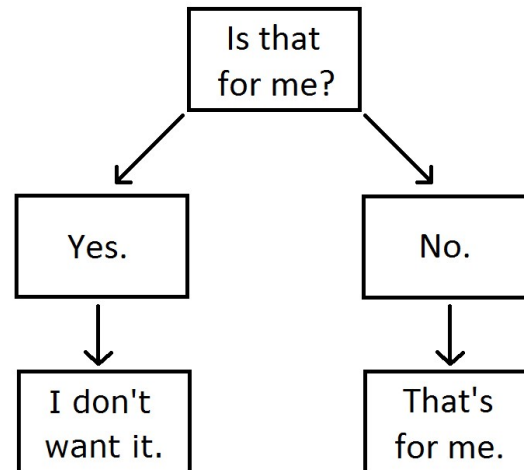


Data-Science 1

beslissingsbomen

My Cat's Decision-Making Tree.



Inhoud

- voorbeeld: ad eater
- voorbeeld: Simpsons
- het ID3 algoritme
- Implementaties
 - ID3
 - ID3Estimator
 - CART

Voorbeeld:
ad eater

Ad eater

- webpagina's bevatten images
- sommige images zijn reclame, andere niet
- kan ik automatisch detecteren wat reclame is?

Ad eater



Ad eater

- probleem: aan de hand van welke parameters kan je bepalen of een beeld reclame is?
 - afmetingen, positie, kleur
 - html attributen
 - alt tag
 - url van het beeld
 - url van de link (als clickable)
 - ...

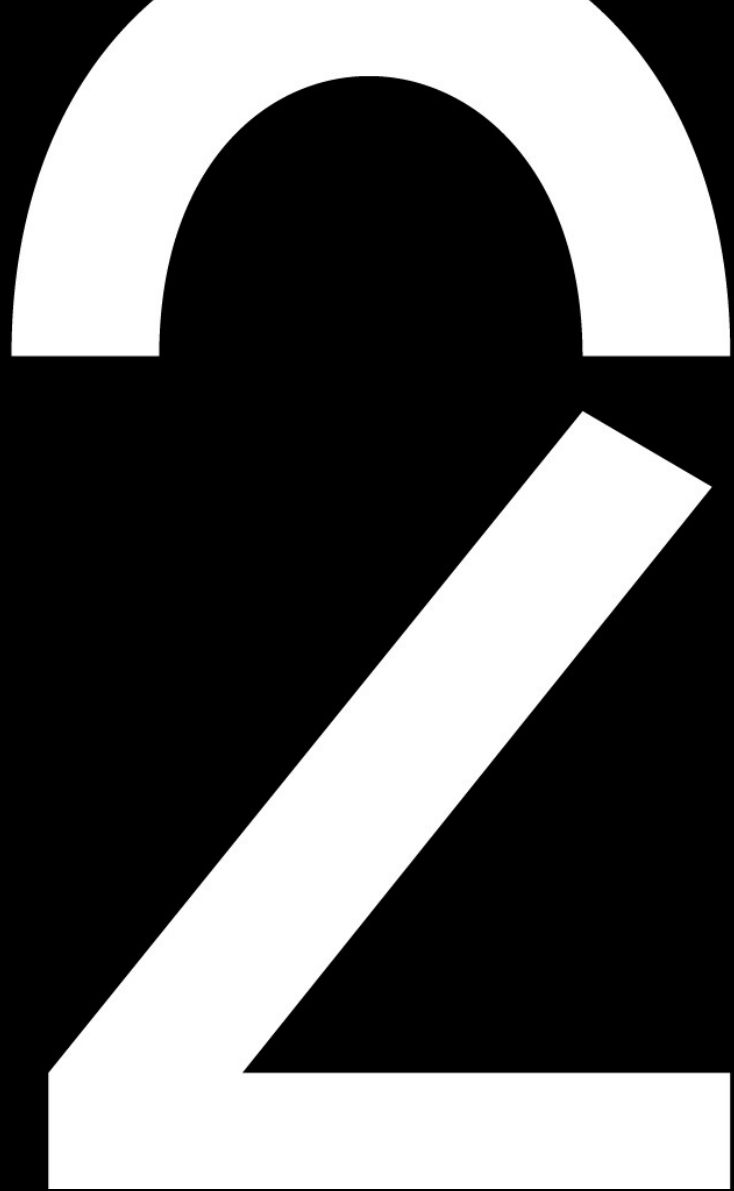
Ad eater










- oplossing
 - zoek een aantal voorbeelden (3279)
 - som alle eigenschappen op (1558)
 - zet alles in een tabel met 3279 rijen en 1558 kolommen
 - bepaal handmatig of deze voorbeelden reclame zijn of niet
 - laat computer hieruit "leren"

Ad eater

- resultaat: regels
 - als
 - aspect ratio > 4.5833
 - alt doesn't contain "to"
 - alt contains "click+here"
 - url doesn't contain "http+www"
 - dan: reclame!

Voorbeeld:
Simpsons



	Naam	Haarlengte (inch)	Gewicht (lbs)	Leeftijd (jaren)	Geslacht
	Homer	0	250	36	M
	Marge	10	150	34	V
	Bart	2	90	10	M
	Lisa	6	78	8	V
	Maggie	4	20	1	V
	Abe	1	170	70	M
	Selma	8	160	41	V
	Otto	10	180	38	M
	Krusty	6	200	45	M



Comic

8











290

38

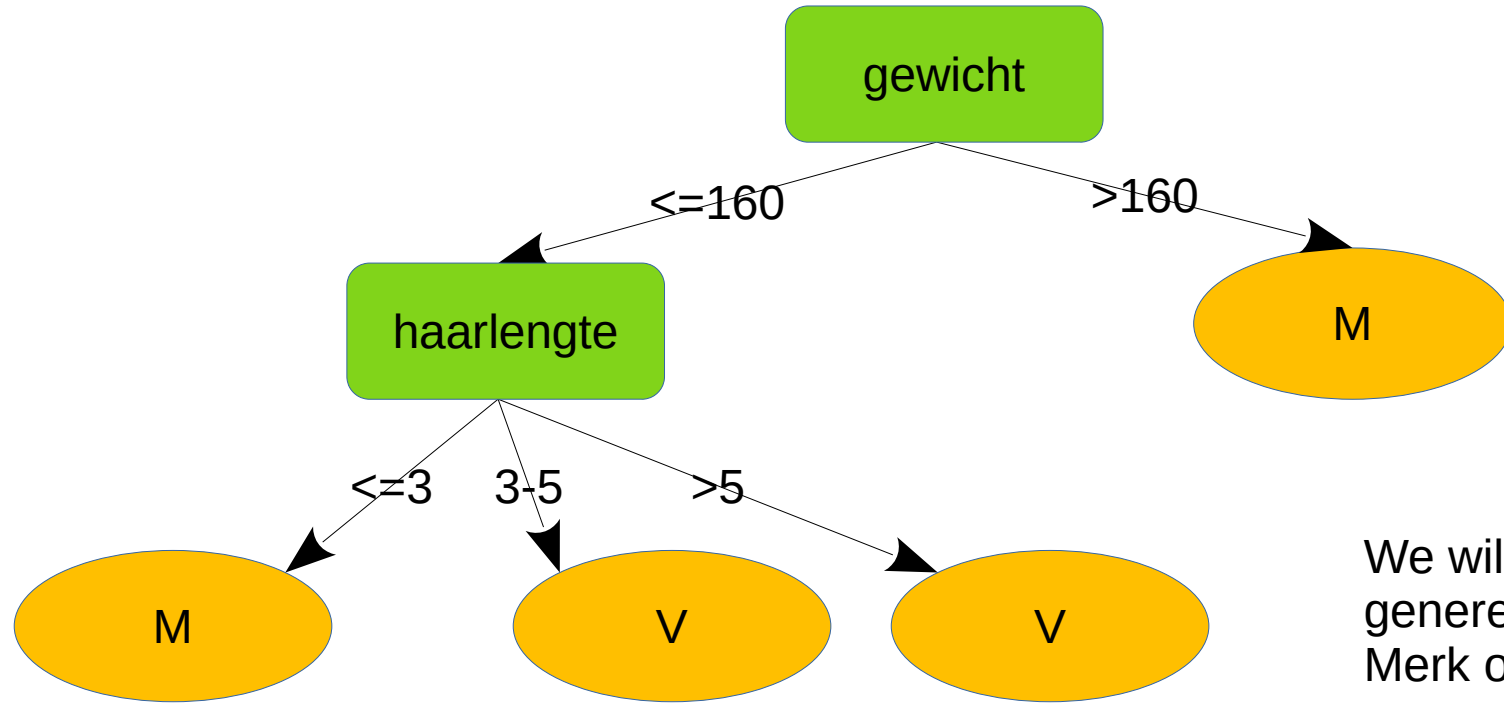
??

Stap 1: discretiseren

- variabelen moeten discreet zijn (continue variabelen zien we later)
- niet te veel verschillende waarden
- we zetten alles om naar nominaal en ordinaal meetniveau
 - hoe kan je een continue variabele omzetten naar een discrete met ordinaal meetniveau? (hint: zie frequenties)

	Naam	Haarlengte (inch)	Gewicht (lbs)	Leeftijd (jaren)	Geslacht
	Homer	<=3	>160	30-40	M
	Marge	>5	<=160	30-40	V
	Bart	<=3	<=160	<=30	M
	Lisa	>5	<=160	<=30	V
	Maggie	3-5	<=160	<=30	V
	Abe	<=3	>160	>40	M
	Selma	>5	<=160	>40	V
	Otto	>5	>160	30-40	M
	Krusty	>5	>160	>40	M
	Comic	>5	>160	30-40	??

Beslissingsboom: resultaat



We willen dit laten genereren.
Merk op: leeftijd speelt geen rol!

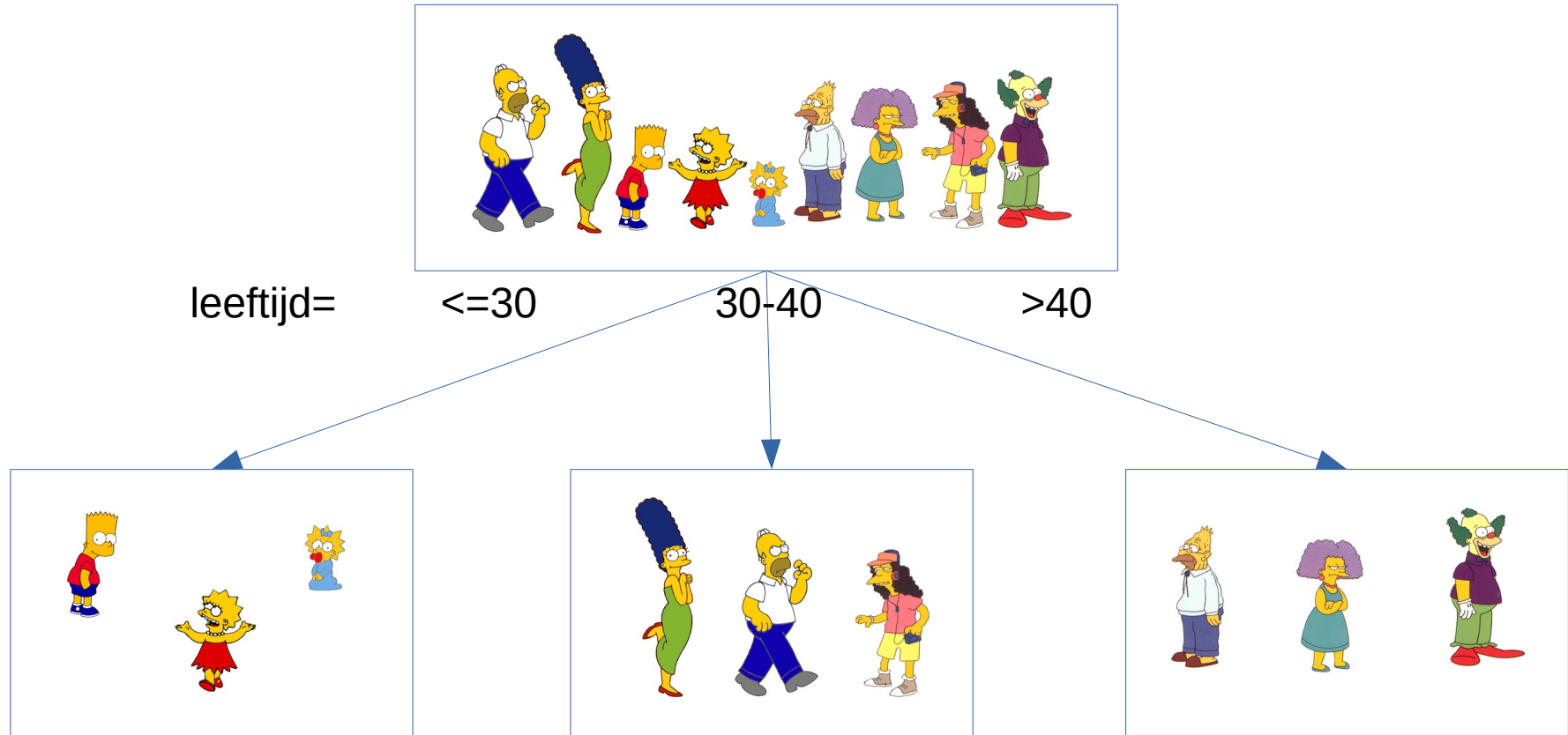
ID3

Hoe een boomstructuur vinden?

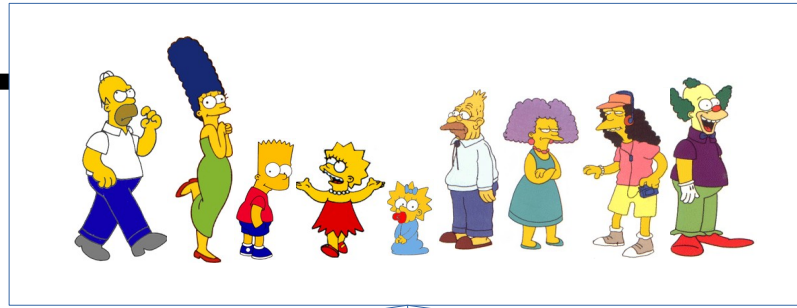
- kies een kolom
- maak een node voor deze kolom
- maak pijlen naar kind-nodes voor iedere mogelijke waarde van deze kolom (vandaar: niet te veel verschillende waarden per kolom)
- per pijl maak je een subtabel waarin enkel die rijen voorkomen met de gekozen waarde
- doe het algoritme recursief voor alle subtabellen
- als alle rijen van de tabel een zelfde uitkomst hebben, stop dan

Voorbeeld Simpsons

- kies kolom "leeftijd"
- leeftijd heeft 3 mogelijkheden: ≤ 30 , 30-40 of > 40
- maak dus een node "leeftijd" met 3 kinderen
- bereken voor het eerste kind een tabel met alle rijen waarbij leeftijd ≤ 30
- bereken voor het tweede kind een tabel met alle rijen waarbij leeftijd 30-40
- bereken voor het derde kind een tabel met alle rijen waarbij leeftijd > 40



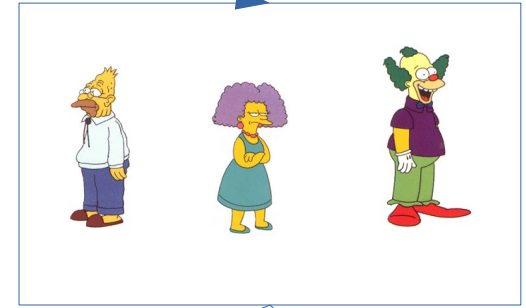
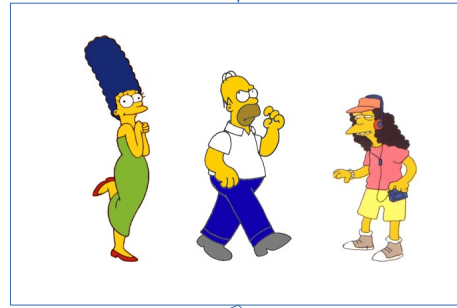
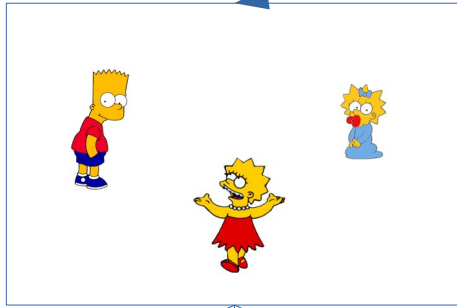
overal zitten nog mannen en vrouwen



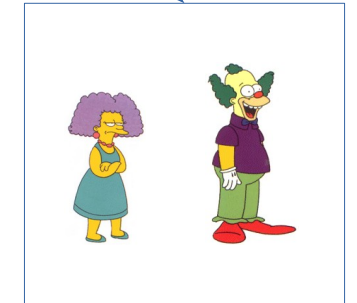
leeftijd=

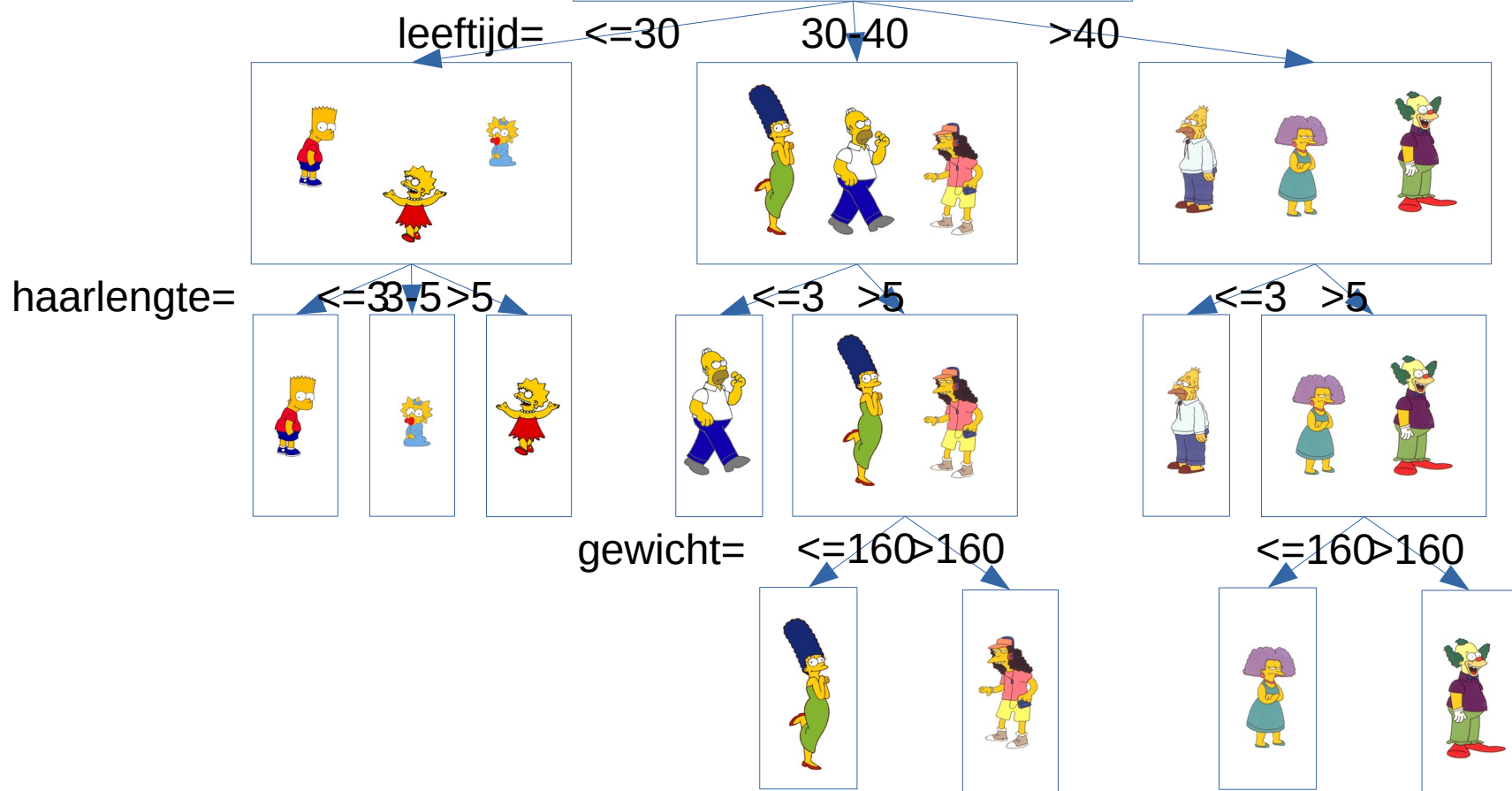
 ≤ 30

30-40

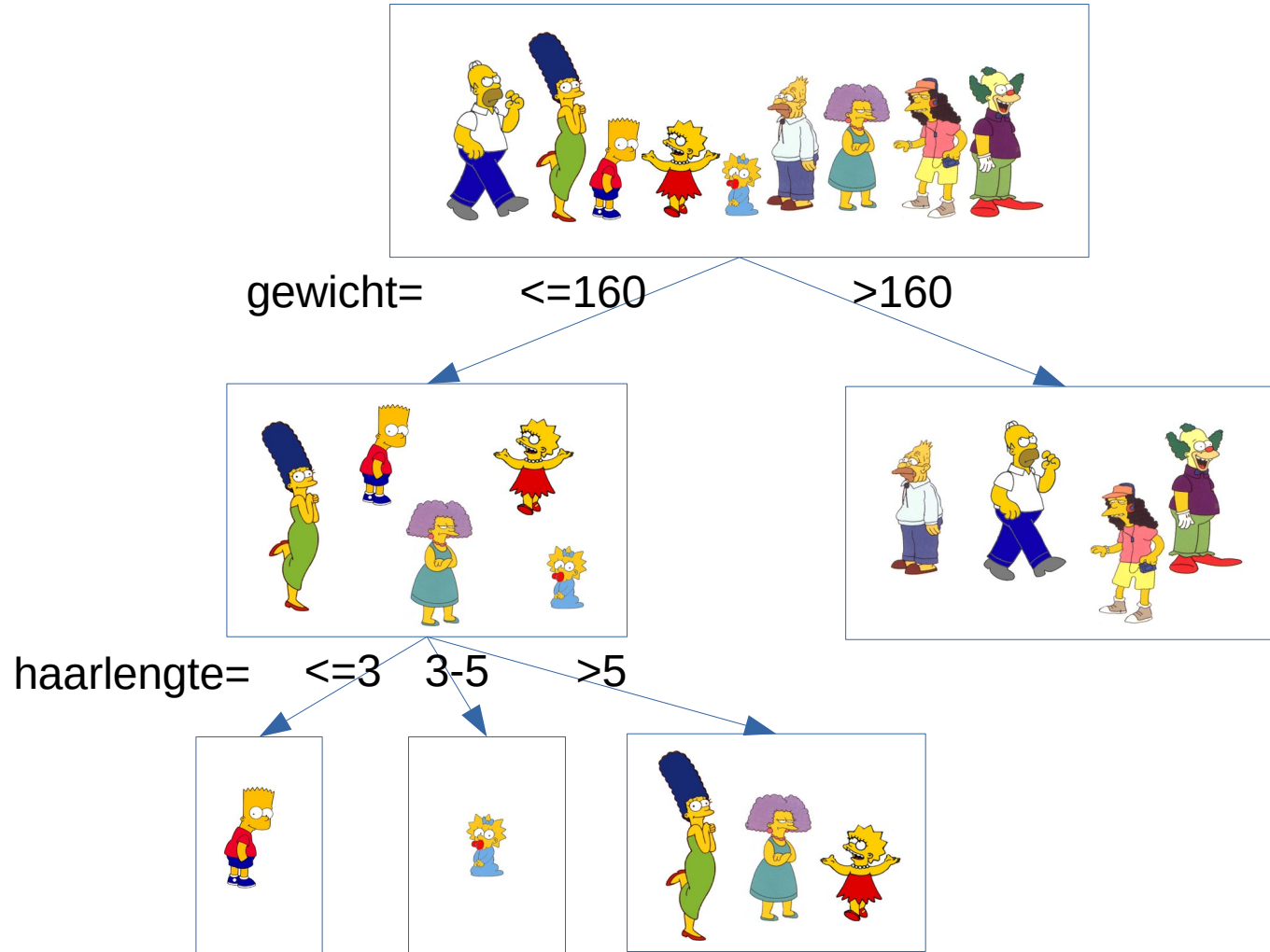
 > 40 

haarlengte=

 ≤ 3 3-5 > 5 ≤ 3 > 5 ≤ 3 > 5 



Als we begonnen met gewicht^{20/35}



Keuze van de kolom

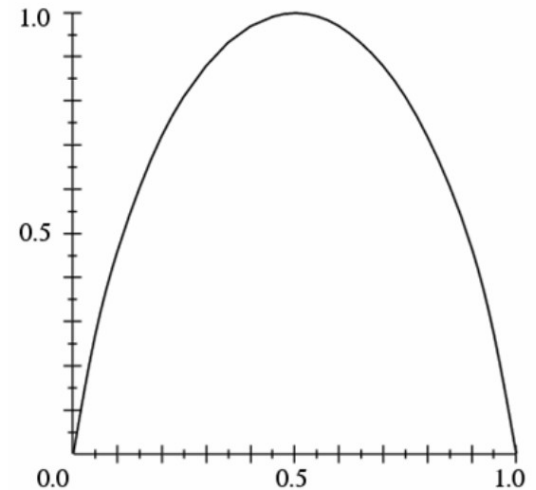
- waarom begonnen we met "leeftijd"?
 - “gewicht” zou beter geweest zijn
- hoe kunnen we dit weten?
- we zoeken een zo klein mogelijke boomstructuur

Keuze van de kolom

- zoek de kolom met het grootste onderscheidend vermogen
- men noemt dit "**information gain**"
- $Gain(kolom) = E(tabel) - \sum_{waarden\ van\ kolom} (p/n) \cdot E(subtabel)$
 - subtabel bevat enkel een bepaalde waarde voor de gegeven kolom
 - p is het aantal rijen in de subtabel
 - n is het totaal aantal rijen
 - E(tabel) is de entropie van die tabel

Entropie

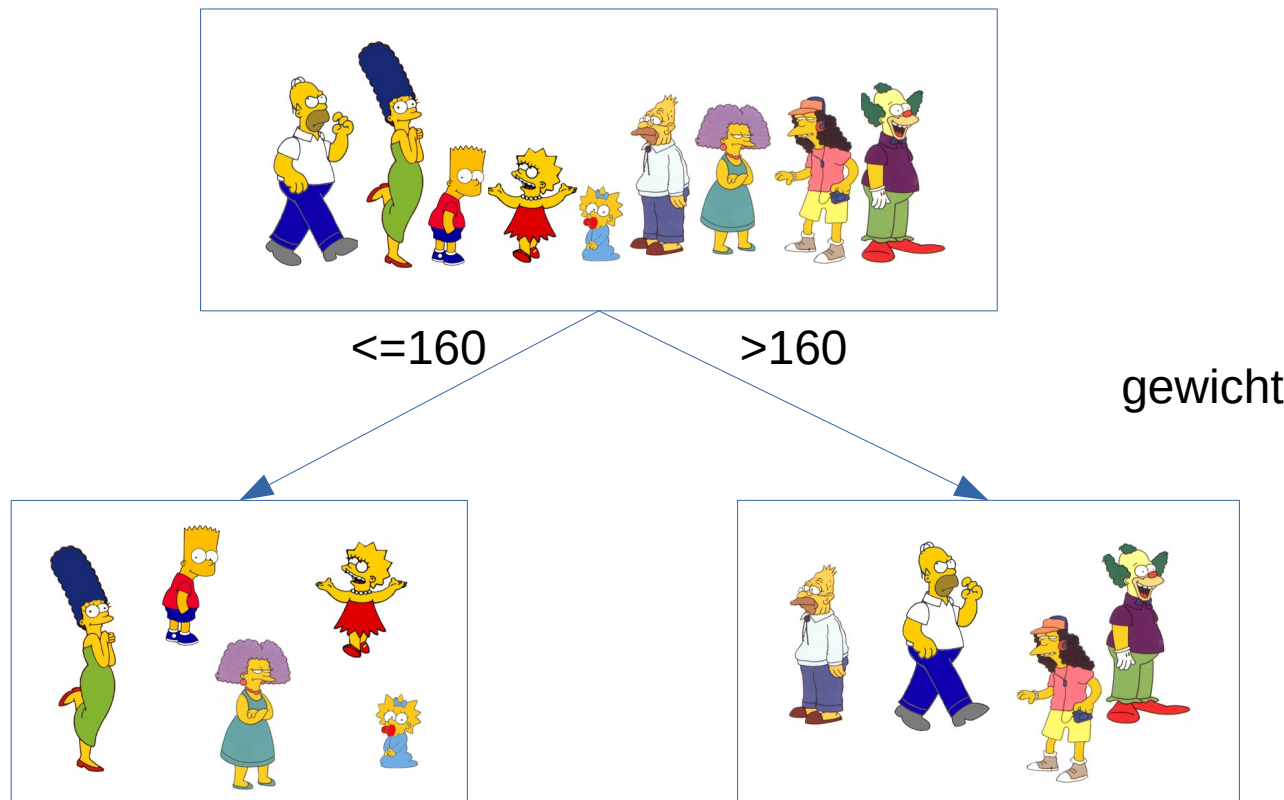
- maat voor "chaos" in de kolom met resultaten (meestal laatste kolom)
- allemaal dezelfde waarde: $E(\text{tabel})=0$
- als alle waarden evenveel voorkomen: $E(\text{tabel})=1$
- formule:
$$E(\text{tabel}) = \sum_{\text{waarden}} -(p/n) \cdot \log_2(p/n)$$
 - p = aantal rijen met de gegeven waarde in de eindkolom
 - n = aantal rijen in de tabel
- voorbeeld (simpsons heeft 4 vrouwen en 5 mannen):
$$E(\text{simpsons}) = -(4/9) \cdot \log_2(4/9) - (5/9) \cdot \log_2(5/9)$$



$$\text{Gain}(\text{gewicht}) = 0,991 - (5/9) \cdot 0,722 - (4/9) \cdot 0 = 0,59$$

24/35

$$E(\text{simpsons}) = -(4/9) \cdot \log_2(4/9) - (5/9) \cdot \log_2(5/9) = 0,991$$



$$E(\text{gewicht} \leq 160) = -(4/5) \cdot \log_2(4/5) - (1/5) \cdot \log_2(1/5) = 0,722$$

$$E(\text{gewicht} > 160) = -(0/4) \cdot \log_2(0/4) - (4/4) \cdot \log_2(4/4) = 0$$

Andere Gains

- we bekomen volgende gains (oefening!):
 - $\text{Gain}(\text{haarlengte}) = 0,452$
 - $\text{Gain}(\text{gewicht}) = 0,590$
 - $\text{Gain}(\text{leeftijd}) = 0,073$
- gewicht heeft hoogste gain
- opmerkingen
 - $\text{Gain}(\text{geslacht}) = 0,991$
 - hoogst mogelijke gain = entropie

Implementaties



ID3: zelf gemaakt

- je kan het ID3 algoritme eenvoudig zelf programmeren (zie python code)
- tabel mag enkel discrete waarden bevatten
- functies
 - `calculate_entropy(target)`
 - `calculate_information_gain(data, column_name, target)`
 - `id3(data, target)`

Problemen

- als waarden continu zijn
 - moeilijk om klassen op voorhand te bepalen
- als waarden ontbreken...
- als er inconsistenties zijn (twee rijen met zelfde waarden en ander resultaat)
- als de boom te groot wordt: minder overzichtelijk
 - dus eventueel boom ergens afkappen

Andere algoritmes

- andere algoritmes kunnen:
 - continue variabelen automatisch opsplitsen
 - omgaan met ontbrekende waarden
 - de boom afkappen indien te complex

ID3Estimator

- installeer de library “decision-tree-id-fork”
 - versie 0.0.15
- deze kan zowel discrete als continue variabelen aan
- zie python voor gebruik

CART

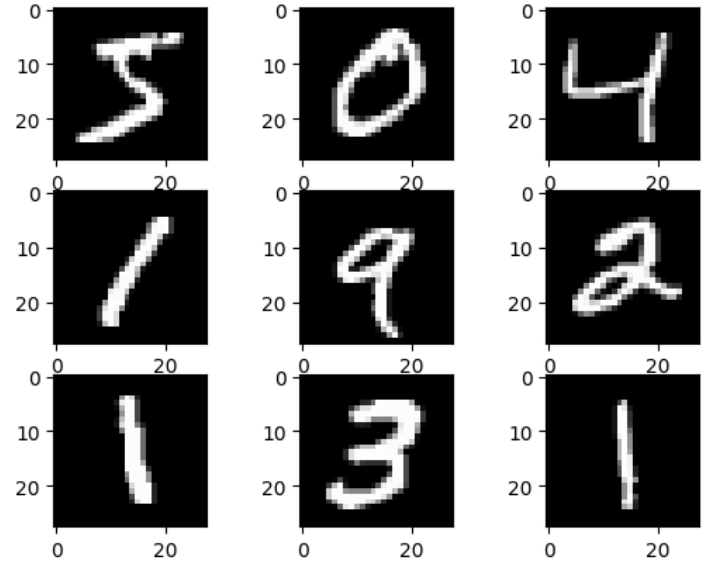
- onderdeel van sklearn (Scikit-Learn)
- is heel snel
- kan enkel continue variabelen aan
 - gebruik volgnummer voor ordinale variabelen
 - gebruik “one-hot encoding” voor nominale variabelen
- gebruik “random_state” parameter om algoritme deterministisch te maken
- zie python

Toepassingen

- voorspellen van borstkanker
- credit approval
- leeftijd dieren schatten adv eigenschappen
- spam detecteren
- inzicht krijgen in factoren die inkomen bepalen
- ...

Karakter herkenning

- download mnist databank
- maak een decision tree die kan voorspellen welk cijfer er in een afbeelding staat
- test uit op de test-databank
- hoe accuraat is deze voorspelling?



Oefeningen

Oefeningen

- beslissingsbomen
 - simpsons
 - play ball
 - scores voor vakken voorspellen
 - bank