Arbre de décision

Dr. Ilham KADI

i.kadi@emsi.ma

2023/2024

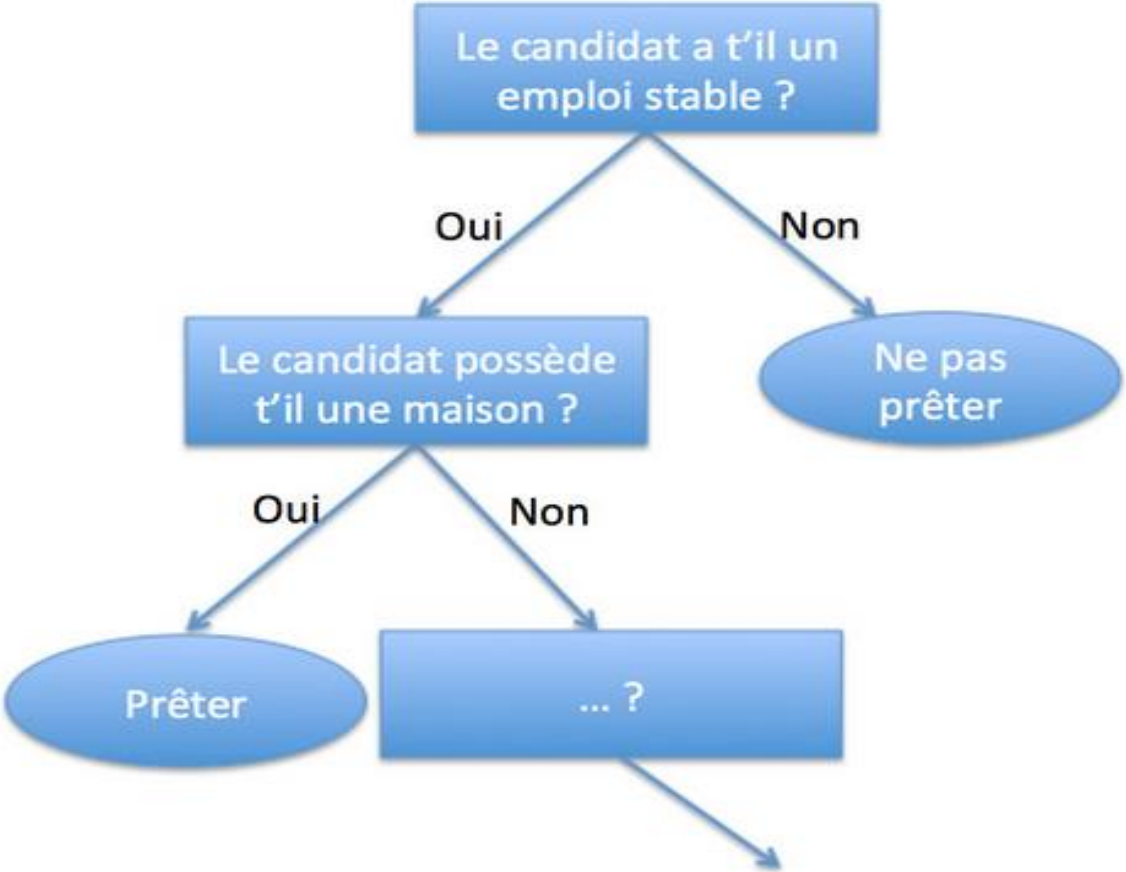
Arbre de décision: Définition

Les arbres de décision sont des règles de classification qui basent leur décision sur une suite de tests associés aux attributs, les tests étant organisés de manière arborescente.

Arbre de décision: Principe

Prédire la valeur d'un **attribut** (variable cible) à partir d'un **ensemble de valeurs d'attributs** (variables prédictives).

- Une méthode simple, supervisée, et très connue de classification et de prédiction.
- Un arbre est équivalent à un ensemble de règles de décision : un modèle facile à comprendre.



Exemple : Prêt bancaire

Arbre de décision: vocabulaire

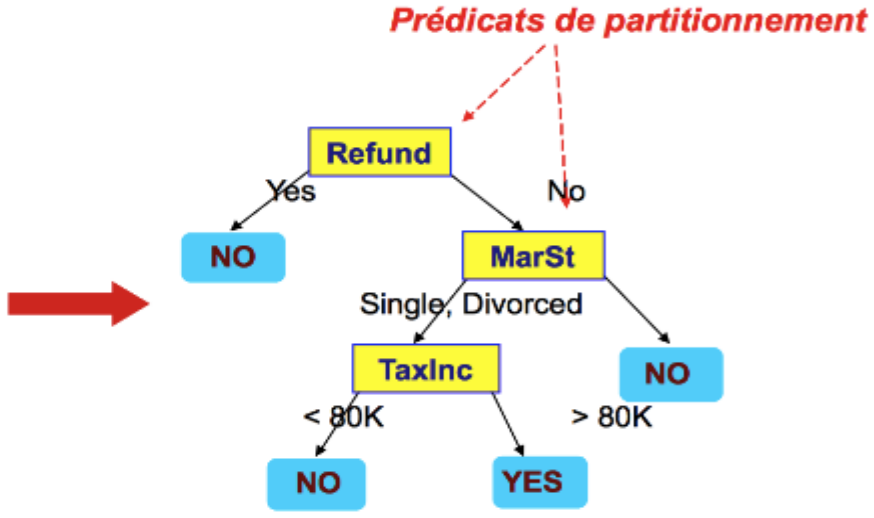
L'ensemble des nœuds se divise en trois catégories :

- Nœud *racine:* l'accès à l'arbre se fait par ce nœud
- Nœuds *internes* : les nœuds qui ont des descendants (ou *enfants*), qui sont à leur tour des nœuds
- Nœuds *terminaux* (ou *feuilles*) : nœuds qui n'ont pas de descendant.

Arbre de décision: Exemple

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Ensemble d'apprentissage



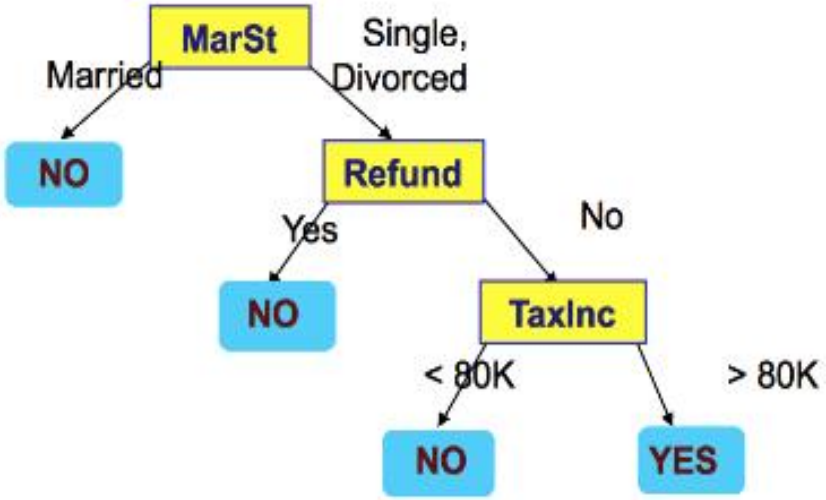
Modèle: Arbre de décision

Refund Noeuds attributs

Noeuds classes

Arbre de décision: Exemple

Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Pour des mêmes données, plusieurs arbres sont possibles !

Arbre de décision: induction

Construction de l'arbre en découpant successivement les données en fonction des variables prédictives.

- Soit D l'ensemble d'enregistrements (données d'apprentissage) qui amène au nœud t.
- Algorithme générique : Segmenter(D)
  - Si tous les enregistrements de D appartiennent à la même classe de variable à prédire y alors t est une feuille labelisée comme yt
  - Si D contient des enregistrements appartenant à plusieurs classes :
    - Pour chaque attribut prédictif A, évaluer la qualité de découpage selon A.
    - Utiliser l'attribut donnant la meilleure découpe pour découper l'ensemble de données en sous ensembles.
    - Appliquer la procédure de manière récursive sur les sous ensembles obtenus.

Arbre de décision: induction

Problèmes fondamentaux pour construire l'arbre

- Choix de l'attribut discriminant.
- Affectation d'un label à une feuille.
- Arrêt de la procédure de segmentation (i.e. profondeur de l'arbre). Si un arbre est trop profond, il est trop complexe et trop adapté à l'ensemble d'apprentissage, i.e. pas assez généraliste.
- Choix des bornes de discrétisation (i.e. comment découper les valeurs d'un attribut continu).

Arbre de décision: Construction

Pureté d'un nœud

- Un nœud est **pur** si tous les individus associés appartiennent à la même classe.

Exemple:

- Construire un arbre de décision qui classe et détermine les caractéristiques des clients qui consultent leurs comptes sur internet.

Arbre de décision: Construction

Client	M	A	R	E	I
1	moyen	moyen	village	oui	oui
2	élevé	moyen	bourg	non	non
3	faible	âgé	bourg	non	non
4	faible	moyen	bourg	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevè	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non

- M: moyenne des montants sur le compte
- A: âge du client
- R: lieu de résidence du client
- E: le client fait des études supérieures ?
- I: le client consulte ses comptes sur Internet ?

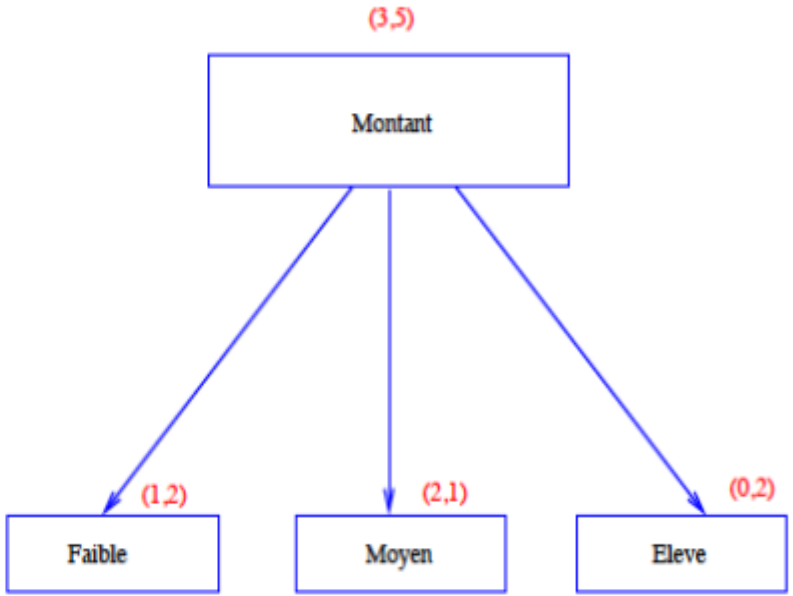
Arbre de décision: Construction

- Construction descendante.
- Au début, tous les individus sont regroupés.
- Est-ce que le nœud initial est un nœud terminal ou est-ce qu'on peut construire un test sur une variable qui permettra de mieux discriminer les individus ?
- Quatre constructions possibles suivant les variables Montant (M), Age(A), Résidence (R), et Etudes (E).

Arbre de décision: Construction

Construction selon la variable Montant (M)

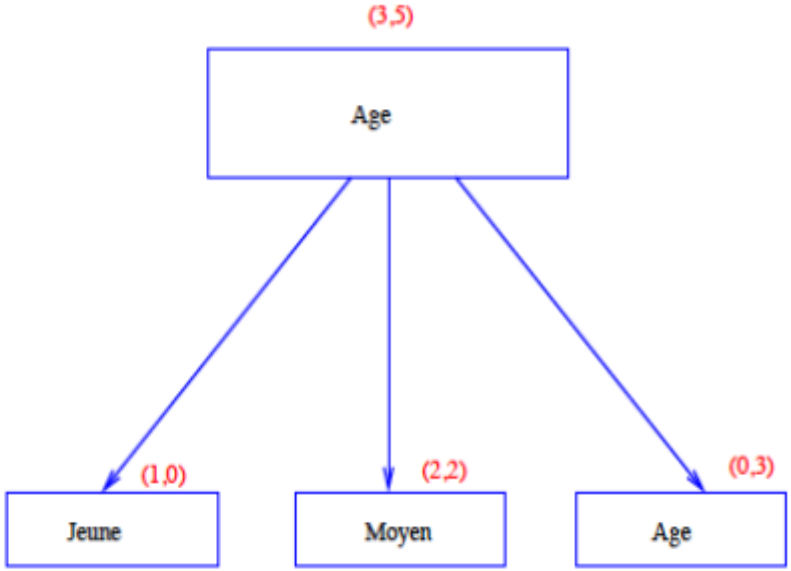
Client	M	I
1	moyen	oui
2	élevé	non
3	faible	non
4	faible	oui
5	moyen	oui
6	élevé	non
7	moyen	non
8	faible	non



Arbre de décision: Construction

Construction selon la variable Age (A)

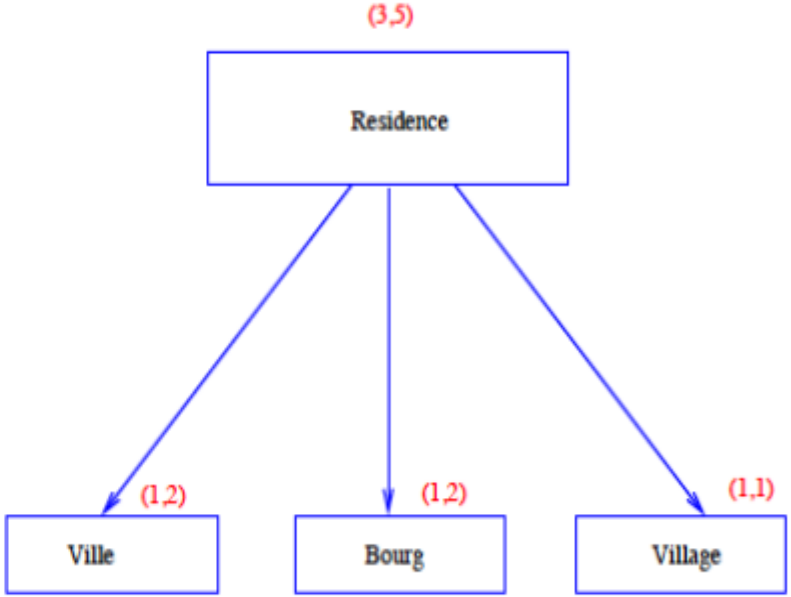
Client	A	I
1	moyen	oui
2	moyen	non
3	âgé	non
4	moyen	oui
5	jeune	oui
6	âgé	non
7	âgé	non
8	moyen	non



Arbre de décision: Construction

Construction selon la variable Résidence (R)

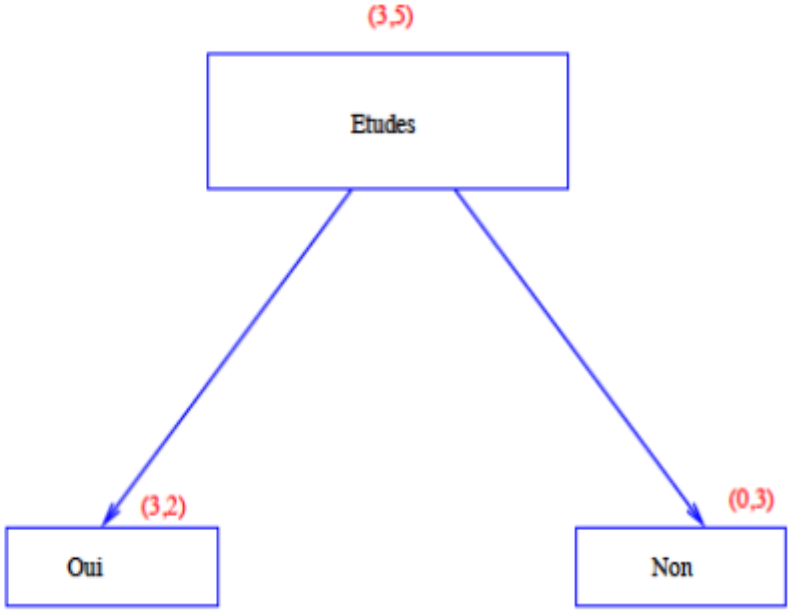
Client	R	I
1	village	oui
2	bourg	non
3	bourg	non
4	bourg	oui
5	ville	oui
6	ville	non
7	ville	non
8	village	non



Arbre de décision: Construction

Construction selon la variable Etudes (E)

Client	E	I
1	oui	oui
2	non	non
3	non	non
4	oui	oui
5	oui	oui
6	oui	non
7	oui	non
8	non	non



Arbre de décision: Construction

Quel test choisir ?

- Un test est intéressant s'il permet une bonne discrimination.
- Sur R, aucune discrimination sur aucune branche : on ne gagne rien avec ce test !
- Sur A, deux nœuds sur trois sont purs.
- Comment écrire cela de manière algorithmique et mathématique ?

Variable test	Composition noeuds
Montant (M)	(1,2),(2,1),(0,2)
Age (A)	(1,0),(2,2),(0,3)
Résidence (R)	(1,2),(1,2),(1,1)
Etudes (E)	(3,2),(0,3)

Arbre de décision: Construction

- On a besoin de comparer les différents choix possibles.
- Introduire des fonctions permettant de mesurer le degré de désordre dans les différentes classes (pureté d'un nœud)
- Propriétés des fonctions (degré de désordre) :
  - Le minimum est atteint lorsque tous les nœuds sont purs.
  - Le maximum est atteint lorsque les individus sont équirépartis entre les classes.
- Exemples de fonctions :Indice de Gini, Entropie,...

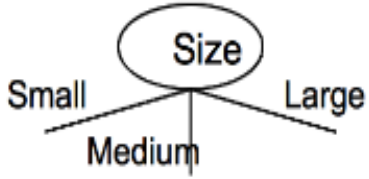
Choix de l'attribut discriminant

- Comment spécifier la condition de test ?

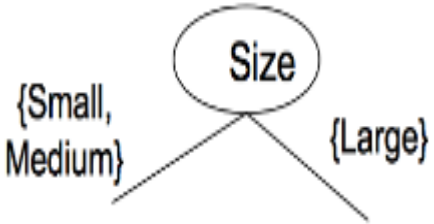
1. Dépend du type d'attribut
   - Nominal
   - Ordinal
   - Continu
2. Dépend du nombre de façon de diviser
   - Division en 2
   - Division en n

Attribut nominal / Ordinal

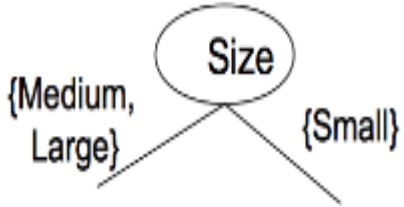
- **Division multiple** : autant de partitions que de valeurs distinctes.



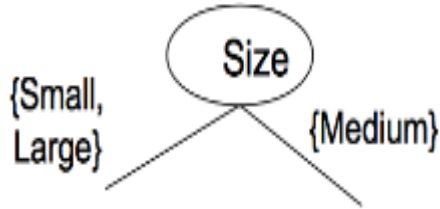
- **Division binaire** : Division des valeurs en deux sous-ensembles ⇒ Trouver le partitionnement optimal.



OU

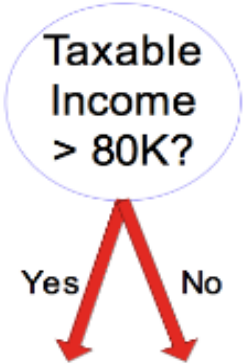


OU

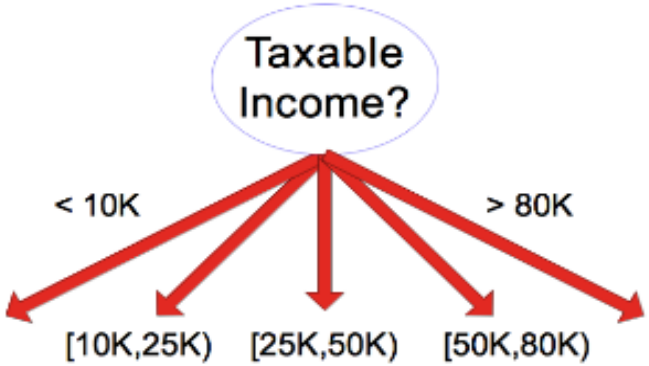


Attribut continu

- Différentes manières de discrétiser :
  - Discrétisation pour former un attribut ordinal.
  - Décision binaire



(i) Binary split



(ii) Multi-way split

Choix de l'attribut discriminant

- On privilégie les nœuds avec des distributions homogènes.

C0: 5
C1: 5

Non homogène
Impureté forte

C0: 9
C1: 1

Homogène
Impureté faible

- Mesure du désordre d'un nœud:
  - Indice de Gini
  - Entropie
  - Taux de classification

Mesure du désordre : GINI

- Pour un nœud ***t*** donné : Dépend du type d'attribut.

$$GINI(t) = 1 - \sum_j p(j|t)^2$$

- Avec *p(j|t)* la fréquence relative de la classe j au nœud t.
  - Maximum : $1 - \frac{1}{n_c}$ quand tous les enregistrements sont distribués de manière égale parmi toutes les classes.
  - Minimum : 0.0 quand tous les enregistrements appartiennent à une classe.

Mesure du désordre : GINI

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – (1/6)² – (5/6)² = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)² – (4/6)² = 0.444

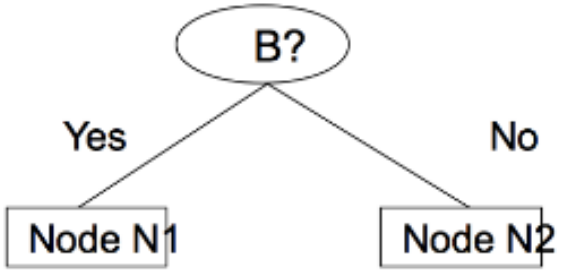
Mesure du désordre : Gain de GINI

- Utilisé dans les algorithmes CART, SPLIQ et SPRINT.
- Calculer le **Gain de Gini** et choisir le nœud qui a le plus petit Gain,
- Quand un nœud ***p*** est divisé en ***k*** partitions, la qualité de la division est calculée par

$$Gain\ de\ GINI_p = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

- Avec:
  - n_i : nombre d'enregistrements au nœud ***i***.
  - n: nombre d'enregistrements au nœud ***p***.

Attributs binaires : indice de Gini



Gini(N1)= 1-(5/7)^2 −(2/7)^2
= 0,42

Gini(N2)= 1-(1/5)^2 −(4/5)^2
= 0,32

	N1	N2
C1	5	1
C2	2	4
Gain= 0,37		

Gain_de_GINI(B)
= (7/12)*0,42 + (5/12)*0,32
= 0,37

Attributs catégoriques : indice de Gini

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Attributs continus : indice de Gini

- Pour chaque attribut:
  - Trier les attributs par valeurs
  - Scanner linéairement les valeurs , en calculant l'indice de Gini
  - Choisir la position qui a le plus petit indice de Gini

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
Taxable Income										
Sorted Values →	60	70	75	85	90	95	100	120	125	220
Split Positions →	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	3 0
No	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	7 0
Gini	0.420	0.400	0.375	0.343	0.417	0.400	*0.300*	0.343	0.375	0.400	0.420

Mesure du désordre : Entropie

- Calculer l'Entropie à un nœud t:

$$Entropie(t) = -\sum_{i=1}^{k} p(j|t) \log_2 p(j|t)$$

- Calculer le Gain d'entropie à un nœud t

$$Gain\ d'Entropie(t) = Entropie(S) - \sum_{i=1}^{k} p(j|t) Entropie(t_i)$$

- Sélectionner l'attribut A ayant le meilleur Gain d'entropie dans l'ensemble de données S.

Mesure du désordre : Entropie

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Arbre de décision: Construction

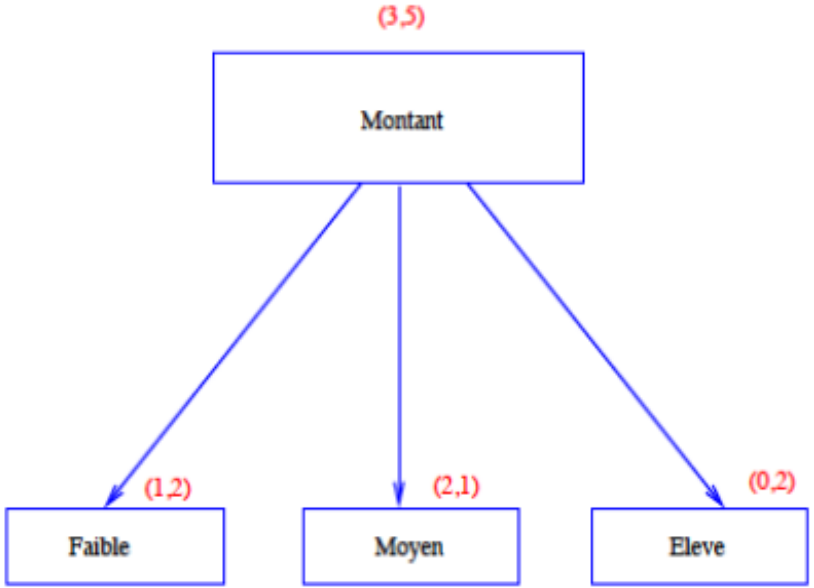
Client	M	A	R	E	I
1	moyen	moyen	village	oui	oui
2	élevé	moyen	bourg	non	non
3	faible	âgé	bourg	non	non
4	faible	moyen	bourg	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevè	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non

- M: moyenne des montants sur le compte
- A: âge du client
- R: lieu de résidence du client
- E: le client fait des études supérieures ?
- I: le client consulte ses comptes sur Internet ?

Retour sur l'exemple

- Tester sur la variable Montant (M) : on considère le nœud 0, (3,5) avec comme fonction l'entropie.

Client	M	I
1	moyen	oui
2	élevé	non
3	faible	non
4	faible	oui
5	moyen	oui
6	élevé	non
7	moyen	non
8	faible	non



Retour sur l'exemple

Gain(O, M) = Entropie(O) – (3/8*E(Faible) + 3/8*E(Moyen) + 2/8*E(Elevé))

Entropie(Faible) = –(1/3)*log(1/3) – (2/3)*log(2/3) = 0.64

Entropie(Moyen) = –(2/3)*log(2/3) – (1/3)*log(1/3) = 0.64

Entropie(Elevé) = –(2/2)*log(2/2) = 0

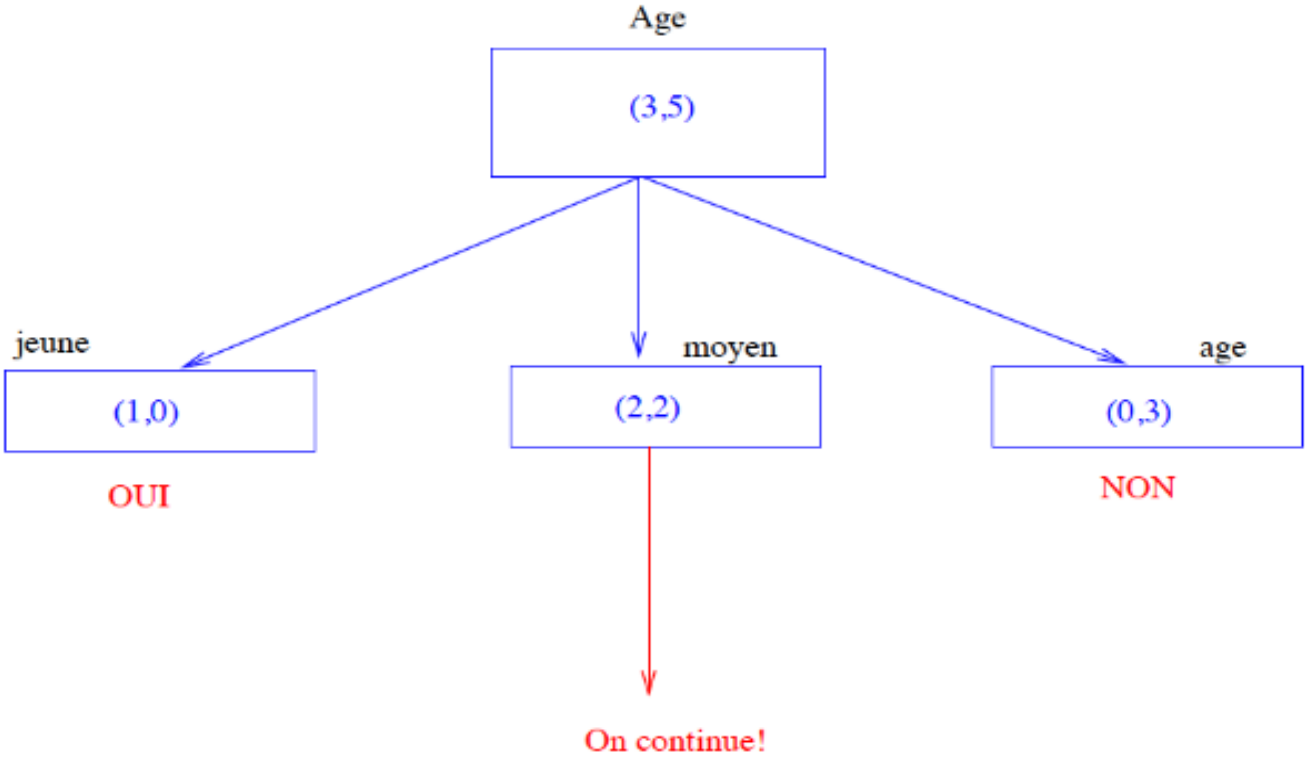
Gain(O, M) = Entropie(O) – 0.48

Retour sur l'exemple

Variable test	Gain
Montant (M)	$Gain(0, M) = Entropie(0) - 0.48$
Age (A)	$Gain(0, A) = Entropie(0) - 0.35$
Résidence (R)	$Gain(0, R) = Entropie(0) - 0.65$
Etudes (E)	$Gain(0, R) = Entropie(0) - 0.42$

- Choix de l'attribut âge (A)

Retour sur l'exemple



Suite de la construction

Client	M	A	R	E	I
1	moyen	moyen	village	oui	oui
2	élevé	moyen	bourg	non	non
3	faible	âgé	bourg	non	non
4	faible	moyen	bourg	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevé	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non

⇓

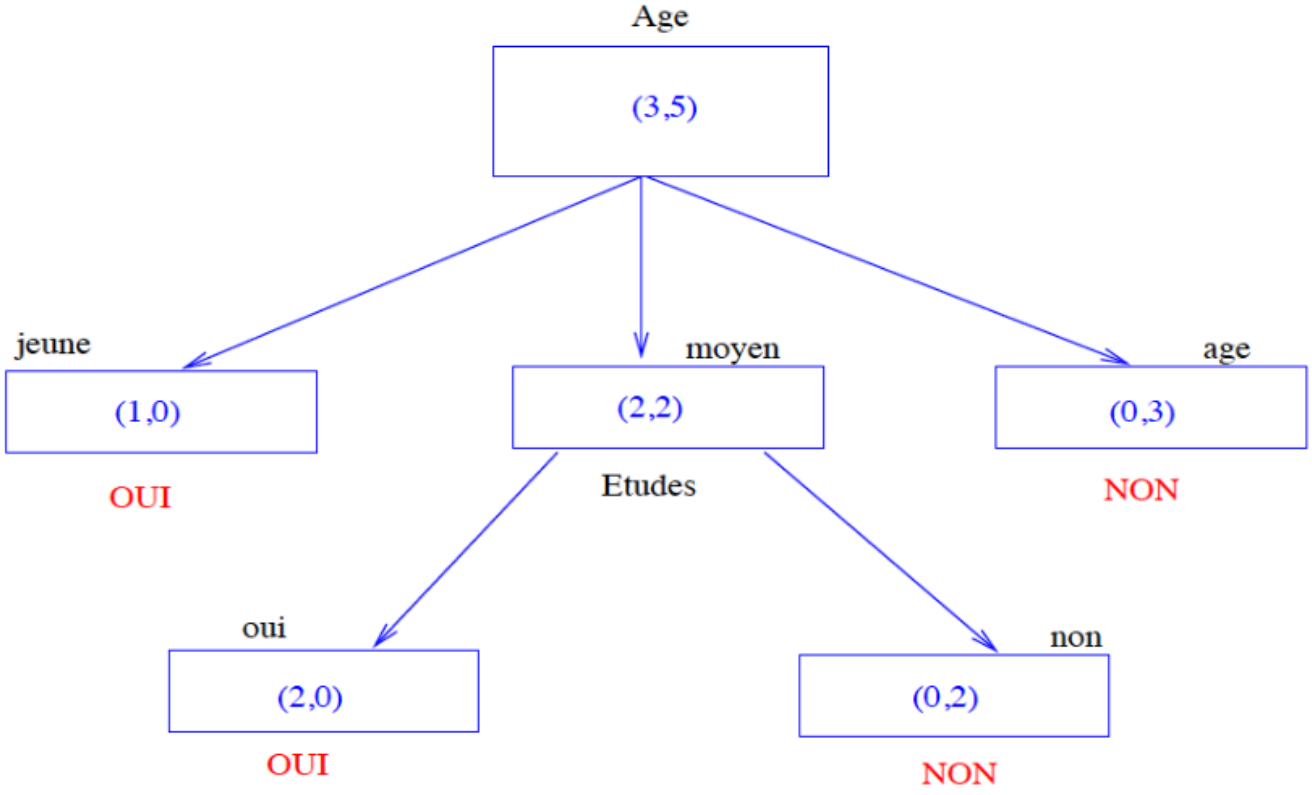
Client	M	R	E	I
1	moyen	village	oui	oui
2	élevé	bourg	non	non
4	faible	bourg	oui	oui
8	faible	village	non	non

Suite de la construction

Variable test	Composition noeuds
Montant (M)	(1,1),(1,0),(0,1)
Résidence (R)	(1,1),(1,1)
Etudes (E)	(2,0),(0,2)

- Quel test choisir ?
- Calcul du gain pour chaque test.

Suite de la construction



Apprentissage des arbres de décision : Généralités

- Arbre de décision parfait, i.e. tous les exemples sont bien classifiés => n'existe pas toujours.
- Le meilleur arbre est l'arbre le plus petit parfait.
- L'objectif est d'obtenir l'arbre le plus petit possible (facilitant la recherche) tout en établissant un compromis entre les taux d'erreur sur l'ensemble d'apprentissage et sur l'ensemble de test afin de pouvoir généraliser.

Algorithmes ID3

- ID3 considère en entrée un ensemble d'attributs A, un attribut cible c (appelés classe) et un ensemble d'échantillons E.
- ID3 fonctionne exclusivement avec des attributs qualitatifs.
- A chaque étape de la récursion, il détermine l'attribut qui maximise le gain d'informations.
- Cet attribut est alors retenu pour la branche en cours puisqu'il permet de classer plus facilement l'ensemble des données à ce niveau de l'arbre.

Algorithmes C4.5

- Amélioration de l'algorithme ID3.
- Permet de traiter des attributs quantitatifs.
- C4.5 utilise la fonction du gain d'entropie pour le choix des attributs à considérer à chaque niveau de l'arbre.
- C4.5 a été amélioré sous l'appellation C5, qui permet une rapidité d'exécution et une efficacité d'utilisation de la mémoire plus élevée.

Algorithmes CART

- CART (Classification And Regression Trees) permet la construction d'un arbre de décision.
- CART pose seulement de questions-test binaires (arbres binaires).
- Il utilise la fonction GINI et fonctionne aussi pour des attributs aux valeurs continues.
- CART cherche tous les attributs et tous les seuils pour trouver celui qui donne la meilleure homogénéité du découpage.

Surapprentissage

- Dans le cas extrême où l'arbre a autant de feuilles qu'il y a d'individus dans la population (d'enregistrements dans le jeu de données).
- L'arbre ne commet alors aucune erreur sur cet échantillon puisqu'il en épouse toutes les caractéristiques, mais il **n'est pas généralisable** à un autre échantillon.
- Ce problème, nommé surapprentissage ou surajustement (overfitting).

Surapprentissage

- Veiller à construire un arbre qui soit le plus petit possible en assurant la meilleure performance possible.
- Plus un arbre sera petit, plus il sera stable dans ses prévisions futures.
- Pour éviter le sur-apprentissage (créer un arbre avec une grande profondeur), on peut utiliser la technique d'élagage:
  - Le pré-élagage
  - Le post-élagage

Le pré-élagage

- Consiste à proposer des critères d'arrêt lors de la phase d'expansion.
- Lorsque le groupe est d'effectif trop faible, ou lorsque l'homogénéité d'un sous-ensemble a atteint un niveau suffisant, on considère qu'il n'est plus nécessaire de séparer l'échantillon.
- Un autre critère souvent rencontré dans ce cadre est l'utilisation d'un test statistique.

Le post-élagage

- Consiste à construire l'arbre en deux temps :
  - Construire d'abord l'arbre dont les feuilles sont le plus homogènes possibles.
  - Puis réduire l'arbre, ie. éliminer les branches qui n'améliorent pas la performance de l'arbre.
- Selon les cas, cette seconde portion des données est désignée par le terme d'échantillon de validation ou échantillon de test, introduisant une confusion avec l'échantillon utilisé pour mesurer les performances des modèles.
- Le terme d'échantillon d'élagage permet de le désigner sans ambiguïté