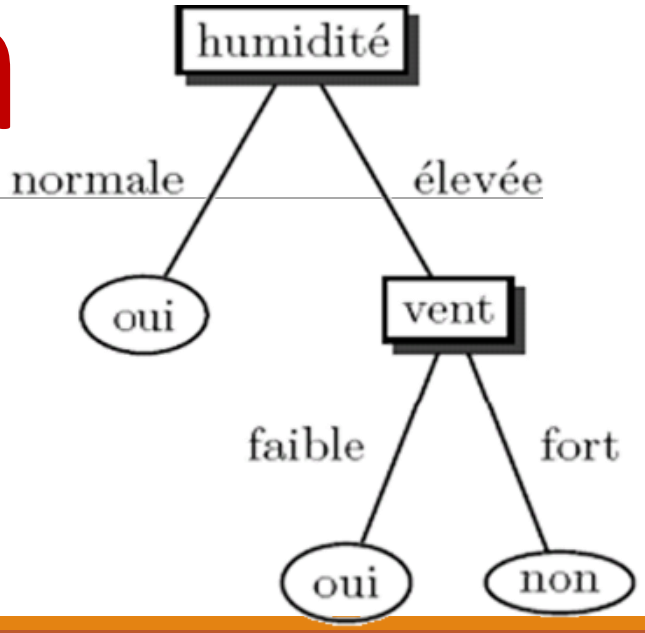
Arbres de décision

HADJADJ ABDELHALIM

Centre universitaire de Mila
Master I: apprentissage Automatique



Arbre de décision

Un arbre de décision est construit à partir d'un ensemble D composé de N exemples d'apprentissage, caractérisés par d attributs pouvant être **qualitatifs ou quantitatifs**.

Chaque exemple x est associé à une classe y, qui représente la variable cible et appartient à l'espace Y. L'arbre ainsi généré respecte la structure suivante :

- Chaque **nœud** représente un test basé sur la valeur d'un ou plusieurs attributs.
- Chaque **branche** issue d'un nœud correspond à une ou plusieurs valeurs possibles du test effectué.
- Chaque **feuille** est associée à une valeur spécifique de la variable cible.

Utilisation des arbres de décision (AD)

Un arbre de décision peut être utilisé de plusieurs façons :

1. **Classification** : Il permet d'assigner une nouvelle donnée à une catégorie spécifique en suivant les tests définis dans l'arbre.
2. **Estimation d'attributs** : Lorsqu'une valeur d'attribut est manquante, l'arbre peut aider à l'inférer en fonction des autres caractéristiques observées.
3. **Extraction de règles** : L'arbre peut être converti en un ensemble de règles logiques facilitant l'interprétation du modèle.
4. **Analyse de la pertinence des attributs** : Il permet d'identifier les attributs les plus influents dans la prise de décision, en fonction de leur impact sur la classification.

Construction d’un arbre de décision

Exemple :

•Considérons un ensemble de jours, où chaque jour représente un exemple.

• Chaque jour est décrit par plusieurs attributs météorologiques,

•tels que **le ciel, la température, l’humidité de l’air et la force du vent**.

•La variable cible est **« jouer au tennis ? »**, dont les valeurs possibles sont **{oui, non}**.

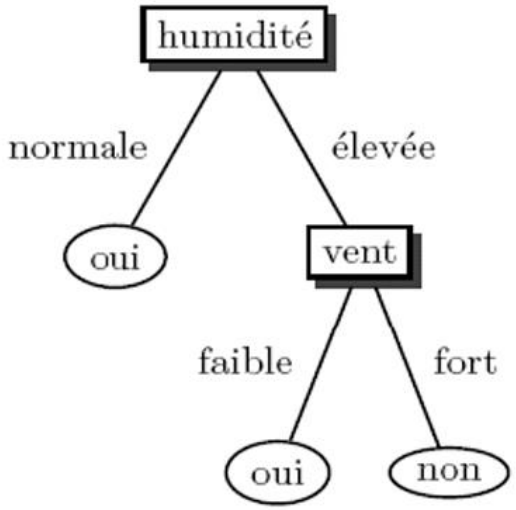
•Une fois l’arbre de décision établi, il permettra de déterminer si une nouvelle journée, en fonction de ses conditions météorologiques, est propice ou non pour jouer au tennis.

Construction d'un AD

Jour	Ciel	Température	Humidité	Vent	Jouer au tennis ?
1	Ensoleillé	Chaude	Élevée	Faible	Non
2	Ensoleillé	Chaude	Élevée	Fort	Non
3	Couvert	Chaude	Élevée	Faible	Oui
4	Pluie	Tiède	Élevée	Faible	Oui
5	Pluie	Fraîche	Normale	Faible	Oui
6	Pluie	Fraîche	Normale	Fort	Non
7	Couvert	Fraîche	Normale	Fort	Oui
8	Ensoleillé	Tiède	Élevée	Faible	Non
9	Ensoleillé	Fraîche	Normale	Faible	Oui
10	Pluie	Tiède	Normale	Faible	Oui
11	Ensoleillé	Tiède	Normale	Fort	Oui
12	Couvert	Tiède	Élevée	Fort	Oui
13	Couvert	Chaud	Normale	Faible	Oui
14	Pluie	Tiède	Élevée	Fort	Non

Construction d'un AD

L'exemple de ligne 1 du tableau sera classé comme «oui» et les exemples 2 et 5 seront classés « non » et « oui ».



Jour	Ciel	Température	Humidité	Vent
1	Ensoleillé	Chaude	Élevée	Faible

Complexité et algorithmes de construction des arbres de décision

- La construction d'un arbre de décision optimal, c'est-à-dire minimisant le nombre d'erreurs de classification, est un **problème NP-complet**.
- Il est donc irréaliste de toujours espérer obtenir un arbre parfaitement optimal pour un ensemble de données donné. L'objectif est plutôt de construire un arbre **correct et efficace**.
- Plusieurs algorithmes ont été développés pour générer des arbres de décision, notamment **CART, ID3 et C4.5** :
  - **ID3** fonctionne uniquement avec des **attributs catégoriels**.
  - **C4.5**, en plus des attributs catégoriels, prend également en charge les **attributs numériques**.

Construction récursive d'un arbre de décision

Dans cet exemple, nous supposons que tous les attributs sont **nominaux**. Les algorithmes **ID3 et C4.5** construisent l'arbre de décision de manière récursive en procédant comme suit :

- **Choix de la racine** : Sélection d'un attribut à placer à la racine de l'arbre. Le nombre de branches issues de cette racine correspond au nombre de valeurs possibles de l'attribut choisi.
- **Création des branches** : Chaque branche représente une valeur de l'attribut et est associée à l'ensemble des exemples où cet attribut prend cette valeur.
- **Construction récursive** : Pour chaque sous-ensemble d'exemples associé à une branche, on construit un sous-arbre en excluant l'attribut déjà utilisé comme racine et en répétant le processus jusqu'à atteindre un critère d'arrêt (exemple : tous les exemples d'un sous-ensemble appartiennent à la même classe).

Arbres de décision: Algorithmes

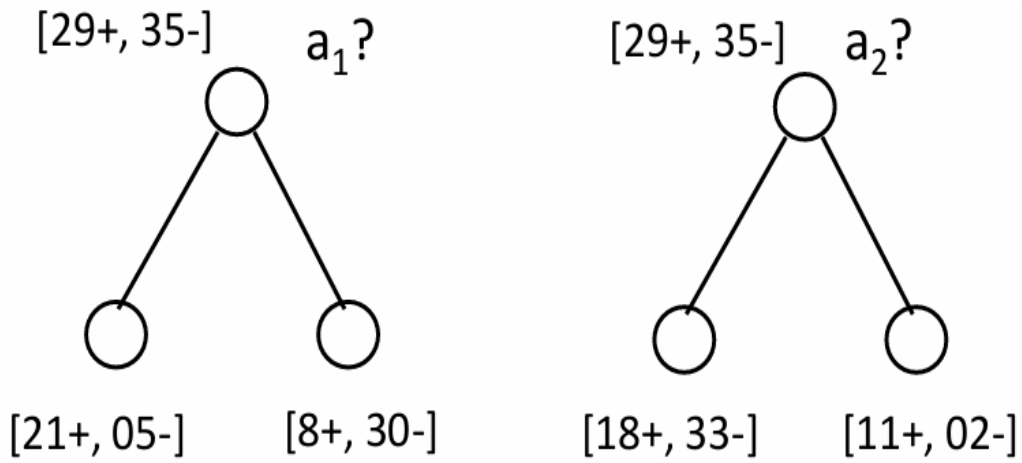
Nœud = racine;

1- a ← meilleur attribut de décision pour prochain nœud;
2- Assigner a comme attribut de décision au nœud;
3- Pour chaque valeur de a, créer un nœud descendant;
4- Trier les exemples d'entrainement dans les feuilles de l'arbre;
5- **Si** (les exemples sont parfaitement classés)
STOP;
sinon
Itérer sur un nœud feuille.
fin

Sélection de l'attribut de décision:

L’objectif est de déterminer si l’attribut **a1** ou **a2** est le plus pertinent pour diviser les données de manière efficace.

Quel attribut choisir : **a1 ou a2** ?



Construction récursive d'un arbre de décision

Sélection de l’attribut de décision

- Les algorithmes de construction des arbres de décision suivent une approche **descendante**, en sélectionnant à chaque étape l’attribut qui permet de **mieux segmenter** l’ensemble des exemples.
- Le choix de l’attribut repose généralement sur des **métriques** évaluant l’**homogénéité** de la variable cible dans les sous-groupes formés après chaque division.
- Parmi ces métriques, le **gain d’information**, basé sur le concept d’**entropie** en théorie de l’information, est couramment utilisé pour mesurer la pertinence d’un attribut.

Construction récursive d'un arbre de décision

L’entropie est une mesure de l’incertitude associée à une variable aléatoire x. Elle quantifie la quantité moyenne d’information nécessaire pour encoder les valeurs de cette variable.

Elle est définie par la formule :

$$H(\mathrm{x}) = -\sum_{\mathrm{x}=x_i} p(\mathrm{x} = x_i)\log_2(p(\mathrm{x} = x_i))$$

où :

- p(x) est la probabilité d’occurrence de la valeur x.
- log2p(x) est le logarithme en base 2 de cette probabilité.
- La somme est effectuée sur toutes les valeurs possibles de x.

L’entropie est mesurée en **bits** et représente le nombre moyen de bits requis pour coder x.

Sélection de l'Attribut de Décision

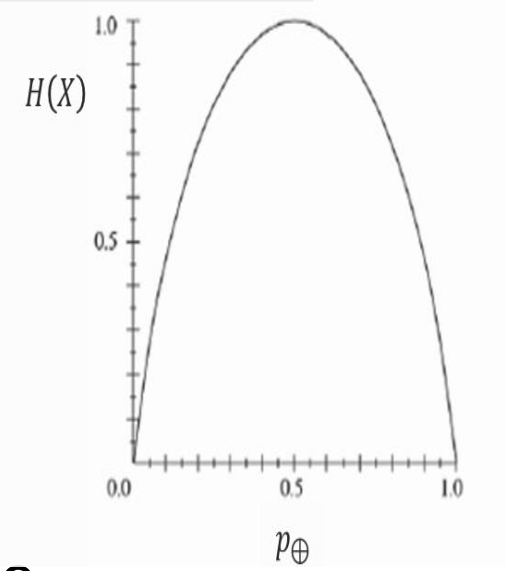
- Pour K = 2 (deux classes), les éléments de la classe C_1 seront dénotés par $\oplus$ et ceux de la classe C_2 par $\ominus$. On aura:

$$H(\mathrm{x}) = -p_{\oplus}\log_2(p_{\oplus}) - p_{\ominus}\log_2(p_{\ominus})$$

- $p_{\oplus}$ et $p_{\ominus}$ sont les proportions des deux classes dans le nœud, avec $p_{\oplus} + p_{\ominus} = 1$. On aura aussi:

- $0 \leq H(\mathrm{x}) \leq 1$.
- Si $p_{\oplus} = 0$ ou $p_{\ominus} = 0$, alors $H(\mathrm{x}) = 0$.
- Si $p_{\oplus} = p_{\ominus} = 0.5$, alors $H(\mathrm{x}) = 1$ (entropie maximale).

Sélection de l’Attribut de Décision



Sélection de l’Attribut de Décision (Cas Binaire)

•Lorsqu’il n’existe que deux classes, **l’entropie** suit une courbe caractéristique illustrant H(X) en fonction de p, où p représente la probabilité d’appartenance à une classe donnée.

•**L’entropie** évalue le niveau d’impureté d’un nœud, c'est-à-dire l’hétérogénéité des classes dans un sous-ensemble d’exemples.

•Un nœud est pur lorsque tous les exemples appartiennent à la même classe (entropie = 0), et maximalement impur lorsque les classes sont également réparties (entropie = 1).

Sélection de l'Attribut de Décision

- Soit une population d'exemples $\mathcal{D}$. Le gain d'information de $\mathcal{D}$ par rapport à un attribut a_j donné est la variation d'entropie causée par la partition de $\mathcal{D}$ selon a_j.

$$Gain(\mathcal{D}, a_j) = H(\mathcal{D}) - \sum_{v \in valeurs(a_j)} \frac{|\mathcal{D}_{aj=v}|}{|\mathcal{D}|} H(\mathcal{D}_{aj=v})$$

- $\mathcal{D}_{aj=v} \subset \mathcal{D}$ est l'ensemble des exemples ayant $a_j = v$.

- $|\mathcal{D}|$ indique la cardinalité de l'ensemble $\mathcal{D}$.

Gain d'Information pour le Choix de l'Attribut

- Si un attribut a_j peut prendre **trois valeurs distinctes** v_1, v_2, v_3
- l'ensemble D est divisé en **trois sous-ensembles** :

$$\mathcal{D}_{aj=v1}, \mathcal{D}_{aj=v2}, \mathcal{D}_{aj=v3}$$

- Le **gain d'information** est calculé comme suit :

$$H(\mathcal{D}) - \frac{|\mathcal{D}_{aj=v1}|}{|\mathcal{D}|} H(\mathcal{D}_{aj=v1}) - \frac{|\mathcal{D}_{aj=v2}|}{|\mathcal{D}|} H(\mathcal{D}_{aj=v2}) - \frac{|\mathcal{D}_{aj=v3}|}{|\mathcal{D}|} H(\mathcal{D}_{aj=v3})$$

- Ce **gain** correspond à la **différence entre l'entropie initiale** de D et **l'entropie moyenne** des sous-ensembles après la division selon a_j.
- **Plus le gain d'information est élevé**, plus la partition est efficace car elle **réduit l'impureté** et **rend les sous-ensembles plus homogènes**.

Gain d'Information pour le Choix de l'Attribut

Exercice:

- Dans notre exemple, il existe 9 $\oplus$ et 5 $\ominus$. Parmi ces exemples, 6 $\oplus$ et 2 $\ominus$ prennent la valeur «alpha» pour l'attribut a, tandis que les autres exemples prennent la valeur «beta» pour cet attribut.
- Calculer le gain d'information $Gain(\mathcal{D}, a)$ pour l'attribut a, si on choisi de le placer en racine.

$$Gain(\mathcal{D}, a_j) = H(\mathcal{D}) - \sum_{v \in valeurs(a_j)} \frac{|\mathcal{D}_{aj=v}|}{|\mathcal{D}|} H(\mathcal{D}_{aj=v})$$

$$H(\mathrm{x}) = -\sum_{\mathrm{x}=x_i} p(\mathrm{x} = x_i) \log_2(p(\mathrm{x} = x_i))$$

Jour	Ciel	Température	Humidité	Vent	Jouer au tennis ?
1	Ensoleillé	Chaude	Élevée	Faible	Non
2	Ensoleillé	Chaude	Élevée	Fort	Non
3	Couvert	Chaude	Élevée	Faible	Oui
4	Pluie	Tiède	Élevée	Faible	Oui
5	Pluie	Fraîche	Normale	Faible	Oui
6	Pluie	Fraîche	Normale	Fort	Non
7	Couvert	Fraîche	Normale	Fort	Oui
8	Ensoleillé	Tiède	Élevée	Faible	Non
9	Ensoleillé	Fraîche	Normale	Faible	Oui
10	Pluie	Tiède	Normale	Faible	Oui
11	Ensoleillé	Tiède	Normale	Fort	Oui
12	Couvert	Tiède	Élevée	Fort	Oui
13	Couvert	Chaud	Normale	Faible	Oui
14	Pluie	Tiède	Élevée	Fort	Non

Gain d'Information pour le Choix de l'Attribut

$$Gain(\mathcal{D}, a) = H(D) - \frac{8}{14} H(\mathcal{D}_{a=alpha}) - \frac{6}{14} H(\mathcal{D}_{a=beta})$$

$$H(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

$$H(\mathcal{D}_{a=alpha}) = -\frac{6}{8}\log_2\left(\frac{6}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right) = 0.811$$

$$H(\mathcal{D}_{a=beta}) = -\frac{3}{6}\log_2\left(\frac{3}{6}\right) - \frac{3}{6}\log_2\left(\frac{3}{6}\right) = 1$$

$$Gain(\mathcal{D}, a) = 0.940 - \frac{8}{14}0.811 - \frac{6}{14}1 = 0.940 - 0.8920 = 0.048$$

Jour	Ciel	Température	Humidité	Vent	Jouer au tennis ?
1	Ensoleillé	Chaude	Élevée	Faible	Non
2	Ensoleillé	Chaude	Élevée	Fort	Non
3	Couvert	Chaude	Élevée	Faible	Oui
4	Pluie	Tiède	Élevée	Faible	Oui
5	Pluie	Fraîche	Normale	Faible	Oui
6	Pluie	Fraîche	Normale	Fort	Non
7	Couvert	Fraîche	Normale	Fort	Oui
8	Ensoleillé	Tiède	Élevée	Faible	Non
9	Ensoleillé	Fraîche	Normale	Faible	Oui
10	Pluie	Tiède	Normale	Faible	Oui
11	Ensoleillé	Tiède	Normale	Fort	Oui
12	Couvert	Tiède	Élevée	Fort	Oui
13	Couvert	Chaud	Normale	Faible	Oui
14	Pluie	Tiède	Élevée	Fort	Non

a	y
6 ⊕ → alpha	9 ⊕
2 ⊖ → alpha	
3 ⊕ → beta	5 ⊖
3 ⊖ → beta	

Sélection de l'Attribut de Décision (Cas Binaire)

L'algorithme **ID3** suit un processus itératif pour choisir l'attribut optimal à placer à la racine d'un arbre de décision :

1. **Sélection de l'attribut optimal** :
    1. Calculer le **gain d'information** pour chaque attribut.
    2. Choisir l'attribut avec le **gain d'information maximal** et le placer à la racine.
2. **Construction des sous-arbres** :
    1. Diviser l'ensemble des exemples en sous-groupes en fonction des valeurs de l'attribut choisi.
    2. Appliquer récursivement l'algorithme ID3 à chaque sous-groupe.
3. **Critère d'arrêt** :
    1. L'algorithme s'arrête lorsque le **gain d'information devient négligeable** ou lorsque tous les exemples d'un nœud appartiennent à la même classe.

Analyse d'un Exemple d'Arbre de Décision

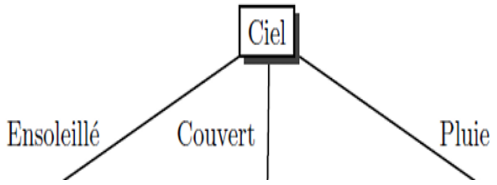
Dans le cas de l'exemple **« Jouer au tennis ? »** :

- Comme les exemples ne sont **ni tous positifs (⊕), ni tous négatifs (⊖)** et qu'il reste encore des attributs disponibles, il est nécessaire de **calculer les gains d'information** pour chaque attribut.
- L'attribut ayant le **gain d'information maximal** est sélectionné comme **racine de l'arbre de décision**.
- Dans ce cas, l'attribut **« Ciel »** est choisi comme racine.

Attribut	Gain
Ciel	0, 246
Humidité	0, 151
Vent	0, 048
Température	0, 029

Construction de l’Arbre de Décision

- L’attribut **« Ciel »** peut prendre **trois valeurs** : **Ensoleillé, Pluie, et Couvert**.
- Pour la branche **« Ensoleillé »**, l'algorithme **ID3** est appliqué récursivement sur les **5 exemples** : $x(1)$, $x(2)$, $x(8)$, $x(9)$, et $x(11)$.
- À ce stade, les **gains d’information** des **trois attributs restants** sont calculés.
- L’attribut **« Humidité »** ayant le **gain d’information maximal**, il est sélectionné pour la suite de la construction de l’arbre.

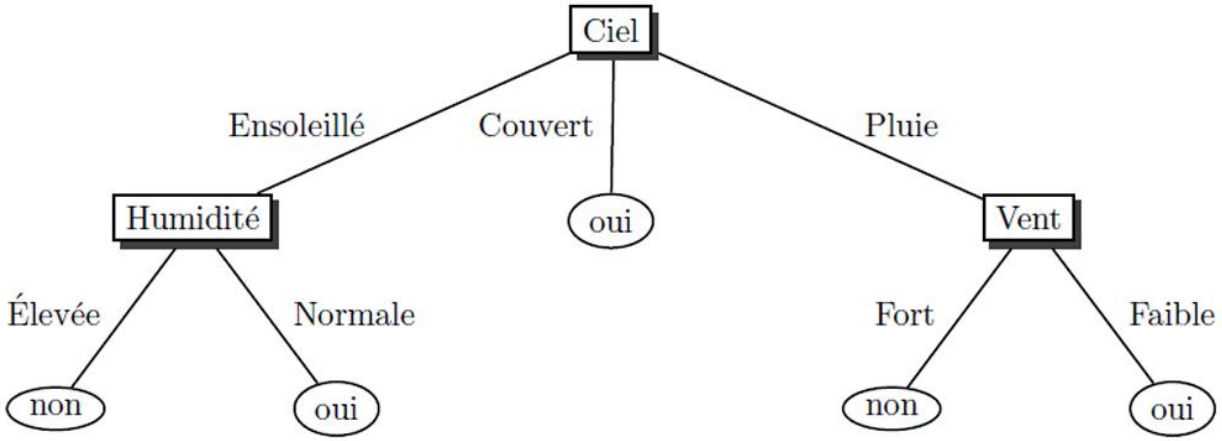


Jour	Ciel	Température	Humidité	Vent	Jouer au tennis ?
1	Ensoleillé	Chaude	Élevée	Faible	Non
2	Ensoleillé	Chaude	Élevée	Fort	Non
3	Couvert	Chaude	Élevée	Faible	Oui
4	Pluie	Tiède	Élevée	Faible	Oui
5	Pluie	Fraîche	Normale	Faible	Oui
6	Pluie	Fraîche	Normale	Fort	Non
7	Couvert	Fraîche	Normale	Fort	Oui
8	Ensoleillé	Tiède	Élevée	Faible	Non
9	Ensoleillé	Fraîche	Normale	Faible	Oui
10	Pluie	Tiède	Normale	Faible	Oui
11	Ensoleillé	Tiède	Normale	Fort	Oui
11	Ensoleillé	Tiède	Normale	Fort	Oui
12	Couvert	Tiède	Élevée	Fort	Oui
13	Couvert	Chaud	Normale	Faible	Oui
14	Pluie	Tiède	Élevée	Fort	Non

Attribut	Gain
Humidité	0, 970
Vent	0, 570
Température	0, 019

Construction de l’Arbre de Décision

- **Branche « Pluie »** :
  - À partir de la racine, **ID3** est appliqué récursivement sur **5 exemples** : $x(4)$, $x(5)$, $x(6)$, $x(10)$, et $x(14)$.
  - La construction de l’arbre se poursuit en divisant ces exemples selon les attributs restants.
- **Branche « Couvert »** :
  - À partir de la racine, **ID3** est appliqué récursivement sur **4 exemples** : $x(3)$, $x(7)$, $x(12)$, et $x(13)$.
  - Tous ces exemples appartenant à la classe $\oplus$ **(« oui »)**, on affecte directement cette classe à cette feuille de l’arbre.



Analyse de l’Arbre de Décision

À partir du graphe obtenu, on peut tirer les conclusions suivantes :

- **L'attribut « Température » n'est pas utilisé** : Cela signifie qu'il **n'influence pas** la classification et **n'est pas pertinent** pour décider du jeu de tennis.
- **Si « Ciel » est « Ensoleillé »**, alors **« Vent » n'est pas pertinent** : Cela indique que la décision peut être prise sans considérer cet attribut.
- **Si « Ciel » est « Pluie »**, alors **« Humidité » n'est pas pertinente** : Cela signifie que l'attribut « Humidité » n'apporte pas d'information supplémentaire pour la classification dans ce cas.

la classification d'une nouvelle donnée

Une fois l'AD construit à partir des exemples d'apprentissage D, la classification d'une nouvelle donnée x suit l'algorithme suivant :

Algorithme: entrées (**AD**, *x*)

- **Nc** = *racine* (**AD**)
- *Tant-que* (**Nc** ≠ feuille) *faire*:
  - En fonction de l'attribut testé dans **Nc** et de sa valeur dans **x**, suivre l'une des branches de **Nc**.
  - Le nœud atteint devient **Nc**.
- *Fin tant-que*
- Retourner Étiquette (**Nc**).

Exemple X: Ensoleillé Tiède Normale Fort

Attributs numériques dans les Arbres de Décision (AD)

- Contrairement à ID3, son successeur C4.5 prend en compte les attributs numériques, c'est-à-dire des attributs avec un domaine de valeurs potentiellement infini.
- Mis à part cette différence et quelques ajustements supplémentaires, la construction d'un AD avec C4.5 suit le même principe que celle d'ID3.
- Dans C4.5, un nœud de l'AD peut inclure un test évaluant si la valeur d'un attribut numérique est inférieure ou égale à un certain seuil.
  → Cela revient à créer un nouveau **pseudo-attribut binaire**.

Attributs numériques dans les Arbres de Décision (AD)

- Prenons l'exemple illustratif **Jouer au tennis ?** , dans lequel les attributs **Température** et **Humidité** ont été transformés en valeurs numériques.
- Considérons uniquement les exemples pour lesquels l'attribut **Ciel** a la valeur **Ensoleillé** , Soit $\mathbf{D_{Ciel}}$**=Ensoleillé** l'ensemble des exemples filtrés, avec un seul attribut numérique, défini comme suit :

Jour	Température	« jouer au tennis »
1	27, 5	non
2	25	non
8	21	non
9	19, 5	oui
11	22, 5	oui

Attributs numériques dans les Arbres de Décision (AD)

- Les exemples sont d'abord triés selon la valeur de leur attribut numérique.
- Chaque attribut est associé au numéro de l'exemple correspondant ainsi qu'à la valeur de l'attribut cible.
- Un seuil **s** est ensuite déterminé pour partitionner cet ensemble d'exemples.

Température	19,5	21	22,5	25	27,5
Jour	9	8	11	2	1
« jouer au tennis ? »	oui	non	oui	non	non

Attributs numériques dans les Arbres de Décision (ADs)

C4.5 applique les règles suivantes :

- Ne pas séparer deux exemples consécutifs ayant la même classe. Ainsi, la séparation ne peut se faire qu'entre x(9) et x(8), x(8) et x(11), ainsi que x(11) et x(2).
- Si la coupure se fait entre deux valeurs v et w (avec v<wv < w), le seuil s est fixé à v (ou éventuellement à (v + w) / 2).
- Le seuil s est choisi de manière à maximiser le gain d'information.

Attributs numériques dans les Arbres de Décision (ADs)

- **Remarque** :
  - ✓Une fois le seuil s déterminé et le nœud créé, chaque sous-arbre peut à nouveau tester la valeur de cet attribut.
  - ✓Contrairement aux attributs qualitatifs, qui génèrent des nœuds avec autant de branches que de valeurs distinctes, un attribut numérique divise l'ensemble des valeurs en deux parties.
  - ✓Chaque sous-ensemble peut ensuite être affiné jusqu'à ne contenir que des exemples ayant la même classe cible.

Attributs numériques dans les Arbres de Décision (ADs)

- Dans notre exemple, l'entropie de l'ensemble est donnée par:

$$H(\mathcal{D}) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) \approx 0.971$$

- Pour $s = 21$, le gain d'information est:

$$Gain(\mathcal{D}, Température, s = 21)$$
$$= H(\mathcal{D}) - \left(\frac{2}{5}H\left(\mathcal{D}_{Température \leq 21}\right) + \frac{3}{5}H\left(\mathcal{D}_{Température > 21}\right)\right)$$

$$H\left(\mathcal{D}_{Température \leq 21}\right) = -\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) \approx 1$$

$$H\left(\mathcal{D}_{Température > 21}\right) = -\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right) \approx 0.39$$

- Il s'en suit que:

$$Gain(\mathcal{D}, Température, s = 21) = 0.971 - \left(\frac{2}{5}\cdot 1 + \frac{3}{5}\cdot 0.39\right) \approx 0.337$$

Température	19,5	21	22,5	25	27,5
Jour	9	8	11	2	1
« jouer au tennis ? »	oui	non	oui	non	non

Attributs numériques dans les Arbres de Décision (ADs)

De la même manière, en fonction du seuil, remplir le tableau

Seuil	Gain($\mathcal{D}$, *Température*, s)
$s = 21$	0.337
$s = 19.5$	??
$s = 22.5$	??
S=25	??

Attributs numériques dans les Arbres de Décision (ADs)

- C4.5 applique ce traitement à chaque attribut quantitatif, en identifiant pour chacun le seuil qui maximise le gain d'information.
- Pour chaque attribut numérique, le gain d'information considéré est celui obtenu avec le seuil produisant la plus grande amélioration.
- Finalement, parmi tous les attributs (qu'ils soient quantitatifs ou qualitatifs, ce dernier cas étant traité comme dans ID3), celui qui est retenu est celui qui offre le gain d'information le plus élevé.

Attributs numériques dans les Arbres de Décision (ADs)

Rapport de gain pour C4.5

- Lorsqu'un attribut est numérique ou possède une arité élevée, il est naturellement favorisé pour être utilisé dans les tests des nœuds.
- Pour compenser cet avantage, C4.5 utilise le **rapport de gain** au lieu du simple gain d'information afin de choisir l'attribut le plus pertinent pour chaque nœud.
- Le rapport de gain est défini par :

$$Rapport\ gain(\mathcal{D}, a) = \frac{Gain(\mathcal{D}, a)}{Division\ Inf(\mathcal{D}, a)}$$

$$Division\ Inf(\mathcal{D}, a) = \sum_{v \in valeurs(a)} \frac{|\mathcal{D}_{a=v}|}{|\mathcal{D}|} \log_2 \left(\frac{|\mathcal{D}_{a=v}|}{|\mathcal{D}|} \right)$$

Attributs numériques dans les Arbres de Décision (ADs)

1. Calcul de l'Entropie Initiale

L'entropie du jeu de données avant toute séparation est :

$$E(S) = -p_{\text{Oui}} \log_2(p_{\text{Oui}}) - p_{\text{Non}} \log_2(p_{\text{Non}})$$

Dans notre tableau, nous avons **9 "Oui"** et **5 "Non"** sur **14 exemples**, donc :

$$E(S) = -\frac{9}{14} \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \log_2\left(\frac{5}{14}\right)$$

$$E(S) \approx -0.643 - 0.485 = 1.029$$

$$Gain(\text{Ciel}) = E(S) - \sum_{i=1}^{k} \frac{|S_i|}{|S|} E(S_i)$$

$$Gain(\text{Ciel}) = 1.029 - \left(\frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971\right)$$

$$Gain(\text{Ciel}) = 1.029 - (0.3475 + 0 + 0.3475) = 0.294$$

Rapport de gain pour C4.5

Jour	Ciel	Température	Humidité	Vent	Jouer au tennis ?
1	Ensoleillé	27,5	85	Faible	Non
2	Ensoleillé	25	90	Fort	Non
3	Couvert	26,5	86	Faible	Oui
4	Pluie	20	96	Faible	Oui
5	Pluie	19	80	Faible	Oui
6	Pluie	17,5	70	Fort	Non
7	Couvert	17	65	Fort	Oui
8	Ensoleillé	21	95	Faible	Non
9	Ensoleillé	19,5	70	Faible	Oui
10	Pluie	22,5	80	Faible	Oui
11	Ensoleillé	22,5	70	Fort	Oui
12	Couvert	21	90	Fort	Oui
13	Couvert	25,5	75	Faible	Oui
14	Pluie	20,5	91	Fort	Non

$$GD(\text{Ciel}) = -\sum_{i=1}^{k} \frac{|S_i|}{|S|} \log_2\left(\frac{|S_i|}{|S|}\right)$$

$$GD(\text{Ciel}) = -\left(\frac{5}{14} \log_2\left(\frac{5}{14}\right) + \frac{4}{14} \log_2\left(\frac{4}{14}\right) + \frac{5}{14} \log_2\left(\frac{5}{14}\right)\right)$$

$$GD(\text{Ciel}) \approx -(0.530 + 0.523 + 0.530) = 1.583$$

Calcul du Rapport de Gain

$$\text{Rapport de Gain}(\text{Ciel}) = \frac{0.294}{1.583} \approx 0.186$$

Gestion des valeurs d'attributs manquantes (VAM)

L'algorithme **ID3** ne prévoit pas de solution pour gérer les valeurs d'attributs manquantes, tandis que **C4.5** intègre un mécanisme spécifique à ce problème.

- Deux situations sont à distinguer :
  - Certains attributs des exemples d'apprentissage ne sont pas renseignés.
  - Certains attributs de la donnée à classifier restent absents.

Gestion des valeurs d'attributs manquantes (VAM) dans la phase d'apprentissage

Plusieurs approches générales peuvent être adoptées :

- **Exclusion des exemples incomplets** :
  On élimine les exemples présentant des valeurs manquantes, bien que cela réduise la taille de l'ensemble d'apprentissage.
- **Valorisation de l'absence comme information** :
  L'absence d'une valeur est considérée comme une information à part entière. On ajoute alors une valeur spécifique à l'ensemble des valeurs possibles de l'attribut, indiquant qu'elle est inconnue.
- **Imputation par la modalité majoritaire** :
  La valeur manquante est remplacée par la valeur la plus fréquente pour cet attribut parmi les exemples présents dans le nœud.

Gestion des valeurs d’attributs manquantes (VAM)

Gestion des valeurs d’attributs manquantes dans la phase d’apprentissage

- Dans chaque nœud, les différentes valeurs observées pour un attribut sont pondérées en fonction de la proportion d'exemples d'apprentissage présentant chacune de ces valeurs.
  - C4.5 adopte cette approche, en attribuant à chaque valeur un poids spécifique basé sur sa fréquence dans le nœud.
  - Ainsi, des fractions d'exemples interviennent dans la construction de l’arbre, nécessitant une adaptation du calcul du gain d'information.

Gestion des valeurs d'attributs manquantes durant la phase d'apprentissage

- Pour le calcul du gain d'information, seuls les exemples pour lesquels l'attribut est renseigné sont considérés.
- Soit X l'ensemble des exemples présents dans le nœud courant (celui pour lequel l'attribut à tester est en cours de détermination), et soit D_{sans} ?⊂D l'ensemble des exemples dont l'attribut est effectivement valué.

- On redéfinit: $H(\mathcal{D}) = H(\mathcal{D}_{sans?})$ et on a:

$$Gain(\mathcal{D},a) = \left(H(\mathcal{D}) - \sum_{v \in valeurs(a)} \frac{|\mathcal{D}_{sans?,a=v}|}{|\mathcal{D}_{sans?}|} H(\mathcal{D}_{sans?,v=a}) \right) \frac{|\mathcal{D}_{sans?}|}{|\mathcal{D}|}$$

Gestion des valeurs d'attributs manquantes (VAM) durant la phase d'apprentissage

Exemple : Supposons que $x^{(12)}$ ait ? a la place de « *Couvert* » comme valeur de son attribut « **Ciel** ».

$$H(\mathcal{D}) = -\left(\frac{8}{13}\log_2\left(\frac{8}{13}\right) - \frac{5}{13}\log_2\left(\frac{5}{13}\right)\right) \approx 0.961$$

$$Gain(\mathcal{D}, Ciel)$$

Ensoleillé

$$\approx \frac{13}{14}\left(0.961 - \frac{5}{13}\left(-\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right)\right.$$

couvert

$$- \frac{3}{13}\left(-\frac{3}{3}\log_2\left(\frac{3}{3}\right) - \frac{0}{3}\log_2\left(\frac{0}{3}\right)\right)$$

pluie

$$\left. - \frac{5}{13}\left(-\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right)\right) \approx 0.199$$

Jour	Ciel	Température	Humidité	Vent	Jouer au tennis ?
1	Ensoleillé	27, 5	85	Faible	Non
2	Ensoleillé	25	90	Fort	Non
3	Couvert	26, 5	86	Faible	Oui
4	Pluie	20	96	Faible	Oui
5	Pluie	19	80	Faible	Oui
6	Pluie	17, 5	70	Fort	Non
7	Couvert	17	65	Fort	Oui
8	Ensoleillé	21	95	Faible	Non
9	Ensoleillé	19, 5	70	Faible	Oui
10	Pluie	22, 5	80	Faible	Oui
11	Ensoleillé	22, 5	70	Fort	Oui
12	?	21	90	Fort	Oui
13	Couvert	25, 5	75	Faible	Oui
14	Pluie	20, 5	91	Fort	Non

Gestion des valeurs d'attributs manquantes (VAM) durant la phase d'apprentissage

- L'attribut offrant le gain d'information maximal, **« Ciel »**, est placé à la racine de l'arbre.
- L'exemple **12** est réparti avec des poids de **5/13**, **3/13**, et **5/13** sur les branches respectives **« Ensoleillé »**, **« Couvert »**, et **« Pluie »**.
- Les autres exemples sont affectés directement à leur branche respective avec un poids de **1** chacun.

Exemple : Dans la branche **« Ensoleillé »**, la proportion des exemples classés $\oplus$ sera d'environ **0.44**, correspondant au rapport **(5/13+2)/(5/13+2+3)**, tandis que la proportion des $\ominus$ sera **3/(5/13+2+3)** .

$$\text{Proportion des } \oplus = \frac{\text{Nombre d'exemples } \oplus + \text{Poids de } x(12)}{\text{Nombre total d'exemples} + \text{Poids de } x(12)}$$

$$\text{Proportion des } \ominus = \frac{\text{Nombre d'exemples } \ominus}{\text{Nombre total d'exemples} + \text{Poids de } x(12)}$$

$$\frac{5/13+2}{5/13+2+3} \approx 0.44$$

$$\frac{3}{5/13+2+3} \approx 0.56 \, .$$

Gestion des valeurs d'attributs manquantes (VAM) en phase de classification

- Lors de la descente dans l'arbre, si un nœud teste un attribut dont la valeur est inconnue, **C4.5 estime la probabilité** pour la donnée de suivre chacune des branches, en se basant sur la répartition des exemples d'apprentissage couverts par ce nœud.
- La donnée est alors **partiellement répartie** sur plusieurs branches, selon ces probabilités.
- Une fois arrivée aux feuilles, **C4.5 détermine la classe la plus probable** en sommant les poids associés à chaque classe.
- **La classe prédite est celle dont le poids total est maximal.**

Gestion des VAM en phase de classification

– Exemple

Considérons la classification de la donnée :

(Ciel = ?, Température = Tiède, Humidité = ?, Vent = Faible)

- La valeur de **« Ciel »** étant inconnue, on détermine la répartition des exemples :
  - **5/14** pour **Ensoleillé**,
  - **4/14** pour **Couvert**,
  - **5/14** pour **Pluie**.
- La classification se poursuit en affectant ces poids aux branches correspondantes :
  - **5/14** des données poursuivent vers le nœud testant **« Humidité ».**
  - **4/14** suivent la branche **« Couvert »** qui mène directement à la classe **« Oui ».**
  - **5/14** continuent vers le nœud testant **« Vent ».**

Gestion des VAM en phase de classification

– Suite de l'exemple

•La valeur de **« Humidité »** étant également inconnue, nous utilisons la répartition des exemples dans la branche **« Ensoleillé »** :

- **3/5** des exemples ont pour cible **« Non »**.
- **2/5** des exemples ont pour cible **« Oui »**.

•Comme **5/14** de l'exemple a suivi cette branche depuis la racine, on calcule :

- **(5/14) × (3/5) = 3/14** atteignant l'étiquette **« Non »**.
- **(5/14) × (2/5) = 1/7** atteignant l'étiquette **« Oui »**.

Attribution de la classe en fonction des VAM

•L'attribut **« Vent »** ayant la valeur **« Faible »**, les **5/14** de l'exemple ayant suivi cette branche depuis la racine sont classés comme **« Oui »**.

Résumé :

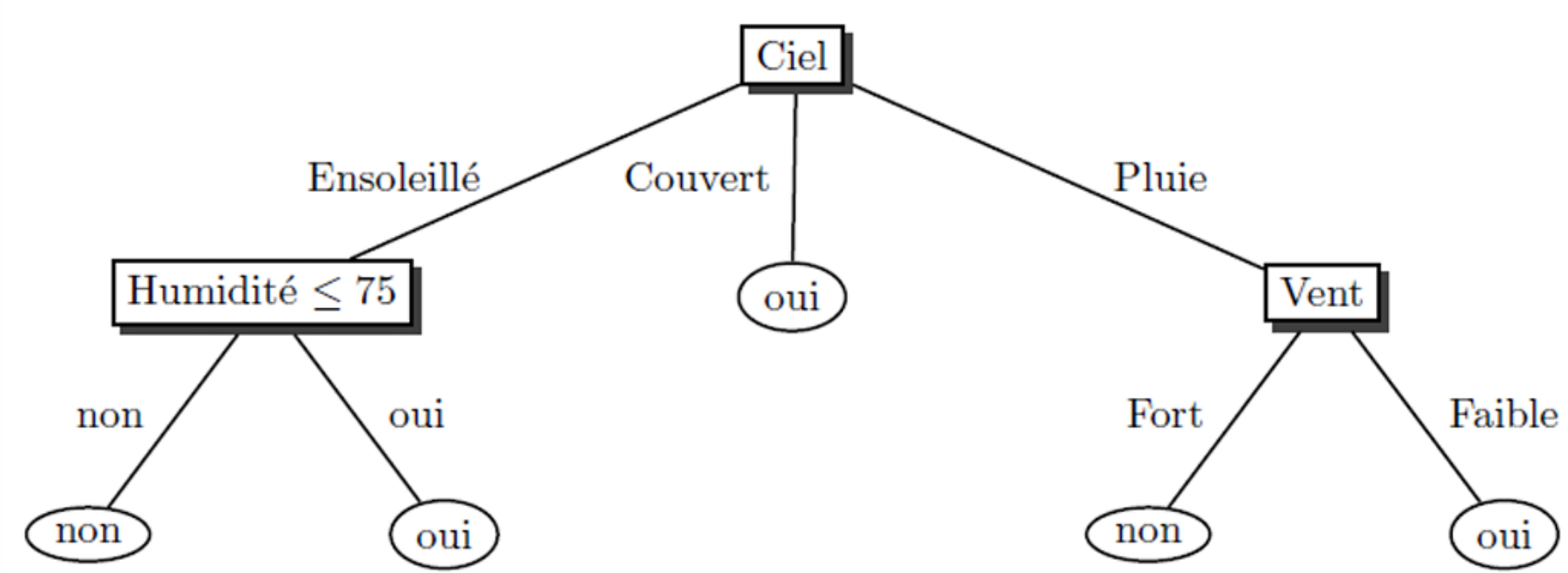
•**3/14** des exemples atteignent l'étiquette **« Non »**.

•**1/7 + 4/14 + 5/14 = 11/14** des exemples atteignent l'étiquette **« Oui »**.

Conclusion : La classe la plus probable pour la donnée est **« Oui »**.

Gestion des valeurs d'attributs manquantes (VAM)

(Ciel =?, Température = Tiède, Humidité =?, Vent = Faible)?



Validation d’un classificateur

Validation d’un arbre de décision

- Une fois un arbre de décision construit, il est crucial de le valider en estimant la probabilité que la classe prédite pour une donnée soit correcte. Cette probabilité est une variable aléatoire dont il faut estimer la valeur.
- L'erreur de classification **E** d'un classificateur correspond à la probabilité qu'il ne prévoie pas correctement la classe d'une donnée appartenant à **D**. Le taux de succès est alors **1 - E**.
- L'erreur d’apprentissage $\mathbf{E_{app}}$ est évaluée à l’aide des exemples d’apprentissage $\mathbf{D_{app}}$: elle correspond à la proportion d’exemples dont la classe est mal prédite par le classificateur.

Validation d’un arbre de décision

– Remarques

- ✓**E_{app}** n’est pas un bon estimateur de l’erreur pouvant être commise sur de nouvelles données.
- ✓L’apprentissage doit être capable de se généraliser à des données inédites : c’est l’objectif principal !
- ✓Un algorithme d’apprentissage est considéré comme efficace uniquement s’il parvient à appliquer correctement ce qu’il a appris à de nouvelles données.

Validation d’un arbre de décision

Pour évaluer la performance d’un arbre de décision (AD), on distingue :

- **Le jeu d’apprentissage (D_{app})** : ensemble d’exemples utilisés pour construire l’AD.
- **Le jeu de test (Dtest)** : ensemble d’exemples dont les classes sont connues. On applique l’AD construit avec D_{app} pour vérifier si la classification est correcte.
- **En l’absence d’un jeu de test**, on divise D_{app} en deux sous-ensembles :
  - Une partie pour l’apprentissage.
  - Une partie pour l’évaluation .

Mesure de la qualité d’un classificateur

Une fois un classificateur construit à partir d’un jeu d’apprentissage **D_{app}**, il est nécessaire d’évaluer sa performance.

Dans le cas d’une classification binaire, on définit :

- **VP (Vrais Positifs)** : nombre d’exemples de classe positive correctement classés comme positifs.
- **VN (Vrais Négatifs)** : nombre d’exemples de classe négative correctement classés comme négatifs.

Mesure de la qualité d'un classificateur

- **FP (Faux Positifs)** : nombre d'exemples de classe négative incorrectement classés comme positifs.
- **FN (Faux Négatifs)** : nombre d'exemples de classe positive incorrectement classés comme négatifs.

Ces valeurs sont généralement organisées dans un **tableau de contingence**, qui, pour **K > 2 classes**, est appelé **matrice de confusion**.

	+	− ← classe prédite
+	VP	FN
−	FP	VN
↑ classe		

Mesure de la qualité d'un classificateur

- Si seuls les éléments de la **diagonale principale** de la matrice de confusion sont non nuls, alors **tous les exemples sont correctement classés** (classification parfaite).
- Deux mesures essentielles pour évaluer un classificateur sont la **précision** et le **rappel** :
  - **Précision des positifs** = $\frac{vp}{vp + fp}$
  - **Précision des négatifs** = $\frac{vn}{vn + fn}$
  - **Rappel des positifs** = $\frac{vp}{vp + fn}$
  - **Rappel des négatifs** = $\frac{vn}{vn + fp}$

Mesure de la qualité d’un classificateur

- **Précision** : mesure la proportion d'exemples réellement positifs (ou négatifs) parmi ceux classés comme tels.
- **Rappel** : mesure la proportion d'exemples réellement positifs (ou négatifs) parmi l'ensemble des exemples de cette classe.
- **F-mesure (F1-score)** : il est souvent plus pratique d'utiliser une seule métrique synthétique. La **F-mesure** est définie par :

$$F = \frac{2 \times rappel \times précision}{rappel + précision} = \frac{2 \times vp}{2 \times vp + fp + fn}$$

Exercice : Évaluation d'un classificateur binaire

- Soit un classificateur binaire tel que le nombre de l'ensemble de test N = 100, le nombre d'éléments de la classe $\oplus = 60$, le nombre d'éléments de la classe $\ominus = 40$.

  - Résultats de classification pour la classe $\oplus$:
    - $VP = 40 \oplus$
    - FN = 20 $\ominus$
  - Résultats de classification pour la classe $\ominus$:
    - VN = 30 $\ominus$
    - FP = 10 $\oplus$

Précision = $\frac{vp}{vp+fp} = 0.8$

Rappel = $\frac{vp}{vp+fn} = 0.67$

- Calculer les mesures de qualité suivantes: Précision, Rappel, et F.

$$F = \frac{2 \times rappel \times précision}{rappel + précision} = \frac{2 \times vp}{2 \times vp + fp + fn} = 0.72$$

Validation croisée

- Est une méthode plus avancée pour évaluer la performance d'un classificateur.
- Elle consiste à diviser l'ensemble des exemples en n sous-ensembles mutuellement exclusifs.
- Il est important de veiller à ce que chaque classe soit représentée avec la même fréquence dans les n sous-ensembles.
  - Par exemple, si n=3, on obtient trois sous-ensembles A, B et C. On entraîne alors l'arbre de décision AD_{AUB} sur l'union des ensembles A et B, puis on évalue son taux d'erreur sur C, noté E_C, qui correspond au nombre d'exemples de C mal classés par AD_{AUB}.

Validation croisée

•Ensuite, on entraîne l'arbre de décision AD_{CUB}, puis on évalue son taux d'erreur sur A, noté E_A.
•De même, on construit l'arbre AD_{AUC} en utilisant A et C pour l'apprentissage, puis on mesure le taux d'erreur sur B, noté E_B.
•Enfin, le taux d'erreur global E est estimé en prenant la moyenne des trois erreurs obtenues :

$$E = \frac{E_A + E_B + E_C}{3}$$

Le Sur-apprentissage dans les Arbres de Décision

Lors de l'entraînement d'un arbre de décision (*AD*), on peut analyser l'évolution de deux types d'erreurs :

- **L'erreur d'apprentissage (E_{app})** : correspond au taux d'erreur sur les données utilisées pour entraîner le modèle.
- **L'erreur réelle (E)** : estimerait l'erreur que le modèle ferait sur de nouvelles données.

Lorsque le nombre d'exemples d'entraînement augmente :

- E_{app} diminue continuellement, car l'arbre s'adapte de mieux en mieux aux exemples fournis.
- E diminue au début, mais finit par se stabiliser, voire augmenter si le **modèle devient trop complexe**.

Exemple : Dans un problème de classification comme *"Jouer au tennis ?"*, si on construit un arbre de décision avec un nombre croissant d'exemples (1, 2, 3... jusqu'à 14), on observera que E_{app} diminue continuellement, tandis que *E* atteint un minimum avant de remonter.

Sur-apprentissage

- Si un arbre de décision devient trop complexe en essayant de s'adapter parfaitement aux données d'entraînement, $\boldsymbol{E_{app}}$ peut atteindre **0**. Cela signifie que le modèle classe **parfaitement** tous les exemples connus.
- Cependant, cela ne garantit pas une bonne généralisation, car si les données contiennent des incohérences (exemple : deux instances ayant les mêmes caractéristiques mais appartenant à des classes différentes), il est impossible de construire un arbre parfait. Dans ce cas, $\boldsymbol{E_{app}}$ reste légèrement supérieur à 0.
- Le véritable critère pour mesurer la qualité du modèle est *E* :
  - Tant que ***E* diminue**, le modèle s'améliore.
  - Dès que **E commence à augmenter**, cela signifie que l'arbre est devenu trop spécifique aux données d'apprentissage et perd en généralisation.

Impact du sur-apprentissage sur un modèle

- Un modèle **optimal** est obtenu au moment où ***E*** est au **plus bas**.
- Si l'apprentissage continue au-delà de ce point, le modèle devient **trop complexe** et commence à mémoriser les spécificités des exemples au lieu de capturer les tendances générales.
- Ce phénomène, appelé **sur-apprentissage**, entraîne :
  - Une **perte de capacité de généralisation**, car le modèle est trop ajusté aux données d'entraînement.
  - Une **augmentation de la probabilité d'erreur** sur de nouvelles données.
  - Un manque de **recul** du modèle par rapport aux données, ce qui réduit sa robustesse.

L'Élagage dans les Arbres de Décision

L'élagage est une technique utilisée pour **réduire la complexité** d'un arbre de décision (*AD*) en supprimant certaines branches. Il a deux objectifs principaux :

1. **Simplifier l'arbre de décision** pour améliorer sa lisibilité et son efficacité.
2. **Réduire le sur-apprentissage** afin d'augmenter la capacité du modèle à généraliser sur de nouvelles données, ce qui permet de **réduire le taux d'erreur**.

Il existe **deux types d'élagage** :

- **Élagage lors de la construction** : on arrête la croissance de l'arbre avant qu'il ne devienne trop complexe.
- **Élagage après la construction** (comme dans l'algorithme **C4.5**) : l'arbre est d'abord entièrement construit, puis certaines branches sont supprimées pour améliorer la généralisation.

Techniques d'Élagage dans C4.5

L'algorithme **C4.5** propose deux approches d'élagage après construction :

1. **Remplacement d'un sous-arbre** : une sous-partie de l'arbre est remplacée par une feuille unique.
2. **Promotion d'un sous-arbre** : plusieurs nœuds sont fusionnés en un seul nœud, simplifiant ainsi la structure.

Dans le premier cas (remplacement d'un sous-arbre) :

- On examine les nœuds en remontant depuis les feuilles jusqu'à la racine.
- À chaque nœud, un test est effectué pour décider s'il doit être remplacé par une feuille.

Processus de Décision pour l'Élagage

La décision de transformer un nœud est basée sur le **taux d'erreur** estimé du nœud et de ses fils.

1. **Estimation des taux d'erreur** :
    1. L'algorithme **C4.5** estime l'erreur de chaque nœud fils.
    2. Il pondère ces erreurs en fonction du nombre d'exemples couverts par chaque fils.
2. **Comparaison des taux d'erreur** :
    1. Si le taux d'erreur global des fils est **supérieur** à celui du nœud parent, alors le nœud parent est **remplacé par une feuille**.

Exemple d'Élagage

Le taux d'erreur global des fils est calculé comme suit :

Fils	Nombre d'exemples	Nombre d'exemples mal classés	Taux d'erreur
1er fils	6	2	0.47
2ème fils	2	1	0.72
3ème fils	6	2	0.47

Le taux d'erreur global des fils est calculé comme suit :

$$\frac{6}{14} \times 0.47 + \frac{2}{14} \times 0.72 + \frac{6}{14} \times 0.47 = 0.504$$

Le taux d'erreur du nœud parent est évalué avec les **14 exemples** dont **5 sont mal classés** :

$$\frac{5}{14} = 0.46$$

Puisque **0.46 < 0.504**, **C4.5 remplace ce nœud par une feuille** étiquetée avec la classe majoritaire parmi les **14 exemples**.

Référence

- The StatQuest Illustrated Guide to Machine Learning!!!, By Josh Starmer, Ph.D
- Studies in Computational Intelligence Volume 498 Series Editor J. Kacprzyk, Warsaw, Poland
- Machine Learning A Probabilistic Perspective Kevin P. Murphy
- Christopher M. Bishop Pattern Recognition and Machine Learning
- Cours Aissa Boulmerka, Associate Professor, The National School of Artificial Intelligence, Algeria