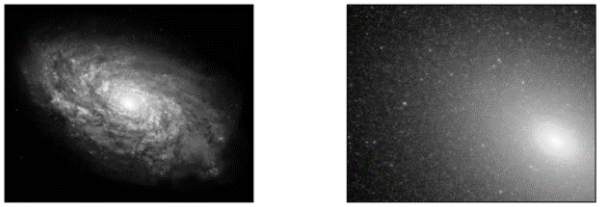
6 Classificeren

6.1 Inleiding

Classificeren is het groeperen van objecten in vooraf gedefinieerde categorieën. Voorbeelden zijn o.a. het detecteren van spam-berichten op basis van de header en content, categoriseren van cellen als kwaadaardig op basis van een MRI scan, classificeren van sterrenstelsels op basis van hun vorm.



(a) A spiral galaxy. (b) An elliptical galaxy.

Figuur 1 - Classificatie van sterrenstelsels. *Overgenomen uit Introduction to Data Mining (p. 145) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.*

6.1.1 Definitie

De input voor een classificatietaak is een verzameling records. Elk record wordt ook wel een instantie of voorbeeld genoemd, dit record wordt gekarakteriseerd door een **tupel** $(\vec{x}, y)$ met $\vec{x}$ de verzameling van attributen (vector) en y het class label.
Het class label is altijd een discreet attribuut.

Classificeren is het leren van een doelfunctie f die elke verzameling attributen $\vec{x}$ afbeeldt op een vooraf gedefinieerd class label y.

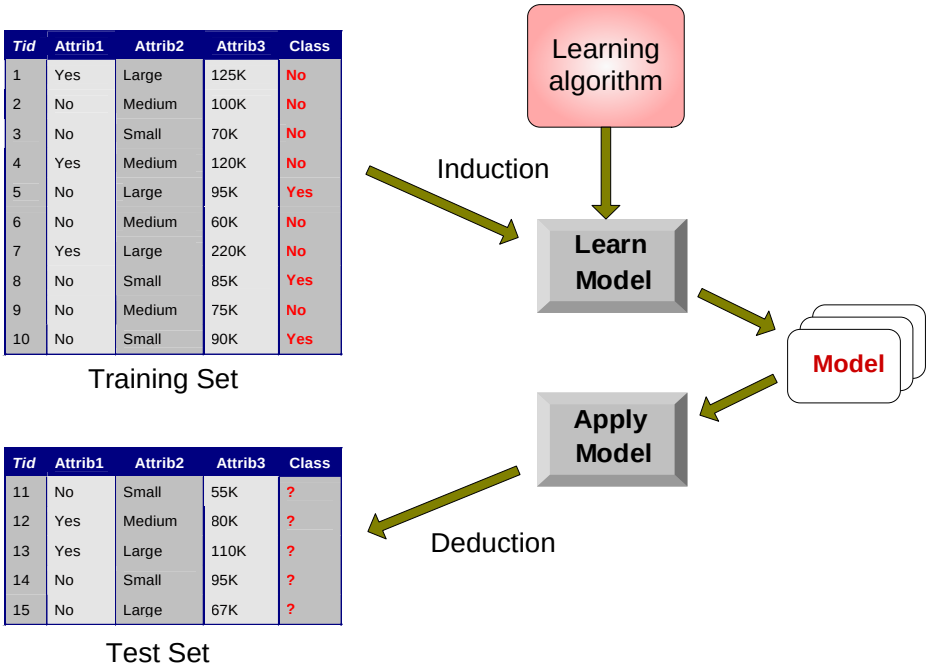
De doelfunctie wordt ook wel het **classificatiemodel** genoemd.

Classificatietechnieken zijn geschikt voor het voorspellen of beschrijven van datasets met binaire of nominale categorieën. Minder voor ordinale categorieën.

6.1.2 Oplossen van een classificatieprobleem

Een **classificatietechniek** (*classification technique of **classifier***) is een systematische benadering om een classificatiemodel op te bouwen vanuit een input dataset. Voorbeelden van technieken zijn beslissingsbomen, regel-gebaseerde classifiers, neurale netwerken, support vector machines en naïve Bayes classifier. Elke techniek gebruik een leeralgoritme om een model te zoeken die het best de relatie tussen de attributenverzameling en het class label van de input data beschrijft.

In Figuur 2 – Algemene oplossingsstrategie classificatieprobleem. *Overgenomen uit Introduction to Data Mining (p. 148) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.* krijg je een algemene oplossingsstrategie te zien voor classificatieproblemen. Eerst en vooral moet je een training set (met records waarvan het klasse label gekend is) verkrijgen. Deze training set wordt gebruikt om een classificatiemodel op te bouwen, dit model wordt dan op de test set losgelaten (in deze test set zijn er nog geen class labels toegevoegd).

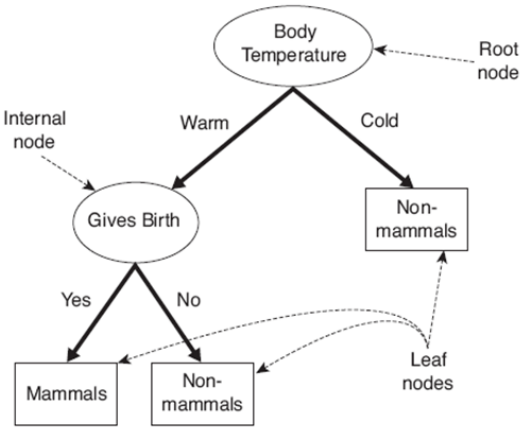


Figuur 2 – Algemene oplossingsstrategie classificatieprobleem. *Overgenomen uit Introduction to Data Mining (p. 148) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.*

6.2 Beslissingsbomen

6.2.1 Hoe werkt een beslissingsboom?

We illustreren de werking aan de hand van een voorbeeld uit de biologie.

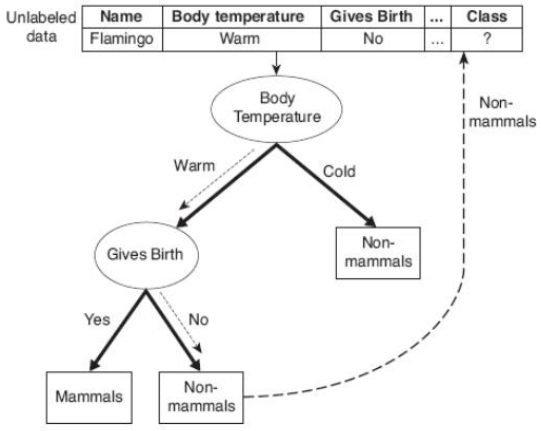


Figuur 3 - Beslissingsboom voor classificatie van zoogdieren. *Overgenomen uit Introduction to Data Mining (p. 151) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.*

In figuur 3 merk je op dat de beslissingsboom bestaat uit drie soorten nodes:

- Een **root node**, deze node heeft geen inkomende pijlen en 0 of meer uitgaande pijlen.
- Een **interne node**, deze node heeft precies één inkomende pijl en twee of meer uitgaande pijlen. Deze bevatten selectievoorwaarden om records van elkaar te onderscheiden (denk aan de *if-else* structuur).
- Een **blad**- (*leaf node*) of **eind-node**, deze heeft precies één inkomende pijl en geen uitgaande pijlen. Aan de bladnode wordt een class label toegekend.

Op basis van deze beslissingsboom kunnen we een nieuw test record classificeren (label toekennen). We starten in de root node en werken onze weg naar beneden tot we een bladnode tegenkomen. In figuur 4 wordt op deze manier aan het record van de flamingo een label toegekend.



Figuur 4 - Classificatie van een nieuw record. *Overgenomen uit Introduction to Data Mining (p. 152) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.*

6.2.2 Hoe bouwen we een beslissingsboom?

Er zijn in principe vele manieren waarop een beslissingsboom kan opgebouwd worden op basis van een gegeven verzameling van attributen. Efficiënte algoritmes bestaan om binnen aanvaardbare tijd een beslissingsboom op te bouwen.

Het algoritme van Hunt is een bekend algoritme en ligt tevens aan de basis van veel bestaande beslissingsboomalgoritmes zoals ID3, C4.5 en CART. In dit algoritme wordt een beslissingsboom opgebouwd door op een recursieve manier de training records te partitioneren in zuiverdere deelverzamelingen.

Algoritme van Hunt
Stel D_t de verzameling van training records geassocieerd met node t en $y = \{y_1, y_2, \dots, y_c\}$ de klasse labels. De recursieve definitie van het algoritme van Hunt is dan: Stap 1: Als alle records in D_t tot dezelfde klasse y_t behoren, dan is t een bladnode met label y_t. Stap 2: Als D_t records bevat die tot meer dan één class behoren wordt een attribuuttestvoorwaarde geselecteerd om de records te partitioneren in kleinere deelverzamelingen. Een "*child*" node wordt gemaakt voor elke uitkomst van de testvoorwaarde en de records in D_t worden op basis van de uitkomst verdeeld onder de *children*. Het algoritme wordt dan herhaald op elke *child* node.

Voorbeeld 1:

We willen voorspellen, op basis van de gegevens in onderstaande dataset, als een aanvrager voor een lening op correcte wijze de lening terugbetaalt of dat hij/zij een wanbetaler wordt.

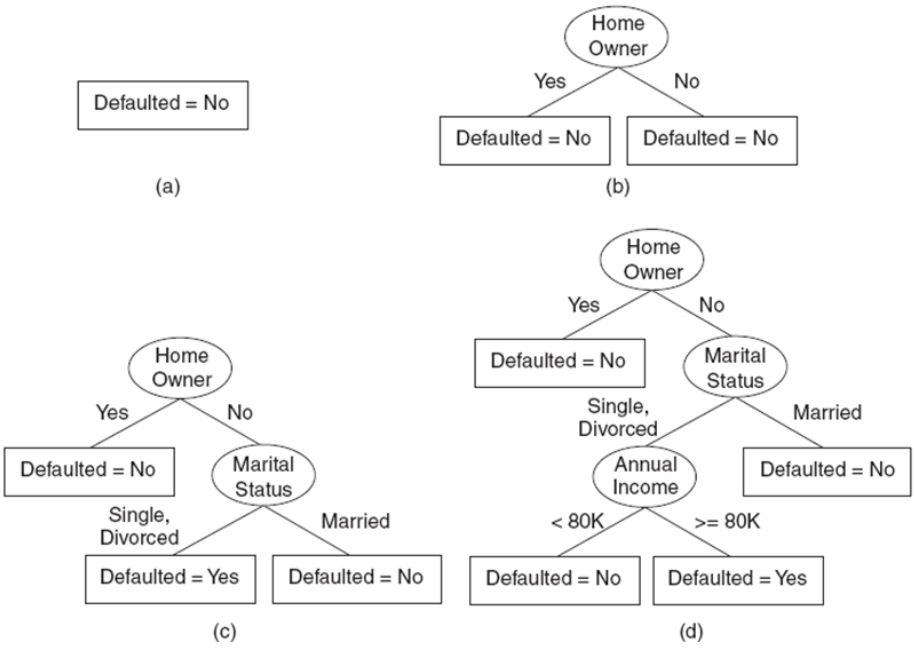
	binary	categorical	continuous	class
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Figuur 5 - Training set aanvrager lening. *Overgenomen uit Introduction to Data Mining (p. 153) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.*

De initiële boom voor het classificatieprobleem bevat één enkele node met al class label `Defaulted = No`,
dit betekent dat de meeste leners hun schulden tijdig en correct terugbetalen.

De records worden vervolgens onderverdeeld op basis van de `Home Owner` testvoorwaarde. Waarom precies `Home Owner` wordt gebruikt komt later aan bod.

Het algoritme van Hunt wordt vervolgens op elk child van de root node recursief uitgevoerd tot we de uiteindelijke boom krijgen die in figuur 6 te zien is.



Figuur 6 - Volledige beslissingsboom volgens het algoritme van Hunt. *Overgenomen uit Introduction to Data Mining (p. 154) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.*

Een beslissingsboomalgoritme (zoals b.v. Hunt) moet wel rekening houden met volgende issues:

1) Hoe moeten de training records opgesplitst worden? Hoe specificeren we de **attribuuttestvoorwaarde** en hoe bepalen we de beste split?
2) Hoe en wanneer moet de opsplitsingsprocedure **stoppen**?

6.2.3 Het bepalen van de attribuuttestvoorwaarde

Binaire attributen

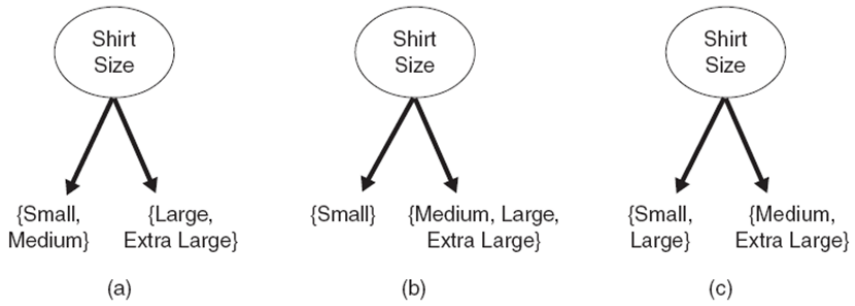
De testvoorwaarde genereert twee mogelijke uitkomsten.

Nominale attributen

Omdat een nominaal attribuut meerdere waarden kan hebben kan de testvoorwaarde op meerdere manieren worden weergegeven, het aantal uitkomsten hangt af van het aantal verschillende waarden voor de corresponderende attributen. Als een attribuut zoals *burgerlijke stand* drie verschillende waarden heeft – *alleenstaand, gehuwd* of *gescheiden* – zal de testvoorwaarde een drievoudige split voorstellen.

Ordinale attributen

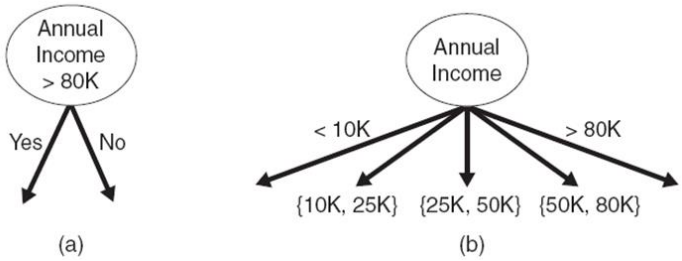
Ordinale attributen kunnen zowel binaire als meervoudige splits voorstellen. Deze attributen kunnen gegroepeerd worden zolang de groepering de orde-eigenschappen van de attribuutwaarden niet in het gedrang brengt. In figuur 7-a en 7-b wordt een correcte, orde bewarende, split voorgesteld. In figuur 7-c niet, waarom?



Figuur 7 - Verschillende vormen van een split voor ordinale attributen. *Overgenomen uit Introduction to Data Mining (p. 157) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.*

Continue attributen

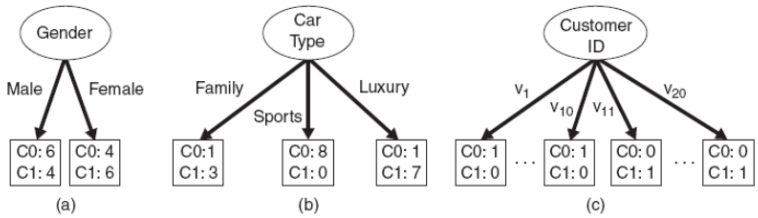
Voor continue attributen zal de testvoorwaarde uitgedrukt worden als een vergelijkingstest $(A < v)$ of $(A \geq v)$ met binaire uitkomsten of een bereik met uitkomsten van de vorm $v_i \leq A \leq v_i + 1$ voor $i = 1, \ldots, k$. Het verschil wordt weergegeven in figuur 8.



Figuur 8 - Testvoorwaarden voor continue attributen. *Overgenomen uit Introduction to Data Mining (p. 157) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.*

6.2.4 Maten voor het bepalen van de beste splitsing

Stel $p(i|t)$ de fractie van records die behoren tot de klasse i in een gegeven node t. Soms wordt de referentie naar de node weggelaten en schrijven we de fractie als p_i. In een twee-klassenprobleem wordt de verdeling van de klasse geschreven als (p_0, p_1), met $p_1 = 1 - p_0$. Om dit te illustreren beschouw de testvoorwaarden in figuur 9. De klassenverdeling voor splitsing is $(0{,}5; 0{,}5)$ omdat er een gelijk aantal records zijn die tot beide klassen behoren. Splitsen we de data gebruik makend van het `Gender` attribuut, dan zijn de klassenverdelingen voor de child nodes $(0{,}6; 0{,}4)$ en $(0{,}4; 0{,}6)$ respectievelijk. De klassen zijn niet meer gelijk verdeeld, maar de child nodes bevatten nog steeds records van beide klassen. Splitsen we echter op het tweede attribuut, `Car Type`, dan krijgen we een meer zuivere opdeling.



Figuur 9 - Meervoudige vs. binaire splitsing. *Overgenomen uit Introduction to Data Mining (p. 158) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.*

De maten die gebruikt worden om de beste splitsing te bepalen zijn vaak gebaseerd op de **graad van onzuiverheid** (*degree of impurity*) van de child nodes. Hoe lager de onzuiverheidsgraad is, hoe meer scheef de klassenverdeling zal zijn.

Zo zal een node met klassenverdeling (0,1) nul onzuiverheid hebben, terwijl een node met een klassenverdeling van (0,5; 0,5) de hoogste onzuiverheid heeft. De meest gebruikte maten voor onzuiverheid zijn: (c stelt het aantal klassen voor)

Entropie

$$\text{Entropie}(t) = H(t) = -\sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t)$$

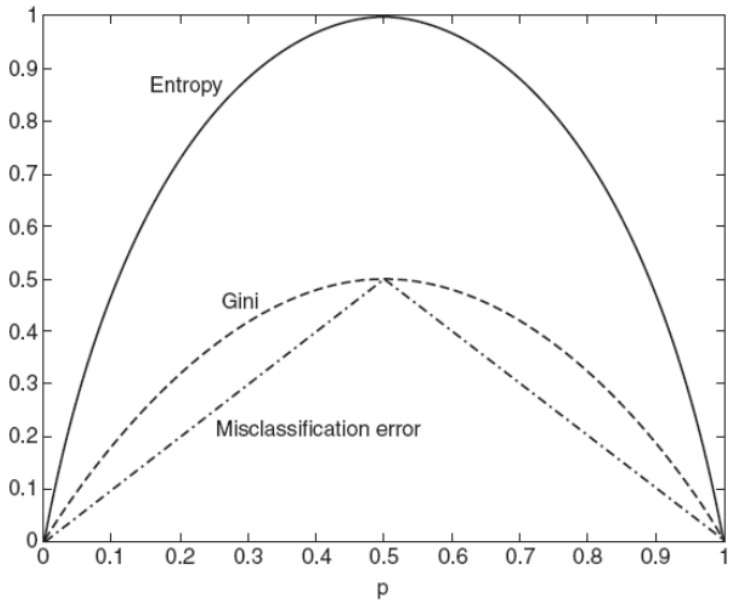
Gini

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2$$

Classificatiefout

$$\text{classificatiefout}(t) = 1 - \max_i [p(i|t)]$$

In figuur 10 worden de waarden van de onzuiverheidsmaten voor binaire classificatieproblemen weergegeven. p refereert naar de fractie van records die behoren tot één van de twee klassen.



Figuur 10 - Vergelijking van de onzuiverheidsmaten voor binaire classificatieproblemen.
Overgenomen uit Introduction to Data Mining (p. 158) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.

Merk op dat alle drie de maten hun maximum bereiken als de klassenverdeling uniform is (dus als $p = 0{,}5$). De minimale waarden worden bereikt wanneer alle records tot dezelfde klasse behoren (dus als p gelijk is aan 0 of 1).

Voorbeeld 1:

Node N_1	Aantal
Klasse=0	0
Klasse=1	6

Gini $= 1 - (0/6)^2 - (6/6)^2 = 0$
Entropie $= -(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0$
Fout $= 1 - \max[0/6, 6/6] = 0$

Voorbeeld 2:

Node N_2	Aantal
Klasse=0	1
Klasse=1	5

Gini $= 1 - (1/6)^2 - (5/6)^2 = 0{,}278$
Entropie $= -(1/6)\log_2(1/6) - (5/6)\log_2(5/6) = 0{,}650$
Fout $= 1 - \max[1/6, 5/6] = 0{,}167$

Voorbeeld 3:

Node N_3	Aantal
Klasse=0	3
Klasse=1	3

Gini $= 1 - (3/6)^2 - (3/6)^2 = 0{,}5$
Entropie $= -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$
Fout $= 1 - \max[3/6, 3/6] = 0{,}5$

De node in voorbeeld 1 heeft de laagste graad van onzuiverheid, gevolgd door voorbeeld 2 en 3.

Om te bepalen hoe goed een testvoorwaarde presteert moeten we de graden van onzuiverheid van de parent node (vóór splitsing) met de graad van onzuiverheid van de child nodes (na splitsing). Hoe groter het verschil, hoe beter de testvoorwaarde. De **informatiewinst Δ**, is een kenmerk dat gebruikt kan worden om te bepalen hoe goed een splitsing is:

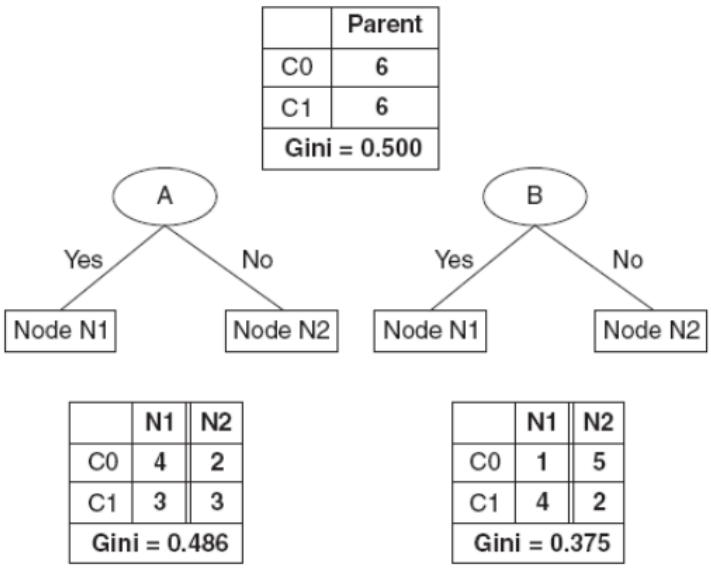
$$\Delta = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j)$$

met $I(\cdot)$ **de onzuiverheidsgraad van een gegeven node**, N het totale aantal records in de parent node, k is het aantal attribuutwaarden en $N(v_j)$ het aantal records van de child node, v_j.

Beslissingsboomalgoritmes kiezen de testvoorwaarde die de winst maximaliseert. Wordt entropie gebruikt als maat voor onzuiverheid, dan spreken we over de informatiewinst Δ_{info}.

6.2.5 Splitsen van binaire attributen

Beschouw het schema in figuur 11. Veronderstel dat er twee manieren zijn om de data op te splitsen in kleinere deelverzamelingen. Vooraleer we splitsen is de Gini-index 0,5 want er zijn een gelijk aantal records voor beide klassen. Als attribuut A gekozen wordt om de data te splitsen, wordt de Gini-index voor node N_1 0,4898 en voor node N_2 0,480. Het gewogen gemiddelde van de Gini-index voor de afstammelingen is dan $(7/12).0{,}4898 + (5/12).0{,}480 = 0{,}486$. Een analoge berekening gaat op als attribuut B gekozen wordt. Maar omdat B een lagere Gini-index heeft verkiezen we deze splitsing boven die op basis van A.

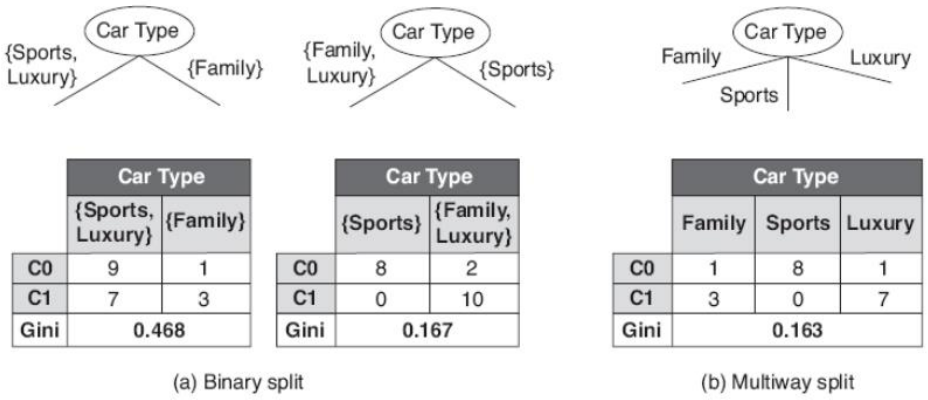


Figuur 11 - Splitsen van binaire attributen. *Overgenomen uit Introduction to Data Mining (p. 161) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.*

6.2.6 Splitsen van nominale attributen

Een nominaal attribuut kan zowel binaire of meervoudige splitsingen produceren, zie ook figuur 12. De berekening van de Gini-index voor een binaire split is gelijkaardig aan die van binaire attributen. Voor de eerste binaire groepering van het `Car Type` attribuut is de Gini-index van {`Sports, Luxury`} gelijk aan 0,4922 en de Gini-index van {`Family`} is 0,3750. Het gewogen gemiddelde is dan

$$16/20 \,.\, 0{,}4922 + 4/20 \,.\, 0{,}3750 = 0{,}468$$



	Car Type	
	{Sports, Luxury}	{Family}
C0	9	1
C1	7	3
Gini	0.468	

	Car Type	
	{Sports}	{Family, Luxury}
C0	8	2
C1	0	10
Gini	0.167	

	Car Type		
	Family	Sports	Luxury
C0	1	8	1
C1	3	0	7
Gini	0.163		

(a) Binary split

(b) Multiway split

Figuur 12 - Splitsing van nominale attributen. *Overgenomen uit Introduction to Data Mining (p. 161) door Tan P., Steinbach M., Kumar V., 2006, Pearson Education. Copyright 2006 Pearson Education Inc.*

Analoog is voor de tweede binaire groepering van {`Sports`} en {`Family, Luxury`} het gewogen gemiddelde van de Gini-index gelijk aan 0,167. De twee groepering heeft een lagere Gini-index omdat de corresponderende deelverzamelingen meer zuiver zijn.

Voor de meervoudige splitsing is de Gini-index nodig van elke attribuutwaarde. Omdat, $\text{Gini}(\{\texttt{Family}\}) = 0{,}378$, $\text{Gini}(\{\texttt{Sports}\}) = 0$ en $\text{Gini}(\{\texttt{Luxury}\}) = 0{,}219$, het gewogen gemiddelde is dan:

$$4/20 \,.\, 0{,}375 + 8/20 \,.\, 0 + 8/20 \,.\, 0{,}219 = 0{,}163$$

De meervoudige splitsing heeft een lagere Gini-index en dit is niet verrassend want de tweevoudige splitsing voegt sommige uitkomsten die apart staan in de meervoudige splitsing samen, wat resulteert in minder zuivere deelverzamelingen.

6.2.7 Oefeningen

1) Beschouw volgende training records van een binair classificatieprobleem.

Customer ID	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1

a. Bereken de Gini-index voor de volledige verzameling aan training records.

b. Bereken de Gini-index voor het `Customer ID` en het `Gender` attribuut.

c. Bereken de Gini-index voor het `Car Type` en `Shirt Size` attribuut gebruik makend van meervoudige splitsing.

d. Welk van de attributen verkies je?

2) Beschouw volgende training records van een binair classificatieprobleem.

Instance	a_1	a_2	a_3	Target Class
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	−
4	F	F	4.0	+
5	F	T	7.0	−
6	F	T	3.0	−
7	F	F	8.0	−
8	T	F	7.0	+
9	F	T	5.0	−

a. Wat is de entropie van de verzameling aan training records rekening houdend met de positieve klasse.?

b. Wat is de informatiewinst van a_1 en a_2 in relatie tot de training records?

c. Voor a_3, een continu attribuut, bereken de informatiewinst voor elke mogelijke splitsing.

d. Wat is de beste splitsing (tussen a_1, a_2 en a_3) rekening houdend met de informatiewinst?

e. Wat is de beste splitsing (tussen a_1 en a_2) rekening houdend met de classificatiefout?

f. Wat is de beste splitsing (tussen a_1 en a_2) rekening houdend met de Gini-index?

3) Stel dat je op een onbewoond eiland zit. Er groeien vele soorten paddenstoelen op het eiland, er is geen andere voedselbron. Sommige van de paddenstoelen zijn giftig, andere niet (dit is d.m.v. trial en error onderzocht door je voorgangers). Je bent de enige overlevende en hebt enkel onderstaande data om je beslissing te maken:

Example	*NotHeavy*	*Smelly*	*Spotted*	*Smooth*	*Edible*
A	1	0	0	0	1
B	1	0	1	0	1
C	0	1	0	1	1
D	0	0	0	1	0
E	1	1	1	0	0
F	1	0	1	1	0
G	1	0	0	1	0
H	0	1	0	0	0
U	0	1	1	1	?
V	1	1	0	1	?
W	1	1	0	0	?

Je wee dat paddenstoelen A, B en C eetbaar zijn, D tot en met H niet. Wat zou je dan durven besluiten over U, V en W? Denk eraan je leven kan er van afhangen…

a. Wat is de entropie van "`Edible`"?
b. Welk attribuut verkies je als root node van de beslissingsboom? (Je kan dit in principe zelfs visueel uit de dataset halen…)
c. Wat is e informatiewinst van het attribuut dat je in b. verkiest?
d. Bouw een beslissingsboom om paddenstoelen als giftig of niet te classificeren.
e. Wat beslis je dan over U, V en W?

4) Van een 800-tal exoplaneten werd nagegaan of zij bewoonbaar (`habitable`) zijn of niet. Men onderzoekt dit op basis van een aantal features zoals "`Size`" en "`Orbit`". Elke rij geeft ook weer of de planeet "`Habitable`" is of niet. Zo zijn er bijvoorbeeld in de tabel 20 grote "Big" planeten die dicht "`Near`" rond hun ster draaien die "`habitable`" zijn.

Size	Orbit	Habitable	Count
Big	Near	Yes	20
Big	Far	Yes	170
Small	Near	Yes	139
Small	Far	Yes	45
Big	Near	No	130
Big	Far	No	30
Small	Near	No	11
Small	Far	No	255

Bouw een beslissingsboom op basis van deze data, maak hiervoor gebruik van de classificatiefout als informatiewinstcriterium.