INTRODUCTION AU DATA MINING

6 séances de 3 heures

mai-juin 2006

EPF - 4ème année - Option Ingénierie d'Affaires et de Projets

Bertrand LIAUDET

PHASE 4 : MODELISATION NON-SUPERVISEE - 3 : LES ARBRES DE DECISION

Présentation

Vocabulaire des arbres

Arbre, nœud, racine, feuille, branche

Un arbre est constitué de nœuds connectés par des branches. Un arbre de décision est constitué de nœuds de décision.

La connexion entre les nœuds est orientée : chaque nœud est connecté à un et un seul nœud parent, sauf le nœud racine qui n'a pas de parent ; chaque nœud peut être connecté à 0 ou n nœuds-enfants. De ce fait, un arbre n'est pas un réseau.

Un nœud qui n'a pas de nœuds enfants est appelé « nœud feuille » ou « feuille ».

Variable cible et variables prédictives

Un arbre de décision travaille sur une variable cible avec plusieurs variables prédictives.

Chaque nœud non-feuille correspond à une variable prédictive. Chaque nœud feuille correspond à la variable cible.

Chaque branche correspond à une valeur (ou un ensemble de valeurs) pour la variable prédictive parent.

Chemin, prédiction

Un chemin est un parcours du nœud racine jusqu'à un nœud feuille. Sur chaque branche, le chemin précise la valeur que prend la variable prédictive du nœud à l'origine de la branche.

Un chemin se termine par un nœud feuille qui précise la ou les valeurs prévues pour les enregistrements de la variable cible pour ce chemin particulier.

Feuille pure

Un nœud feuille est pur si les valeurs de la variable cible sont les mêmes pour tous les enregistrements de ce nœud, autrement dit si le chemin (donc le n-uplet de valeurs pour le n-uplet de prédicteurs) détermine la valeur de la variable cible.

Exemple

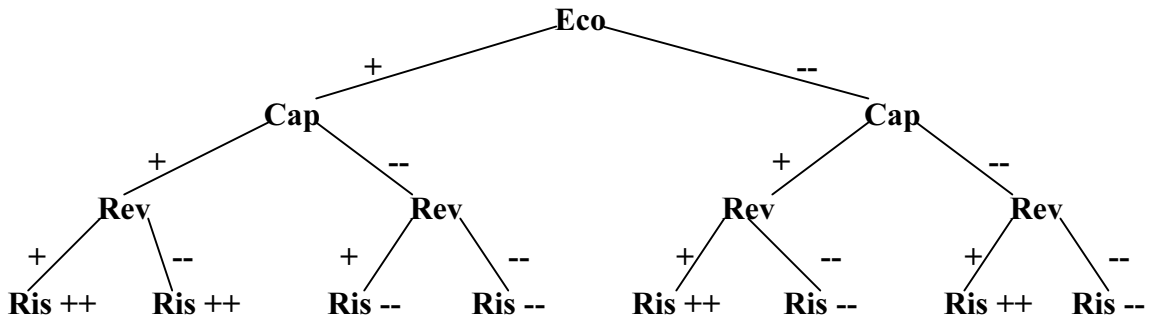
On considère trois variables prédictives : économie, capital et revenu.

Les variables prédictives ont deux valeurs possibles : faible (--) et forte (+).

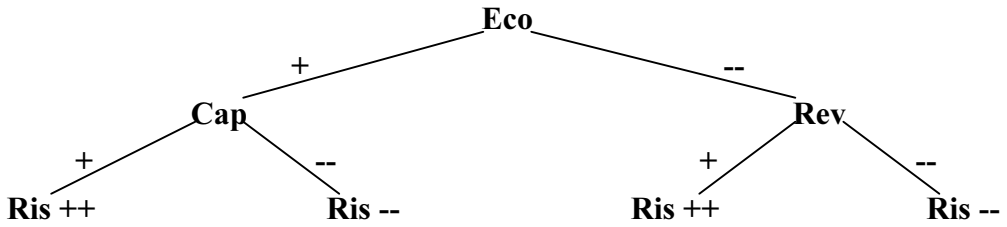
La variable cible est : risque (pour un crédit).

Les valeurs possibles pour la variable cible sont : faible (--) et forte (+).

Un arbre de décision systématique :



Avantageusement remplacé par :



On a constaté que :

Règle 1 :	(Eco +, Cap +)	=>	Ris ++ quel que soit Rev
Règle 2 :	(Eco +, Cap --)	=>	Ris -- quel que soit Rev
Règle 3 :	(Eco -, Rev +)	=>	Ris ++ quel que soit Cap
Règle 4 :	(Eco -, Rev --)	=>	Ris -- quel que soit Cap

Arbre de décision et SQL : tri et group by

En première approximation, on peut dire qu'un arbre de décision correspond à un group by fait sur toutes les variables prédictives et en dernier sur la variable cible ; les valeurs possibles pour chaque variable ayant été regroupées en sous ensembles (toutes les variables sont donc ramenées à des variables catégorielles). Les branches de l'arbre correspondent à l'un de ces sous-ensembles.

♥ ***Exemple 4-2-1***

Dans l'exemple précédent, on peut faire :

ArbreClients = Select Eco, Cap, Rev, Ris, count(*)
From Clients
Group by Eco, Cap, Rev, Ris

Eco	Cap	Rev	Ris	Count(*)
+	+	+	+	3
+	+	-	+	2
+	-	+	-	3
+	-	-	-	1
-	+	+	+	4
-	+	-	-	2
-	-	+	+	1
-	-	-	-	3

Les effectifs donnés par le count(*) correspondent au nombre d'enregistrements qui ont les caractéristiques de la ligne.

Pour savoir si on a des feuilles pures ou pas, il faut travailler sur la table ArbreClients.

Il faut savoir si le n-uplet correspondant aux prédicteurs est unique ou pas.

Select Eco, Cap, Rev, count(*)
From ArbreClients
Group by Eco, Cap, Rev

Ainsi on obtient :

Eco	Cap	Rev	Count(*)
+	+	+	1
+	+	-	1
+	-	+	1
+	-	-	1
-	+	+	1
-	+	-	1
-	-	+	1
-	-	-	1

=> toutes les feuilles sont pures.

Si on était parti du tableau suivant :

Eco	Cap	Rev	Ris	Count(*)
+	+	+	+	10
+	+	-	+	4
+	+	-	-	6
+	-	+	-	4
+	-	-	-	8

On aurait obtenu le résultat suivant :

Eco	Cap	Rev	Count(*)
+	+	+	1
+	+	-	2
+	-	+	1
+	-	-	1

On a une feuille pure si le résultat du group by est 1 (une valeur pour la variable cible).

Le triplet (+, +, -) a deux occurrences : ce n'est pas une feuille pure.

En revenant au tableau précédent, on peut dire que :

(Eco +, Cap +, Rev -) => Ris + avec un seuil de confiance de 40 %

(Eco +, Cap +, Rev -) => Ris - avec un seuil de confiance de 60 %

Algorithmes

Algorithme supervisé

Les algorithmes de data mining correspondant à un arbre de décision sont des algorithmes supervisés (il existe une variable cible).

Ensemble d'apprentissage

Un ensemble de données avec les valeurs de la variable cible est fourni à l'algorithme : c'est l'ensemble d'apprentissage.

L'ensemble d'apprentissage doit être riche et varié pour donner une classification exploitable.

Les arbres de décision apprennent par l'exemple : si les exemples manquent systématiquement dans l'ensemble d'apprentissage concernant une catégorie particulière de données, la segmentation et la prévision seront problématiques ou impossibles pour cette catégorie de données.

Variables catégorielles

Les prédicteurs et la variable cible doivent être des variables catégorielles. Si on a des variables continues, il faudra les discrétiser.

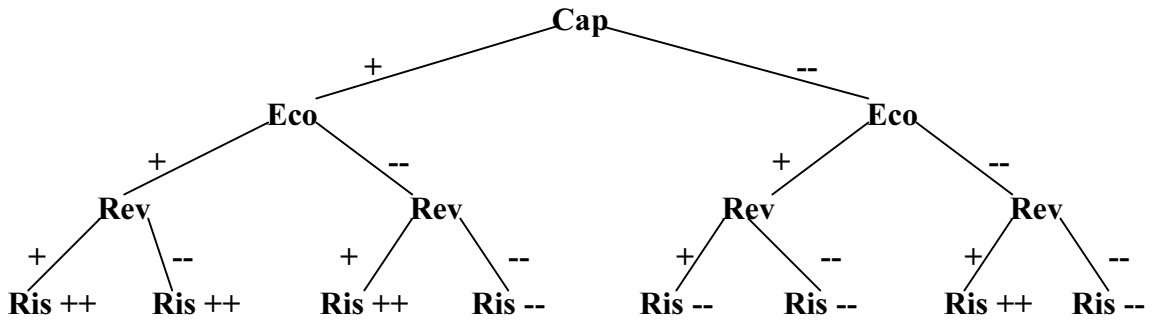
Paramètres de la construction d'un arbre : nœud et branche (« scission »)

Deux paramètres vont intervenir pour construire l'arbre de décision :

- L'ordre dans lequel on parcourt les branches de l'arbre. Autrement dit, quelle variable on traite à chaque nœud.
- La sélection des catégories pour les prédicteurs et la variable cible. Autrement dit, quelle branche choisit-on à chaque nœud : c'est le problème de la scission.

Le but d'un algorithme d'arbre de décision est de créer un ensemble de nœuds feuilles qui soient le plus pures possible avec le moins de branches possibles et des branches les plus courtes possibles.

Reprise du premier exemple en changeant l'ordre des variables :



On peut voir que :

Règle a :	(Cap+, Eco+)	=>	(Ris+)
Règle b:	(Cap-, Eco+)	=>	(Ris-)

et

Règle c:	(Cap+, Eco-, Rev+)	=>	(Ris+)
Règle d:	(Cap+, Eco-, Rev-)	=>	(Ris-)
Règle e:	(Cap-, Eco-, Rev+)	=>	(Ris+)
Règle f:	(Cap-, Eco-, Rev-)	=>	(Ris-)

Règle a = Règle 1 vue précédemment.

Règle b = Règle 2 vue précédemment.

Règle c et e = Règle 3 vue précédemment.

Règle d et f = Règle 4 vue précédemment.

On peut donc revenir aux 4 règles vues précédemment :

Règle 1 :	(Eco +, Cap +)	=>	Ris ++ quel que soit Rev
Règle 2 :	(Eco +, Cap --)	=>	Ris -- quel que soit Rev
Règle 3 :	(Eco -, Rev +)	=>	Ris ++ quel que soit Cap
Règle 4 :	(Eco -, Rev --)	=>	Ris -- quel que soit Cap

Les algorithmes existants : CART et C4.5

Il existe plusieurs algorithmes de fabrication d'arbre de décision.

Citons particulièrement :

- Le CART : méthode des arbres de segmentation et de régression (1984). Le CART fabrique des arbres binaires (toujours deux branches par nœuds non-feuilles).
- L'algorithme C4.5. Le C4.5 fabrique des arbres qui ne sont pas nécessairement binaires (0 à n branches par nœud).

L'algorithme de segmentation et de régression : CART

Principe de l'algorithme CART

Les principes d'un algorithme d'arbre de décision suivent les paramètres de la construction d'un arbre :

Choix des variables : nœud

L'algorithme partitionne l'ensemble d'apprentissage en sous-ensemble d'enregistrements avec des données identiques pour la variable cible (autrement dit, fait un tri par variable cible, puis par prédicteurs).

Choix des valeurs pour les variables : branche (scission)

L'algorithme par de la racine de l'arbre. À chaque nœud de décision, l'algorithme fait une recherche exhaustive sur toutes les variables avec toutes les valeurs de scission possibles. L'algorithme choisit ensuite la scission optimale selon un critère défini. Il n'y a qu'une scission par nœud puisque l'arbre est binaire.

Précision sur l'algorithme CART

Mesure de la qualité d'une scission

$$\varphi(s \mid t) = 2P_G P_D \sum(i=1 \text{ à } nbClasses) \mid P(i \mid t_G) - P(i \mid t_D) \mid$$

$\varphi(s \mid t)$: mesure de la qualité d'une scission au nœud t

La meilleure scission parmi toutes les scissions possibles au nœud t est celle qui a la plus grande valeur pour $\varphi(s \mid t)$.

t_G : nœud enfant gauche du nœud t — t_D : nœud enfant droit du nœud t

P_G : (nb enregistrements à t_G) / nbTotal — P_D : (nb enregistrements à t_D) / nbTotal

nbTotal : nombre d'enregistrements dans tout l'ensemble d'apprentissage

nbClasses : nombre de classes de valeurs pour la variable cible.

$P(i \mid t_G)$: (nb enregistrements pour la classe i à t_G) / (nb enregistrements à t)

$P(i \mid t_D)$: (nb enregistrements pour la classe i à t_D) / (nb enregistrements à t)

Nombre d'enregistrements sur un nœud

Le nombre d'enregistrement sur un nœud correspond au nombre d'enregistrement restant près les décisions déjà prises.

Par exemple, pour le tableau suivant, si le nœud racine concerne la variable Eco, et que la scission vers le nœud droit se fait sur le critère = Faible, alors, il reste 3 enregistrements sur le nœud droit, et donc 5 sur le nœud gauche.

Client	Eco	Cap	Rev	Ris
1	Moyen	Elevé	75	+
2	Faible	Faible	50	-
3	Elevé	Moyen	25	-
4	Moyen	Moyen	50	+
5	Faible	Moyen	100	+
6	Elevé	Elevé	25	+
7	Faible	Faible	25	-
8	Moyen	Moyen	75	+

Quelles sont les scissions candidates au nœud racine ?

On passe en revue toutes les variables prédicteurs et pour chaque variable, on passe en revue toutes les dichotomies possibles de classes de valeurs.

Pour le revenu : on a une variable continue (non catégorielle). L'algorithme utilisera chaque valeur réelle comme limite possible entre deux classes de valeurs. L'analyste peut aussi choisir de catégoriser la variable.

Pour chaque scission, on calcule le nombre d'enregistrements pour le nœud droit et pour le nœud gauche, ce qui permet de calculer P_G et P_D (P_G = nb enregistrements G / nb enregistrements total qui vaut 8).

Scission candi-date	Nœud enfant de gauche t_G	Nb enregis-trements G	P_G	Nœud enfant de droite t_D	Nb enregis-trements D	P_D	$2*P_G*P_D$
1	Eco = Faible	3	0,375	Eco ∈ {Moyen, Elevé}	5	0,625	0,469
2	Eco = Moyen	3	0,375	Eco ∈ {Faible, Elevé}	5	0,625	0,469
3	Eco = Elevé	2	0,250	Eco ∈ {Faible, Moyen}	6	0,750	0,375
4	Cap = Faible	2	0,250	Cap ∈ {Moyen, Elevé}	6	0,750	0,375
5	Cap = Moyen	4	0,500	Cap ∈ {Faible, Elevé}	4	0,500	0,500
6	Cap = Elevé	2	0,250	Cap ∈ {Faible, Moyen}	6	0,750	0,375
7	Rev <= 25	3	0,375	Rev > 25	5	0,625	0,469
8	Rev <= 50	5	0,625	Rev > 50	3	0,375	0,469
9	Rev <= 75	7	0,875	Rev > 75	1	0,125	0,219

Ensuite, toujours pour chaque scission, on compte le nombre d'enregistrement pour chaque valeur possible de la variable cible (+ ou - pour le risque).

Ca nous permet de calculer $P(i \mid t_G)$ qui vaut ce nombre diviser par le nombre d'enregistrements pour la scission (Nb Ris / Nœud G) et (Nb Ris / Nœud D).

On peut ensuite calculer, pour chaque scission, la somme de la valeur absolue des différences entre $P(i \mid t_G)$ et$P(i \mid t_D)$.

♥ ***Exemple de calcul : Arbre / Arbre CART.xls***

La mesure de la qualité de la scission, $\varphi(s \mid t)$, est enfin donnée par la multiplication de $\sum$ et de $2P_GP_D$

Scission	Risque	Nb enr.	Nœud G	Nb Ris G	$P(i \mid t_G)$	Nœud D	Nb Ris D	$P(I \mid t_D)$	$Abs(P(i \mid t_G)-P(i \mid t_D))$	$\sum$	$2P_GP_D$	$\varphi(s \mid t)$
1	+	8	3	1	0,333	5	4	0,800	0,467	0,933	0,469	0,438
	--	8		2	0,667		1	0,200	0,467			
2	+	8	3	3	1,000	5	2	0,400	0,600	1,200	0,469	0,563
	--	8		0	0,000		3	0,600	0,600			
3	+	8	2	1	0,500	6	4	0,667	0,167	0,333	0,375	0,125
	--	8		1	0,500		2	0,333	0,167			
4	+	8	2	0	0,000	6	5	0,833	0,833	1,667	0,375	**0,625**
	--	8		2	1,000		1	0,167	0,833			
5	+	8	4	3	0,750	4	2	0,500	0,250	0,500	0,5	0,250
	--	8		1	0,250		2	0,500	0,250			
6	+	8	2	2	1,000	6	3	0,500	0,500	1,000	0,375	0,375
	--	8		0	0,000		3	0,500	0,500			
7	+	8	3	1	0,333	5	4	0,800	0,467	0,933	0,469	0,438
	--	8		2	0,667		1	0,200	0,467			
8	+	8	5	2	0,400	3	3	1,000	0,600	1,200	0,469	0,563
	--	8		3	0,600		0	0,000	0,600			
9	+	8	7	4	0,571	1	1	1,000	0,429	0,857	0,219	0,188
	--	8		3	0,429		0	0,000	0,429			

Cf. Arbre / Arbre CART.xls

On voit que c'est la scission numéro 4 qui a la mesure de qualité la plus élevée.

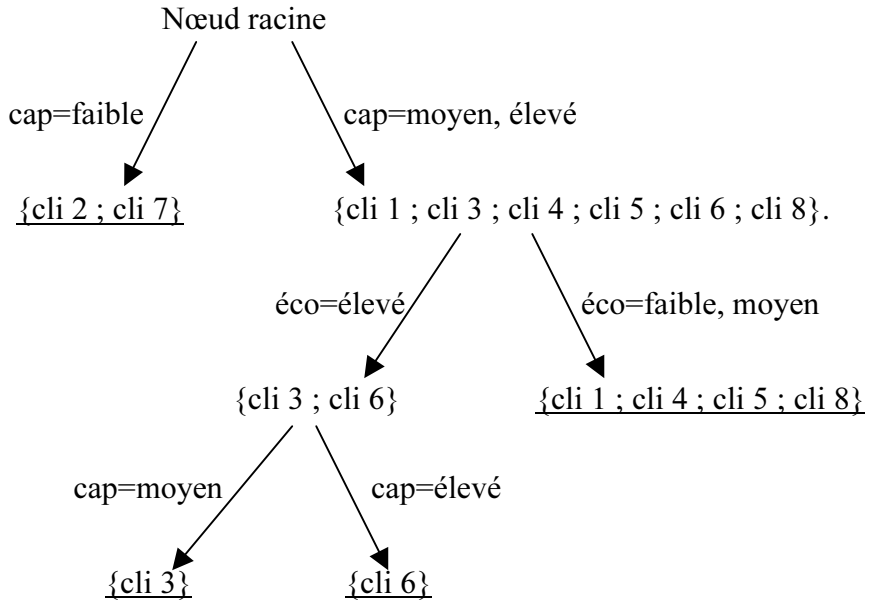
L'algorithme du CART va donc choisir la 4ème scission. Ce qui produit un nœud enfant de gauche {cli 2 ; cli 7}et un nœud enfant de droite {cli 1 ; cli 3 ; cli 4 ; cli 5 ; cli 6 ; cli 8}.

On peut ensuite appliquer le calcul au nœud enfant de gauche {cli 2 ; cli 7}: on n'obtiendra que des valeurs de qualité à 0 car le risque est le même pour les deux enregistrements de ce nœud : mauvais. On a donc un nœud feuille.

On applique ensuite le calcul au nœud enfant de droite {cli 1 ; cli 3 ; cli 4 ; cli 5 ; cli 6 ; cli 8}. C'est la 3ème scission (Économie = élevé) ou la 7ème (Revenu <= 25) qui donne la meilleure qualité (0,444). L'algorithme a choisi la 3ème scission comme nouveau nœud. On produit donc un nœud enfant de gauche {cli 3 ; cli 6}, et un nœud enfant de droite {cli 1 ; cli 4 ; cli 5 ; cli 8}.

On applique ensuite le calcul au nœud enfant de gauche {cli 3 ; cli 6}. Il n'y a que deux scissions valides et elles ont la même qualité. De plus, elles sont finalement équivalentes. Qu'on prenne l'une ou l'autre, on se retrouve avec deux feuilles et un individu dans chaque feuille. On arrive donc à une feuille à gauche avec des capitaux élevés{cli 6}, et une feuille à droite avec des capitaux moyens {cli 3}.

On applique ensuite le calcul au nœud enfant de droite {cli 1 ; cli 4 ; cli 5 ; cli 8}. Tous les individus de cet ensemble ont la même valeur de risque : satisfaisant. De ce fait, les scissions valides valent toutes 0. On a affaire à une feuille.



Interprétation

Pour que la qualité de la scission, $\varphi(s \mid t)$, soit le plus grand possible, il faut que $\sum$ et $2*P_G*P_D$ soient le plus grands possibles.

Examinons d'abord la $\sum$. C'est le nombre de classes pour la variable cible * Abs($P(i \mid t_G)$-$P(i \mid t_D)$). Dans notre exemple, la variable cible peut prendre deux valeurs : ris-- (i=1 par exemple) et ris ++ (i=2). Le nombre de classe, c'est donc 2. Pour une scission donnée, si la tous les éléments du côté gauche ont la même valeur (par exemple ris --) pour la variable cible, alors $P(1 \mid t_G) = 1$, et $P(2 \mid t_G) = 0$. Si on a la même chose en inversé du côté droit, alors on aura Abs($P(i \mid t_G)$-$P(i \mid t_D)$)=1, et donc la $\sum$ vaudra 2, ce qui sera sa valeur maximum.

Cela veut donc dire que la répartition s'est fait de telle sorte que tous les ris-- sont d'un côté et tous les ris++ sont de l'autre côté.

Pour que $2*P_G*P_D$ soit le plus grand possible, il faut que le nombre d'enregistrements du nœud de gauche (NG) * nombre d'enregistrements du nœud de droite (ND) soit le plus grand possible. Or NG = nombre total d'enregistrements du nœud (NT) - ND. C'est donc quand ND = NG = NT/2 qu'on a le $2*P_G*P_D$ le plus grand possible. Quand NG = ND, et que NG = NTR (nombre total d'enregistrements du nœud racine), on a le maximum possible pour $2*P_G*P_D$, à savoir 2*0,5*0,5 = 0,5.

Cela veut dire qu'on a tendance à favoriser les scissions qui répartissent les enregistrements du nœud parent en à peu près autant d'enregistrements dans chacun des nœuds enfants et de telle sorte qu'une seule valeur de la variable cible soit représenté d'un côté, et une autre valeur de l'autre côté.

A noter en conclusion la valeur maximum pour la qualité de la scission est donc de 1.

Les règles de décision : si antécédent(s) alors conséquence

Les arbres de décision permettent de construire des règles de décision.

Les règles de décision sont une représentation « lisible » de l'arbre de décision.

Il y a autant de règles de décision que de branches dans l'arbre de décision.

Les règles de décision apparaissent sous la forme de :

si antécédent alors conséquence

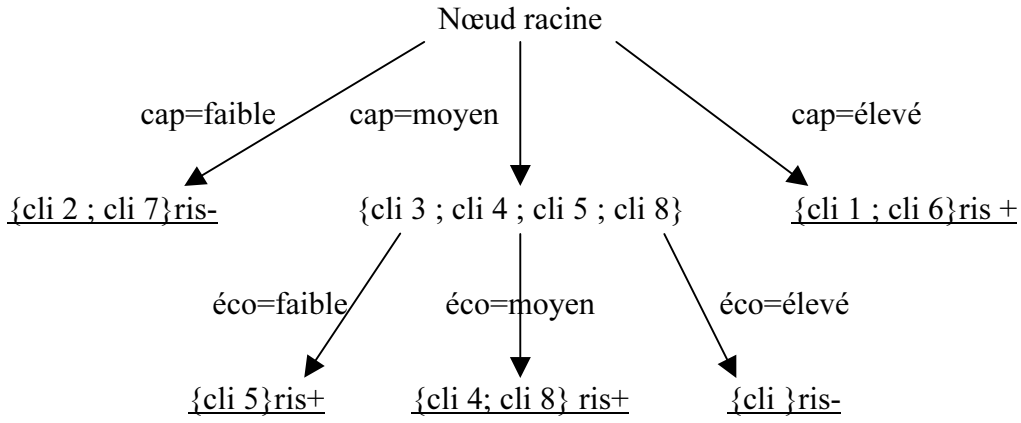
L'antécédent correspond à la branche de l'arbre.

La conséquence correspond à la feuille au bout de la branche.

N°	Antécédent	Conséquence	Population	Seuil de confiance
R1	Si capital = faible	Mauvais risque	2/8	1,00
R2	Si capital = moyen ou élevé Et Economie = faible ou moyen	Risque satisfaisant	4/8	1,00
R3	Si capital = moyen Et Economie = élevé	Mauvais risque	1/8	1,00
R4	Si capital = élevé Et Economie = élevé	Risque satisfaisant	1/8	1,00

Différence avec l'algorithme du C4.5

L'algorithme du C4.5 travaille sur des arbres n-aires.
De ce fait, il aboutira dans l'exemple traité à l'arbre suivant :



Et aux règles de décision suivantes :

	Antécédent	Conséquence	Population	Seuil de confiance
Ra	Si capital = faible	Mauvais risque	2/8	1,00
Rb	Si capital = moyen Et économie = faible ou moyen	Risque satisfaisant	3/8	1,00
Rc	Si capital = moyen Et économie = élevé	Mauvais risque	1/8	1,00
Rd	Si capital = élevé	Risque satisfaisant	2/8	1,00

Comparaison entre les deux tableaux :
R1 = Ra
R3 = Rc
Dans le CART, la segmentation se fait en final d'abord sur les économies.
Dans le C4.5, la segmentation se fait d'abord sur le capital.

Le C4.5 donne un résultat plus précis que le CART. Avec R2 et R4 du CART, on sait que si le capital est élevé, le risque est satisfaisant, quelle que soit l'économie. Mais on ne sait pas sur quelle population. Le C4.5 permet de le savoir.

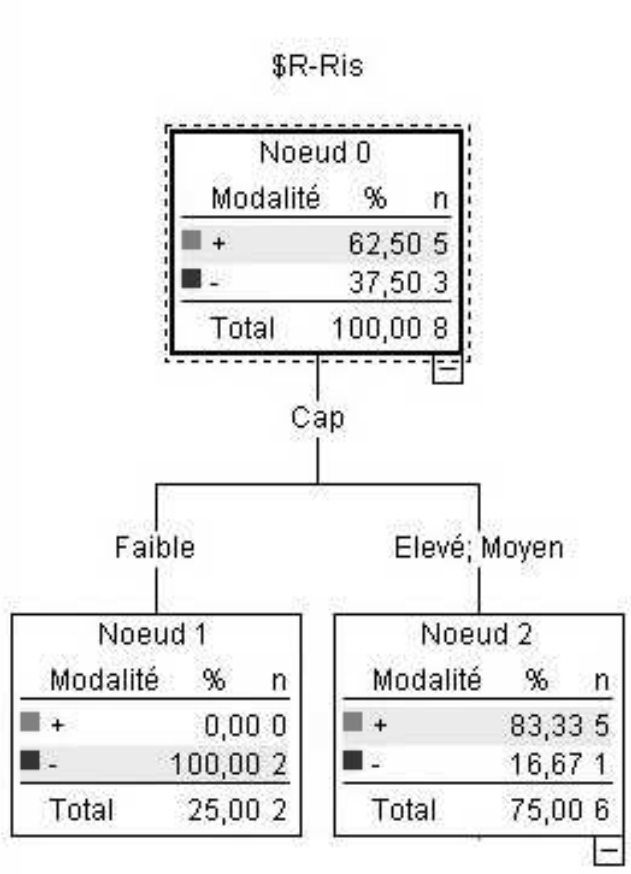
Conclusion : le C4.5 est un algorithme plus complexe (arbre n-aire au lieu de binaire) mais qui donne des résultats plus précis.

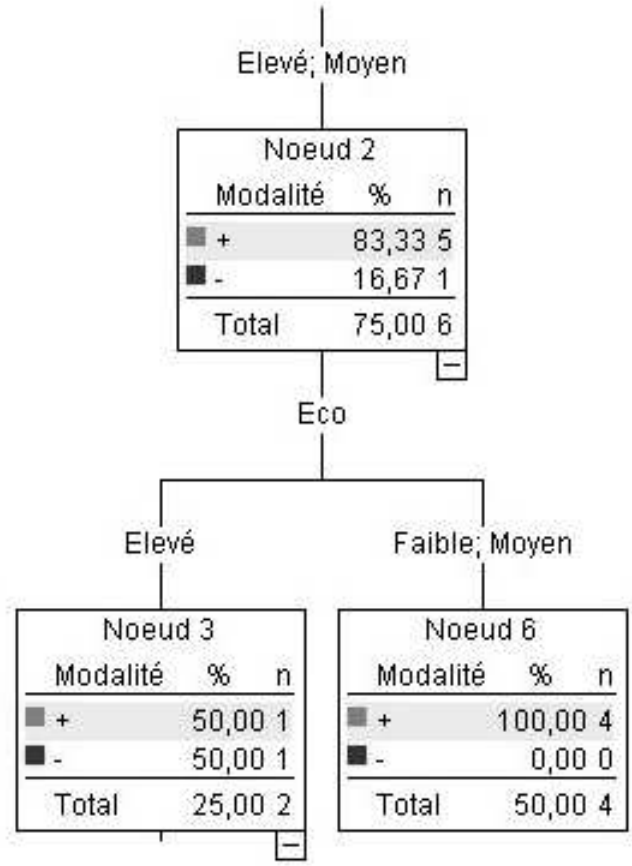
Exemple avec Clémentine

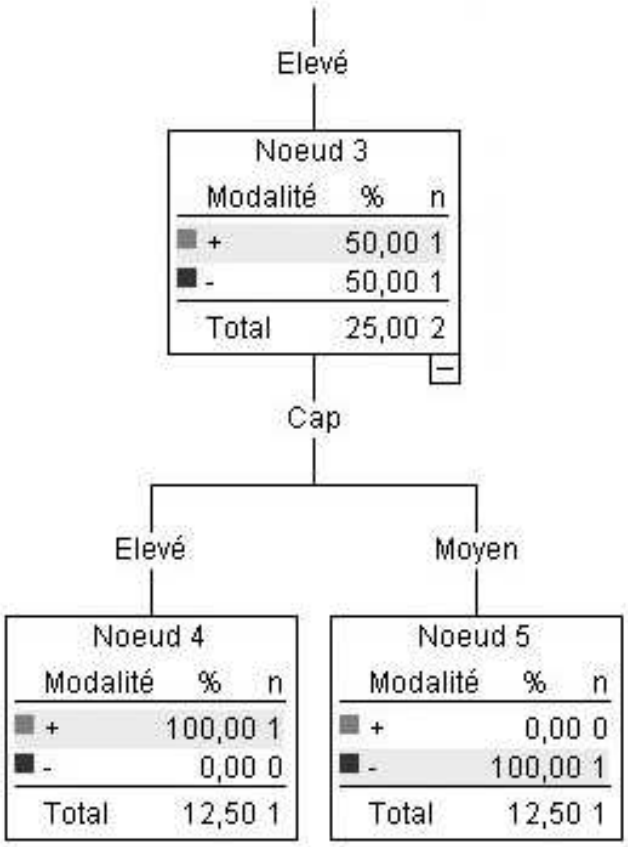
♥ ***Exemple 4-2-2***

Cap in ["Faible"] [Mode : -]	=>	-	(2 ; 1,0)
Cap in ["Elevé" "Moyen"] [Mode : +]			(6)
Client <= 3.500 [Mode : -]			(2)
Client <= 2 [Mode : +]	=>	+	(1 ; 1,0)
Client > 2 [Mode : -]	=>	-	(1 ; 1,0)
Client > 3.500 [Mode : +]	=>	+	(4 ; 1,0)

L'arbre de décision précise combien d'enregistrements sont concernés et le pourcentage précis




Elevé; Moyen
Noeud 2
Modalité % n
+ 83,33 5
- 16,67 1
Total 75,00 6
Eco
Elevé
Faible; Moyen
Noeud 3
Modalité % n
+ 50,00 1
- 50,00 1
Total 25,00 2
Noeud 6
Modalité % n
+ 100,00 4
- 0,00 0
Total 50,00 4


Elevé
Noeud 3
Modalité % n
+ 50,00 1
- 50,00 1
Total 25,00 2
Cap
Elevé
Moyen
Noeud 4
Modalité % n
+ 100,00 1
- 0,00 0
Total 12,50 1
Noeud 5
Modalité % n
+ 0,00 0
- 100,00 1
Total 12,50 1

Cas où les feuilles ne sont pas pures : taux d'erreur de segmentation

On n'arrive pas forcément à des feuilles pures, c'est-à-dire à des nœuds dont tous les éléments ont la même valeur pour la variable cible.

Par exemple, dans une feuille, on a 10 enregistrements, mais 6 seulement donnent le résultat attendu (par exemple un risque satisfaisant) et 4 un résultat autre. Le taux d'erreur de segmentation est alors de 40% pour cette feuille, puisque 4 enregistrements sur 10 sont mal classés.

L'algorithme CART va aussi calculer le taux d'erreur pour l'arbre entier. C'est la moyenne des taux d'erreur des feuilles, les taux individuels étant pondérés en proportion du nombre d'enregistrements dans chaque feuille.

♥ ***Exemple 4-2-3***

Arbre de décision sur les données du churn

Conso de jour en minutes <= 264.450 [Mode : False.]					3 122	
	Appels au service client <= 3.500 [Mode : False.]				2 871	
		International in ["no"] [Mode : False.]			2 604	
			Conso de jour en minutes <= 223.250 [Mode : False.]			=> False. (2 221; 0,973)
			Conso de jour en minutes > 223.250 [Mode : False.]		383	
				Conso de soirée en minutes <= 259.800 [Mode : False.]		=> False. (332; 0,898)
				Conso de soirée en minutes > 259.800 [Mode : True.]		=> True. (51; 0,667)
		International in ["yes"] [Mode : False.]			267	
			Nb d'appels internationaux <= 2.500 [Mode : True.]			=> True. (51; 1,0)
			Nb d'appels internationaux > 2.500 [Mode : False.]		216	
				Conso internationale en minutes <= 13.100 [Mode : False.]		=> False. (173; 0,96)
				Conso internationale en minutes > 13.100 [Mode : True.]		=> True. (43; 1,0)
	Appels au service client > 3.500 [Mode : True.]				251	
		Conso de jour en minutes <= 160.200 [Mode : True.]				=> True. (102; 0,873)
		Conso de jour en minutes > 160.200 [Mode : False.]				=> False. (149; 0,745)
Conso de jour en minutes > 264.450 [Mode : True.]					211	
	Mail in ["no"] [Mode : True.]				158	
		Conso de soirée en minutes <= 187.750 [Mode : False.]				=> False. (57; 0,561)
		Conso de soirée en minutes > 187.750 [Mode : True.]				=> True. (101; 0,95)
	Mail in ["yes"] [Mode : False.]					=> False. (53; 0,887)

Ce tableau est le résultat de l'arbre de décision pour le churn

On a une première division en fonction de la consommation par jour :

- 3122 enregistrements pour une consommation <= 264,45
- 211 enregistrements pour une consommation > 264,45

(pour un total de 3122 + 211 = 3333).

L'arbre nous dit combien on a de churn à chaque étape :

- 483 au départ (pour 2850 non churn)
- 356 à consommation <= 264,45 (pour 2766 non churn)
- 127 à consommation > 264,45 (pour 84 non churn)