** Classification des livres pour la recommandation **

** **

**** Dyhia****

** préparée à partir de 13/10/2023**

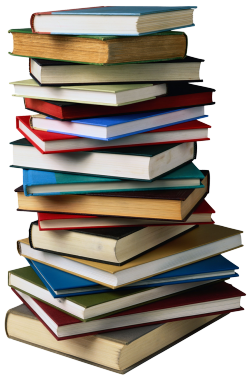


Figure 1: Recommandation de livres

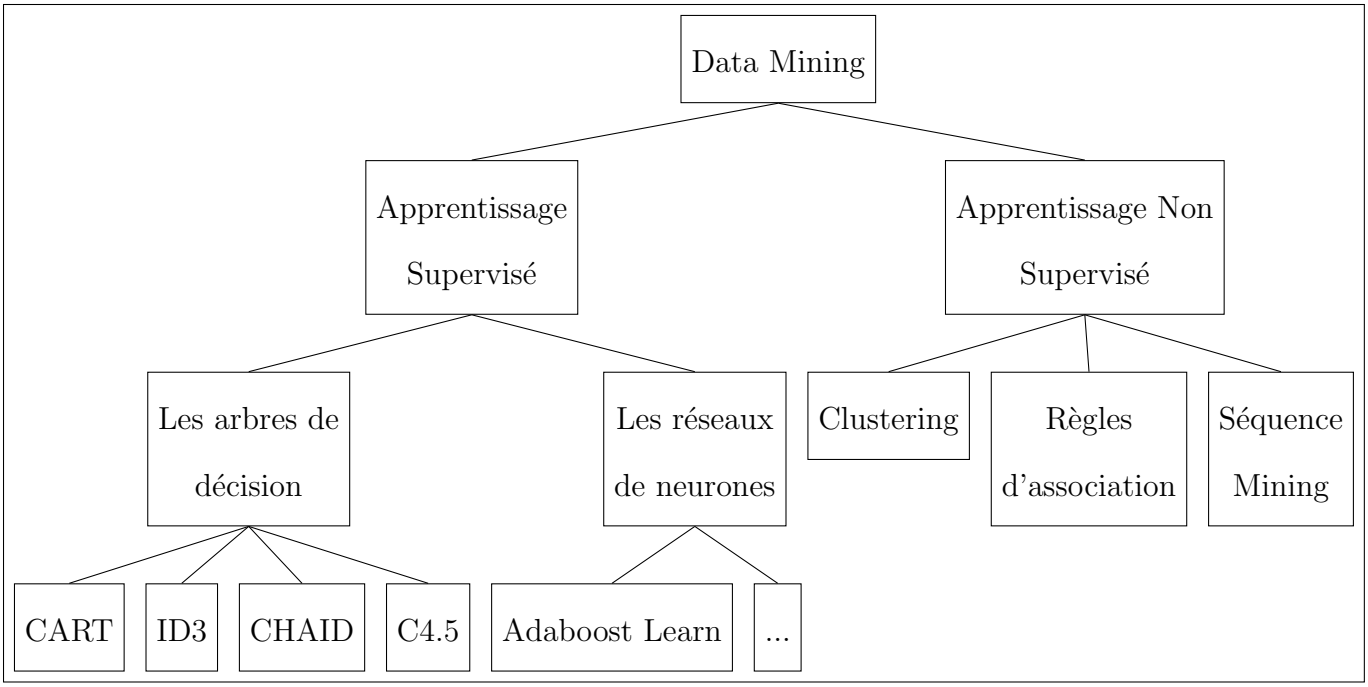
Figure 2: Arbre de décisions

1 Les méthodes d'apprentissages

1.1 Introduction

Le DATAMINING est un ensemble de techniques et de méthodes du domaine des statistiques, mathématiques et d'informatiques qui permettent d'extraire des informations précises d'une grande quantité de données (grandes BDDs). Il représente aussi un procédé essentiel pour l'aide à la décision et un moyen efficace de développement. Parmi les outils utilisés dans ce processus on trouve les arbre de décision.

1.2 Types d'apprentissage



1.3 Les arbres de décisions

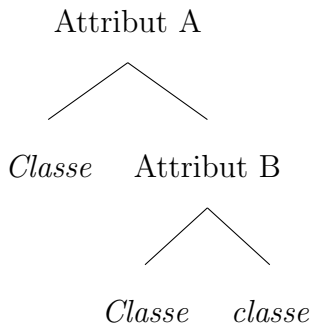
1.3.1 Définition:

Ensemble de règles de classification basant leur décision sur des tests associées au attributs organisés de manière arborescente.

1.3.2 Domaines d'applications:

- La fouille de données
- la médcine
- L'astronomie
- La météorologie

1.4 structure d'un arbre de décision



1.4.1 Types d'arbres de décision:

1. Les arbres de régressions: La variable à expliquer est quantitative, en fonction de variables explicatives quantitatives et /ou qualitatives.
2. Les arbres de classification: La variable à expliquer est qualitative, en fonction de variables explicatives quantitatives et /ou qualitatives.

1.4.2 Apprentissage d'arbre de décision:

Les trois étapes essentielles sont:

- Décider si un nœud est terminal.

- Si un nœud n'est pas terminale, lui associe un test.
- Si un nœud est terminal, lui affecter une classe.

1.5 Algorithme d'apprentissage générique:

Entrée : échantillon S
Debut: Initialiser l'arbre courant à l'arbre vide:
La racine est le nœud courant;
repeat
  décider si le nœud courant est terminal
  if *le nœud est terminal* **then**
    Lui affecter une classe
  end
  else
    sélectionner un test et créer autant de nouveaux nœud fils qu'il y a de réponses possibles au test
  end
until *Obtenir l'arbre de décision;*;
End

1.5.1 Algorithmes d'arbres de décision:

- ID3
- C4.5
- CHAID
- CART

1.5.2 Avantages et Inconvénients des arbres de décision:

- Les avantages:
  1. La rapidité.
  2. Facile à comprendre.

- Les inconvénients:

  1. Problème de stabilité.
  2. Peu de performance lorsqu'il y a beaucoup de classe.

2 CART

2.1 Définition:

CART (Classification And Regression Tree) est un algorithme d'apprentissage automatique, développé en 1984 par Breiman, Friedman, Olshen et Stone qui génère un arbre de décision binaire et offre la possibilité de:

- Manipuler les valeurs qualitative et quantitative.
- Gérer les attributs avec des valeurs manquantes, nulles.
- Elaguer l'arbre.

2.2 Indice de Gini:

$$Gini(s) : 1 - \sum_j^x freq(c_j, S)^2$$

CrièreGini(P)=$Gini(p) - (p_{Gauche} x Gini(P1) + P_{droite} x Gini(P2))$

- Critère Gini(P):critère Gini de l'attribut à la position P
- P (gauche) : proportion des élément s dans le jeux d'apprentissage associé à P qui vont au (Nœud P1).
- P (droite) : proportion des élément s dans le jeux d'apprentissage associé à P qui vont au (Nœud P2).

- C : Classe.
- S : échantillon à tester.

2.2.1 Algorithme de construction d'un arbre CART:

Phase d'expansion: on dispose d'un échantillon S découpé en un ensemble d'apprentissage A et un ensemble de test T.

- Entrée: ensemble d'apprentissage A
- On utilise la fonction Gini
- Décider si un nœud est terminal : Un nœud à la position P est terminal si $Gini(p) \leq i_0$ ou $N(p) \leq n_0$ où i_0 et n_0 sont des paramètres à fixer.
- Sélectionner un test à associer pour un nœud : On choisit le test qui maximise$\delta(p, T)$,où P est une position, T un test, P_{Gauche} (ou P_{droite}) la proportion d'éléments qui vont sur la position P1 (ou P2).

  $\delta(p, T) = Gini(p)(PgGini(p1) + PdGini(p2))$
- Affecter une classe à une feuille : on choisit la classe majoritaire.
- Sortie: un arbre de décision.

2.2.2 Phase de l'élagage:

- **Entrée** : l'arbre de décision obtenu dans la phase d'expansion
- On construit une suite d'arbres $t_0, t_1, t_2, \ldots t_k$
- On calcule pour chaque t_j l'erreur apparente sur l'ensemble T.

- La suite est donné par:
  - t_0 est l'arbre obtenu dans la phase d'expansion.
  - t_k est une feuille
  - À l'étape t_i : pour toute position P de t_i on calcule g(p) et on choisit la position P qui minimise g(p).
  - L'arbre $t_i + 1$ est un élagué de $t_i$en position P.
- **Sortie:** l'arbre de la suite dont l'erreur apparente est minimale.

2.2.3 Les avantages et les inconvénients:

Les avantages:

- Simple à comprendre, interpréter, visualiser.
- Peu d'effort pour la préparation des données.

Les inconvénients:

- Binarisation pas toujours approprié.
- les arbres de décision peuvent être instables car y a de petites variations dans les données.

3 CART

3.1 Définition:

CART (Classification And Regression Tree) est un algorithme d'apprentissage automatique, développé en 1984 par Breiman, Friedman, Olshen et Stone qui génère un arbre de décision binaire et offre la possibilité de:

- Manipuler les valeurs qualitative et quantitative.
- Gérer les attributs avec des valeurs manquantes, nulles.
- Elaguer l'arbre.

3.2 Indice de Gini:

$$Gini(s) : 1 - \sum_j^x freq(c_j, S)^2$$

$$CrièreGini(P) = Gini(p) - (p_{Gauche} x Gini(P1) + P_{droite} x Gini(P2))$$

- $CritèreGini(P)$:critère Gini de l'attribut à la position P
- $P(gauche)$: proportion des élément s dans le jeux d'apprentissage associé à P qui vont au (Nœud P1).
- $P(droite)$: proportion des élément s dans le jeux d'apprentissage associé à P qui vont au (Nœud P2).
- C : Classe.
- S : échantillon à tester.

3.2.1 Algorithme de construction d'un arbre CART:

Phase d'expansion: on dispose d'un échantillon S découpé en un ensemble d'apprentissage A et un ensemble de test T.

- Entrée: ensemble d'apprentissage A
- On utilise la fonction Gini

- Décider si un nœud est terminal : Un nœud à la position P est terminal si $Gini(p) \leq i_0$ ou $N(p) \leq n_0$ où i_0 et n_0 sont des paramètres à fixer.
- Sélectionner un test à associer pour un nœud : On choisit le test qui maximise$\delta(p, T)$,où P est une position, T un test, P_{Gauche} (ou P_{droite}) la proportion d'éléments qui vont sur la position P1 (ou P2).

  $\delta(p, T) = Gini(p)(PgGini(p1) + PdGini(p2))$
- Affecter une classe à une feuille : on choisit la classe majoritaire.
- Sortie: un arbre de décision.

3.2.2 Phase de l'élagage:

- **Entrée** : l'arbre de décision obtenu dans la phase d'expansion
- On construit une suite d'arbres $t_0, t_1, t_2, \ldots t_k$
- On calcule pour chaque t_j l'erreur apparente sur l'ensemble T.
- La suite est donné par:
  - t_0 est l'arbre obtenu dans la phase d'expansion.
  - t_k est une feuille
  - À l'étape t_i : pour toute position P de t_i on calcule g(p) et on choisit la position P qui minimise g(p).
  - L'arbre $t_i + 1$ est un élagué de $t_i$en position P.
- **Sortie:** l'arbre de la suite dont l'erreur apparente est minimale.

3.2.3 Les avantages et les inconvénients:

Les avantages:

- Simple à comprendre, interpréter, visualiser.
- Peu d'effort pour la préparation des données.

Les inconvénients:

- Binarisation pas toujours approprié.
- les arbres de décision peuvent être instables car y a de petites variations dans les données.

4 Gestion des données manquantes:

Les données manquantes posent plusieurs problèmes pour construire un arbre de décision.

Solution CART utilise une procédure de variables substituts pour s'affranchir du problème.

4.1 Exemple applicatif:

Exemple d'arbre de classification

Ce jeu d'apprentissage décrit les conditions de météo pour savoir si les conditions sont idéales pour aller jouer au Golf

Problématique: Peut-on aller jouer au Golf ou pas ? Soit le tableau (ensemble S de prévisions météorologiques):

Jeux d'apprentissage					
Exemple	Condition	Température	Humidité	Vent	Classe
x1	Soleil	Chaud	Élevée	Faible	Non
x2	Soleil	Chaud	Élevée	Fort	Non
x3	Nuage	Chaud	Élevée	Faible	Oui
x4	Pluie	Doux	Élevée	Faible	Oui
x5	Pluie	Froid	Normale	Faible	Oui
x6	Pluie	Froid	Normale	Fort	Non
x7	Nuage	Froid	Normale	Fort	Oui
x8	Soleil	Doux	Élevée	Faible	Non
x9	Soleil	Froid	Normale	Faible	Oui
x10	Pluie	Doux	Normale	Faible	Oui
x11	Soleil	Doux	Normale	Fort	Oui
x12	Nuage	Doux	Élevée	Fort	Oui
x13	Nuage	Chaud	Normale	Faible	Oui
x14	Pluie	Doux	Élevée	Fort	Non

Table 1: Exemple d'un jeu d'apprentissage

On calcule l'index de Gini par rapport à l'objectif de classification :
$1 - ((9/14)^2 + (5/14)^2) = 0.459$

Déterminer le test pour l'attribut Condition

Test attribut condition		
Ensemble gauche	Ensemble droit	Index Gini pour l'attribut condition
Soleil	Nuage ou Pluie	0.065
Nuage	Soleil ou pluie	0,102(la valeur qui maximise l'indice de Gini)
Pluie	Soleil ou nuage	0,002

Table 2: Choix de répartition pour le nœud Condition.

Déterminer le test pour l'attribut Température :

Test attribut Température		
Ensemble gauche	Ensemble droit	Index Gini pour l'attribut condition
Chaud	Doux ou froid	0.0163
Doux	Chaud ou froid	0,0008
Froid	Doux ou chaud	0,0091

Table 3: Choix de répartition pour le noeud Humidité

Déterminer le test pour les attributs Humidité et Vent:

Test attribut Humidité et vent			
Attribut	Ensemble gauche	Ensemble droit	Index de gini
Humidité	Élevée	Normale	0.091
Vent	Faible	Fort	0.031

Table 4: Gini d'index pour l'attribut humidité et vent

On choisit l'attribut qui maximise l'indice de Gini

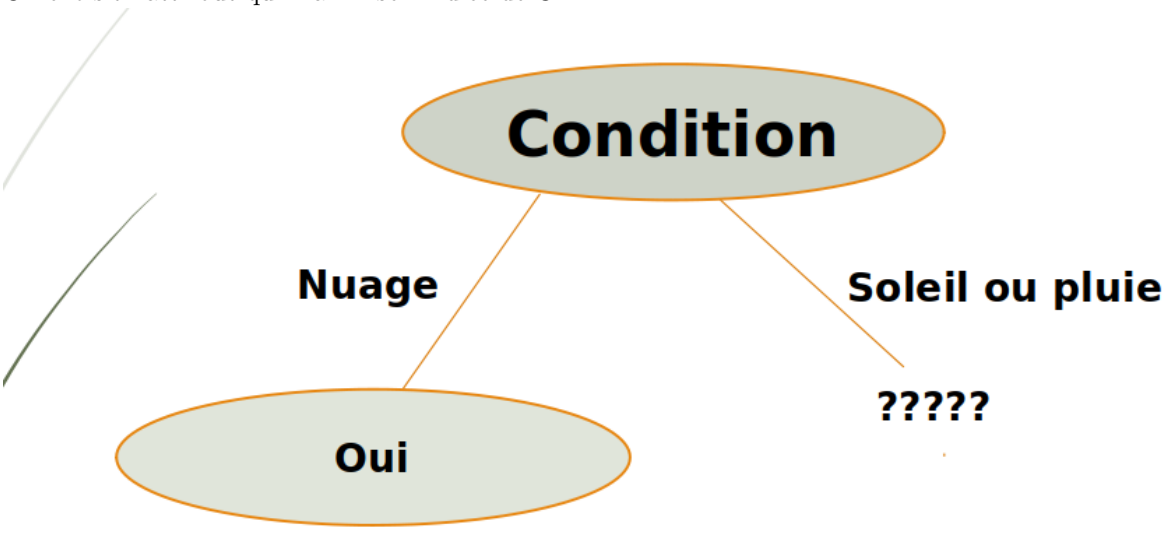


Figure 3: ** Premier niveau de l'arbre CART **

On calcule l'index Gini par rapport aux classes de table5

$$1 - ((5/10)^2 + (5/10)^2) = 0.5$$

On Choisit comme attribut Humidité(maximise la fonction de gini)

On choisit Température comme racine du noeud

Jeux d'apprentissage					
Exemple	Condition	Température	Humidité	Vent	Classe
x1	Soleil	Chaud	Élevée	Faible	Non
x2	Soleil	Chaud	Élevée	Fort	Non
x3	Nuage	Chaud	Élevée	Faible	Oui
x4	Pluie	Doux	Élevée	Faible	Oui
x5	Pluie	Froid	Normale	Faible	Oui
x6	Pluie	Froid	Normale	Fort	Non
x8	Soleil	Doux	Élevée	Faible	Non
x9	Soleil	Froid	Normale	Faible	Oui
x10	Pluie	Doux	Normale	Faible	Oui
x11	Soleil	Doux	Normale	Fort	Oui
x13	Nuage	Chaud	Normale	Faible	Oui
x14	Pluie	Doux	Élevée	Fort	Non

Table 5: Jeu d'apprentissage avec 'soleil ou pluie

Déterminer le test pour l'attribut Température

Test attribut condition		
Ensemble gauche	Ensemble droit	Index Gini pour l'attribut Température
Chaud	Doux ou froid	0.125
Doux	Chaud ou froid	0,020
Froid	Doux ou chaud	0,023

Table 6: Choix de répartition pour le nœud température

Déterminer le test pour les attributs Humidité et Vent

Test attribut Humidité et vent			
Attribut	Ensemble gauche	Ensemble droit	Index de gini
Humidité	Élevée	Normale	0.180
Vent	Faible	Fort	0.083

Table 7: Calculde Gini humidité et vent

pour la branche gauche Humidité= Élevée, les attributs restants sont Température et vent

Jeux d'apprentissage					
Exemple	Condition	Température	Humidité	Vent	Classe
x1	Soleil	Chaud	Élevée	Faible	Non
x2	Soleil	Chaud	Élevée	Fort	Non
x4	Pluie	Doux	Élevée	Faible	Oui
x8	Soleil	Doux	Élevée	Faible	Non
x14	Pluie	Doux	Élevée	Fort	Non

Table 8: Jeu d'apprentissage avec Humidité = 'Elevée

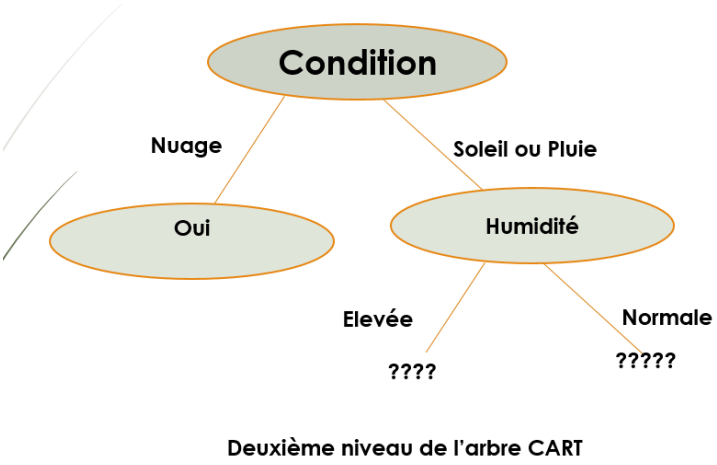


Figure 4: Deuxième niveau de l'arbre CART

On calcule l'index Gini par rapport aux classes de la table 8
$1 - ((4/5)^2 + (1/5)^2) = 0.32$
Les deux attributs ont seulement 2 valeurs:

Calcul de l'index de Gini			
Attribut	Ensemble gauche	Ensemble droit	Index de gini
Température	Chaud	Doux	0.053
Vent	Faible	Fort	0.053

Table 9: Calcul de l'index Gini

Pour la branche droite Température = Doux, il reste seulement l'attribut Vent

Jeux d'apprentissage					
Exemple	Condition	Température	Humidité	Vent	Classe
x4	Pluie	Doux	Élevée	Faible	Oui
x8	Soleil	Doux	Élevée	Faible	Non
x14	Pluie	Doux	Élevée	Fort	Non

Table 10: Jeu d'apprentissage pour la Température = 'Doux'

On calcule l'index Gini par rapport aux classes de la table 10:
$1 - ((2/3)^2 + (1/3)^2) = 0.44$

Calcul de l'index de Gini			
Attribut	Ensemble gauche	Ensemble droit	Index de gini
Vent	Faible	Fort	0.111

Table 11: Calcul l'indice Gini pour la branche Température = 'Doux'

On choisit Température comme racine du noeud

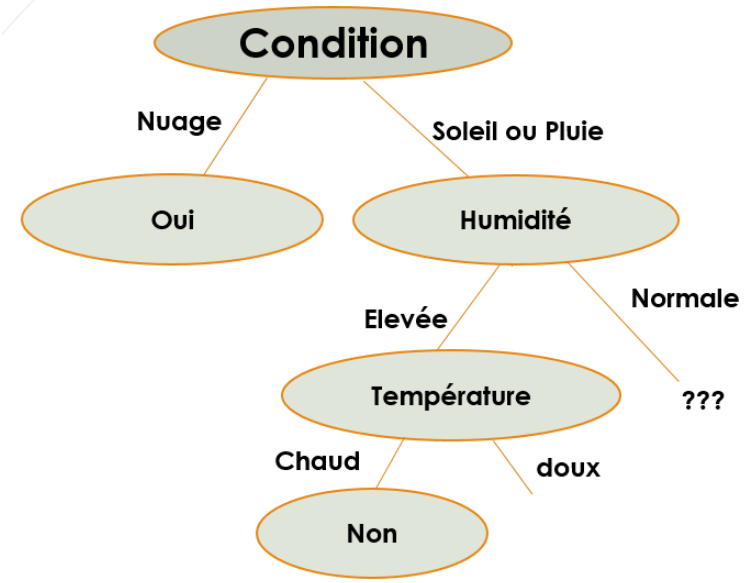


Figure 5: ** Troisième niveau de l'arbre CART **

Il reste 2 attributs Température et Vent:

Jeux d'apprentissage					
Exemple	Condition	Température	Humidité	Vent	Classe
x5	Pluie	Froid	Normale	Faible	Oui
x6	Pluie	Froid	Normale	Fort	Non
x9	Soleil	Froid	Normale	Faible	Oui
x10	Pluie	Doux	Normale	Faible	Oui
x11	Soleil	Doux	Normale	Fort	Oui

Table 12: Jeu d'apprentissage pour Humidité = 'Normale'

On calcule L'index de Gini par rapport aux classes au jeu de la table 12
$1 - ((4/5)^2 + (1/5)^2) = 0.32$
Les deux attributs ont seulement 2 valeurs:

Calcul de l'index de Gini			
Attribut	Ensemble gauche	Ensemble droit	Index de gini
Température	Froid	Doux	0.053
Vent	Faible	Fort	0.119

Table 13: Calcul de l'index Gini pour la branche Humidité = 'Normale'.

D'autres attributs

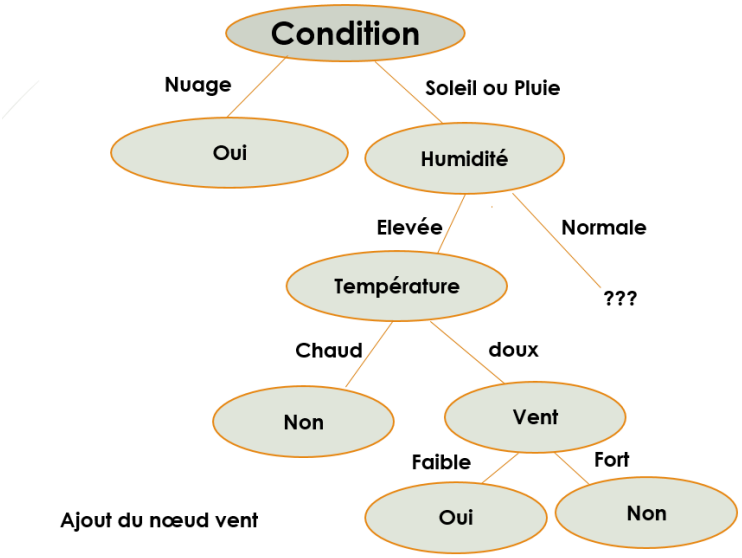


Figure 6: ** Quatrième niveau de l'arbre CART **

L'attribut Température a seulement 2 valeurs selon la table 14:

Jeux d'apprentissage					
Exemple	Condition	Température	Humidité	Vent	Classe
x6	Pluie	Froid	Normale	Fort	Non
x11	Soleil	Doux	Normale	Fort	Oui

Table 14: Jeu d'apprentissage pour la branche Vent = 'Fort

On calcule l'index Gini par rapport aux classes de la table 14
$1 - ((1/)^2 + (1/2)^2) =$
L'attribut température a seulement 2 valeurs selon la table 14

Calcul de l'index de Gini			
Attribut	Ensemble gauche	Ensemble droit	Index de gini
Température	Froid	Doux	0.5

Table 15: Calcul de l'index Gini pour la branche Vent = 'Fort'

On choisit l'attribut vent comme racine: La figure est comme suit:

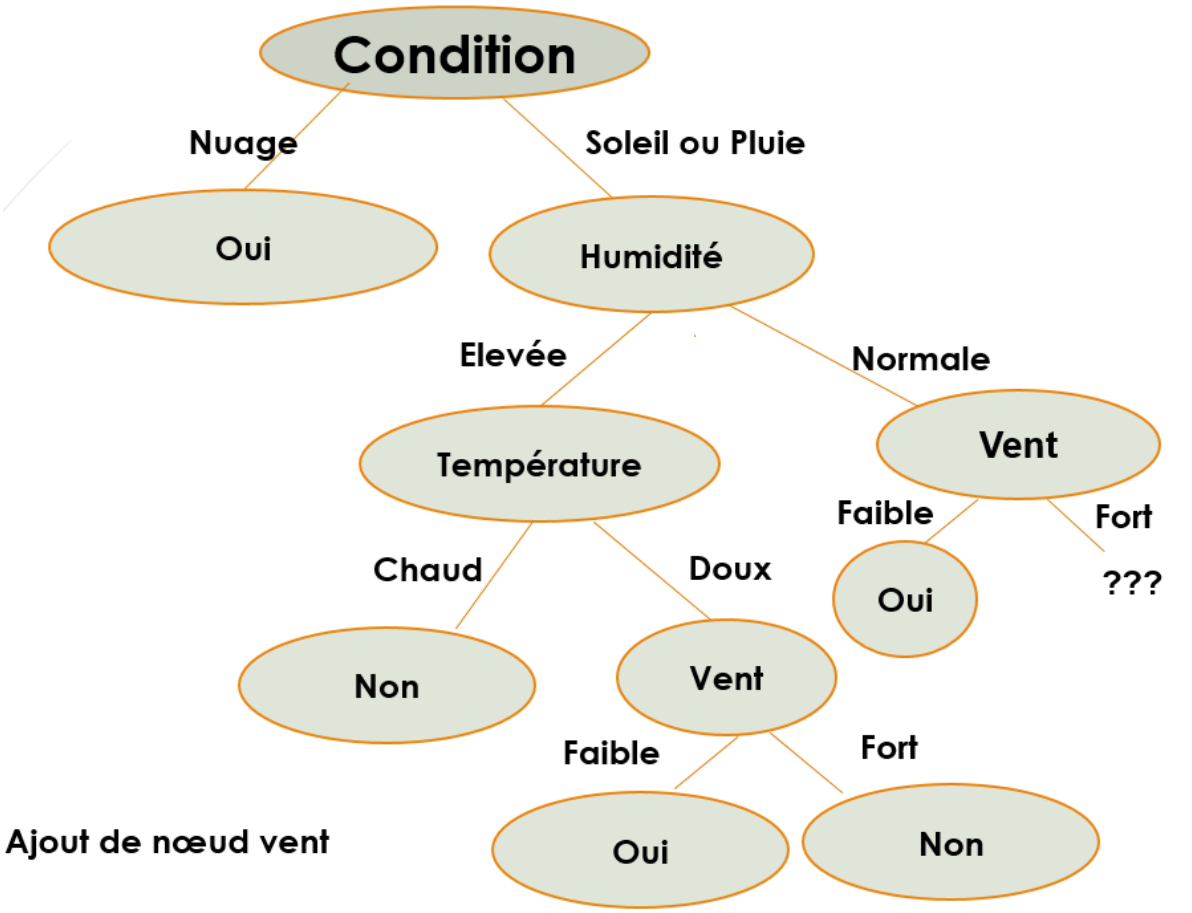


Figure 7: ** Ajout du noeud vent **

5 Partie élagage:

On utilise la formule suivante:

Critère (Tk,d)=$\frac{MC(d,k)-MCT(Tk)}{N(k)*(Nt(d,k)-1)}$

- MC(d,k):nombre d'exemples mal classés par le noeud d de l'arbre Tk quand on fait l'hypothèse qu'il a été transformé en feuille.

Pour les branches Température= Froid et Température= Doux, Il ne reste plus d'attribut à traiter, donc on se retrouve dans un cas de noeud terminal.

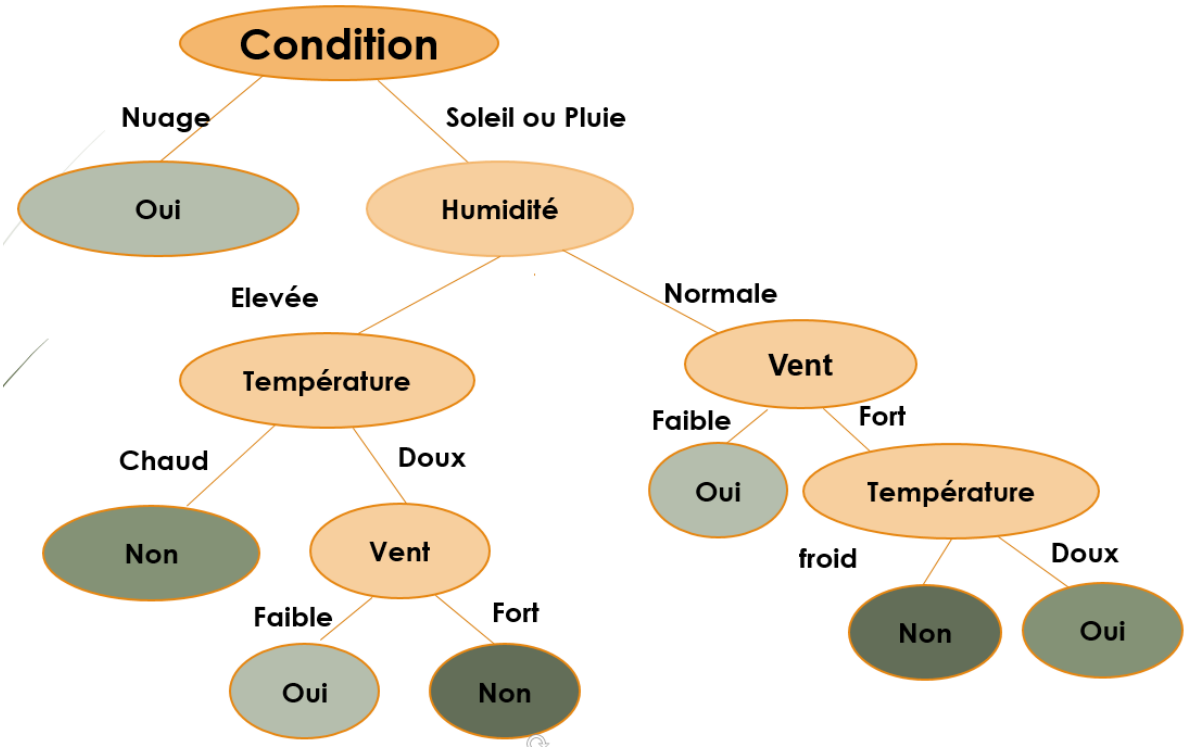


Figure 8: **Arbre de décision complet CART**

- MCT(Tk):nombre d'exemples mal classés par les feuilles du noeud Tk
- N(k): nombre de feuille de tk.
- Nt(d,k):nombre de feuilles du sous arbre situé sous le noeud d.

Le processus d'élagage se termine lorsque la racine de l'arbre est une feuille. Ensuite on choisit le sous arbre qui convient soit par échantillon test ou bien validation croisée.

Les avantages et les inconvénients:

Les avantages:

- Simple à comprendre, interpréter, visualiser.

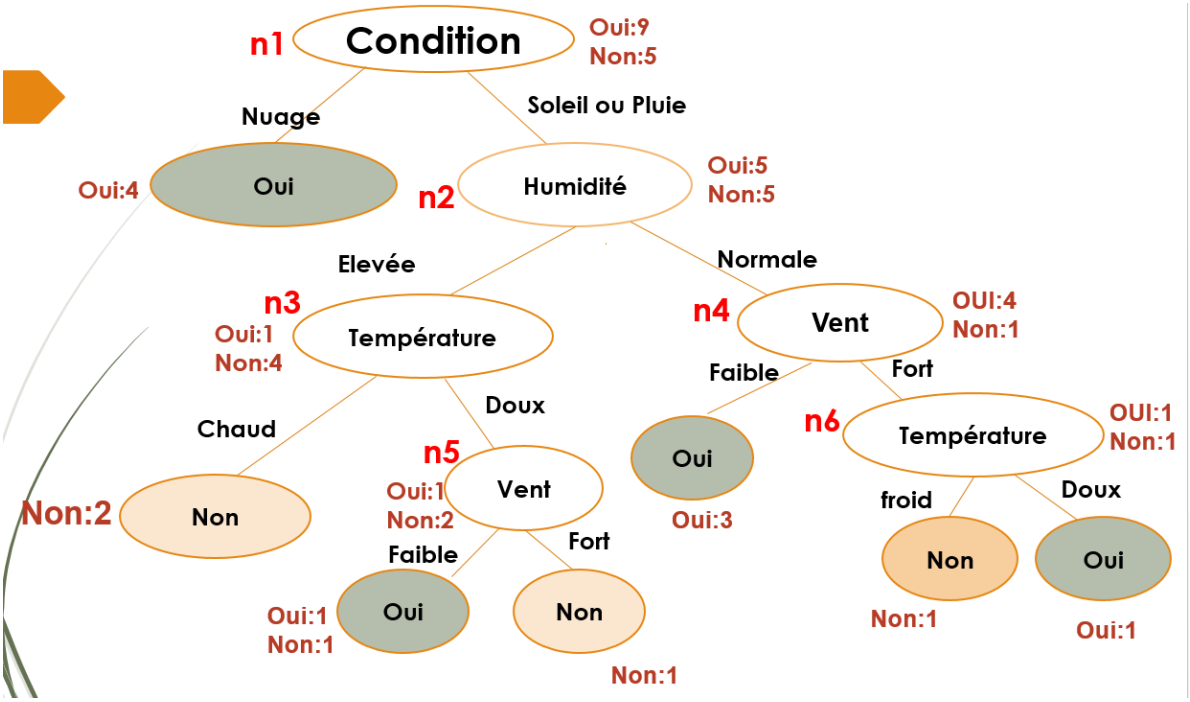


Figure 9: Arbre T0

On cherche à trouver le noeud qu'on va élaguer pour passer de l'arbre originaire T0 à T1. Il faut donc mesurer le critère d'élagage et de choisir le noeud qui minimesera la fonction.

- Peu d'effort pour la préparation des données.

5.1 Les inconvénients

- Binarisation pas toujours appropriée
- les arbres de décision peuvent être instables car y a de petites variations dans les données.

6 Conclusion

Cette méthode n'est pas conditionnée par des types ou structures de données particulières (ce qui fait leurs succès dans l'exploitation de mégadonnées). Elles conduisent à des segmen-

- 0.0095238 = Critère (To,n1) = 5-1/7*(7-1)
- 0.112857 = Critère ((To, n2)= 5-1/7 * (6-1)
- 0= Critère (To, n3) = 1-1/7* (3-1)
- 0.007142 = Critère (Tmax, n4) = 1-0/7 * (3-1)
- 0= Critère (Tmax, n5) = 1-1/7 * (2-1)
- 0.14 28714 = Critère (Tmax, n6) = 1-0/7 * (2-1)

Le noeud n3 est remplacé par une feuille, la classe attribuée à cette feuille est la classe qui représente la plus fréquente, soit Non

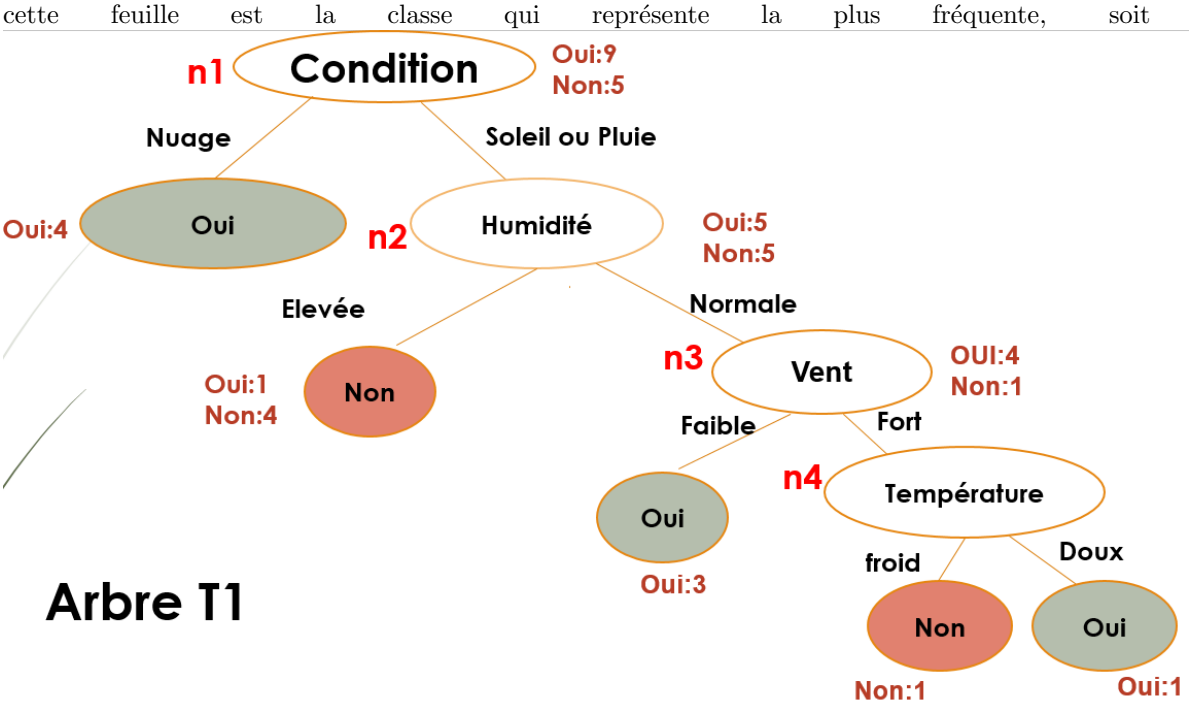


Figure 10: Arbre T1

tations invariantes aux transformations de variables pour lesquelles on dispose d'estimateurs du taux de mauvais classement. Enfin (et surtout?) elles produisent des résultats simples à interpréter et à utiliser.

- 0.5=Critère (T1, n1) = 5-1/5 * (5 -1).
- 0.266666 = Critère (T1, n2) = 5-1/5 *(4-1).
- 0.006666 = Critère (T1, n3) = 1-0/5 x (3-1).
- 0.1 =Critère (T1, n4) = 1-0/5 x (2-1).

On choisit d'élaguer le noeud n3 pour passer de T1 à T2. La feuille sera représentée par Oui:

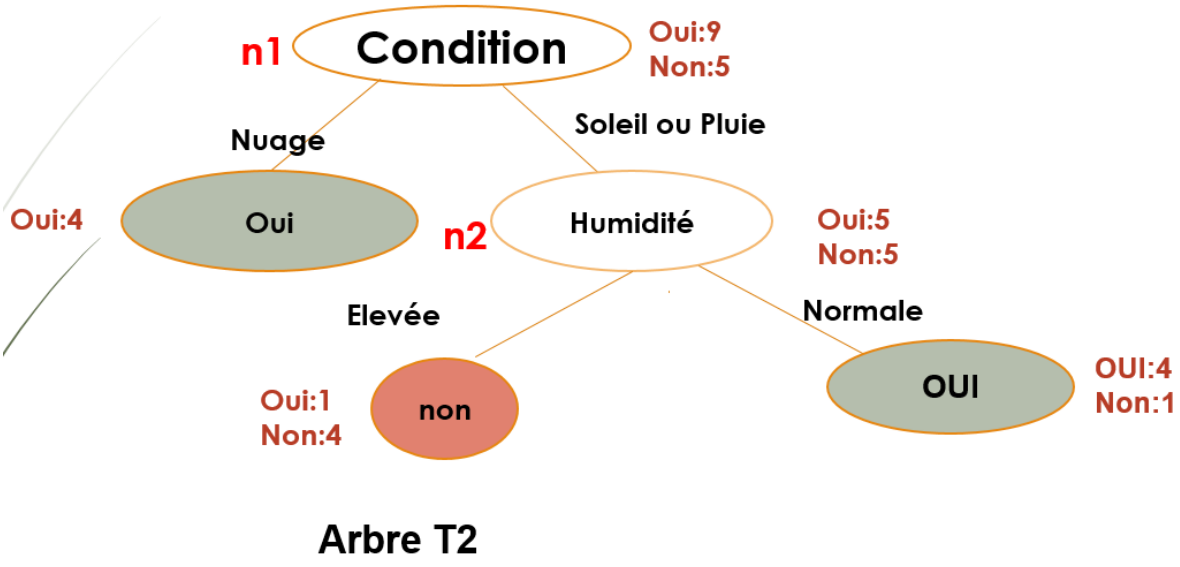


Figure 11: Arbre T2

On recommence le test:

- 0.5 = Critère (T2, n1)=5-1/3 * (3 -1)
- 1= Critère (T2, n2) =5-2/3 * (2-1)

On choisit d'élaguer le noeud n1 pour passer de T2 à T3:

Figure 12: Arbre T3