réduit le plus l'entropie.

4.2 C4.5 (Successeur de ID3)

4.2.1 Origine

Également développé par Ross Quinlan, C4.5 est une amélioration de ID3.

4.2.2 Critère de division

Utilise le **gain ratio** (une normalisation du gain d'information) pour éviter le biais en faveur des attributs avec beaucoup de valeurs.

4.2.3 Types de problèmes

Principalement pour la **classification**, mais peut être adapté pour la régression.

4.2.4 Caractéristiques

— Supporte les attributs **numériques** et **catégoriels**.
— Gère les **valeurs manquantes** en utilisant des techniques de substitution.
— Génère des arbres **non binaires**.
— Inclut un mécanisme de **élagage** (pruning) pour réduire le surajustement.

4.2.5 Fonctionnement

1. Calcule le gain ratio pour chaque attribut.
2. Choisit l'attribut avec le gain ratio maximal.
3. Pour les attributs numériques, détermine un seuil optimal pour la division.
4. Répète le processus récursivement.

4.2.6 Avantages

— Plus robuste que ID3 grâce au gain ratio et à la gestion des valeurs manquantes.
— Moins sujet au surajustement grâce à l'élagage.

4.2.7 Exemple

Si vous avez un attribut numérique comme "Âge", C4.5 déterminera un seuil optimal (par exemple, "Âge $\leq$ 30") pour diviser les données.

4.3 CART (Classification and Regression Trees)

4.3.1 Origine

Développé par Breiman et al. en 1984.

4.3.2 Critère de division

Utilise l'**indice de Gini** pour la classification et la **variance réduite** pour la régression.

4.3.3 Types de problèmes

Classification et **régression**.

4.3.4 Caractéristiques

— Génère des arbres **binaires** (chaque nœud a exactement deux branches).
— Supporte les attributs numériques et catégoriels.
— Ne gère pas directement les valeurs manquantes (nécessite un prétraitement).
— Inclut un mécanisme d'élagage pour éviter le surajustement.

4.3.5 Fonctionnement

1. Calcule l'indice de Gini pour chaque attribut.
2. Choisit l'attribut et le seuil qui minimisent l'impureté.
3. Divise le nœud en deux sous-ensembles.
4. Répète le processus récursivement.

4.3.6 Encodage des variables catégorielles

Pour les attributs catégoriels, CART utilise un encodage binaire (comme le **one-hot encoding**) pour les transformer en variables binaires.

4.3.7 Exemple

Pour un attribut catégoriel comme "Couleur" (Rouge, Vert, Bleu), CART peut créer des divisions binaires comme "Couleur = Rouge" vs "Couleur $\neq$ Rouge".

4.4 SPRINT (Scalable Parallelizable Induction of Decision Trees)

4.4.1 Origine

Conçu pour être efficace sur de grands ensembles de données.

4.4.2 Critère de division

Peut utiliser l'indice de Gini ou l'entropie.

4.4.3 Types de problèmes

Principalement pour la **classification**.

4.4.4 Caractéristiques

— Conçu pour les **grands ensembles de données**.
— Utilise des techniques de **parallélisation** pour accélérer la construction de l'arbre.
— Supporte les attributs numériques et catégoriels.
— Gère les valeurs manquantes.

4.4.5 Fonctionnement

1. Divise les données en plusieurs partitions.
2. Applique l'algorithme d'arbre de décision en parallèle sur chaque partition.
3. Combine les résultats pour former l'arbre final.

4.4.6 Avantages

— Très efficace pour les données massives.
— Adapté aux environnements distribués (comme Hadoop ou Spark).

4.4.7 Exemple

Dans un environnement distribué, SPRINT peut traiter des millions de lignes de données en divisant le travail entre plusieurs nœuds de calcul.

4.5 Comparaison des algorithmes

Algorithme	**Critère de division**	**Types de problèmes**	**Valeurs manquantes**
ID3	Gain d'information	Classification	Non
C4.5	Gain ratio	Classification	Oui
CART	Indice de Gini / Variance	Classification, Régression	Non (prétraitement requis)
SPRINT	Indice de Gini / Entropie	Classification	Oui

TABLE 2 – Comparaison des algorithmes d'arbres de décision

4.6 Exemples concrets

4.6.1 ID3

Utilisé pour des problèmes simples de classification avec des attributs catégoriels, comme la prédiction du temps (Ensoleillé, Pluvieux, Nuageux) en fonction de l'humidité et de la température.

4.6.2 C4.5

Idéal pour des problèmes plus complexes avec des attributs numériques et catégoriels, comme la prédiction du risque de crédit en fonction de l'âge, du revenu, et de l'historique de crédit.

4.6.3 CART

Utilisé pour des problèmes de classification et de régression, comme la prédiction du prix d'une maison en fonction de sa taille, de son emplacement, et de son âge.

4.6.4 SPRINT

Adapté aux grands ensembles de données, comme l'analyse de millions de transactions pour détecter des fraudes.

5 Conclusion

Les arbres de décision sont des outils puissants et intuitifs utilisés dans le domaine du machine learning pour résoudre des problèmes de classification et de régression. Grâce à leur simplicité et à leur capacité à produire des modèles facilement interprétables, les arbres de décision sont largement utilisés dans des applications variées allant de la médecine à la finance.

Les principaux avantages des arbres de décision sont leur facilité de compréhension et leur capacité à gérer aussi bien des données numériques que catégorielles. De plus, ils ne nécessitent pas de normalisation des données, ce qui les rend particulièrement pratiques dans des situations où les données sont peu préparées.

Cependant, les arbres de décision présentent aussi certaines limites. Ils peuvent facilement devenir trop complexes et susceptibles de sur-ajuster (overfitting) les données d'entraînement, ce qui réduit leur capacité à généraliser sur de nouvelles données. Pour cette raison, il est souvent nécessaire d'utiliser des techniques telles que l'élagage (pruning) ou des ensembles d'arbres (comme les forêts aléatoires) pour améliorer la performance et éviter le sur-apprentissage.

Enfin, bien que les arbres de décision soient très efficaces pour certains types de données, ils ne conviennent pas nécessairement à tous les types de problèmes. Ils peuvent, par exemple, être moins performants que d'autres algorithmes de machine learning, comme les réseaux neuronaux ou les machines à vecteurs de support (SVM), lorsque les relations entre les données sont complexes et non linéaires.

En conclusion, les arbres de décision restent un modèle fondamental et très utile dans l'arsenal des outils de machine learning, et leur compréhension est essentielle pour aborder des projets d'intelligence artificielle de manière solide et efficace.