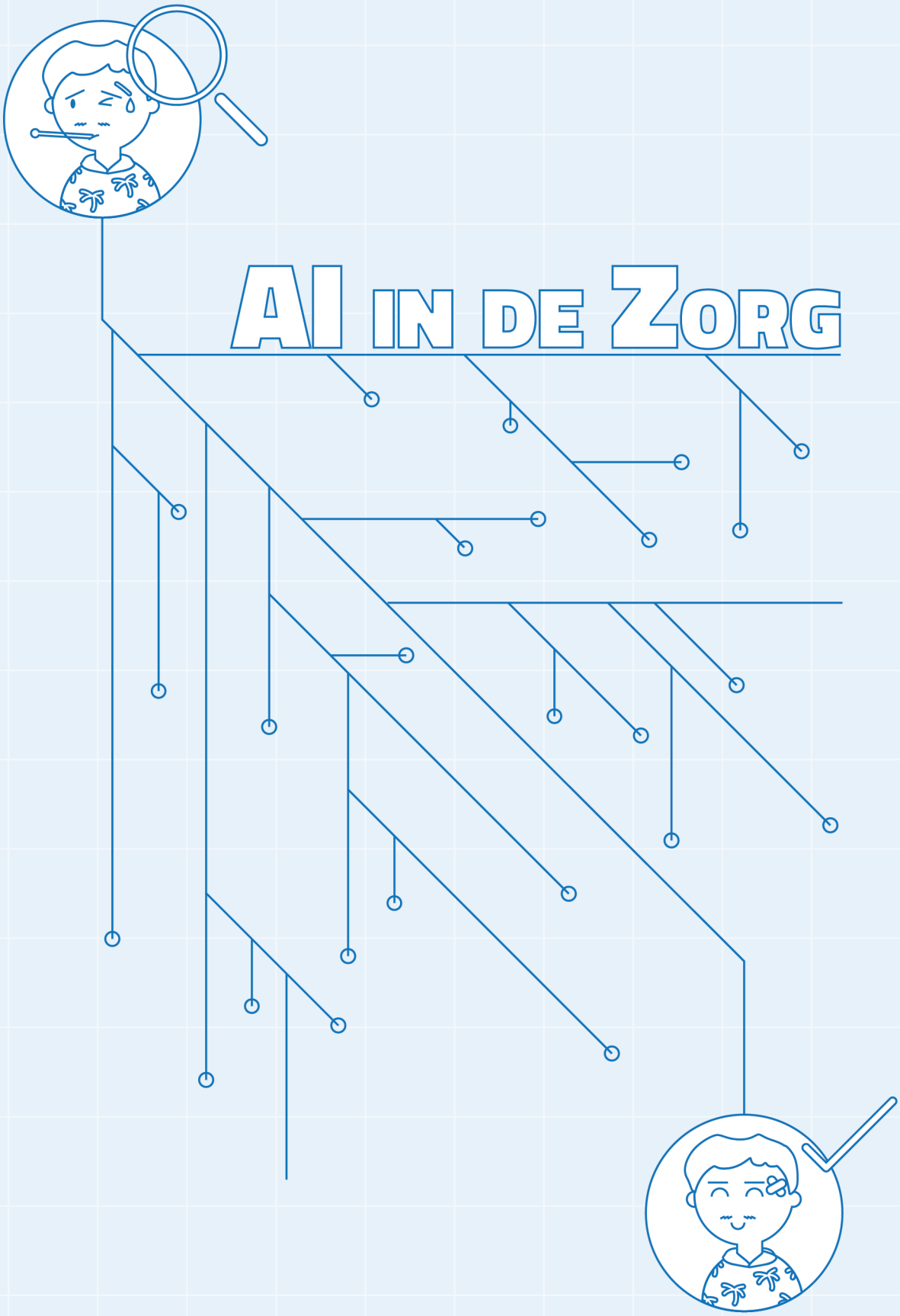


AI in de Zorg

Leerlingencursus - Finaliteit doorstroom

Auteur: Natacha Gesquière
Coauteurs: Tom Neutens, Francis wyffels

Natacha Gesquière

AI in de Zorg
Beslissingsbomen in de zorgsector

Leerlingencursus - Finaliteit doorstroom

Versie 1.0

Coauteurs: Tom Neutens, Francis wyffels

CC BY-SA, 2023, Natacha Gesquière

Coauteurs: Tom Neutens, Francis wyffels

Corrector: Annick Dehennin

Illustrator: Margot De Saegher

AI in de Zorg maakt deel uit van AI Op School.

AI Op School werd in 2020 bekroond met een M-Award voor Mediawijs Educatief Project.

Het project KIKS van AI Op School won in 2020 de Koningin Paolaprijs voor het Onderwijs.

Deze cursus kwam tot stand met (financiële of logistieke) steun van de Universiteit Gent, Dwengo vzw, de cel iSTEM inkleuren en de Vlaamse Overheid.

D/2023/Dwengo/2
ISBN 9789464441024
NUR 950

Eerste druk, maart 2023

Versie 1.0

INHOUDSOPGAVE

1 Artificiële intelligentie **9**

2 Maatschappelijke aspecten **21**

3 Grafen **29**

4 Beslissingsbomen **47**

5 Computationeel denken **63**

6 Binaire beslissingsbomen construeren **67**

7 De notebooks **99**

8 Een beslissingsboom in Python **103**

9 Early Warning System **113**

Index **125**

Versie 1.0

Versie 1.0

INLEIDING

Artificiële intelligentie vindt steeds meer haar weg naar de zorgsector. Logistieke diensten in ziekenhuizen kunnen voordeel halen uit artificiele intelligentie, het potentieel van elektronische patiëntendossiers wordt ermee verbeterd, dokters wenden artificieel intelligente systemen aan voor intakegesprekken of ter ondersteuning bij het stellen van een diagnose.

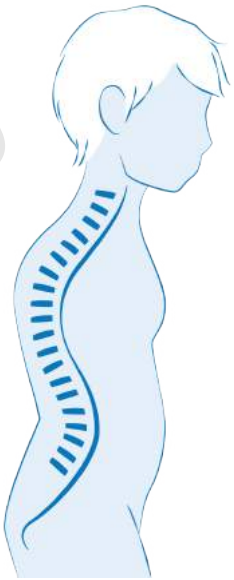
Het volgende voorbeeld illustreert hoe een artificieel intelligent systeem een arts kan helpen bij het bepalen van een therapie.

Kyfose of 'bolle rug' is een afwijking aan de ruggengraat die zich manifesteert als een abnormale ronding van de bovenrug, zoals te zien op Figuur 1.

Kyfose kan op elke leeftijd voorkomen, maar uit zich vaak bij jongvolwassenen. De patiënt kan ervoor kiezen om zich te laten opereren, maar vaak is de aandoening nog steeds aanwezig na de operatie. De arts schat op voorhand in of een operatie aangewezen is of niet, en kan daarbij geholpen worden door een AI-systeem, bijvoorbeeld een beslissingsboom.

De beslissingsboom in Figuur 2 'voorspelt' of een patiënt zal geholpen zijn door een operatie of niet. Deze beslissing gebeurt op basis van verschillende patiëntenkenmerken, namelijk de leeftijd van de patiënt in maanden, het aantal wervels betrokken bij de operatie en het nummer van de eerste wervel waarop geopereerd wordt.

Deze patiëntenkenmerken worden voor een specifieke patiënt ingegeven in het AI-systeem en het systeem geeft aan aan de arts of een operatie voor die patiënt aangewezen is.



Figuur 1: Bolle rug.

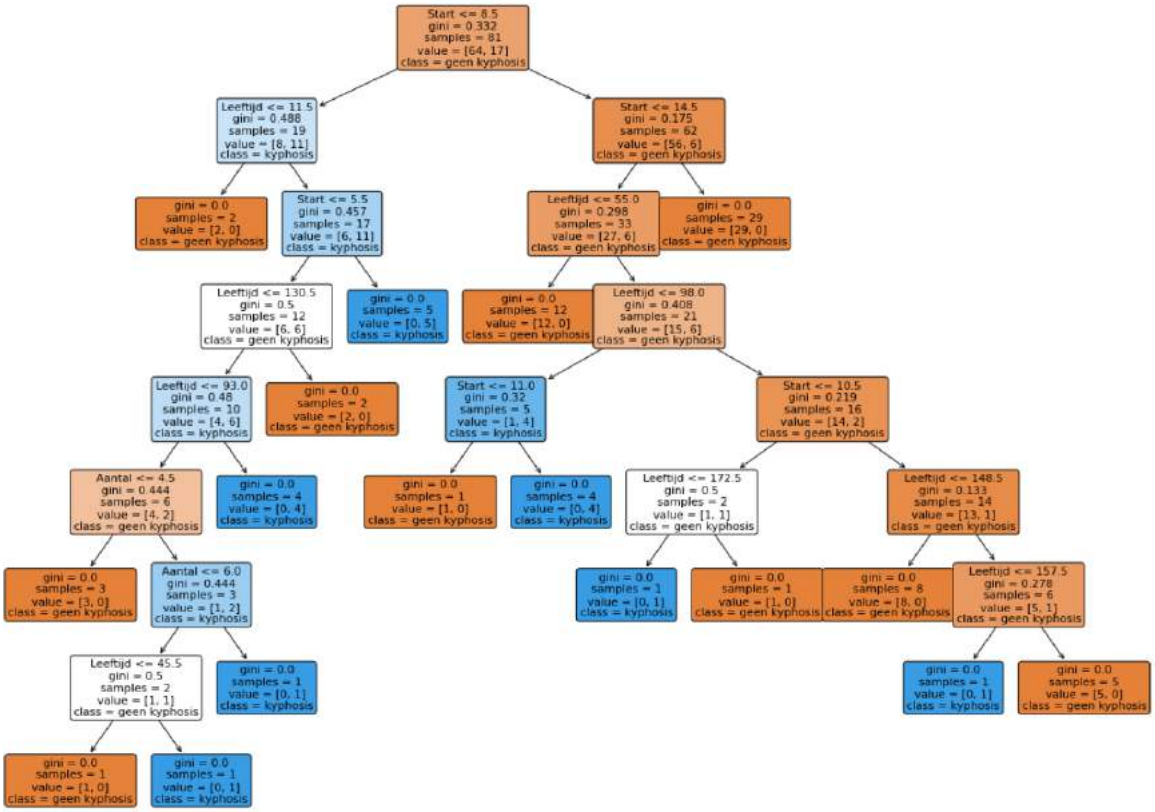
Hoe zo'n beslissingsboom tot stand komt, ontdek je zelf in hoofdstukken 6 en 8.

Zo'n beslissingsboom vormt een regelgebaseerd AI-systeem en is zeer transparant: het is duidelijk hoe het systeem tot zijn beslissing is gekomen. Om de beslissingsboom te construeren werd een lerend AI-systeem aangewend: de beslissingsboom is gebaseerd op gelabelde data, relevante patiëntenkenmerken gekoppeld aan het feit of de operatie resultaat opleverde of niet.

In hoofdstuk 1 kan je meer lezen over wat begrippen zoals 'regelgebaseerd', 'lerend AI-systeem' en 'gelabeld' inhouden.

In hoofdstuk 4 krijg je nog heel wat voorbeelden van beslissingsbomen uit de zorgsector te zien.

Versie 1.0



Figuur 2: Beslissingsboom operatie ‘bolle rug’. Gebaseerd op de dataset op `https://www.kaggle.com/code/data855/kyphosis-disease-classification`.

Versie

Uitdaging

Je beschikt over gelabelde data uit de zorgsector. Laat een computer jou helpen om deze data zo voor te stellen dat je er beslissingen mee kan nemen betreffende een diagnose of een behandeling.

Versie 1.0

Versie 1.0

HOOFDSTUK
EEN

ARTIFICIËLE INTELLIGENTIE

1.1 Toepassingen van AI

Opdracht 1.1.1: de term AI

Wat roept de term AI bij jou op?

Kunstmatige intelligentie (KI) of artificiële intelligentie (AI) komt volop aan bod in de media. Populaire thema's zijn: chatbots, zelfrijdende auto's, virtuele assistenten en het verlies aan jobs.

AI steekt her en der zijn kop op in de steeds meer gedigitaliseerde maatschappij en is al aanwezig in het dagelijks leven:

- gezichtsherkenning op Facebook;
- Google Translate;
- vragen stellen aan Siri;
- de chatbot van bol.com;
- suggesties op Netflix of TikTok ...

Opdracht 1.1.2: toepassingen van AI

- Welke andere toepassingen van AI ken je?
- Met welke toepassingen ben je zelf al in contact gekomen?
- Zoek op het world wide web, in kranten of tijdschriften een toepassing die je nog niet kende.

1.2 Wat is AI?

Kunstmatige intelligentie wordt in veel en sterk van elkaar verschillende domeinen ingezet. Denk bv. aan de robotica, de financiële wereld, kunst en talen. Elk domein kent wel toepassingen. Het is dus zeker niet eenvoudig om AI eenduidig te definiëren. Er zijn al tal van

definities voor AI geformuleerd. In de kader 'AI' worden er enkele opgesomd.

AI

- "Kunstmatige intelligentie verwijst naar systemen die intelligent gedrag vertonen door hun omgeving te analyseren en – in zekere mate zelfstandig – actie te ondernemen om specifieke doelstellingen te verwezenlijken. Op KI gebaseerde systemen kunnen louter softwarematig zijn en actief zijn in de virtuele wereld (bijvoorbeeld stemgestuurde assistenten, software voor beeldanalyse, zoekmachines en systemen voor spraak- en gezichtsherkenning), maar KI kan ook in hardwareapparaten worden geïntegreerd (bijvoorbeeld geavanceerde robots, zelfrijdende auto's, drones of toepassingen van het internet der dingen)". Dit is de definitie die de Europese Commissie hanteerde in 2018 (PwC, 2018).
- "Artificiële intelligentie is niet-biologische intelligentie" (Tegmark, 2017).
- "AI verwijst naar computers die de dingen doen die men met verstand doet. Sommige van deze zaken (bv. redeneren) worden doorgaans omschreven als 'intelligent'. Andere (bv. zicht) niet. Maar allemaal vereisen ze psychologische vaardigheden die het voor een mens en een dier mogelijk maken om hun doelen te bereiken, zoals perceptie, associatie, voorspelling, planning en het controleren van een motor" (Boden, 2016).

Opdracht 1.2.1: definitie van AI

Welke definitie in de kader 'AI' spreekt jou het meest aan? Beargumenteer.

1.3 AI in de zorgsector

AI is ook al doorgedrongen in de **gezondheidszorg**.

Er zijn bijvoorbeeld AI-systemen die specialisten evenaren in de diagnose van huidkanker, en er is een app om een migraineaanval te voorspellen.

Meer over de migraine-app in hoofdstuk 4.

Opdracht 1.3.1: AI in de gezondheidszorg

Ga op het world wide web, in kranten en tijdschriften op zoek naar een AI-toepassing in de zorgsector. Omschrijf kort wat de toepassing doet en leg uit waarom deze toepassing je aanspreekt.

1.4 Brede, enge AI en AGI

Men is optimistisch over de toekomstige ontwikkelingen binnen AI, maar **realiteitszin** is op zijn plaats. Men moet beseffen dat sommige zaken veel tijd zullen vergen, bv. vertalen tussen om het even welke talen, het begrijpen van taal, zelfrijdende auto's en robotverpleegkundigen. En hoewel er ongetwijfeld spannende ontwikkelingen zijn op het vlak van AI, staat de kunstmatige intelligentie vooralsnog **mijlenver af van menselijke intelligentie**. Een AI-systeem is gericht op het invullen van een bepaalde taak en is maar goed in datgene waarvoor het ontworpen is.

Een mens kan behoorlijk veel bijleren uit een beperkt aantal voorbeelden, omdat hij ook zijn reeds opgedane kennis en vaardigheden benut. Een mens is bovendien zeer flexibel: mensen kunnen zich snel aanpassen aan veranderende omstandigheden en kunnen adequaat reageren op onverwachte gebeurtenissen. Vandaag de dag kan een computer dat niet.

Men maakt een onderscheid tussen *brede AI*, *enge AI* en *AGI* (zie kader 'Brede, enge AI en AGI').

Brede AI, enge AI en AGI

Bij enge AI (*narrow AI*) gaat het over AI-systemen die de specifieke taken doen waarvoor ze ontworpen werden, bv. een spamfilter.
Brede AI (*general AI*) betreft systemen die al meer diverse taken aankunnen, denk bv. aan wat een robotbutler verondersteld wordt te kunnen: personen en voorwerpen herkennen, gesproken vragen begrijpen, zich voortbewegen in een ongekende ruimte ...
Bij de zogenaamde *Artificial General Intelligence* (AGI) gaat het over computersystemen met dezelfde capaciteiten als een mens. Vooralsnog is de kennis nog niet aanwezig om dergelijke systemen te ontwerpen.

Voorlopig zit AI nog maar op het niveau van de enge AI.

Opdracht 1.4.1: autonoom voertuig

Gaat het bij de zelfrijdende auto om de enge AI, de brede AI of de AGI? Leg uit.

Discussie 1.4.2: jobs

Wat is je mening over de volgende stelling?
"Er zullen ongetwijfeld jobs verdwijnen door de opkomst van AI, maar er zullen ook nieuwe jobs door ontstaan".

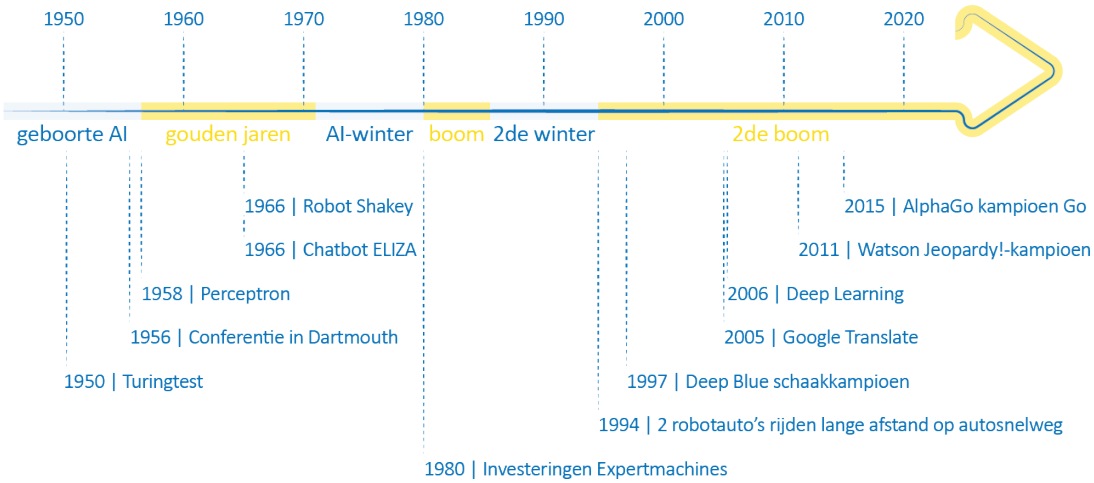
1.5 Geschiedenis van AI

In de negentiende eeuw ontwierp Charles Babbage de 'Analytical Machine', die beschouwd wordt als de eerste computer, hoewel het toestel nooit fysiek gerealiseerd werd. Ada Lovelace publiceerde in die context het eerste computerprogramma. Verkeerdelijk ziet men AI vaak als iets van de laatste jaren, maar **Ada Lovelace** dacht er in de negentiende eeuw al over na. Als onderzoeksdomein bestaat AI reeds van in de jaren 50 van de vorige eeuw. AI is dus geen nieuw fenomeen (zie Figuur 1.1).

Lovelace gebruikte bij dat eerste computerprogramma concepten uit de computerwetenschappen, zoals variabelen, lussen en geneste lussen; ze wordt dan ook aangezien als de eerste computerprogrammeur.
Lovelace vermeldde dat zo'n machine nooit origineel uit de hoek zou komen, maar enkel zou uitvoeren wat haar was opgedragen. Maar ze zei ook dat, als men erin zou slagen om verhoudingen tussen muzikale tonen op een abstracte manier weer te geven, de machine dan in staat zou zijn om muziek te maken (Wolfram, 2015).

In 1950 vroeg **Alan Turing** zich af of een machine er in een conversatie zou kunnen in slagen een mens te doen geloven dat zij een mens is. Bij de *Turingtest* stelt een mens vragen aan een ongekende partij in een andere kamer, die zowel een mens als een machine kan zijn, om dan te besluiten of die een mens is of niet. Hiermee gaf Turing een eerste aanzet tot het onderzoeksdomein dat we nu kennen als kunstmatige intelligentie (KI of AI).

De term AI werd geponeerd op een **conferentie in Dartmouth** in 1956, toen vooraanstaande onderzoekers waaronder John McCarthy,



Figuur 1.1: Geschiedenis van AI.

Marvin Minsky, Claude Shannon en Nathaniel Rochester (zeer optimistisch) stelden dat "elk aspect van leren of elk ander kenmerk van intelligentie in principe zo precies kan worden beschreven dat er een machine gemaakt kan worden om het na te bootsen".

In 1958 ontwierp Frank Rosenplatt het **Perceptron**, het eerste *neuraal netwerk*.

Zie paragraaf 1.6 voor meer uitleg over neurale netwerken.

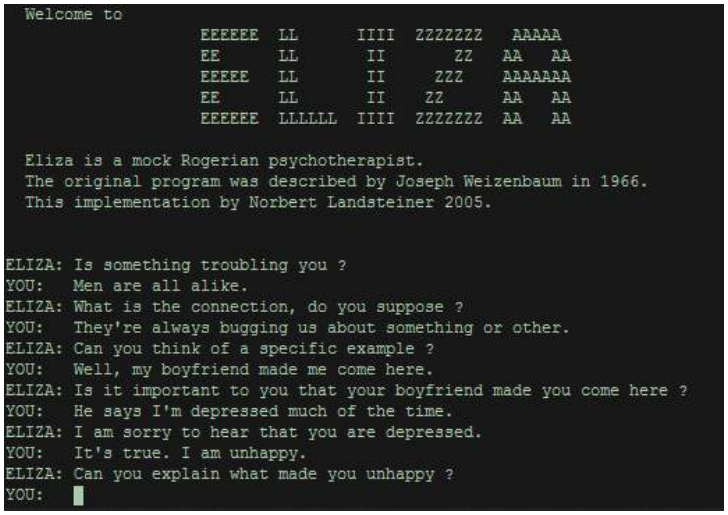
De robot **Shakey** uit 1966 werd gecreëerd aan het Artificial Intelligence Center van het Stanford Research Institute en was de eerste robot die instructies kon analyseren en opbreken in deelinstructies (zie Figuur 1.2).

Een voorbeeld van een opdracht die aan Shakey werd gegeven: "Duw de doos van het platform". Shakey kijkt rond, identificeert een platform met een doos erop en lokaliseert een ramp om het platform op te kunnen rijden. Shakey duwt dan de ramp tot aan het platform, rijdt op de ramp tot op het platform en duwt de doos eraf (Wikipedia, 2019).

Figuur 1.2: Robot Shakey (Nardone, 2007).

Op het Massachusetts Institute of Technology (MIT) ontwierp Weizenbaum software voor de gezondheidszorg, **ELIZA**, waarmee een Rogeriaanse psychotherapeut bij een intakegesprek van een nieuwe patiënt werd nagespeeld (zie Figuur 1.3). Zulke therapeuten nemen een passieve rol op en bouwen hun vragen aan de patiënt zo op dat ze de antwoorden van de patiënt erin verwerken. ELIZA kan beschouwd worden als de eerste chatbot (Güzeldere & Franchi, 1995).

Versie 1.0



Figuur 1.3: Conversatie met ELIZA (ELIZA, 2018).

Opdracht 1.5.1 Chat zelf met ELIZA

Surf naar `https://web.njit.edu/~ronkowit/eliza.html` en chat zelf eens met ELIZA (NJIT IST, 2018).
Of chat in het Nederlands via `https://www.eclecticenergies.com/nederlands/psyche/eliza` (Berkers, 2020).

Kan ELIZA jou overtuigen als psychotherapeut?

Tussen 1974 en 1980 hadden overheid en investeerders veel kritiek op het onderzoek naar AI, en investeringen bleven uit. Het onderzoeksdomein belandde in een '**AI-winter**' waarbij de ontwikkelingen in AI een **terugval** kenden (zie Figuur 1.4). In die tijd speelde het **gebrek aan rekenkracht** de onderzoekers parten, maar ondanks alles boekten de onderzoekers toch vooruitgang.

In de jaren 80 werden wel weer investeringen gedaan. **Expertsystemen** die de besluitvorming van menselijke experts om zeer specifieke problemen op te lossen simuleerden, zoals het diagnosticeren van besmettelijke ziektes of het identificeren van chemische componenten, waren toen zeer populair. Toch kwam er tussen 1987 en 1993 een **tweede AI-winter**. Omdat de desktopcomputers van Apple en IBM steeds sneller en krachtiger werden, was er geen nood meer aan gespecialiseerde AI-hardware, waardoor de markt instortte (Lim, 2018).

Merk op dat er al heel snel sinds het ontstaan van het onderzoeksdomein AI aandacht was voor mogelijke medische toepassingen. Het expertsysteem MYCIN diende om bacteriën te identificeren die bloedinfecties veroorzaken en ernstige ziektes zoals hersenvliesontsteking. Het expertsysteem PUFF interpreteerde labogegevens i.v.m. de werking van de longen.

Sinds **midden jaren 90** is er een opleving.

In 1994 reden voor het eerst twee zelfrijdende auto's op de autosnelweg in de buurt van Parijs, te midden van het drukke verkeer.

In 1997 versloeg IBM Deep Blue de heersende schaakkampioen Garry Kasparov.

In 2006 werd een online versie van Google Translate gelanceerd.

Er kon vertaald worden van het Engels naar het Arabisch en vice versa.

In 2011 werd IBM's Watson kampioen in de quiz 'Jeopardy!'. Naast kennis was hierbij ook *natural language processing* belangrijk.

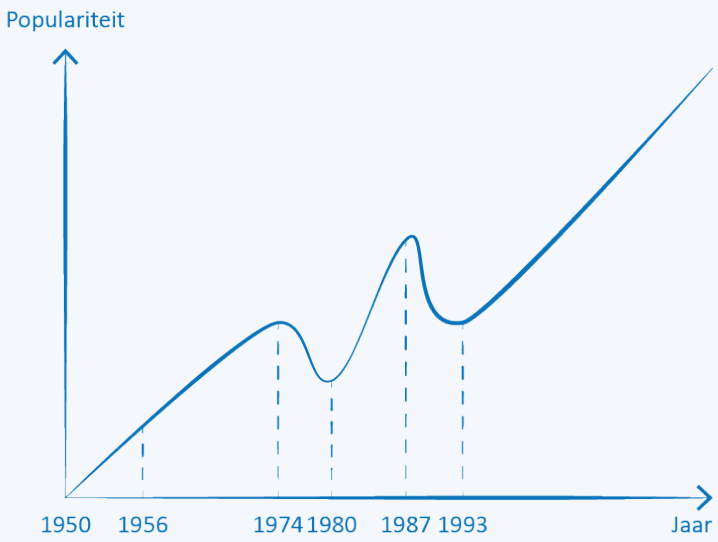
In 2016 won Google DeepMind's AlphaGo van Lee Sedol, een van de beste go-spelers ter wereld; Lee Sedol stopte in 2019 als professionele speler, omdat hij niet meer de beste kon zijn (BBC, 2019).

Bekijk een fragment uit 1990 van Waagstuk!, de Vlaamse versie van Jeopardy!, via `https://youtu.be/FBlBArYvUz0`.

Go is een Aziatisch bordspel voor twee spelers dat al meer dan 2000 jaar bestaat. Het spelbord heeft 19 op 19 velden. Het spel begint met een leeg bord en het is de bedoeling om zoveel mogelijk territorium af te bakenen door het leggen van stenen. Bekijk in een GIF-afbeelding de eerste zetten van een mogelijke go-partij op `https://nl.wikipedia.org/wiki/Go_(bordspel)`.

Opdracht 1.5.2: AI-winters

De ups en downs in de ontwikkelingen van het domein van AI worden geïllustreerd in Figuur 1.4, vanaf het ontstaan tot nu.
Duid de AI-winters aan.



Figuur 1.4: Ups en downs (Lim, 2018).

De laatste jaren boekte men dankzij **snellere hardware** en **zeer grote datasets** veel vooruitgang in het machinaal leren (*machine learning*, ML). Door de ontwikkelingen in de **game-industrie** (zie Figuur 1.5) en in de **dataopslag**, maar vooral door de mogelijkheden die het **internet** biedt om zeer grote datasets te verzamelen en te delen, is ML prominent aanwezig sedert het begin van de jaren 2010.

ML toonde bv. opmerkelijke resultaten op het vlak van spraak- en beeldherkenning, wat vaardigheden zijn waarvan de computer de menselijke prestaties tot op vandaag moeilijk kan evenaren.

Big data speelt hierin een belangrijke rol. Big data zijn data afkomstig van e-mails, video's, brieven, rapporten, blogs, socialemediaposts, sensoren, camera's ... (Berbers et al., 2017). Ze hebben dus allerlei formaten, zoals tekst, afbeeldingen en geluid, en het kunnen ook persoonsgegevens zijn. Big data gaat om grote hoeveelheden data

Figuur 1.5: Voor meer rekenkracht werd binnen de game-industrie de GPU ontwikkeld. Hierdoor kunnen games worden ontwikkeld met realistisch ogende graphics zoals in Shadow of the Tomb Raider © Square Enix (2019).

die digitaal beschikbaar zijn, zodat computersystemen ze kunnen doorzoeken, analyseren en verwerken. Big data houdt dus bijvoorbeeld in dat AI-systemen data van verschillende bronnen met elkaar in verband kunnen brengen, data van allerlei oorsprong en formaat.

Opdracht 1.5.3: wisselwerking tussen technologie en maatschappij

Illustreer de wisselwerking tussen nieuwe ontwikkelingen in wetenschappen en technologie en de maatschappij met concrete voorbeelden. Toon daarbij hoe wetenschappen en technologie de maatschappij kunnen beïnvloeden, alsook hoe de maatschappij wetenschappelijke en technologische ontwikkelingen beïnvloedt.

Sommigen veronderstellen dat de snelle ontwikkelingen in AI zullen leiden tot het ontstaan van kunstmatig intelligente systemen die zichzelf zullen verbeteren zonder menselijke tussenkomst. Systemen die zó intelligent zijn dat ze de maatschappij kunnen sturen en zullen overnemen. De creatie van zo'n artificiële superintelligentie is gekend als de '*technologische singulariteit*'. De technologische singulariteit is een hypothese. Sommigen denken dat het nooit zal gebeuren, maar anderen verwachten dat deze singulariteit al in deze eeuw zal worden bereikt.

Momenteel zijn we daar nog ver vanaf. Merk op dat een schaakcomputer, die als zeer slim wordt aangezien, alleen maar kan schaken.

Versie 1.0

1.6 Soorten AI

Om AI in te zetten voor een bepaald probleem, worden gegevens over dat probleem aan de computer gegeven (*input*). Vervolgens verwerkt het AI-systeem deze data en komt er een uitvoer (*output*). Men kan daarbij het probleem **kennisgebaseerd** of **datagebaseerd** aanpakken. In het eerste geval zal men **regels** opstellen; in het tweede geval zal men een **lerend systeem** gebruiken (zie de kader 'Kennis- en datagebaseerde AI-systemen').

Opdracht 1.6.1: soorten AI

In paragraaf 1.5 maakte je kennis met heel wat toepassingen van AI.
Herken je in deze voorbeelden de kennisgebaseerde en datagebaseerde AI-systemen? Geef van elk een voorbeeld.

Kennis- en datagebaseerde AI-systemen

Een kennisgebaseerde aanpak houdt in dat men de kennis van menselijke experts zoveel mogelijk in regels probeert te gieten om de kennis van deze experts eigen te maken aan een zogenaamd expertsysteem. Bij kennisgebaseerde systemen bestaat het algoritme uit expliciet geprogrammeerde regels. Men spreekt ook van regelgebaseerde systemen.

Een datagebaseerde aanpak houdt in dat een lerend systeem met statische methodes patronen in relevante data opspoort. Men spreekt dan van machinaal leren (*machine learning*, ML).

Er bestaan ook hybride AI-systemen. Deze zijn een combinatie van een kennis- en datagebaseerde aanpak.

Regelgebaseerde systemen kwamen veelvuldig voor in de voorbeelden van paragraaf 1.5. Robot Shakey, ELIZA en Deep Blue zijn voorbeelden van zulke systemen. DeepMinds AlphaGo en Google Translate zijn daarentegen lerende systemen.

Een voorbeeld van een regelgebaseerd systeem dat in de geneeskunde veelvuldig wordt gebruikt, is een **beslissingsboom**. Door de **kennis van artsen** te gebruiken, kan men een structuur opbouwen die toelaat om via de antwoorden op eenvoudige vragen bijvoorbeeld tot een bepaalde diagnose te komen bij nieuwe medische situaties die zich aandienen. Beslissingsbomen zijn zeer **transparante** systemen.

Beslissingsbomen komen verder in deze cursus volop aan bod.
Zo'n beslissingsboom kan ook worden opgesteld m.b.v. machinaal leren: gebaseerd op de data kan het systeem leren welke waarden van de parameters bepalend zijn.

Niet alles is echter in regels te gieten. Denk maar eens aan de talloze manieren waarop het weer beschreven kan worden. Machinaal leren, met **statistische methodes** leren uit data, kan dat deels opvangen. ML-systemen worden opgebouwd met lerende algoritmes (zie kader 'lerende algoritmes') en omdat ze gebruikmaken van statistiek, zullen ze nooit honderd procent accuraat zijn. En aangezien het lerende systemen zijn, neemt een ML-systeem zijn beslissingen dus niet op basis van vooraf in detail geprogrammeerde instructies.

Voorbeeld: het regent pijpenstelen; het giet water, het is hard aan het regenen.

Voorbeelden van datagebaseerde AI-systemen in de zorgsector vind je in de volgende paragraaf.

Lerende algoritmes

Lerende algoritmes zijn algoritmes waarbij het machinaal leren-systeem zelfstandig de parameters die aanwezig zijn in het algoritme, stap voor stap aanpast om geleidelijk aan te komen tot betere prestaties, en dat op basis van de data.

Omdat lerende algoritmes zelf de parameters aanpassen en het niet altijd duidelijk is welke stappen ze daarvoor juist hebben ondernomen, zijn ML-systemen **niet transparant**.

Na het opbouwen van een regelgebaseerd systeem of na afloop van het leerproces van een ML-systeem beschikt men over een model dat kan worden gebruikt om **voorspellingen** te doen over niet eerder geziene data.

'Voorspellen' betekent dat er uit voorbije tendensen cijfers voor

Versie 1.0

de toekomst afgeleid worden of dat een object bij een bepaalde categorie ('klasse' genoemd) wordt ingedeeld. Het voorspellen op basis van tendensen is een **regressieprobleem**. Het voorspellen van een klasse is een **classificatieprobleem**.

In plaats van categorieën spreekt men in deze context eerder van klassen.

1.7 Voorbeelden van lerende systemen binnen de zorgsector

- Een regressievoorbeeld is a.d.h.v. *wearables* bekijken of de lichaamsparameters van een patiënt die net ontslagen werd uit het ziekenhuis, niet de verkeerde kant uitgaan.

Lichaamsparameters zijn bv. lichaamstemperatuur, hartslag en bloeddruk. Ze spelen een grote rol bij de EWS, zie ook hoofdstuk 9.

- Een classificatievoorbeeld is nagaan of de evolutie in de waarden van de lichaamsparameters van een patiënt erop wijzen dat die patiënt op intensieve zorg dreigt terecht te komen.

- Omdat men er steeds beter in slaagt om met AI objecten te herkennen, kunnen radiologen een beroep doen op AI-systemen om hen te helpen om medische beelden meer in detail te bekijken en te interpreteren.

- Watson voor Oncologie van IBM is ontwikkeld om de oncoloog te ondersteunen bij het bepalen van de beste behandeling voor de patiënt. Zo is het voor specialisten een uitdaging om de nieuwste ontwikkelingen in hun domein te kunnen volgen. Watson voor Oncologie kwam tot stand in nauwe samenwerking met oncologen. Het AI-systeem leerde bv. hoe de resultaten van bloedafnames en de grootte van tumoren in rekening te brengen voor het bepalen van een behandeling (Cavallo, 2019).

- Een ingreep waarbij een hartklep vervangen moet worden, is minder ingrijpend voor de patiënt wanneer de borstkas niet opengemaakt wordt. Voor zo'n minimaal invasieve hartklepvervanging gebruikt de arts een katheter. Op voorhand bepaalt hij aan de hand van medische beelden de afmetingen van de hartklep en de positie en bereikbaarheid ervan. Dat kost hem al snel een half uur. Men onderzoekt hoe het inzetten van AI deze voorbereidende taken kan versnellen. Het AI-systeem voert de metingen uit in enkele seconden en vaak ook nauwkeuriger dan de arts. Een belangrijke voorwaarde is dat de computer over voldoende voorbeelden beschikt. Een controle door de arts is nog steeds vereist, want bij gevallen die niet of onvoldoende in de voorbeelden aanwezig zijn, kan het algoritme foute informatie geven (Astudillo & Dambre, 2020).

- De operatierobot 'Da Vinci' wordt via een joystick bediend door de chirurg en is dus geen autonoom AI-systeem, maar er is wel een systeem in verwerkt dat een ongewenste bruuske beweging van de chirurg herkent en tempert.

Versie 1.0

- AI kan in het ziekenhuis ook helpen bij logistieke taken. Bijvoorbeeld om de voorraad op peil te houden en operaties zo efficiënt mogelijk in te plannen.

- Bij ontslag uit spoedgevallendiensten krijgen patiënten soms instructies mee naar huis over welke medicatie ze moeten innemen of wanneer ze op consultatie moeten komen. Omdat niet alle patiënten het Engels voldoende machtig zijn, geven spoedgevallendiensten in de VS deze instructies graag mee in de moedertaal van de patiënt. Ze doen daarvoor vaak een beroep op Google Translate, een tool die nog niet voldoende betrouwbaar is voor dergelijke medische toepassingen: de nauwkeurigheid van Google Translate verschilt sterk van taal tot taal, en kan niet altijd goed omgaan met medische vaktaal (Taira et al., 2021).
  Google Translate wordt steeds beter, en kan dus in de toekomst misschien wel geschikt zijn voor deze taak. Zeker bij talen die minder vaak voorkomen, blijft voorzichtigheid geboden.

Zo werd de Engelse instructie “You can take over the counter ibuprofen as needed for pain” naar het Armeens vertaald en kreeg daar de betekenis: “You may take anti-tank missile as much as you need for pain”. De instructie “Do not blow your nose or put pressure on your facial fracture”. In het Farsi werd dit: “Do not explode your nose because it could put pressure on the break in your face”. Bij vertaling naar het Chinees, een taal waarin Google Translate doorgaans goede resultaten geeft, werd voor “blow” een karakter gebruikt dat eerder dient voor het blazen van de wind, waardoor de vertaling mogelijk fout begrepen wordt (Taira et al., 2021).

- De taaltechnologie kent grote vooruitgang omdat lerende AI-systemen steeds beter in staat zijn om taal juist te interpreteren, zij het nog altijd met beperkte mogelijkheden. Deze lerende systemen doen vooral aan patroonherkenning en nemen beslissingen gebaseerd op statistiek. Dit staat ver af van de inhoud van een tekst begrijpen (Dickson, 2019).

  – Sommige systemen kunnen al bruikbare informatie uit teksten halen, wat interessant is om bv. automatisch een elektronisch patiëntendossier te doorlopen (Liu, 2015).

  – Het noodnummer 112 is het enige nummer dat u met een vergrendelde smartphone kunt bellen. Dat kan dus ook per ongeluk gebeuren. Tot 29 procent van de oproepen gebeurt per ongeluk, bv. door zogenaamde broekzakbellers. De 112 werkt volgens het principe ‘*first in first out*’: een oproep die eerst is binnengekomen, wordt door een operator ook eerst behandeld; dus de oproepen komen terecht in een wachtrij. Hazira Digital is een AI-systeem, weliswaar nog in proefstadium, dat 84 % van de broekzakoproepen correct identificeert. Het is ontwikkeld op basis van 300 audiofragmenten van echte en van onvrijwillige oproepen (DataNews, 2021).

Het project ‘Chatbot’, zie `https://dwengo.org/chatbot` geeft inzicht in de werking van taaltechnologie.

- Taaltechnologie ligt ook aan de basis van de chatbots die her en der opduiken, en die ook ingezet worden in de zorgsector.

  – Ping An Good Doctor plaatste een duizendtal onbemande klinieken verspreid over China. Patiënten gaan in gesprek met een AI-dokter die hen naar hun symptomen en ziekteverloop vraagt, en vervolgens een voorlopige diagnose stelt. Een echte dokter vervoegt dan de online consultatie en controleert de diagnose. Mensen maken gretig gebruik van deze ‘one-minute-clinics’ die 24 op 24 uur open zijn. Zo vermijden ze lange wachttijden en verplaatsingen door guur weer. Elke kliniek heeft een groot

versie 1.0

aanbod aan courante medicijnen die de patiënten onmiddellijk kunnen meenemen, wat er niet is wordt binnen het uur geleverd (Koh, 2019).

  - Zo is er de virtuele therapeut Woebot (Woebot Health, 2021).
  - Om tijd te besparen wordt hier en daar het anamnesegesprek bij de dokter vervangen door een chatbotgesprek, bijvoorbeeld met Bingli; de zinvolle informatie wordt doorgespeeld aan de dokter (Bingli, 2021).

- De acteur Val Kilmer werd in 2014 door keelkanker getroffen. Als gevolg van de behandeling is zijn stem aangetast en is hij moeilijk verstaanbaar. Met die stem kan hij niet communiceren en creatief zijn, zoals hij zou willen. Het bedrijf Sonantic wendde AI aan om a.d.h.v. bestaande geluidsopnames van Kilmers stem een model te bouwen die de stem van Kilmer zoveel mogelijk benadert. Het bedrijf had daarvoor geluidsopnames nodig van hoge kwaliteit en met zo weinig mogelijk achtergrondgeluid. De geluidsopnames werden eerst opgekuist: aanwezig achtergrondgeluid werd verwijderd zonder de spraak te vernietigen, en de gesproken tekst werd ook omgezet naar getypte tekst. De audio en corresponderende geschreven tekst werden gekoppeld en deze dataset werd aan de computer gegeven om het model te trainen. Men bekwam meer dan 40 modellen, waaruit de meest expressieve stem werd gekozen. “Het is een ongelofelijk speciaal cadeau om mijn verhaal te kunnen doen met een stem die authentiek en vertrouwd aanvoelt”, zegt Kilmer (Flynn, 2021).

Beluister de AI-stem van Val Kilmer op `https://youtu.be/OSMue60Gg6s`.

Bezoek in dit kader eens de demo van het AI-taalmodel dat getypte tekst omzet naar spraak op `https://valle-demo.github.io/`.

Versie 1.0

Samengevat

De term ‘artificiële intelligentie (AI)’ bestaat reeds sinds de jaren 50 van de vorige eeuw en de ontwikkeling ervan kende ups en downs. De mindere periodes zijn gekend als AI-winters.
Van AI doen veel definities de ronde. AI behelst immers zoveel verschillende aspecten en invalshoeken dat een algemene definitie moeilijk is.

Binnen de AI-systemen onderscheidt men kennis- en datagebaseerde systemen (ook omschreven als respectievelijk regelgebaseerde en lerende systemen).
Men maakt ook een onderscheid tussen enge AI, brede AI en AGI. Momenteel zit AI op het niveau van de enge AI.

AI is al aanwezig in het dagelijks leven en is ook al doorgedrongen in domeinen zoals de juridische wereld, de gezondheidszorg en de kunstwereld.
Hoewel men optimistisch is over de toekomstige ontwikkelingen binnen AI, is enige realiteitszin op zijn plaats.

Na dit hoofdstuk ...

- weet je dat er verschillende soorten AI bestaan: kennisgebaseerde en datagebaseerde systemen, ontwikkeld met regels of lerende algoritmes;
- kan je voorbeelden geven van kennisgebaseerde en datagebaseerde AI-systemen;
- ken je het verschil tussen enge AI, brede AI en AGI;
- weet je dat een lerend systeem zijn beslissingen niet neemt op basis van vooraf in detail geprogrammeerde instructies;
- weet je dat de voorspellingen van een lerend systeem niet met 100 % zekerheid gedaan zijn;
- weet je wat een AI-winter is;
- kan je voorbeelden geven van de wisselwerking tussen nieuwe ontwikkelingen in wetenschappen en technologie, en de maatschappij;
- kan je voorbeelden geven van toepassingen van AI in de zorgsector.

Versie 1.0

HOOFDSTUK
TWEE

MAATSCHAPPELIJKE ASPECTEN

2.1 Nieuwe technologieën

Nieuwe technologieën veranderen de maatschappij: sommige doen dat op een zeer ingrijpende manier.

Volgens Gabriels (2019) "geven technologische ontwikkelingen mee vorm aan hoe wij de wereld zien en beïnvloedt technologie ons denken, ons gedrag en onze waarden en normen". Bijvoorbeeld door echografie te gebruiken, gaan we anders om met een zwangerschap en eventuele complicaties die zich (kunnen) voordoen.

Hoe ingrijpend een nieuwe technologie de maatschappij zal veranderen, is moeilijk in te schatten.

Zie de kader 'Video' over de mobiele telefoon.

- De uitvinding van de telefonie en van de fotografie in de negentiende eeuw hebben bepaald hoe de hedendaagse communicatie eruitziet.

- Toen de eerste auto's begonnen te rijden, had niemand verwacht dat men ooit met zijn allen uren in de file zou staan.

- Wie kon in 1969 vermoeden dat we nu met een smartphone op zak zouden lopen met meer rekenkracht dan de computers die de astronauten naar de maan brachten?

- In 1995 was het moeilijk in te schatten dat het internet vandaag zo'n grote impact zou hebben op het dagelijkse leven.

Video 2.1.1: mobiele telefoon

Mobiel bellen in 1998.
`https://youtu.be/TNwhIHqM60g` (Bromet, 2016).
Mobiel bellen: 1998 vs NU.
`https://youtu.be/mQVkpdzPGtQ` (DWDD, 2019).

Versie 1.0

Video 2.1.2: David Bowie over de impact van het internet.

David Bowie predicted in 1999 the impact of the Internet in BBC.
`https://youtu.be/8tCC9yxUIdw` (Ramos, 2019).

Opdracht 2.1.3: impact van AI

Kun je je vinden in de volgende uitspraak: “De impact van AI zal mogelijk nog groter zijn dan de komst van het internet“?
Beargumenteer.

2.2 *Transparantie*

Het nemen van beslissingen wordt steeds meer geautomatiseerd. Zo wordt er met kunstmatig intelligente systemen bepaald welk nieuws je te zien krijgt op Facebook, welke series Netflix je aanraadt en welke filmpjes op YouTube worden geweigerd, maar ook de banken, de juridische wereld en de overheid maken er gebruik van. Nochtans heeft men niet altijd goed zicht op wat onder de motorkap van een AI-systeem gebeurt, bv. welke algoritmes het systeem aansturen. Daarom klinkt de vraag naar **transparant** steeds luider. Men moet er zich echter wel van bewust zijn dat een transparant algoritme daarom nog geen betrouwbaar algoritme is.

Banken gebruiken algoritmes om te beslissen wie een lening krijgt. Rechters gebruiken algoritmes om de strafmaat te bepalen. De overheid gebruikt algoritmes om sociale media te screenen.

2.3 *Vooringenomenheid*

In tegenstelling tot wat men doorgaans denkt, neemt een AI-systeem geen 100 % ‘objectieve’ beslissingen, maar is het ‘**vooringenomen**’. Vooringenomenheid houdt in dat een bepaalde groep gediscrimineerd wordt. Dat is meestal te wijten aan de data waarmee het systeem werd ontwikkeld. Het is belangrijk dat men deze vooringenomenheid (*bias*) zoveel mogelijk opspoort en tot het minimum herleidt. In de zorgsector kan die vooringenomenheid vergaande gevolgen hebben (zie kader ‘Vooringenomenheid bij toepassingen in de zorgsector’ op p. 24).

Als de gebruikte data gekleurd zijn door aanwezige vooroordelen in de maatschappij, zoals stereotypen, dan zal dit ook doorgegeven worden aan het lerende systeem. Men moet er dus over waken dat een ontwikkeld model bepaalde bevolkingsgroepen niet discrimineert.

Hoewel men een model test alvorens het in gebruik genomen wordt, wordt aanwezige vooringenomenheid niet altijd opgemerkt. De testdata kunnen immers dezelfde bias bevatten als de data voor de ontwikkeling.

Vooringenomenheid komt voor als de data waarmee een lerend systeem leerde, niet representatief zijn. Als men bv. enkel groene appels als voorbeelden geeft aan een lerend systeem, dan zal het ontwikkelde model geen rode appels herkennen. Of als men enkel foto’s van honden onder een stralend blauwe hemel aanbiedt, dan zal het model een hond in de regen niet bij de klasse ‘hond’ indelen. Als men enkel vrouwelijke verplegers in de dataset stopt, dan zullen mannen niet als verpleger worden geclassificeerd.

Versie 1.0

Voorbeelden:

- Tot 2018 gebruikte Amazon een sterk bejubeld maar ondertussen afgevoerd AI-systeem om sollicitanten te beoordelen. Het systeem selecteerde geen vrouwen voor technologische posities. Het was immers getraind met **historische data**: werknemers van de voorbije 10 jaar, voornamelijk mannen, aangezien zij nog steeds de technologiewereld domineren.
- De *Allegheny Family Screening Tool* is een AI-model om de medewerkers van de hotline van de kinderbescherming te helpen bij de beslissing of een melding van kindermishandeling of -verwaarlozing wordt opgevolgd of niet (zie kader 'Allegheny Family Screening Tool').

Allegheny County is een provincie in Pennsylvania gesitueerd rond Pittsburgh (VS).

Allegheny Family Screening Tool

De hotline van de kinderbescherming van *Allegheny County* wordt overspoeld met meldingen van mogelijke kinderverwaarlozing of -mishandeling. De medewerkers die de oproepen beantwoorden, moeten beslissen of een melding verder wordt onderzocht of niet.

Men merkte op dat te veel gevallen die door de medewerkers van de hotline als 'laag risico' aangezien werden, toch in een drama eindigden. Bovendien beschikken de medewerkers, behalve over de oproep zelf, ook over veel extra gegevens over de betreffende gezinnen, maar zijn ze wegens tijdsgebrek niet in staat die grote hoeveelheid aan informatie mee te nemen in hun beslissing. Daarom ontwikkelde men de *Allegheny Family Screening Tool*, een AI-model om de medewerkers van de hotline te helpen bij de beslissing of een melding van kindermishandeling of -verwaarlozing wordt opgevolgd of niet. De tool werd ontwikkeld ter ondersteuning van de te nemen beslissing, maar het advies van de tool naast zich neerleggen kan niet zomaar; de medewerkers moeten zich daarover schriftelijk verantwoorden.

De tool die op een heel open manier werd ontwikkeld, o.a. met inspraak van het publiek en transparantie over de gebruikte data, wordt door sommigen bejubeld, maar krijgt ook kritiek: de tool zou te kampen hebben met vooringenomenheid.
Voorstanders zeggen dat door de tool meer 'hoog risico'-gevallen opgevolgd worden en dat de tool eerlijker is naar raciale aspecten toe; ook de medewerkers van de hotline zijn, zoals alle mensen, vooringenomen, en mogelijk beïnvloed door persoonlijke ervaringen en recente gevallen. Critici menen dat de tool vooringenomen is omdat men een publieke dataset heeft gebruikt die maatschappelijke vooroordelen weerspiegelt. Tegenstanders van de tool zeggen dat door het algoritme kinderen thuis worden weggehaald omdat ze arm zijn en niet omdat ze in gevaar zijn.
De gebruikte dataset betreft historische data die de overheid er sinds 1998 heeft verzameld. Meer gegoede families kunnen het echter beter voor de overheid verbergen als ze hun kinderen verwaarlozen of mishandelen, omdat ze zich bv. voor psychologische hulp, bij financiële problemen of voor een behandeling van een drugsverslaving, wenden tot private instanties, en daarom minder aanwezig zijn in de dataset. Arme families doen een beroep op publieke voorzieningen, bv. voor voeding, huisvesting en zorgverlening, waardoor ze net meer aanwezig zijn in publieke datasets.
Men heeft de tool sinds het eerste gebruik wel al aangepast. Zo neemt het algoritme parameters als 'een beroep doen op voedselhulp' niet meer mee in zijn beslissing.

(Chouldechova et al., 2018; McKenna, 2019; Strong, 2021; Vaithianathan et al., 2019)

Vooringenomenheid bij toepassingen in de zorgsector

Vrouwen en kansengroepen zijn vaak ondervertegenwoordigd bij klinische testen. De deelnemers zijn meestal mannen van een bepaalde leeftijdsgroep. Maar vrouwen reageren soms anders op geneesmiddelen dan mannen. Daarom zijn bijwerkingen op nieuwe medicatie soms onvoldoende gekend voor de vrouwelijke populatie of blijkt achteraf dat de medicatie onvoldoende werkzaam is bij vrouwen (Norori et al., 2021).

Voorbeelden:

- Algoritmes om radiologen te helpen met röntgenfoto's van de borstkas, maar die ontwikkeld werden met data waarin vrouwen ondervertegenwoordigd zijn, presteren slechter op de röntgenfoto's van vrouwen.
- Veel AI-systemen om huidkanker te detecteren zijn ontwikkeld met data vooral afkomstig van blanke huidtypes, en merken huidkanker bij mensen met een donkere huidskleur minder goed op (Kaushal et al., 2020).
- Er zijn algoritmes om slaapstoornissen op te sporen ontwikkeld, met data van jonge, gezonde personen. Deze zijn vaak ontoereikend om een goed beeld te geven van slaapstoornissen bij oudere patiënten (Norori et al., 2021).
- AI wordt ook aangewend om de hoog oplopende kosten in de zorg te verminderen. Er zijn bv. algoritmes ontwikkeld die de zorgnoden van patiënten inschatten gebaseerd op reeds door de patiënt gemaakte kosten voor gezondheid. Deze algoritmes concludeerden foutief dat patiënten met een zwarte huidskleur gezonder waren dan even zieke, blanke patiënten. Het resultaat was dat de algoritmes de blanke patiënten voorrang gaven voor de behandeling van levensbedreigende aandoeningen, zoals diabetes en nierziekte, ook als de zwarte patiënten er duidelijk ernstiger aan toe waren. Toen dit probleem tot uiting kwam, zijn de algoritmes aangepast (Norori et al., 2021; Shin, 2020).

De data waarmee AI-systemen worden ontwikkeld zouden voldoende divers moeten zijn, maar het is voor onderzoekers al heel moeilijk om aan medische datasets te geraken, o.a. door de regels rond privacy en de gevoeligheid van de data (mensen zijn er zelf sterk toe geneigd om hun medische gegevens af te schermen) (Kaushal et al., 2020; Norori et al., 2021).
Bovendien leiden niet enkel de data tot de vooringenomenheid van de algoritmes. Ook de onderzoekers en het medisch personeel die betrokken zijn bij de ontwikkeling van de algoritmes, kunnen onbewust opvattingen en vooroordelen binnenbrengen in de algoritmes. Diversiteit binnen de ontwikkelteams en engagement van patiëntengroepen kunnen helpen om de vooringenomenheid in algoritmes te verminderen (Norori et al., 2021).

Opdracht 2.3.1: vooroordelen in de maatschappij

Welke stereotypen die aanwezig zijn in de maatschappij zouden volgens jou versterkt kunnen worden in een AI-systeem?

Opdracht 2.3.2: vooringenomen AI-systemen

Zoek op het internet, in kranten of tijdschriften een geval van bias in een AI-systeem dat reeds gebruikt wordt (werd).

Video 2.3.3: bias in een onvoldoende getest systeem

Soms worden nieuwe producten al op de markt gebracht ondanks het feit dat ze onvoldoende getest zijn. In de zeepdispenser in het volgende filmpje zit weliswaar geen AI-systeem, maar het filmpje illustreert treffend hoe onvoldoende testen kan leiden tot pijnlijke situaties.

This 'Racist soap dispenser' at Facebook office does not work for black people.
`https://youtu.be/YJjv_OeiHmo` (Futureism, 2017).

2.4 Ethiek

Naast de grote impact die nieuwe technologische ontwikkelingen kunnen hebben, moet men ook aandacht besteden aan de ethische gevolgen (zie kader 'NIPT').

NIPT

Als een erfelijke aandoening voorkomt in een familie, doet men soms genetische tests bij de familieleden. De mensen bij wie de test wordt afgenomen, worden op voorhand goed geïnformeerd, zodat ze kunnen beslissen of ze het genetische onderzoek wel willen ondergaan. Volgende zaken worden uitgebreid besproken:

- Wat kan er aan het licht komen?
- Hoe kunnen eventuele problemen dan worden aangepakt?

De NIPT-test staat hiermee in schril contrast. NIPT (niet-invasieve prenatale test) is een test om te zien of een ongeboren kind aan het syndroom van Down of een andere chromosomale afwijking lijdt. De informatie die ouders voorafgaand aan de NIPT krijgen, is vaak ontoereikend, waardoor de ouders niet altijd beseffen waarvoor ze kiezen. Sommigen staan er bv. niet bij stil dat er na de test ook slecht nieuws kan volgen. En niet alle ouders weten wat NIPT allemaal kan onthullen:

- NIPT kan nog andere afwijkingen opsporen dan chromosomale afwijkingen;
- de test wordt gedaan via het bloed van de moeder, waardoor bv. ook de aanwezigheid van een bepaald type kanker bij de moeder opgespoord kan worden.

De ouders zouden op voorhand moeten kunnen aangeven wat ze wel en niet willen weten.
In België wordt de test ook terugbetaald. Sommigen denken dat hierdoor de druk groter wordt om enkel nog gezonde kinderen op de wereld te zetten (Eckert, 2019; Garderen, 2019).

Net zoals bij andere technologische ontwikkelingen steken ook bij AI ethische dilemma's de kop op. Bovendien kunnen er ook moreel ongewenste effecten optreden waar men bij de ontwikkeling van het AI-systeem niet op bedacht was.

Zo is het zeker nodig om aan de ethische kant van AI-toepassingen in de zorg, zoals de gezelschapsrobot Paro en AI bij het elektronisch patiëntendossier, voldoende aandacht te schenken.

Als men de AI-technologie goed inzet, kan die bijdragen tot een betere wereld. Denk maar aan exoskeletten die mensen met een beperking meer mobiel maken en ouderen die langer thuis kunnen blijven wonen.

Maar zal AI te allen tijde worden ingezet voor een mooiere maatschappij? Bij de ontwikkeling van AI-systemen kan men zich door verschillende motieven laten leiden. Zal men bij de ontwikkeling van een nieuw product steeds de mens centraal zetten, of soms ook het AI-systeem zelf of geldbejag?

Door robots in te zetten in de zorgsector krijgen verplegers meer tijd voor de patiënten. Als men er echter voor kiest om minder personeel aan te werven, dan verliest men de mens uit het oog.

Om de COVID-19-crisis te beteugelen moet de overheid voortdurend een afweging maken tussen o.a. economische belangen, de mentale gezondheid van de bevolking en de verzuchtingen van de zorgverleners in de ziekenhuizen (Knack, 2022; Zorgnet-Icuro, 2021a,b).

Het inzetten van technologie in de geneeskunde kan een positief iets zijn, maar hoe ver wil men daarin gaan? Welk soort cyborg is toelaatbaar? Is een virtuele therapeut zoals Woebot wenselijk? Welke taken krijgen zorgrobots? Hoe bekomt men dat exoskeletten voor iedereen toegankelijk zijn en niet enkel voor mensen met voldoende geld (Gabriels, 2019)?

1.0

Discussie 2.4.1

Welk soort cyborg is toelaatbaar?
Is een virtuele therapeut zoals Woebot wenselijk?
Welke taken krijgen zorgrobots?
Hoe bekomt men dat robotchirurgie en exoskeletten voor iedereen toegankelijk zijn en niet enkel voor mensen met voldoende geld?

Video 2.4.2: robotchirurgie

Duizendste operatie met knierobot.
`https://youtu.be/83CuN_W7z4s` (AZ Delta, 2021).

Getuigenis na twee robotgeassisteerde knieprotheses via het Joint Care Zorgpad in AZ Delta Roeselare.
`https://youtu.be/eQWVaduhAJo` (Orthopedie Roeselare, 2021).

Video 2.4.3: exoskelet

Lopen met dwarslaesie dankzij exoskelet van TU Delft.
`https://youtu.be/2IK7CicZ4EA` (AZ Delta, 2021).

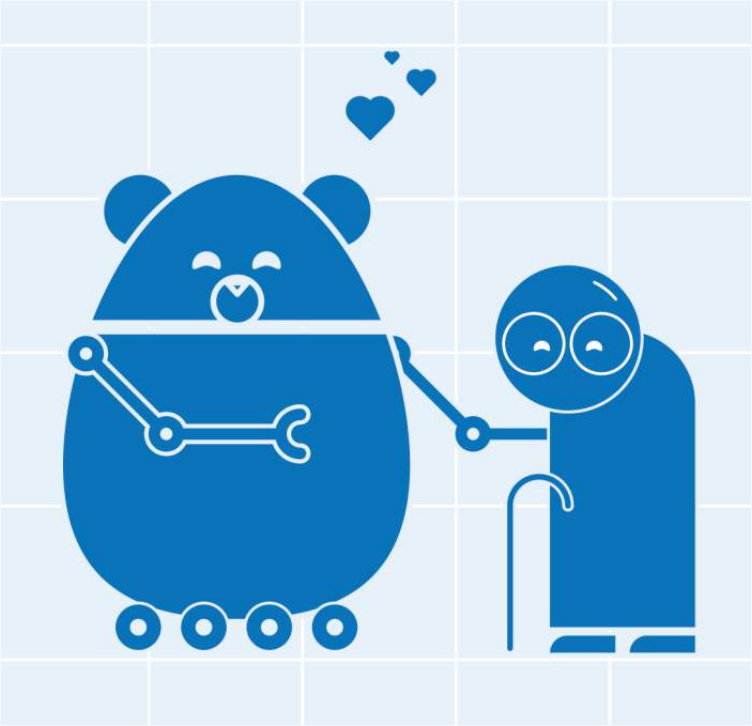
Ontwikkelaars van AI-systemen moeten reeds bij de ontwikkeling bedacht zijn op mogelijke ongewenste effecten. De **morele verantwoordelijkheid** van een AI-systeem dat in de maatschappij wordt gebracht, ligt niet bij het AI-systeem, maar bij de mens.

Ethische kwesties vergen veel tijd. Ze zijn ook vaak complex. Men mag daarom niet wachten tot de technologie klaar is om de nodige ethische discussies te voeren. Er is in elk geval al aandacht voor: *ethical AI*, *responsible AI*, *AI4good*, *trustworthy AI*, *explainable AI* zijn termen die her en der opduiken, met de bijbehorende richtlijnen.

Klasdiscussie 2.4.4

Op `dwengo.org/zorg` vind je een kaartenset met stellingen over huidige en mogelijk toekomstige toepassingen van AI in de zorgsector. De stellingen kunnen bekeken worden vanuit drie perspectieven: zorgverlener, patiënt en kritische burger, en behandelen drie categorieën: technologische vernieuwing, sociale interactie, en data en privacy.
Argumenteer waarom je met een bepaalde stelling al dan niet akkoord gaat.

De figuur toont een van de kaarten. De bijbehorende stelling is: “Gelukkig zijn er knuffelrobots om mensen met dementie en bejaarden gezelschap te houden.”



Voorbeeld van de discussiekaarten AI In de Zorg.

Samengevat

Nieuwe technologieën veranderen de maatschappij en sommige doen dat op een zeer ingrijpende manier. De impact van AI zal mogelijk nog groter zijn dan de komst van het internet. Het nemen van beslissingen wordt immers steeds meer geautomatiseerd. Zo wordt er met kunstmatig intelligente systemen bv. bepaald welke series je worden aangeraden op Netflix en beslissen de banken wie een lening krijgt. Daarom klinkt de vraag naar transparantie steeds luider.

Net zoals bij andere technologische ontwikkelingen steken ook hier ethische dilemma's de kop op. Men is van bij het begin van het ontwikkelproces best op zijn hoede voor potentieel ongewenste effecten; deze komen mogelijk voort uit de 'vooringenomenheid' van het AI-systeem. Deze *bias* moet men zoveel mogelijk controleren en tot het minimum herleiden.
Omdat ethische discussie veel tijd vergen, stelt men die best niet uit tot het AI-systeem af is.

Na dit hoofdstuk ...

- weet je wat bias in een AI-systeem is;
- zie je in dat met de ontwikkeling van AI ook ethische aspecten gemoeid zijn;
- kan je voorbeelden geven van de wisselwerking tussen nieuwe ontwikkelingen in wetenschappen en technologie, en de maatschappij.

Versie

HOOFDSTUK
DRIE

GRAFEN

3.1 Graaf

Spel 3.1.1: Sprouts

Sprouts is een spel met pen en papier, bedacht door de wiskundigen John Conway en Michael Patterson.

Start met enkele punten op een blad papier.

Twee spelers spelen om de beurt en verbinden telkens twee punten met elkaar. Een punt mag ook met zichzelf worden verbonden. Op die verbindingslijn teken je een punt. De lijnen mogen recht of krom zijn, maar ze mogen elkaar niet snijden. Elk punt mag hoogstens drie van die verbindingslijnen hebben.

De eerste speler die geen lijn meer kan tekenen, verliest.

De figuur die verschijnt tijdens het spelen van Sprouts, is een graaf. Het is weliswaar vaak een meervoudige graaf.

Dit hoofdstuk behandelt voornamelijk **simpele** grafen.

Voor meer uitleg over Sprouts zie ook het leerpad 'Grafen' op `dwengo.be/zorg`.

Graaf

Een graaf bestaat uit **knopen** en eventuele verbindingen tussen die knopen, de **bogen**.

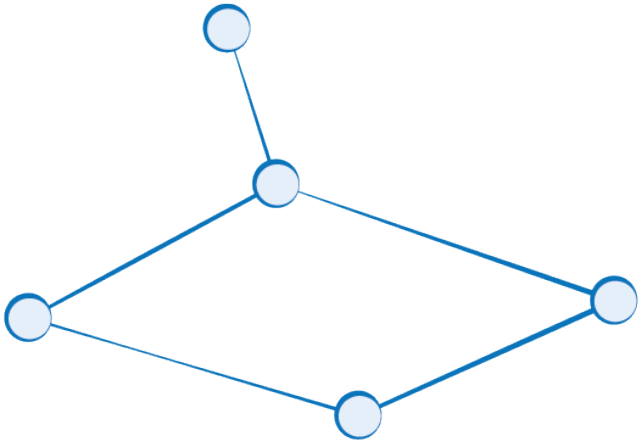
Het aantal bogen in een knoop is de **graad** van die knoop.
Knopen die met elkaar verbonden zijn, zijn **buren**.

Het boek van Veerle Fack (2007) bood inspiratie voor de definities in dit hoofdstuk.

In principe kan je een graaf omschrijven a.d.h.v. zijn knopen en bogen, maar de knopen en bogen kunnen ook voorgesteld worden in een figuur, zoals Figuur 3.1. Die voorstelling is echter niet uniek: eenzelfde graaf kan verschillende voorstellingen hebben.

In Figuur 3.1 zie je een graaf met vijf knopen. Sommige knopen zijn met elkaar verbonden door de bogen. Er is een knoop met graad 3,

er is een knoop met graad 1 en er zijn drie knopen met graad 2.

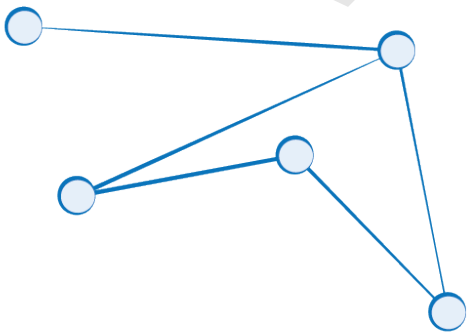


Figuur 3.1: Graaf met 5 knopen en 5 bogen.
Er is 1 knoop met 3 bogen, 1 knoop met 1 boog, en 3 knopen met elk 2 bogen.

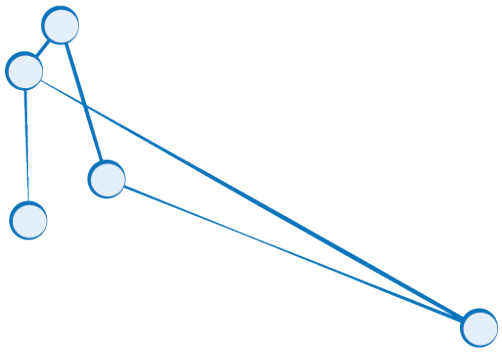
De graaf G uit Figuur 3.1 zou je kunnen omschrijven als de graaf $G = (V, E)$ met als verzameling knopen $V = \{1, 2, 3, 4, 5\}$ en als verzameling bogen $E = \{\{1,2\}, \{1,4\}, \{1,5\}, \{2,3\}, \{3,4\}\}$ (knoop en boog zijn in het Engels resp. *vertex* en *edge*). In Python geef je een graaf op deze manier in: a.d.h.v. zijn verzamelingen knopen en bogen.

Versie 1.0

Diezelfde graaf G vind je ook terug in Figuren 3.2 en 3.3; de knopen en de bogen zijn immers dezelfde als in Figuur 3.1.



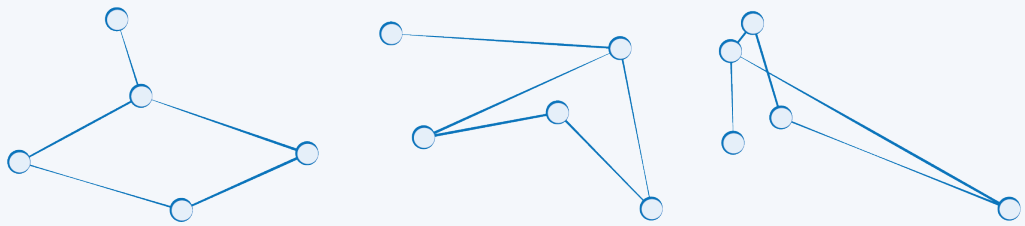
Figuur 3.2: Dezelfde graaf als in Figuur 3.1.



Figuur 3.3: Dezelfde graaf als in Figuur 3.1.

Voorbeeld 3.1.2: dezelfde graaf anders voorgesteld

Ga na dat de grafen in Figuren 3.1, 3.2 en 3.3 dezelfde zijn.
Deze grafen staan hier nog eens getekend:



Je kan de voorstellingen vergelijken door de knopen te kleuren of door ze te nummeren (voor meer uitleg zie ook `dwengo.org/zorg`). Je zal in ieder geval moeten letten op het aantal knopen en de graad van de knopen.

- Je kan met **kleuren** werken. Je geeft elke knoop een andere kleur. Je doet dat op dezelfde manier voor elk van de drie voorstellingen.

  Voorbeeld: Er is één knoop met drie bogen; je kleurt op de drie grafen die unieke knoop met drie bogen groen. Merk op dat van de knopen die verbonden zijn met deze knoop, er slechts één knoop is met slechts één boog; die knoop kleur je geel op de drie voorstellingen. Er is één knoop die niet verbonden is met de groene knoop; die knoop kleur je rood op de drie voorstellingen. De groene knoop is nu nog verbonden met twee knopen met elk twee bogen; de ene kleur je paars en de andere oranje, telkens op de drie voorstellingen.

- Je kan op elk van de drie voorstellingen de knopen op dezelfde manier **nummeren**, zoals vermeld in de opmerking in de kantlijn: graaf met een verzameling knopen $V = \{1, 2, 3, 4, 5\}$ en een verzameling bogen $E = \{\{1,2\}, \{1,4\}, \{1,5\}, \{2,3\}, \{3,4\}\}$.

  Ga bij deze nummering goed na wat bv. het nummer van de enige knoop met drie bogen is, of er knopen zijn met twee bogen, en of er een knoop is met één of geen boog.

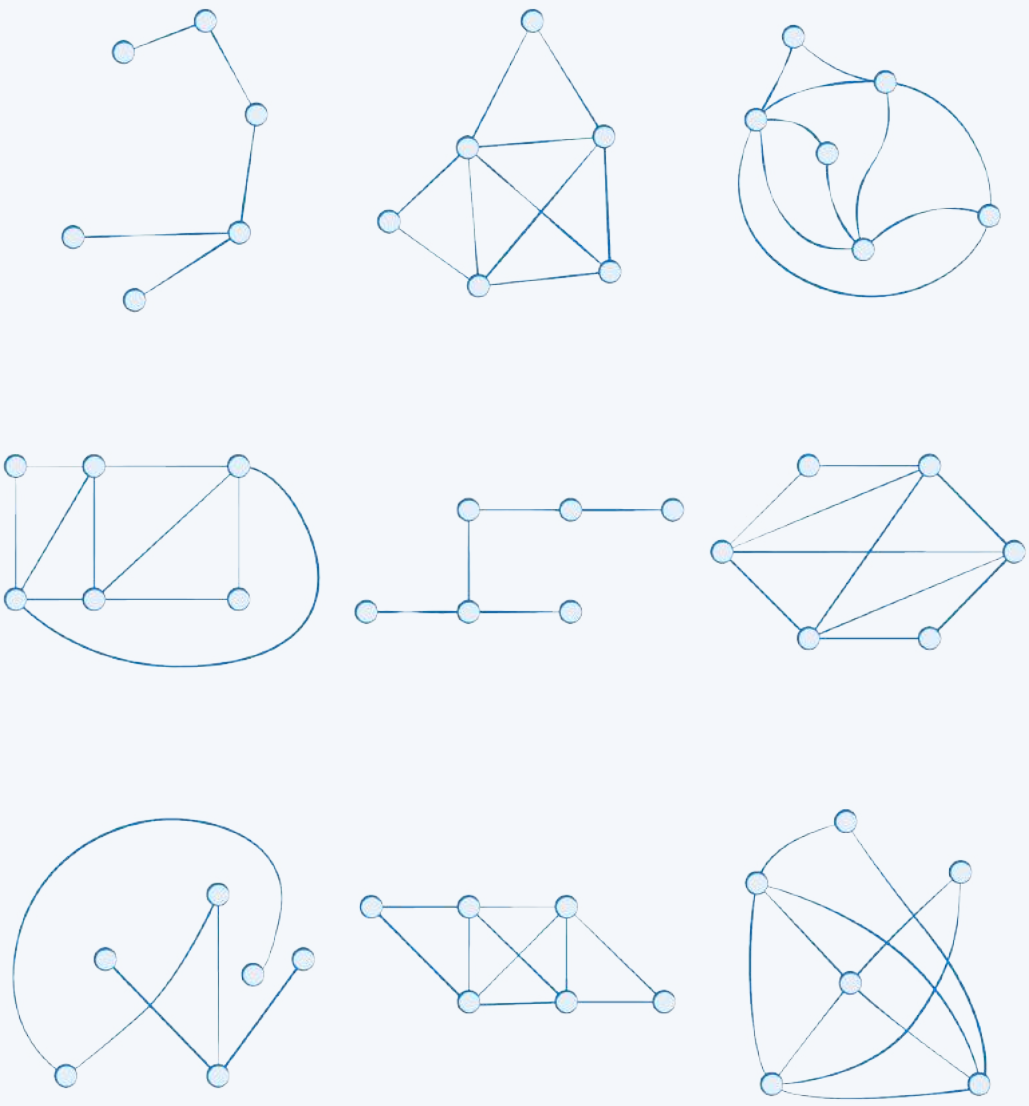
  Voorbeeld: De enige knoop met drie bogen is knoop 1. De enige knoop die niet verbonden is met knoop 1, is knoop 3. Van de knopen die verbonden zijn met knoop 1, is er slechts één die maar één boog heeft, dat is knoop 5. Dan resten er nog twee knopen, beide verbonden met knoop 1 en knoop 3, dat zijn knopen 2 en 4. Je kan dit stramien op elk van de drie grafieken van een graaf toepassen.

Opdracht 3.1.3: algoritme

Formuleer een algoritme in woorden dat een werkwijze weergeeft om na te gaan of twee voorstellingen van een graaf dezelfde graaf betreffen.

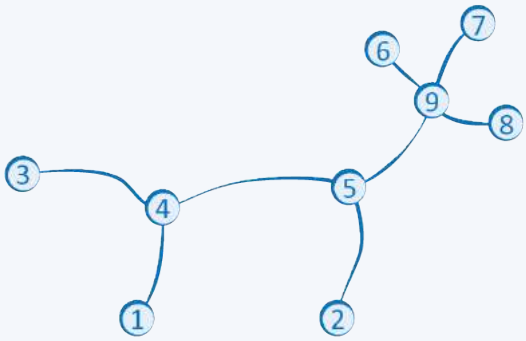
Oefening 3.1.4: wel of niet dezelfde grafen

Hieronder zijn verschillende grafen te zien. Van elke graaf die voorkomt, zijn er meerdere voorstellingen. Welke voorstellingen komen overeen met eenzelfde graaf?



Oefening 3.1.5: verzameling knopen en bogen

Omschrijf de graaf in de figuur door de verzamelingen van de knopen (V) en de bogen (E) te geven.



Oefening 3.1.6: graaf met gegeven knopen en bogen

Teken de graaf (V, E) met:
$V = \{1, 2, 3, 4, 5, 6\}$ en $E = \{\{1,2\}, \{1,4\}, \{1,6\}, \{2,3\}, \{3,4\}, \{3,6\}, \{4,5\}, \{4,6\}, \{5,6\}\}$.

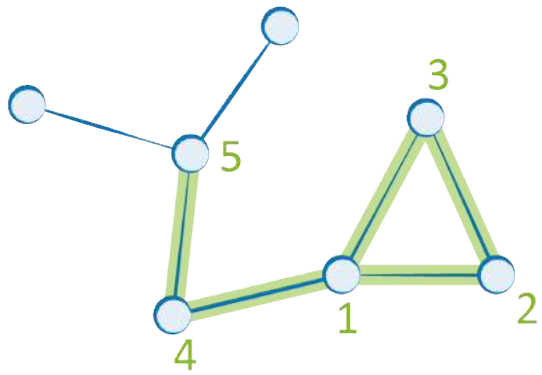
Versie

Voorbeeld: Vriendschappen op Facebook kunnen worden voorgesteld met een graaf (zie Figuur 3.4). De knopen stellen de Facebook-accounts voor. Een boog kan je lezen als 'is vriend van', immers als Margo bevriend is met Jef op Facebook, dan is Jef ook bevriend met Margo.

Figuur 3.4: De mensen met een Facebookaccount en met wie ze vriend zijn (Haydon, 2010).

3.2 Wandeling, pad en cykel

In een graaf kan je een **wandeling** beschouwen, zoals bv. de groene wandeling in Figuur 3.5 met beginknoop 1 en eindknoop 5. Een wandeling heeft steeds een beginknoop en een eindknoop.



Figuur 3.5: Wandeling met beginknoop 1 en eindknoop 5: $(1, \{1,2\}, 2, \{2,3\}, 3, \{3,1\}, 1, \{1,4\}, 4, \{4,5\}, 5)$.

Wandeling

Een wandeling begint en eindigt in een knoop en bestaat uit een lijst van bepaalde bogen die je **van de ene knoop naar de andere** voeren.

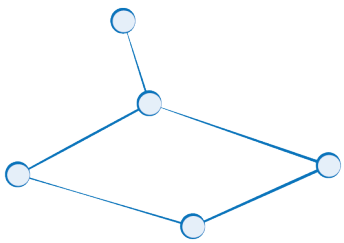
Pad

Een pad is een wandeling waarin alle **bogen verschillend** zijn en een **knoop nooit meer dan één keer** voorkomt. Hierop is één uitzondering toegestaan: de begin- en eindknoop mogen eventueel dezelfde knoop zijn.

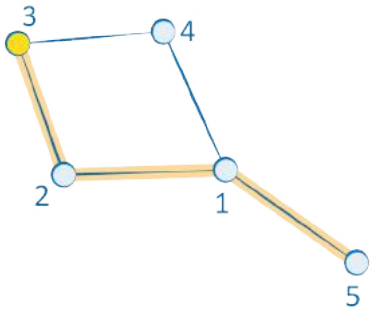
Cykel

Een pad waarvan de **begin- en eindknoop dezelfde** zijn, is een cykel.

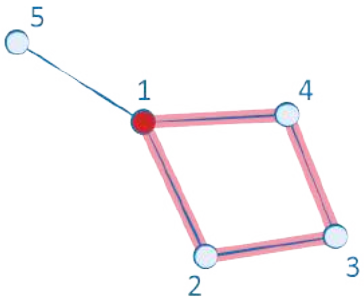
Figuren 3.7 en 3.8 tonen een pad in het geel en een cykel in het bordeauxrood in eenzelfde graaf.



Figuur 3.6: Graaf.



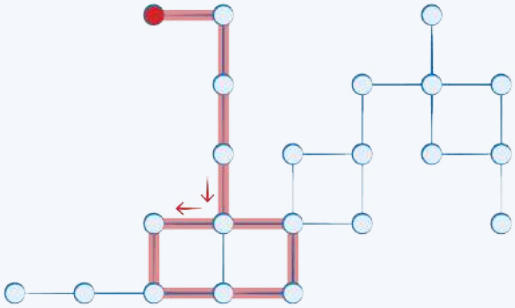
Figuur 3.7: $(3, \{3,2\}, 2, \{2,1\}, 1, \{1,5\}, 5)$. Pad, wandeling.



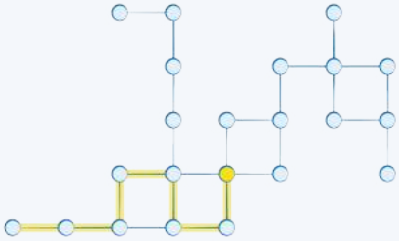
Figuur 3.8: $(1, \{1,2\}, 2, \{2,3\}, 3, \{3,4\}, 4, \{4,1\}, 1)$. Cykel, pad, wandeling.

Oefening 3.2.1: pad

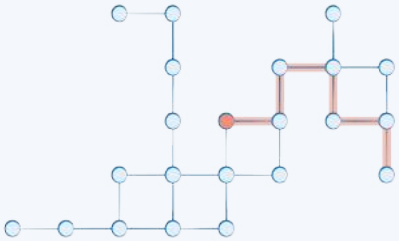
Bekijk de wandelingen op de figuren. De beginknoop van elke wandeling is ingekleurd.
Welke zijn een pad?



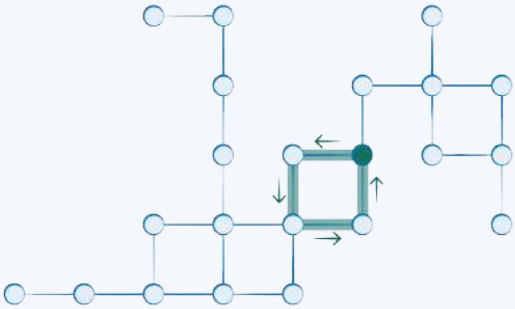
Figuur 3.9: ...



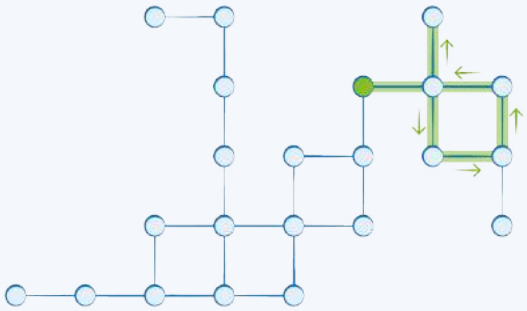
Figuur 3.10: ...



Figuur 3.11: ...



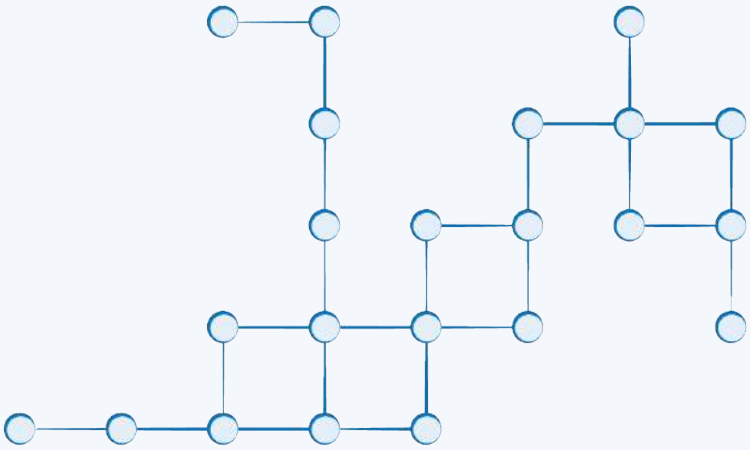
Figuur 3.12: ...



Figuur 3.13: ...

Oefening 3.2.2: wandeling, pad en cykel

Duid hieronder een wandeling aan die geen pad is. Duid een cykel aan.



3.3 *Gerichte graaf*

Een boog verbindt twee knopen eigenlijk op twee manieren, van de ene knoop naar de andere en omgekeerd.

Soms wordt echter een richting opgelegd aan de verbinding, je spreekt dan van een **gerichte boog**.

Dan kan je via die boog slechts van de ene knoop naar de andere, maar niet terug. Je stelt dit voor door een pijl te plaatsen op de boog (zie Figuur 3.14).

Figuur 3.14: Gerichte boog.

Versie 1.0

Gerichte graaf

Een graaf met **enkel** gerichte bogen is een gerichte graaf.

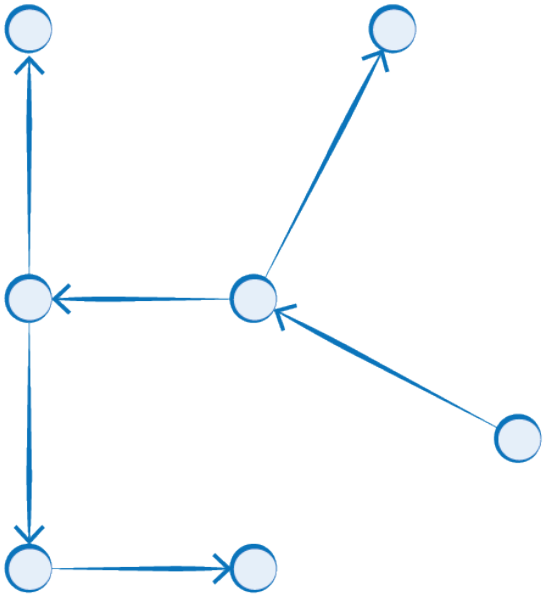
Figuur 3.15 toont een gerichte graaf.

Net zoals in een graaf kunnen een wandeling, een pad en een cykel in een gerichte graaf **gericht** zijn, zoals te zien op Figuur 3.16.

Voorbeeld: Wie wie volgt op Instagram, kan worden voorgesteld met een gerichte graaf (zie Figuur 3.17). De knopen stellen de Instagramaccounts voor. Een boog kan je in de zin van de pijl lezen als 'volgt'. Omdat bv. Axana James volgt, maar James Axana niet, wordt er gebruikgemaakt van gerichte bogen.

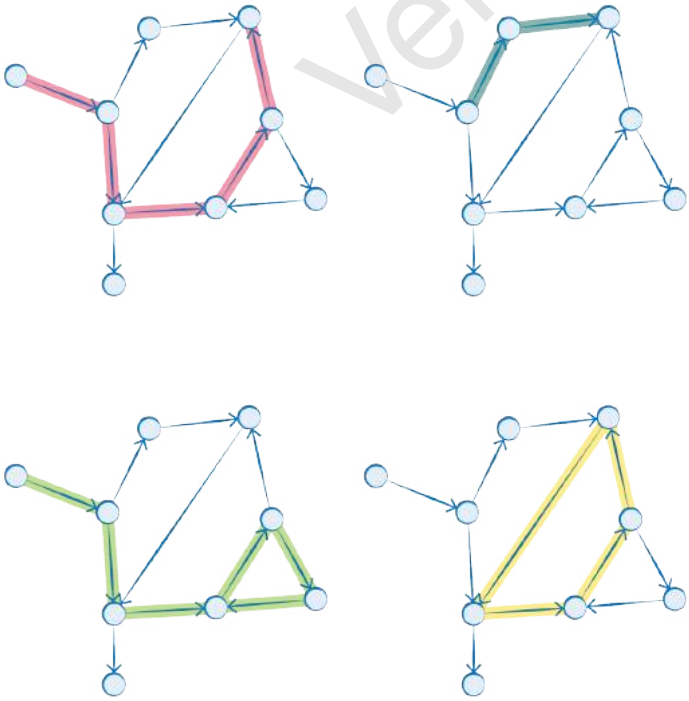
Oefening 3.3.1: Instagram

Vind je een cykel terug in Figuur 3.17? Is er een pad van Axana naar Tom?

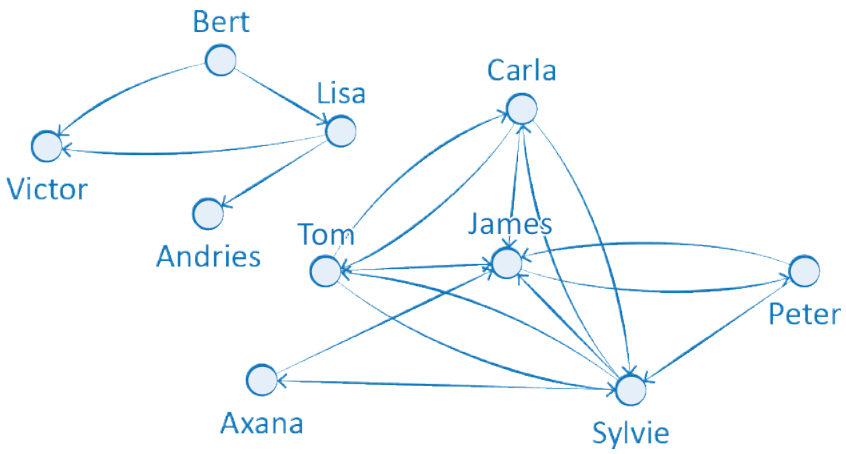


Figuur 3.15: Gerichte graaf.

Versie 1.0



Figuur 3.16: Gerichte wandelingen. De bordeauxrode, de donkerblauwe en de gele wandelingen zijn paden, de groene is geen pad, de gele wandeling is ook een cykel.



Figuur 3.17: Jongeren met een Instagramaccount en wie ze volgen. Een boog lees je in de zin van de pijl als 'volgt'.

3.4 *Samenhangende graaf*

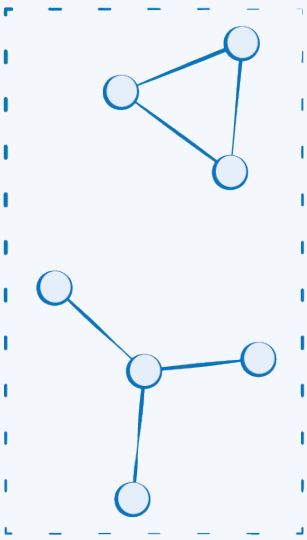
Samenhangende graaf

Een graaf waarbij er **tussen elke twee knopen een wandeling** is, is een samenhangende graaf.

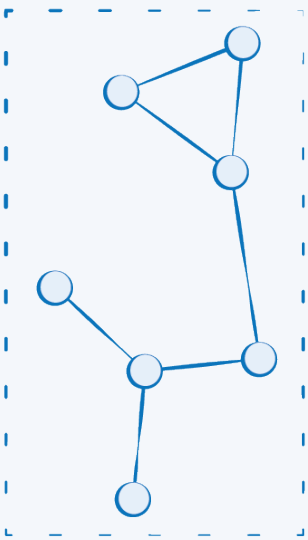
Bij een samenhangende graaf is dus elke knoop via een wandeling bereikbaar vanuit elke andere knoop.

Oefening 3.4.1: samenhangende graaf

De Figuren 3.18 en 3.19 tonen elk één graaf. Zie je welke graaf samenhangend is en welke niet?



Figuur 3.18: ...



Figuur 3.19: ...

3.5 Boom

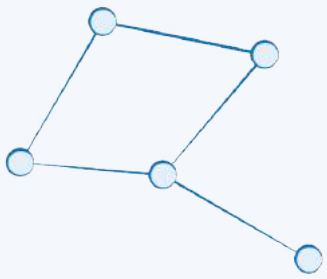
Boom

Een boom is een **samenhangende** graaf **zonder cykels**.

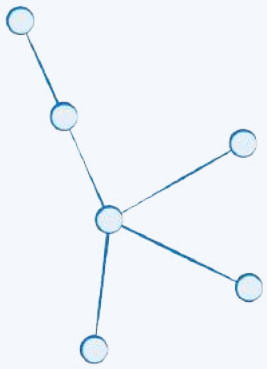
Oefening 3.5.1: boom of geen boom?

Is een van de grafen uit Figuren 3.18 en 3.19 een boom?

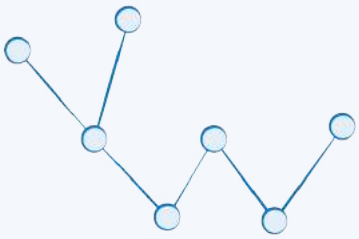
Herken je de bomen in de Figuren 3.20, 3.21 en 3.22?

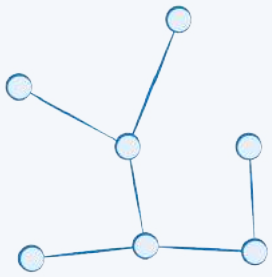


Figuur 3.20: ...



Figuur 3.21: ...





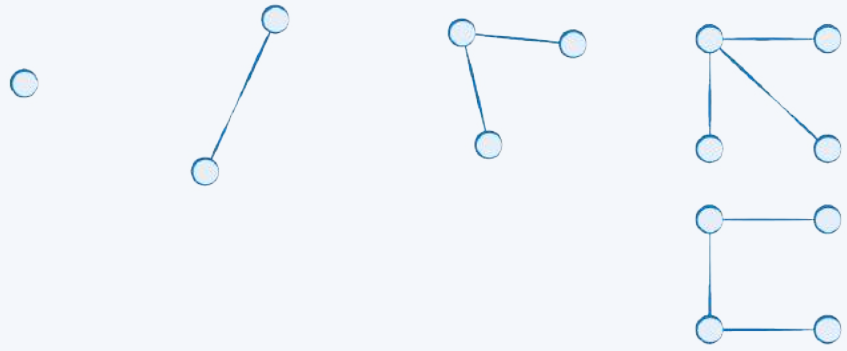
Figuur 3.22: ...

Eigenschap van een boom

Elke twee knopen van een boom zijn door precies één pad verbonden.

Oefening 3.5.2: aantal knopen van een boom

Er is slechts 1 boom met 1 knoop, 1 boom met 2 knopen en 1 boom met 3 knopen. Er zijn 2 bomen met 4 knopen. Bekijk ze in Figuur 3.23.



Figuur 3.23: Bomen met 1, 2, 3 of 4 knopen.

- Er zijn 6 bomen met 6 knopen. Teken ze. Hoeveel bogen hebben ze?
- Hoeveel bomen zijn er met 5 knopen?

3.6 Gewortelde boom

Soms wordt een van de knopen aangeduid als startknoop; deze startknoop wordt de **wortel** van de boom genoemd.

De mappenstructuur op een computer heeft bijvoorbeeld een gewortelde boomstructuur (zie Figuur 3.24).

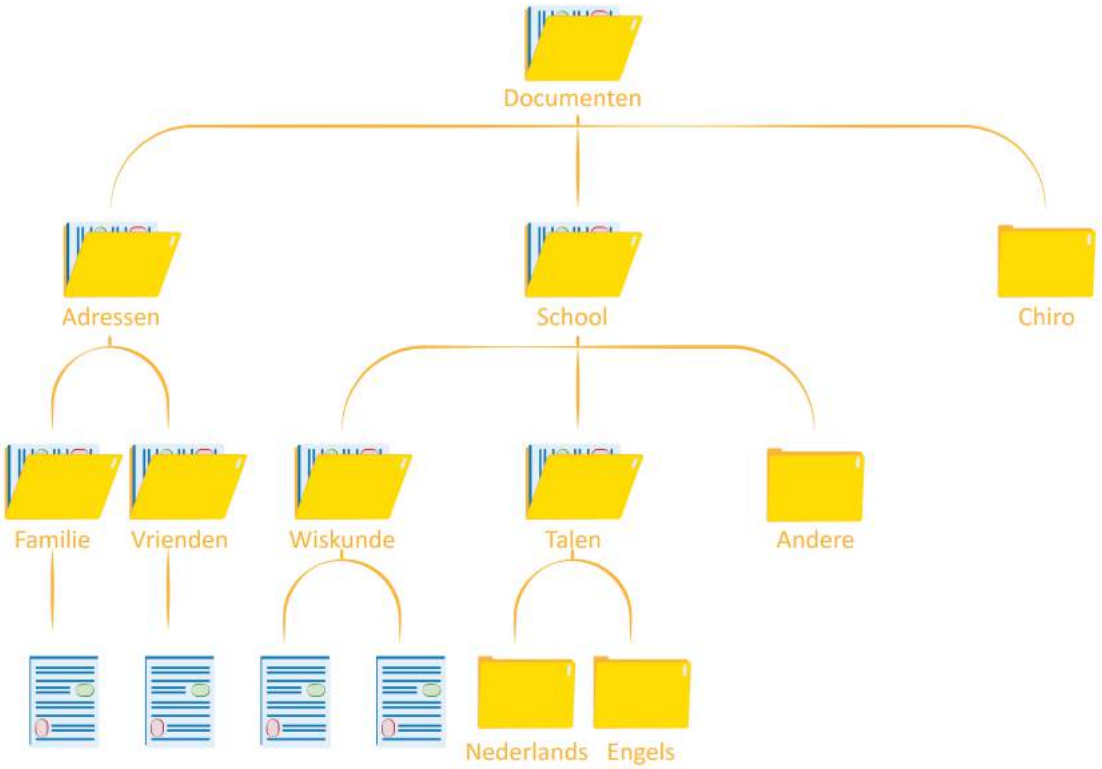
Een gewortelde boom wordt ondersteboven opgebouwd vanuit de wortel, en beschikt over **takken**, knopen en **bladeren**. Deze manier van werken laat toe om een boom op een overzichtelijke manier voor te stellen. Een voorstelling van zo'n boom is te zien in Figuur 3.25.

De bogen van een boom worden ook takken genoemd.

In Figuur 3.26 wordt dezelfde boom voorgesteld.

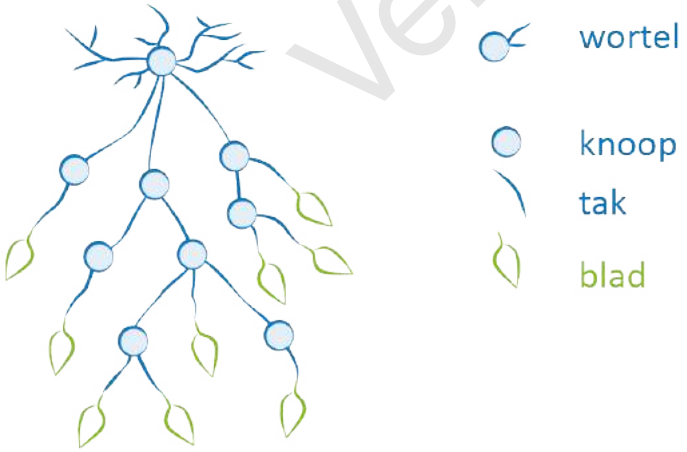
Gewortelde boom

Een gewortelde boom is een boom met de wortel als startknoop, en met **in elke knoop die geen blad is, een vertakking**.

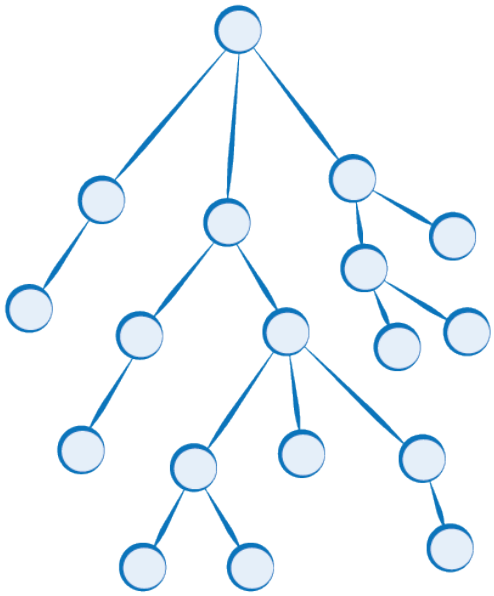


Figuur 3.24: Mappenstructuur op een computer.

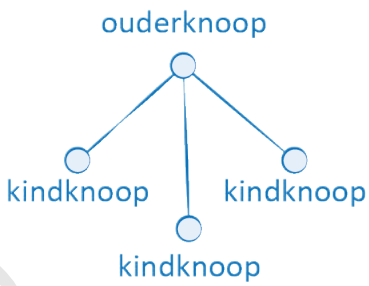
Versie



Figuur 3.25: Gewortelde boom.



Figuur 3.26: Dezelfde gewortelde boom.



Versie 1.0

Door de opbouw vanuit de wortel naar de bladeren toe, treedt er een hiërarchie op. Je kan spreken van ouder- en kindknopen (zie Figuren 3.25 en 3.26).

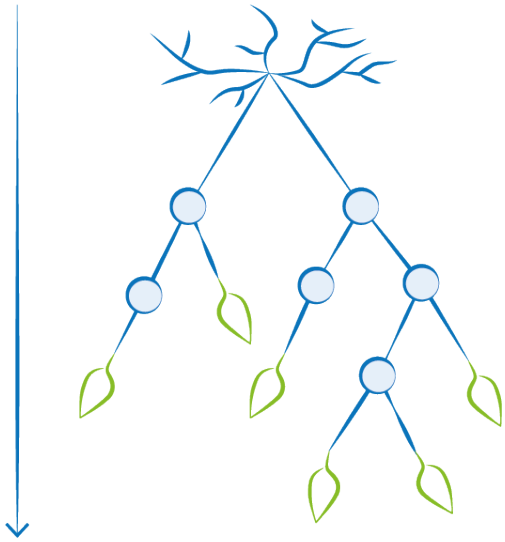
- Een vertakking in een ouderknoop leidt naar de kindknopen van die ouderknoop.
- Een zogenaamde ouderknoop kan één of meerdere kindknopen hebben.
- Een knoop die geen kinderen heeft, is een blad.

3.7 Binaire boom

Binaire boom

Een binaire boom is een **gewortelde boom** waarvan de knopen **ten hoogste twee kinderen** hebben.

De boom in Figuur 3.27 is een binaire boom. De gewortelde boom in Figuur 3.25 is geen binaire boom, want er zijn knopen met drie vertakkingen.



Figuur 3.27: Binaire boom.

Oefening 3.7.1: maximum aantal knopen van een binaire boom

De **diepte** van een knoop is het aantal takken dat doorlopen moet worden om vanuit de wortel tot die knoop te geraken (de wortel heeft diepte 0, de kinderen van de wortel hebben diepte 1, ...). De diepte van een boom is de grootste waarde die voorkomt bij de dieptes van de knopen van de boom.

Hoeveel knopen kan een binaire boom maximaal hebben, als die boom een diepte heeft van 1, 2, 3 of 4?

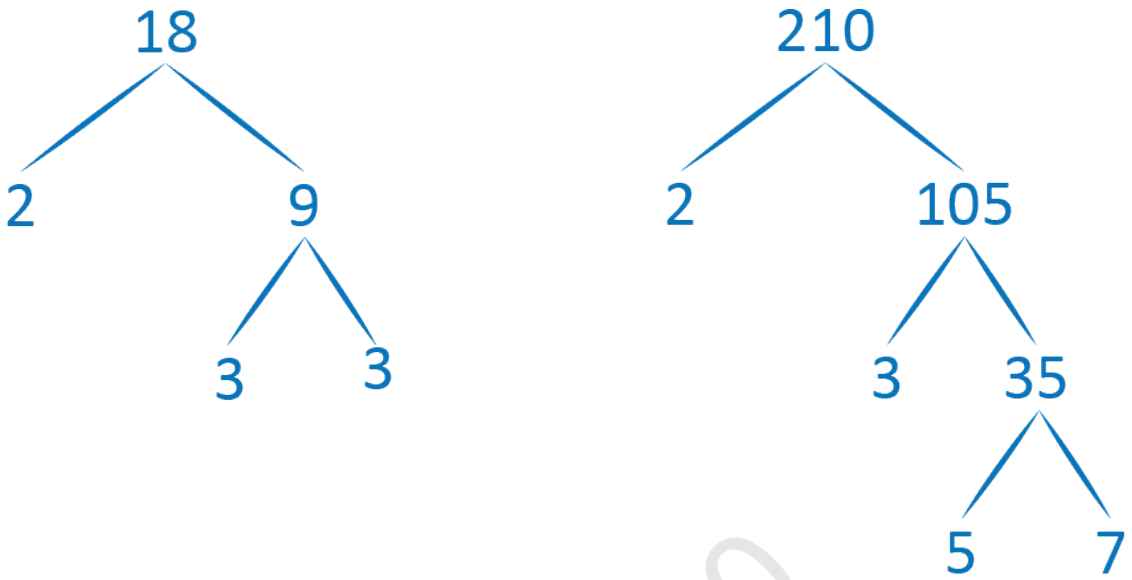
Kan je ook een formule vinden voor het maximum aantal knopen dat een binaire boom met diepte n kan hebben?

3.8 Gerichte boom

Ook een boom kan **gericht** zijn; dat geldt eveneens voor een gewortelde boom.

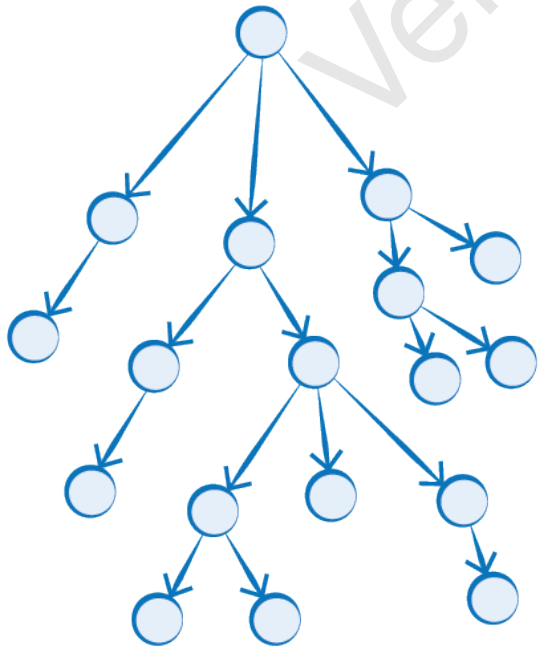
De bomen in Figuur 3.28 zijn binaire bomen die een ontbinding in priemfactoren voorstellen. Deze bomen kan je ook van onder naar boven doorlopen door te vermenigvuldigen. Om de ontbinding in priemfactoren te benadrukken kan je pijlen plaatsen op de takken (naar beneden gericht); je bekomt dan gerichte, binaire bomen.

De gerichte, gewortelde boom in Figuur 3.29 is geen binaire boom, want er zijn knopen met drie vertakkingen.



Figuur 3.28: Binaire bomen die de ontbinding in priemfactoren visualiseren; de bladeren geven de bekomen priemfactoren.
$18 = 2 \cdot 3 \cdot 3$;
$210 = 2 \cdot 3 \cdot 5 \cdot 7$.

Versie 1.0



Figuur 3.29: Gerichte, gewortelde boom.

Oefening 3.8.1: voorstellen van binaire bomen

De grafen hieronder zijn gerichte, binaire bomen. Verduidelijk dit door ze anders voor te stellen, van boven naar onder, van aan de wortel met de opeenvolgende vertakkingen, zoals in Figuur 3.29.

In dit hoofdstuk zag je heel wat soorten grafen passeren. Waarom bestaan al die soorten? Naargelang de aard van de toepassing moet de gebruikte graaf aan andere eisen voldoen, vandaar de noodzaak om steeds nieuwe soorten te bedenken.

Zie hoofdstuk 5 over computationeel denken.

Samengevat

Grafen zijn abstracte structuren die kunnen dienen om bepaalde relaties op een overzichtelijke manier in kaart te brengen. Grafen kennen toepassingen in veel verschillende domeinen, bv. onderzoek naar sociale netwerken. Naargelang de toepassing is een bepaalde soort graaf meer of minder geschikt. In dit hoofdstuk maakte je kennis met grafen, hun voorstelling, verschillende soorten grafen en de gehanteerde terminologie.

Na dit hoofdstuk ...

- weet je dat een graaf meerdere voorstellingen kan hebben;
- weet je dat er verschillende types van grafen zijn;
- ken je de definitie van een graaf, een wandeling, een pad, een cykel, een gerichte graaf, een samenhangende graaf, een boom, een gewortelde boom, een binaire boom;
- weet je wat een knoop, een boog, een buur van een knoop, een wortel, een tak, een blad, een ouderknoop, een kindknoop is;
- herken je de voorstellingen van eenzelfde graaf;
- herken je een wandeling, pad of cykel op een voorstelling van een graaf;
- kan je een wandeling, pad of cykel aanduiden op een voorstelling van een graaf.

Versie 1.0

HOOFDSTUK
VIER

BESLISSINGSBOMEN

4.1 Beslissingsboom

In onze maatschappij zijn er steeds meer data (gegevens) voorhanden. Data zijn echter maar interessant als je er informatie uit wint. Een van de technieken om dat te doen, is een **beslissingsboom**. Een beslissingsboom heeft een gewortelde boomstructuur, die je bijvoorbeeld ook terugvindt bij de mappenstructuur op je computer, en is een gerichte graaf.

Beslissingsboom

Een **beslissingsboom** is een gerichte, gewortelde boom die dient voor classificatie. De beslissingsboom stelt een algoritme voor dat ingevoerde elementen bij een bepaalde categorie indeelt, door opeenvolgende vragen te stellen en te beantwoorden over de kenmerken van die elementen.

Net zoals een andere gewortelde boom wordt een beslissingsboom opgebouwd met een wortel, takken, knopen en bladeren.

Voor uitleg over bomen en gewortelde bomen, zie hoofdstuk 3.

Je kan bv. een **classificatieprobleem** aanpakken met een beslissingsboom. Hoe gaat dat in zijn werk?

Bij een classificatieprobleem wil je de klasse achterhalen waartoe een input behoort. Als je zo'n classificatieprobleem aanpakt met een beslissingsboom, dan is die klasse voorspeld eens je terechtgekomen bent in een blad. Het blad waarin je uiteindelijk belandt, komt overeen met de voorspelde klasse.

Een **input** wordt ingevoerd in de boom langs de wortel. Die input volgt dan een **pad** langs de takken van de boom waarbij verschillende knopen worden gepasseerd. In een knoop wordt een beslissing genomen: er wordt een vraag gesteld en naargelang het antwoord wordt dan een bepaalde tak gevolgd. Op het einde van het gevolgde pad kom je terecht in een bepaald blad. Het blad geeft weer tot welke **klasse** de input behoort.

Maakt de beslissingsboom gebruik van vragen met twee mogelijke antwoorden, bv. ja of nee, dan is de beslissingsboom een binaire beslissingsboom.

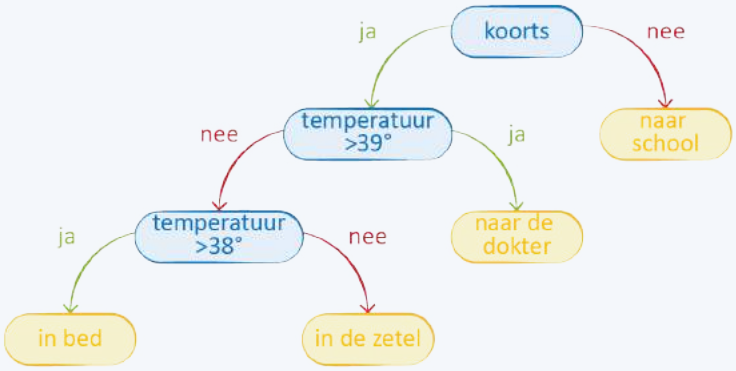
Bij een beslissingsboom heeft elke knoop die geen blad is, minstens twee kinderen.

In de volgende kader staat een voorbeeld van een **classificatieprobleem** dat wordt opgelost a.d.h.v. een binaire beslissingsboom.

Voorbeeld 4.1.1

Stel dat je op een dag opstaat en dat je je niet zo lekker voelt. Je zou dan je lichaamstemperatuur kunnen opnemen en aan de hand van de beslissingsboom in Figuur 4.1 kunnen beslissen wat te doen.

Er zijn vier klassen: 'naar school', 'naar de dokter', 'in de zetel', 'in bed', die je terugvindt in de bladeren van de boom. Er zijn drie knopen: de wortel 'koorts', de knoop 'temperatuur boven 39° Celsius' en de knoop 'temperatuur boven 38° Celsius'. In elke knoop wordt een ja-neevraag gesteld.



Figuur 4.1: Een eenvoudige beslissingsboom.

Oefening 4.1.2

Duid de wortel, knopen, bladeren en takken aan op de beslissingsboom in het voorbeeld.
Is het een binaire boom of niet?
Kleur het pad dat gaat van de wortel tot het blad 'naar de dokter'.

Figuur 4.2 illustreert hoe een bepaalde input aan een AI-systeem gegeven wordt en hoe dat AI-systeem op basis van een beslissingsboom beslist tot welke klasse de input behoort.

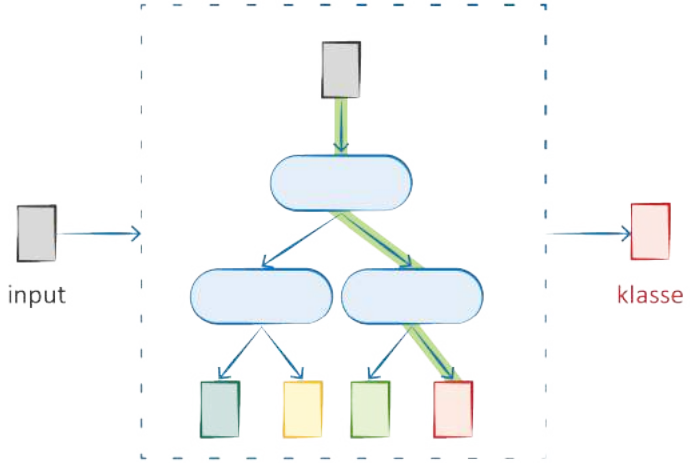
In een beslissingsboom kan je gemakkelijk nagaan welk pad het AI-systeem heeft gevolgd. Zulke systemen zijn dan ook transparant, in tegenstelling tot andere AI-systemen, bv. die die gebruikt worden voor beeldherkenning.

Beslissingsbomen zijn daarom populaire instrumenten binnen de gezondheidszorg. De voorbeelden die nu volgen, illustreren hoe gevarieerd deze worden ingezet en worden vormgegeven.

De voorstelling van een beslissingsboom kan in de praktijk verschillende gedaantes aannemen.

4.2 *Voorbeelden van beslissingsbomen*

Figuren 4.3 en 4.4 tonen een beslissingsboom die meegeeft hoe men op school best handelt met betrekking tot COVID-19.



Figuur 4.2: Via een beslissingsboom wordt de klasse voorspeld waartoe een input behoort.

Figuren 4.5 en 4.6 tonen beslissingsbomen voor eerste hulp, deze vertellen wat te doen in acute, medische situaties.

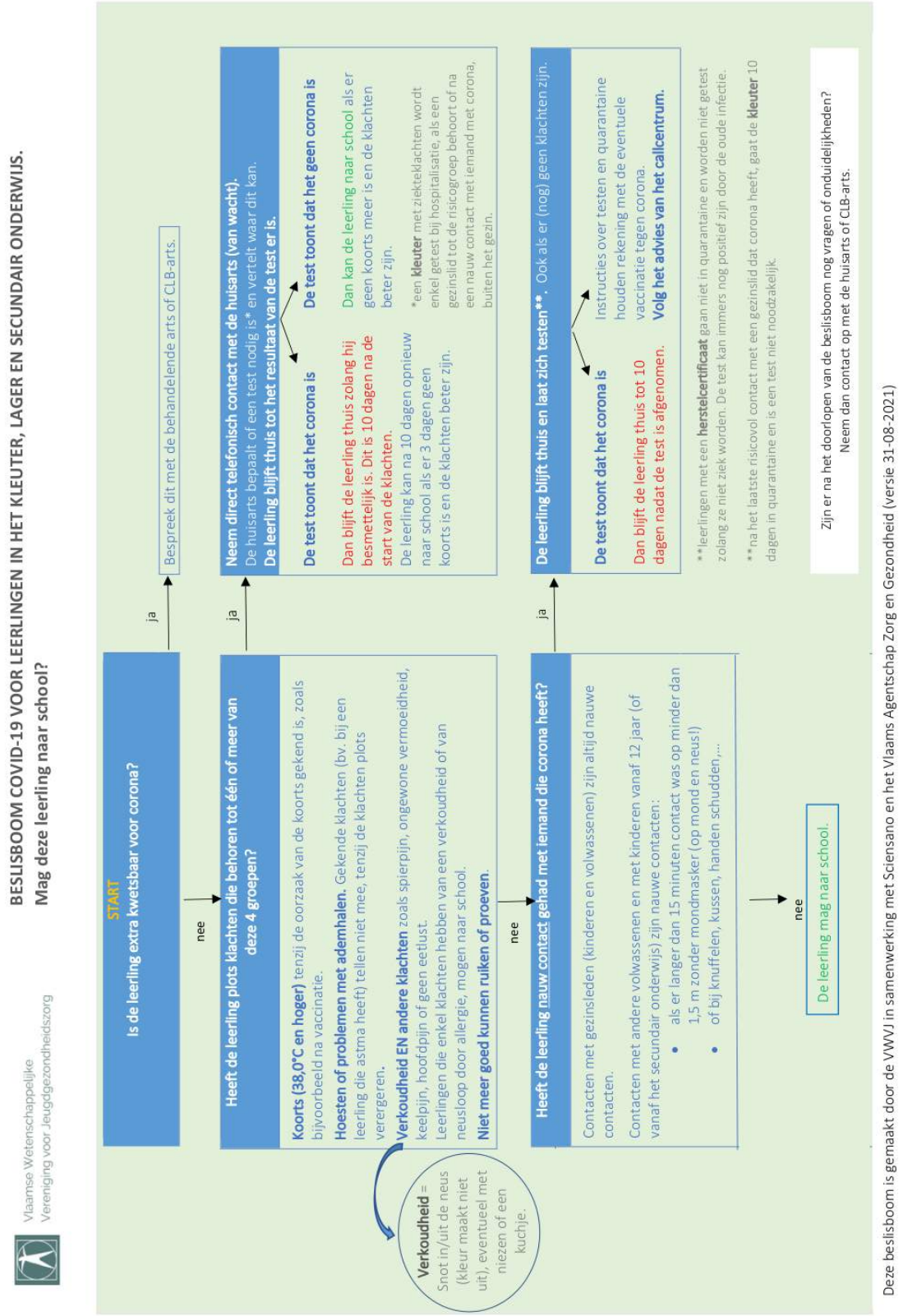
Figuur 4.7 toont een complexer voorbeeld van een beslissingsboom. Deze maakt deel uit van het onderzoeksproject 'mBrain', dat de ontwikkeling van een app die een migraineaanval voorspelt, als doel heeft (zie kader 'mBrain'). Met deze beslissingsboom wordt een probleem van binaire classificatie aangepakt: er zijn slechts twee klassen: 'Migraine' en 'Geen migraine' (Ongenae, 2021, persoonlijke communicatie). In de opbouw van deze beslissingsboom valt het op dat sommige takken van de beslissingsboom samenkomen in eenzelfde knoop. Toch is het wel een binaire boom; deze opbouw vermijdt echter onnodige herhaling en zorgt ervoor dat de boom compact blijft. Figuur 4.8 toont hoe groot de boom anders zou zijn. In de gegeven structuur is het meer transparant hoe het systeem tot een bepaalde beslissing komt.

Versie 1.0

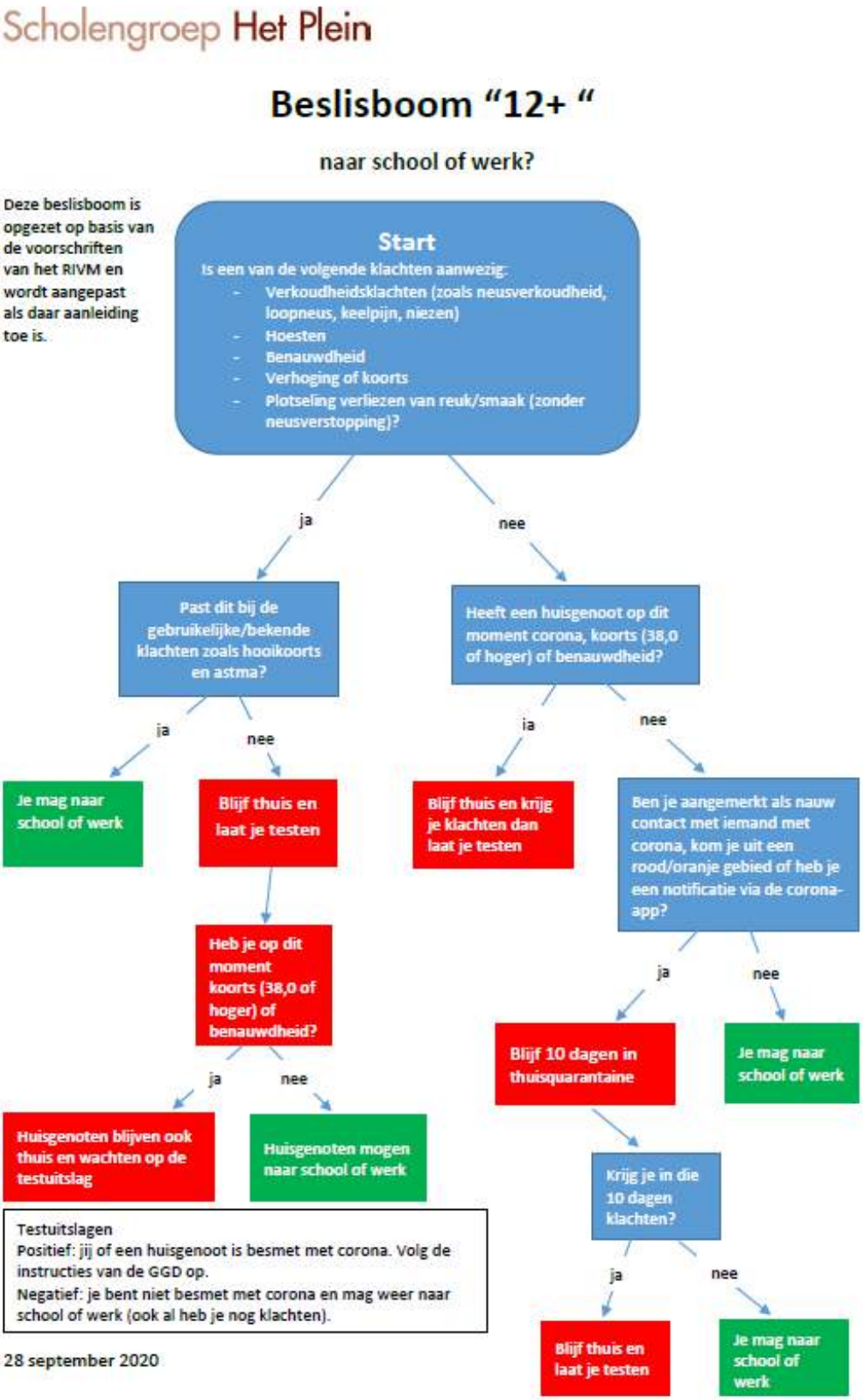
mBrain

Om een diagnose voor migraine, spanningshoofdpijn en andere vormen van hoofdpijn te kunnen stellen, baseert de neuroloog zich op een uitgebreid vraag- en antwoordgesprek en eventueel op een dagboek waarin de patiënt de frequentie van de aanvallen bijhoudt. M.b.v. wearables kunnen hoofdpijnpatiënten beter opgevolgd worden. Ze verzamelen gegevens van de patiënt, waaruit de app mBrain met een lerend systeem afleidt of de patiënt voldoende beweegt, hoe goed die slaapt en hoeveel stress die heeft. De patiënt kan in de app ook het gebruik van medicatie en het voorkomen van hoofdpijn bijhouden. mBrain zal de arts en de patiënt meer inzicht geven in het type hoofdpijn en heeft ook als doel om de triggers te bepalen die een aanval uitlokken en aldus een nieuwe aanval te voorspellen, wat de patiënt de mogelijkheid biedt om bv. preventief medicatie in te nemen (Van Hoecke et al., 2020).

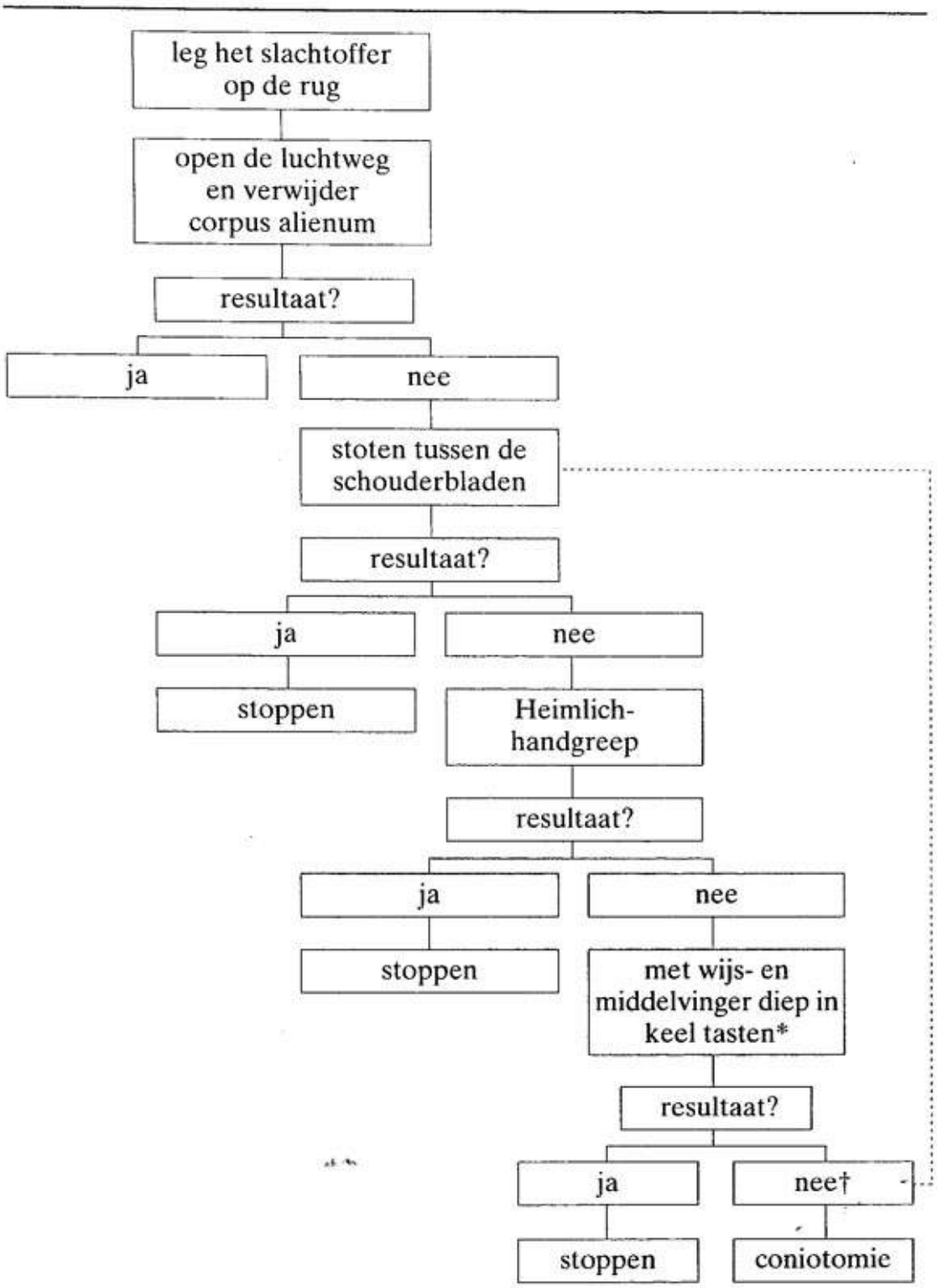
Het project is ontstaan vanuit een maatschappelijke nood: jaarlijks kunnen werknemers in Vlaanderen en Brussel een groot aantal dagen niet gaan werken door migraine. De onderzoekers zetten de nieuwe technologie van wearables en kunstmatige intelligentie in om dit probleem aan te pakken (GRAY, 2021).



Figuur 4.3: Beslissingsboom COVID-19 voor Vlaamse scholen (Vlaamse Wetenschappelijke Vereniging voor Jeugdgezondheidszorg, 2021).



Figuur 4.4: Beslissingsboom COVID-19 voor Nederlandse scholen (VO-Raad, 2020).

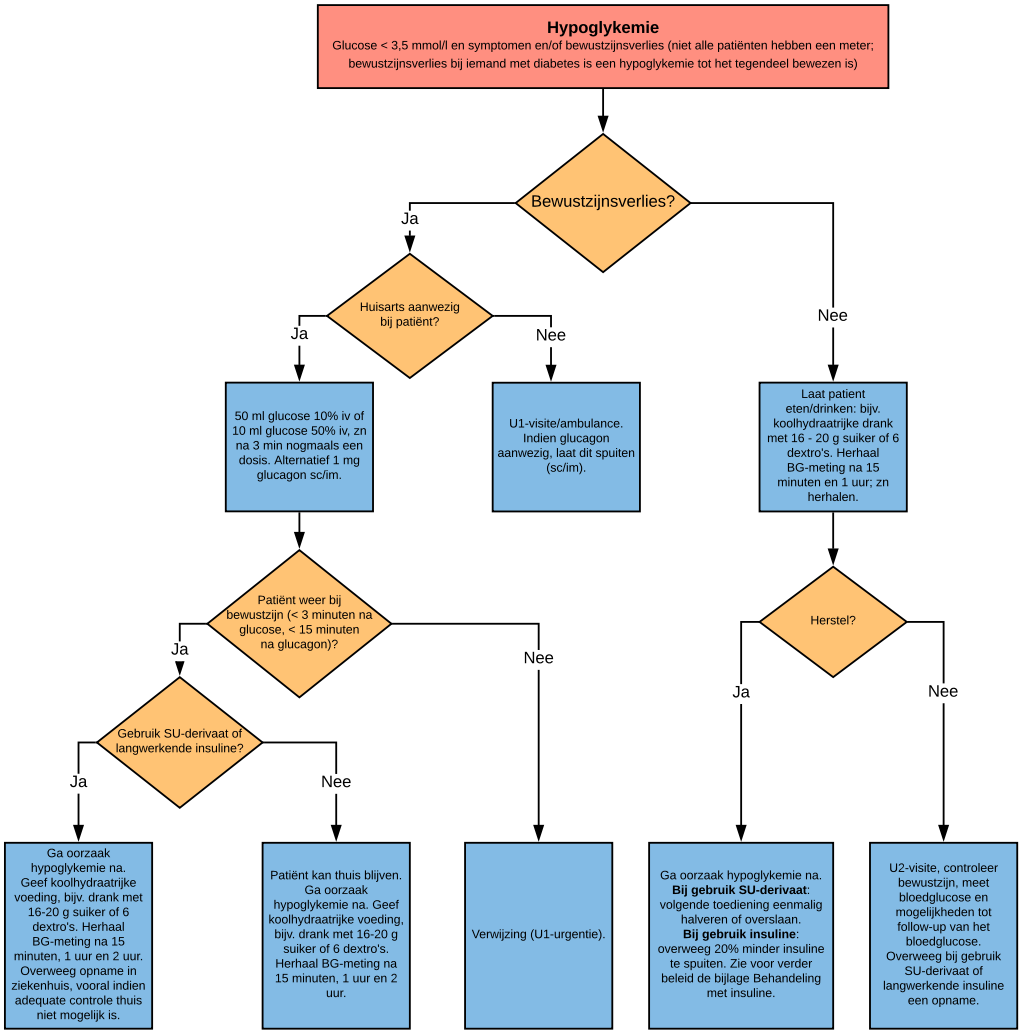


FIGUUR 4. Schema voor de behandeling van een luchtweg-obstructie bij een bewusteloos slachtoffer ouder dan 1 jaar; * bij kinderen alleen wanneer iets zichtbaar is, niet ter exploratie; † nog eenmaal het schema doorlopen.

Figuur 4.5: Te ondernemen actie bij verstikking bij een peuter (Lichtveld & van der Werken, 1996).

nhg
nederlands huisartsen genootschap

Beslisboom Hypoglykemie in de acute fase



BG: bloedglucose

Gebaseerd op: Langerhans School of Diabetes/Nederlands Huisartsen Genootschap. Protocollaire diabeteszorg. Apeldoorn/Utrecht: Langerhans School of Diabetes/Nederlands Huisartsen Genootschap, 2018.

Beslisboom Hypoglykemie in de acute fase

Figuur 4.6: Te ondernemen actie bij suikerpatiënt in acute fase (Barents et al., 2021).

Er worden ook beslissingsbomen ontwikkeld om artsen te helpen bij het stellen van een diagnose. Deze beslissingsbomen worden soms ontwikkeld m.b.v. een AI-systeem. Soms maken ze zelf deel uit van een AI-systeem dat op basis van kenmerken, die de arts invoert of die het systeem zelf uit het elektronisch patiëntendossier haalt, beslist wat een mogelijke diagnose zou kunnen zijn. Deze systemen dienen ter ondersteuning van de arts en zijn niet ontworpen om de expertise van de arts te vervangen.

Figuur 4.9 toont een voorbeeld van zo'n beslissingsboom (nog in onderzoeksfase) voor het stellen van een diagnose voor oogziekten. Deze beslissingsboom werd ontwikkeld m.b.v. artificiële intelligentie a.d.h.v. voorbeelden van de symptomen die patiënten vertonen en de gestelde diagnose (Kabari & Nwachukwu, 2012). De verschillende symptomen waarmee er rekening werd gehouden, zijn opgesomd in Tabel 4.1.

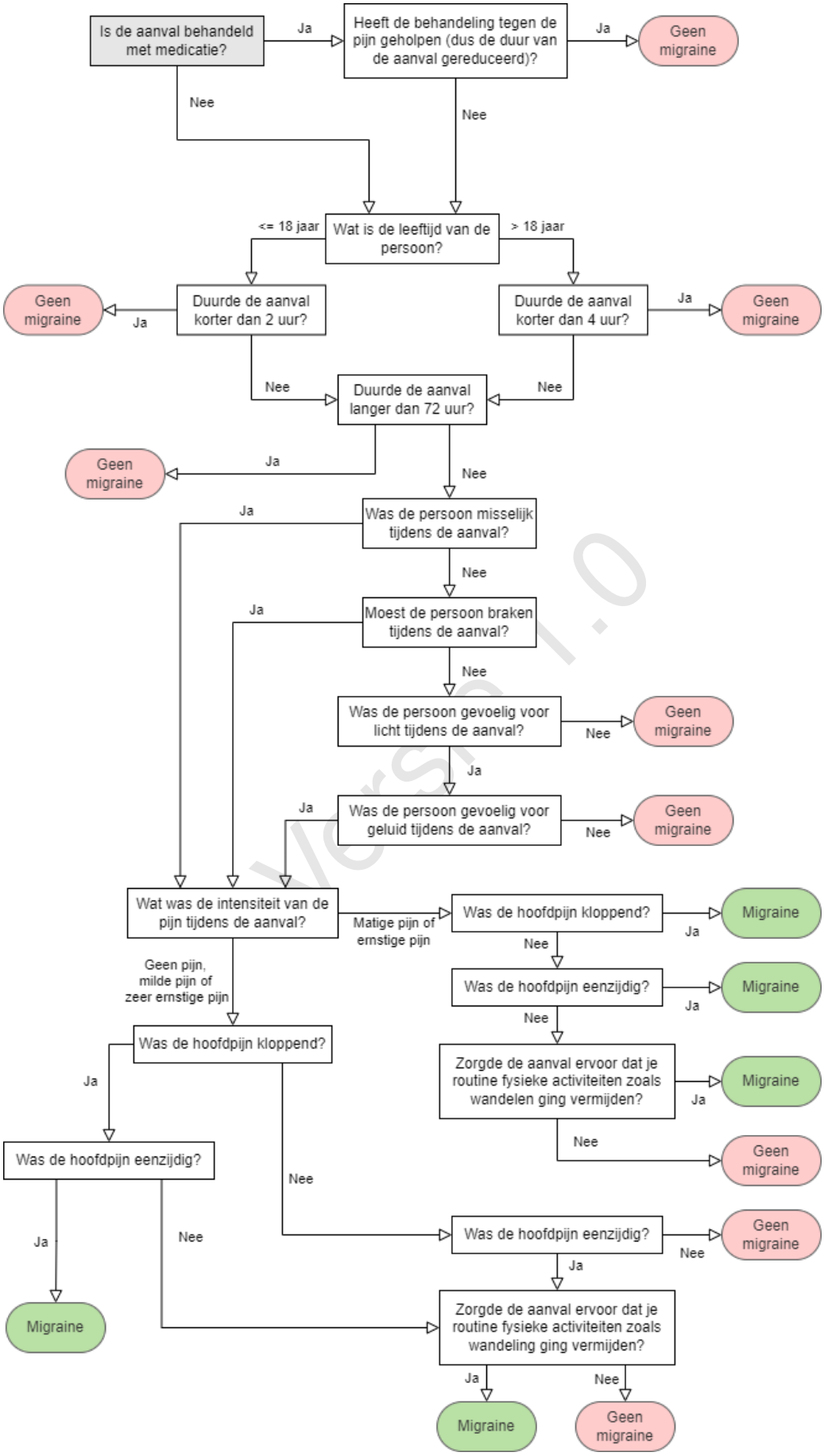
Merk op dat de beslissingsboom sommige symptomen niet in beschouwing neemt.

De opbouw van deze beslissingsboom wijkt af van wat je zou verwachten. Meestal worden enkel de takken getoond die overeenkomen met het antwoord 'ja' op de vraag of een symptoom aanwezig is.

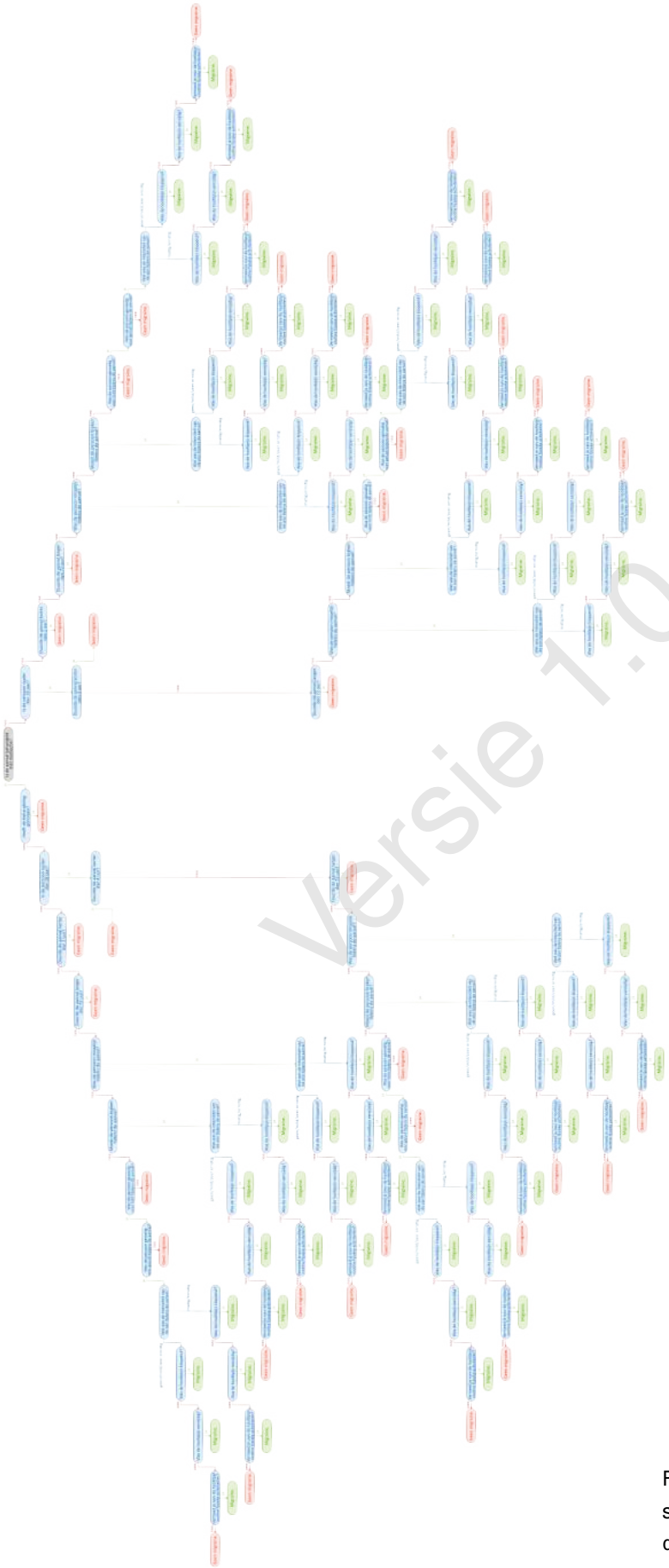
Versie 1.0

X1	Pijn in het oog	**X12**	Zicht herstelt zich traag na blootstelling aan fel licht
X2	Rood of roze oog	**X13**	Irritatie, jeuk, schurend of brandend gevoel aan het oog
X3	Helder licht of speciale zonnebrilglazen verbeteren het zicht	**X14**	Oncomfortabel bij langdurig, geconcentreerd gebruik van het oog
X4	Slecht zicht 's nachts	**X15**	Moeilijk kleuren onderscheiden
X5	Oogproblemen in de familie	**X16**	Zwevende stippen, flitsen, halo's rond licht
X6	Omtrek gezichtsveld is verkleind	**X17**	Tranende ogen of een andere uitscheiding van het oog
X7	Ouder dan 45 jaar	**X18**	Gezwollen oog
X8	Troebel zicht	**X19**	Wazige cornea of wazig oog
X9	Minder troebel zicht door te pinken	**X20**	Gevoelig voor licht
X10	Vervormd beeld	**X21**	Troebel zicht van verre objecten
X11	Vlokkerige substantie voor de ooglens	**X22**	Troebel zicht van dichte objecten

Tabel 4.1: Symptomen bij oogziekte (Kabari & Nwachukwu, 2012).

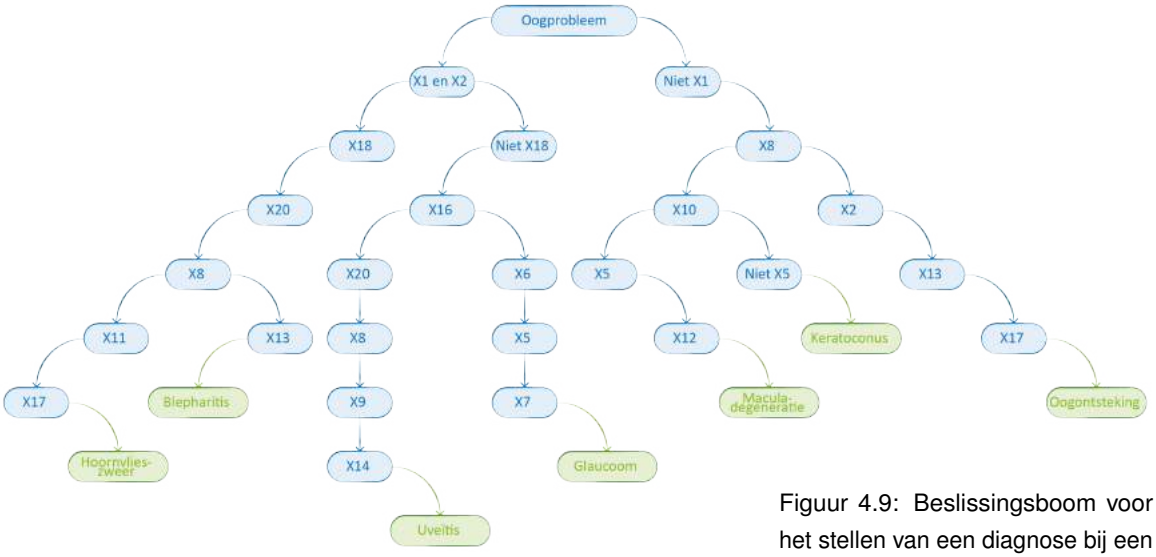


Figuur 4.7: Beslissingsboom migraine (Ongenae, 2021).



Figuur 4.8: Niet-compacte beslissingsboom migraine. Bekijk op dwengo.org/zorg.

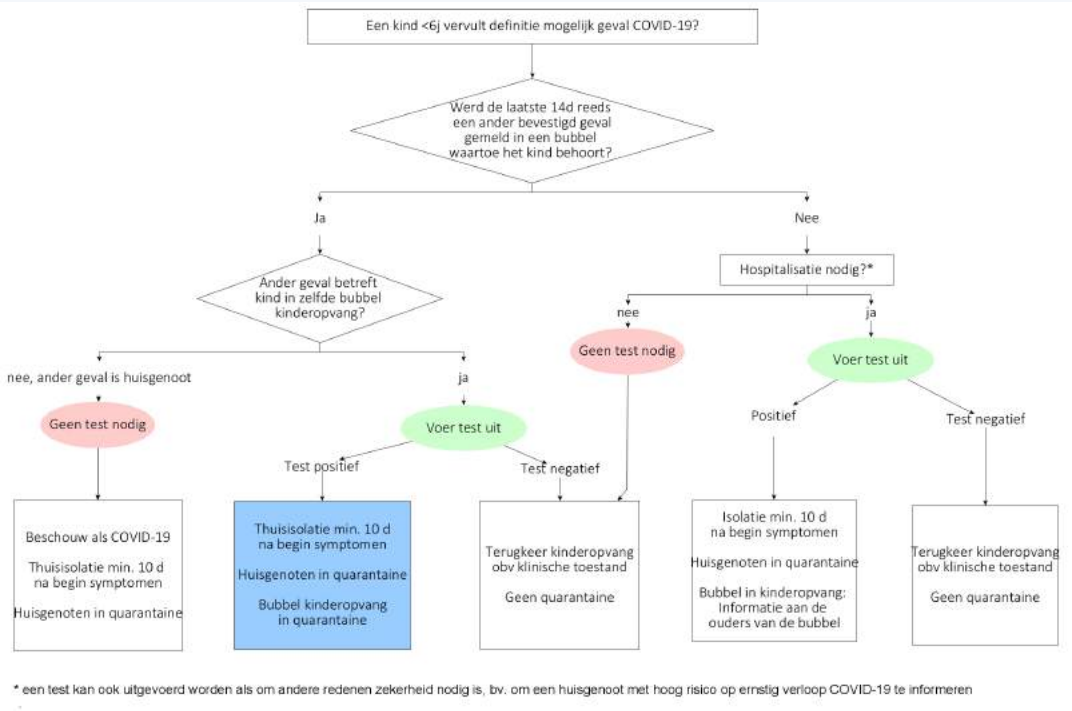
Versie 1.0



Figuur 4.9: Beslissingsboom voor het stellen van een diagnose bij een oogziekte (Kabari & Nwachukwu, 2012).

Oefening 4.2.1

Duid de wortel aan op de volgende beslissingsboom.
Is het een binaire boom of niet?
Kleur de paden die gaan van de wortel tot een blad waarin 'Geen quarantaine' staat.

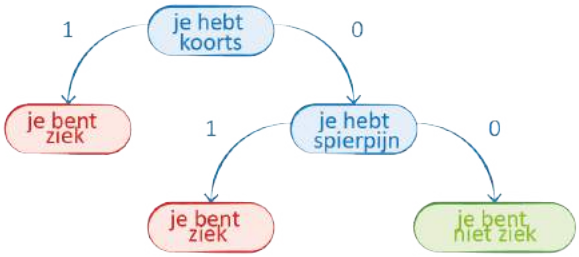


Figuur 4.10: 'Wanneer kinderen onder 6 jaar met symptomen van COVID-19 testen' (Sciensano, 2021).

4.3 Logica

De paden in een beslissingsboom kunnen steeds omgezet worden naar logische uitdrukkingen.

Voorbeeld: In Figuur 4.11 staat een binaire beslissingsboom afgebeeld.



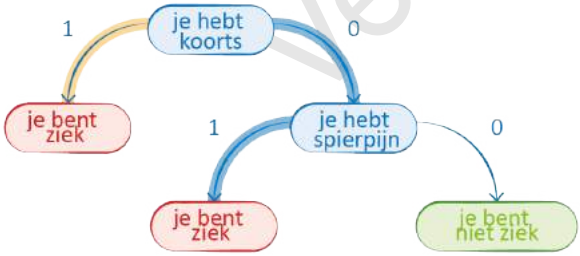
Figuur 4.11: Logische uitdrukkingen gekoppeld aan beslissingsboom.

Uit de bladeren zijn twee klassen af te leiden: ‘je bent niet ziek’ en ‘je bent ziek’, het gaat dus om een probleem van binaire classificatie. In elke knoop staat een uitdrukking die waar of vals (niet waar) kan zijn. het is dus een binaire beslissingsboom. Als de uitdrukking waar is, dan wordt de pijl met een 1 erop gevolgd, in het andere geval wordt de pijl met de 0 erop gevolgd.

Let op: het feit dat het een binaire boom is, is niet gelinkt aan de binaire classificatie.

Denk aan de logische variabelen ‘True’ en ‘False’.

Versie 1.0



Figuur 4.12: Logische uitdrukkingen gekoppeld aan beslissingsboom.

Op Figuur 4.12 is aangeduid wanneer een input ‘terechtkomt’ in de klasse ‘je bent ziek’.

Dat is zo in twee gevallen, namelijk als:

- `je hebt koorts` waar is;
- `je hebt koorts` vals is en `je hebt spierpijn` waar is.

Dus de paden die leiden tot de klasse ‘je bent ziek’, kunnen omgezet worden naar de logische uitdrukking:

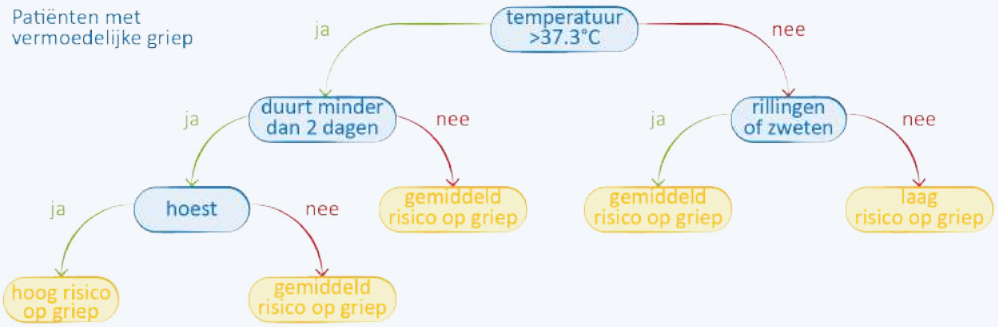
(¬je hebt koorts ∧ je hebt spierpijn) ∨ (je hebt koorts).

Als deze uitdrukking waar is, dan komt de input terecht in de klasse 'je bent ziek'.

Oefening 4.3.1: beslissingsboom diagnose seizoensgriep

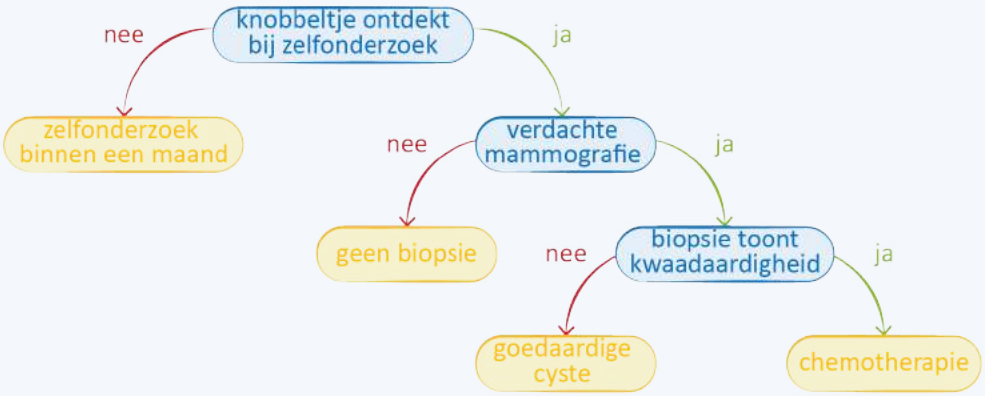
Wanneer komt een input terecht in klasse 'gemiddeld risico op griep'?

- Bekijk de beslissingsboom voor een classificatie in drie klassen (Afonso et al., 2012).
- Duid de paden aan die terechtkomen in een blad 'gemiddeld risico op griep'.
- Geef de logische uitdrukking die in deze beslissingsboom leidt naar 'gemiddeld risico op griep'.



Oefening 4.3.2

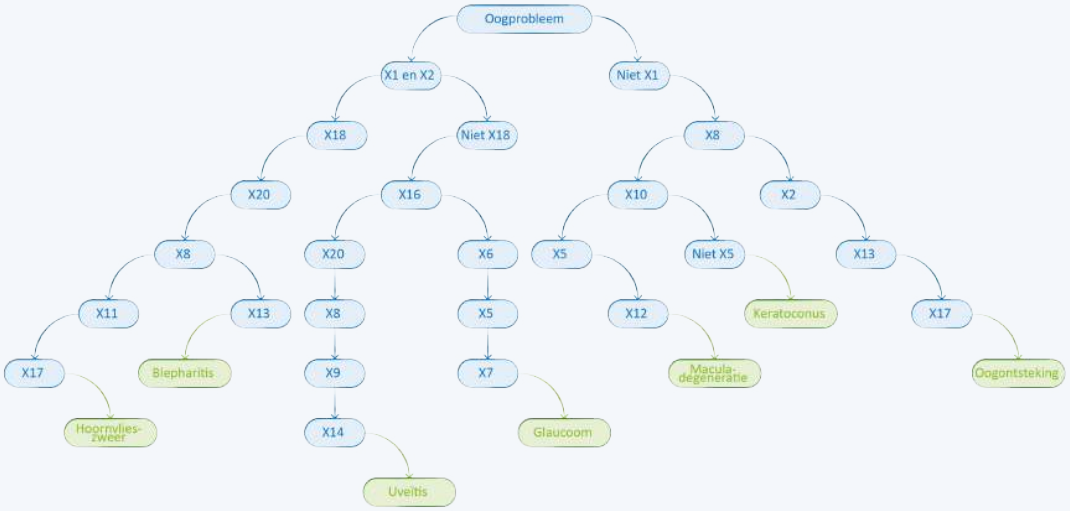
Geef de logische uitdrukkingen die corresponderen met onderstaande beslissingsboom.



Figuur 4.13: Beslissingsboom borstkankerdiagnose (Cruz & Wishart, 2006).

Oefening 4.3.3

Geef de logische uitdrukkingen die corresponderen met de diagnose ‘Glaucoom’ die gesteld is a.d.h.v. onderstaande beslissingsboom.



Figuur 4.14: Beslissingsboom voor enkele oogziekten (Kabari & Nwachukwu, 2012).

Oefening 4.3.4

Een dokter heeft op basis van zijn eigen patiënten een beslissingsboom opgesteld waarmee hij snel kan voorspellen of iemand risico loopt tot het krijgen van een hartaanval. Hij neemt daar verschillende parameters voor in rekening: pijn in de borststreek, het geslacht en of de patiënt rookt en/of voldoende beweging neemt.

In deze beslissingsboom komen de paden die tot de klasse ‘Hartaanval’ leiden, overeen met de logische uitdrukking:
pijn $\vee$ ($\neg$ pijn $\wedge$ $\neg$ man $\wedge$ rookt)
en de paden die tot de klasse ‘Geen hartaanval’ leiden, komen overeen met de logische uitdrukking:
($\neg$ pijn $\wedge$ man) $\vee$ ($\neg$ pijn $\wedge$ $\neg$ man $\wedge$ $\neg$ rookt).

Geef een grafische voorstelling van de beslissingsboom die de dokter opstelde.

4.4 Belangrijke aspecten

Sommige aspecten betreffende de talrijke voorbeelden in dit hoofdstuk zijn van belang voor wat volgt.

Opdracht 4.4.1

- Waarom bestaan er verschillende types van grafen?
- Wat weet je over een graaf van het type 'beslissingsboom'?
- Waarom is een beslissingsboom veelgebruikt in de zorgsector?

Samengevat

Beslissingsbomen worden veel gebruikt in de medische wereld. Ze geven vaak een houvast aan zorgverleners bij hoe te handelen in een specifieke, bv. niet courante, situatie. Sommige beslissingsbomen dienen dan weer om artsen te helpen bij het stellen van een diagnose.

AI-systemen die gebruikmaken van een beslissingsboom, zijn transparant. Het is immers eenvoudig na te gaan welk pad het systeem volgde om tot een bepaalde beslissing te komen.

Na dit hoofdstuk ...

- weet je dat een beslissingsboom een gerichte graaf is;
- herken je een beslissingsboom aan zijn structuur;
- kan je een binaire beslissingsboom onderscheiden van een niet-binaire;
- begrijp je dat een beslissingsboom ingezet kan worden als oplossing voor een classificatieprobleem;
- kan je voorbeelden geven van toepassingen van beslissingsbomen in de zorgsector;
- kan je voorbeelden geven van de wisselwerking tussen nieuwe ontwikkelingen in wetenschappen en technologie, en de maatschappij (meer bepaald de zorgsector);
- kan je de paden in een beslissingsboom omzetten naar logische uitdrukkingen;
- zie je het verband tussen een beslissingsboom en een logische uitdrukking.

Versie 1.0

HOOFDSTUK
VIJF

COMPUTATIONEEL DENKEN

5.1 Computationeel denken

We leven in een technologische kennismaatschappij, snel veranderend door de aanhoudende ontwikkelingen in digitalisering, de niet te stoppen stroom aan informatie en de opkomst van artificiële intelligentie. Het is van belang dat je je weerbaar en flexibel kunt opstellen tegenover nieuwe digitale toepassingen. Weten hoe ze te gebruiken, is niet voldoende. **Computationeel denken is een werkwijze om de digitale wereld te begrijpen en erop in te grijpen.** Door inzichten te verwerven in de concepten die aan de basis liggen van de digitalisering, kan je de mogelijkheden, de beperkingen en de eventuele risico's van deze toepassingen beter inschatten; nieuwe technologieën zijn minder beangstigend. Computationeel kunnen denken kan er zo voor zorgen dat je beter kan functioneren in en kan participeren aan de gedigitaliseerde maatschappij, en dat je de wisselwerking tussen technologische ontwikkelingen en de evoluties in de maatschappij begrijpt.

Het gaat om begrijpen welke impact de digitalisering heeft op jouw eigen leven en in de toekomst als burger gefundeerde beslissingen kunnen nemen.

versie 1.0

5.1.1 Wat houdt computationeel denken in?

Door te leren computationeel denken en handelen verwerf je vaardigheden, zoals het krijgen van een beter overzicht in **complexe problemen** en het kunnen **inzetten van de computer** als hulpmiddel om een probleem op te lossen.

`https://onderwijsdoelen.be/uitgangspunten/4814`

In de definitie (zie kader 'Een definitie') worden vier basisconcepten van computationeel denken naar voren geschoven: **decompositie, abstractie, patroonherkenning en algoritme**. Internationaal is er een consensus dat computationeel denken op deze vier basisconcepten steunt.

Definities van de basisconcepten vind je in paragraaf 5.1.2.

Computationeel denken is echter niet beperkt tot deze basisconcepten, maar omvat nog **andere concepten en praktijken**, zoals problemen oplossen, debuggen, data representeren en modelleren.

Een definitie

Computationeel denken is een methodologie om een complex probleem zo aan te pakken dat je een computer zou kunnen inzetten om het op te lossen.

Computationeel denken steunt op vier basisconcepten: decompositie, abstractie, patroonherkenning en algoritmisch denken, maar omvat nog andere concepten en praktijken.

Voor een bepaald probleem zijn deze basisconcepten vaak verweven met elkaar. Computationeel denken houdt in dat het op te lossen probleem goed begrepen en geformuleerd wordt, er een oplossing bedacht wordt, er een algoritme opgesteld en eventueel geïmplementeerd wordt in de computer.

Een belangrijk aspect is het herkennen van problemen waarbij het interessant is om de computer als hulpmiddel in te zetten.

Voor sommige problemen is dit een efficiënte methodologie om tot een oplossing te komen, of leidt deze methodologie ertoe dat je gemakkelijker de computer kan inzetten om het probleem op te lossen.

Het begrip ‘computer’ wordt gebruikt in de ruime betekenis van het woord: het geheel van hardware en software van computers, smartphones, tablets en andere informatieverwerkende systemen.

De definitie van Denning & Tedre benadrukt het bredere beeld van computationeel denken, zie kader ‘Definitie van Denning en Tedre’.

Definitie van Denning en Tedre

Denning en Tedre geven als definitie van computationeel denken (Denning & Tedre, 2019):

“Computational thinking is the mental skills and practices for

- *designing computations that get computers to do jobs for us, and*
- *explaining and interpreting the world as a complex of information processes.”*

“Computationeel denken omvat de denkvaardigheden en de praktijken voor

- het ontwerpen van berekeningen en instructies die ervoor zorgen dat computers taken voor ons voltrekken, en
- het uitleggen en het interpreteren van de wereld als een complex systeem van informatieprocessen.”

5.1.2 Definities van de basisconcepten

De basisconcepten van computationeel denken

- **Decompositie**:
  - Een probleem doordacht opsplitsen in goedgekozen deelproblemen, zodat elk deel afzonderlijk kan worden aangepakt, en een oplossing van het probleem gemakkelijker bekomen wordt.
- **Patroonherkenning**:
  - Achterhalen dat bepaalde aspecten van een probleem onderling gelijkenissen vertonen, waardoor het vereenvoudigd kan worden.
  - Herkennen dat een probleem gelijkenissen vertoont met een eerder opgelost probleem.
- **Abstractie**:
  - Negeren van informatie die niet nodig is om een probleem op te lossen.
  - Bepaalde details verbergen om in grote lijnen over een probleem te kunnen nadenken.

  Bij het aanpakken van complexe problemen is er nood aan abstractie, omdat het redeneren vlotter verloopt zonder de ballast van irrelevante details.
- **Algoritmisch denken**:
  - Expliciteren van een reeks eenduidige instructies die stapsgewijs moeten worden uitgevoerd.
  - Inzien dat deze reeks instructies en de volgorde ervan essentieel zijn om het gewenste resultaat op te leveren.

5.2 Project 'AI in de Zorg'

5.2.1 Unplugged voorbeeld 'Locked-insyndroom'

Op `dwengo.org/zorg` vind je ook een *unplugged* voorbeeld van computationeel denken uit de zorgwereld, nl. een voorbeeld over het locked-insyndroom.

Een unplugged voorbeeld is een voorbeeld waar geen computer voor nodig is.

Dat voorbeeld maakt duidelijk dat computationeel denken om mensen gaat, en dat men bij het ontwerpen van de oplossing van een probleem best de persoon betrekt voor wie die oplossing bedoeld is. Het illustreert ook dat een bestaande oplossing door computationeel te denken misschien efficiënter kan gemaakt worden.

Zie het leerpad 'Unplugged' op `dwengo.be/zorg`, onderdeel 'Locked-in'.

Voor de oorspronkelijke versie, met dank aan Paul Curzon, zie `https://cs4fndownloads.files.wordpress.com/2016/02/searchingtospeak-booklet.pdf`.

5.2.2 Computationeel denken bij de uitdaging

In de inleiding werd het volgende probleem vooropgesteld:

Uitdaging

Je beschikt over gelabelde data uit de zorgsector. Laat een computer jou helpen om deze data zo voor te stellen dat je er beslissingen mee kan nemen betreffende een diagnose of een behandeling.

Om een oplossing te zoeken voor deze uitdaging ga je in op het brede plaatje van computationeel denken. Je zal data representeren op een geschikte manier om een zorgverlener te helpen bij het stellen van een diagnose of het bepalen van een behandeling. Je zoekt dus een oplossing voor een probleem waar een mens mee geholpen is.

Het creatieve proces waarin je nadenkt over hoe je de computer voor problemen zoals deze inzet, is de essentie van computationeel denken.

Soms komt het voor dat je een bepaalde taak door een computer wil laten doen, maar dat daarvoor nog geen geschikt systeem bestaat. Computationeel denken houdt dan onder meer het volgende in: **je bewust zijn van de beperkingen van het computersysteem dat je ter beschikking hebt, de mogelijkheden van de te gebruiken software en hardware kennen, nadenken over waarom bestaande systemen ontoereikend zijn en het ontwerpen van nieuwe systemen.**

Opdracht: ga de uitdaging aan

Ga de uitdaging aan a.d.h.v. de basisconcepten van computationeel denken.

We helpen je alvast even op weg:

- Welke voorstelling van de data is er geschikt?
- Met welke factoren die het proces bemoeilijken, word je geconfronteerd?

Samengevat

Computationeel denken is een methodologie om een complex probleem zo aan te pakken dat je een computer zou kunnen inzetten om het op te lossen.
Het is een werkwijze om de digitale wereld te begrijpen en erop in te grijpen.
Computationeel denken steunt op vier basisconcepten: decompositie, abstractie, patroonherkenning en algoritmisch denken, maar omvat nog andere concepten en praktijken.

Na dit hoofdstuk ...

- weet je wat computationeel denken inhoudt;
- zie je in hoe computationeel denken helpt om een probleem aan te pakken waarbij je de hulp wenst in te roepen van de computer.

HOOFDSTUK
ZES

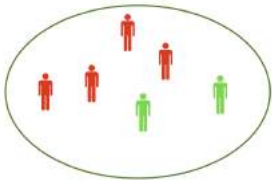
BINAIRE BESLISSINGSBOMEN CONSTRUEREN

6.1 Spreiding over categorieën

Uitdaging

Je beschikt over gelabelde data uit de zorgsector. Laat een computer jou helpen om deze data zo voor te stellen dat je er beslissingen mee kan nemen betreffende een diagnose of een behandeling.

Stel dat een huisarts zijn patiënten in twee categorieën wil opdelen: de ernstig zieke patiënten en de patiënten die niet ernstig ziek zijn (zie Figuur 6.1). Stel dat de arts dat wil doen met een beslissingsboom.



Figuur 6.1: Een groep patiënten die spreiding vertoont over twee categorieën: de rode patiënten zijn ernstig ziek, de groene patiënten zijn niet ernstig ziek.

Het zou goed uitkomen dat het doel reeds na een eerste splitsing bereikt is, zoals in Figuur 6.2. Daar leidt de eerste splitsing tot twee **homogene** verzamelingen, d.w.z. verzamelingen waarvan de elementen alle tot dezelfde categorie behoren.

Een homogene verzameling vertoont geen spreiding.

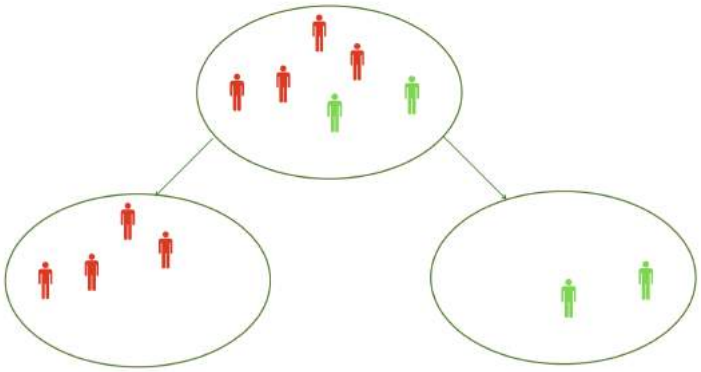
Met een beslissingsboom wil je via de vertakkingen tot homogene verzamelingen komen.

Als een verzameling elementen bevat die niet alle van dezelfde categorie zijn, dan is die verzameling **heterogeen**. De elementen zijn dan gespreid over de categorieën.

Een heterogene verzameling vertoont spreiding.

Een verzameling met tien elementen waarvan er slechts één tot een andere categorie behoort, vertoont minder spreiding dan een waarvan er drie tot een andere categorie behoren.

En wanneer vertoont zo'n verzameling het meest spreiding? Als er van elke categorie evenveel elementen aanwezig zijn.

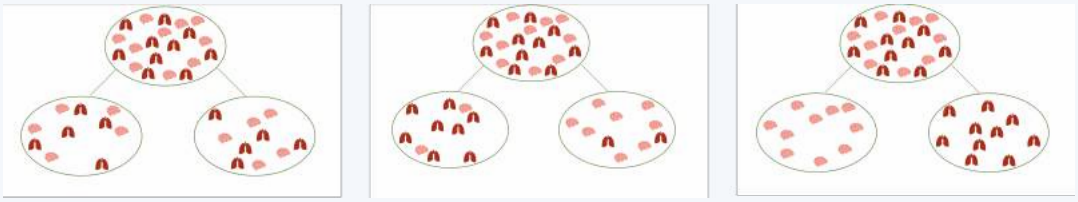


Figuur 6.2: Patiënten verdelen over twee categorieën: de rode zijn de ernstig zieke, de groene de niet ernstig zieke.

Oefening 6.1.1: spreiding

Een huisarts bekijkt welke van zijn patiënten leiden aan een hersen- of longaandoening. A.d.h.v. een bepaalde lichaamsparameter (bv. hartslag, alertheid) wil hij deze patiënten van elkaar onderscheiden. De dokter onderzoekt welke lichaamsparameter de beste scheiding, m.a.w. de minste spreiding over de verzamelingen, oplevert.

Welke van de drie mogelijkheden die voorgesteld zijn in de figuur is de beste? Waarom?



Wat is de beste keuze?

6.2 *Liefst met de computer*

Eenvoudige beslissingsbomen kunnen manueel worden opgemaakt, bv. het stappenplan dat scholen dienen te volgen in geval van een besmetting met COVID-19 op school (zie Figuur 4.3).

Maar als er veel factoren zijn waarmee er rekening gehouden moet worden, dan vergt dat heel wat tijd. Het is dan handig om een computer in te zetten.

Om door een computer een beslissingsboom te laten opmaken, is er een **computerprogramma** nodig. Bijvoorbeeld voor een binaire beslissingsboom moet dat programma de wortel bepalen, de juiste knopen kiezen en de geschikte ja-neevragen vastleggen.

De computer zal er ook moeten in slagen om zo homogene verzamelingen te bekomen. Hij zal er bij elke splitsing naar streven om zo weinig mogelijk spreiding over de categorieën te hebben.

Er is nood aan een methode, een algoritme dat omgezet kan worden naar een computerprogramma.
In dit hoofdstuk worden de concepten behandeld die aan de basis van het algoritme liggen.
Het algoritme is een gulzig algoritme. Het zal voor het bepalen van de splitsing geen rekening houden met de stappen die erna nog komen. In hoofdstuk computationeel denken lees je meer over gretige algoritmes.

Er rijzen hier enkele vragen:

- Hoe weet een computer wat 'de mate van spreiding over de verschillende categorieën' is?

- Hoe maakt een computer ja-neevragen?

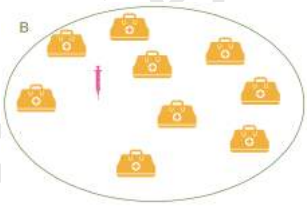
Om deze vragen te beantwoorden, kan je alvast hiermee rekening houden: computers werken met getallen.

Om ondubbelzinnig te kunnen spreken over de spreiding die aanwezig is in een verzameling, geef je de mate van spreiding in een verzameling weer door een getal.

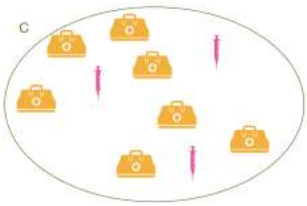
6.3 Een spreidingsparameter

Stel bijvoorbeeld dat je beschikt over **twee categorieën van elementen**: 'dokterstas' en 'injectiespuit'. Uit deze categorieën kies je willekeurig tien elementen. Het is mogelijk dat deze gekozen elementen alle tot dezelfde categorie behoren, maar het is ook mogelijk dat sommige tot de ene categorie behoren en sommige tot de andere categorie.

Figuur 6.3: Homogeen. Geen spreiding.



Figuur 6.4: Een vreemde eend in de bijt. Klein beetje spreiding.

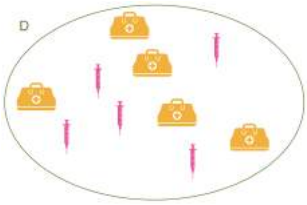


Figuur 6.5: Heterogeen. Nog meer spreiding.

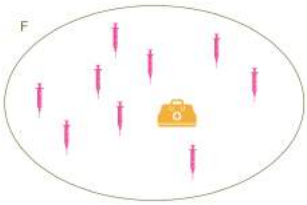
In Figuur 6.3 vertoont de verzameling geen spreiding, want er zitten enkel elementen van de klasse 'dokterstas' in. De spreidingsparameter $s = 0$. Er is geen spreiding van de variabelen.

In Figuur 6.4 vertoont de verzameling wel spreiding: er zitten elementen in van de klasse 'dokterstas', maar er is ook een vreemde eend in de bijt, namelijk een element van de klasse 'injectiespuit'. Er is een lichte spreiding: er is in de verzameling bijna geen spreiding van de variabelen. De spreidingsparameter is hier klein, maar niet nul.

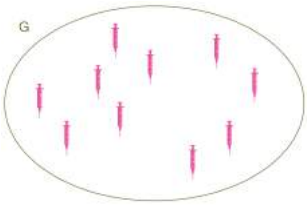
Ook in Figuur 6.5 vertoont de verzameling spreiding. Er zitten in de verzameling vooral elementen van de klasse 'dokterstas', en enkele elementen van de klasse 'injectiespuit'. Er is een grotere spreiding van de variabelen dan in de verzameling van Figuur 6.4. De spreidingsparameter is dus nog wat groter.



Figuur 6.6: Uit elke klasse evenveel elementen. Grootst mogelijke spreiding.



Figuur 6.7: Een vreemde eend in de bijt. Een klein beetje spreiding.



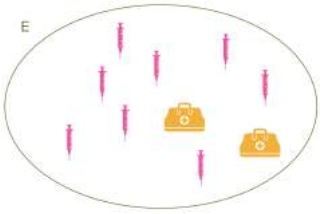
Figuur 6.8: Homogeen. Geen spreiding.

De verzameling in Figuur 6.6 vertoont ook spreiding. Er zitten in de verzameling evenveel elementen van de klasse ‘dokterstas’ als van de klasse ‘injectiespuit’. De spreiding is hier maximaal. Hier heb je te maken met de grootst mogelijke spreiding van de variabelen, namelijk $50\ \% - 50\ \%$. De spreidingsparameter is hier $50\ \%$, dus $s = 0{,}5$. Dat is de grootste spreidingsparameter die kan voorkomen.

In Figuur 6.7 is de spreiding even groot als in Figuur 6.4. Alle elementen behoren tot dezelfde klasse, behalve één. De spreidingsparameter is dezelfde voor deze verzamelingen.

In Figuur 6.8 is de verzameling homogeen. Er is dus geen spreiding, net zoals in Figuur 6.3. Alle elementen behoren tot dezelfde klasse. Net zoals bij Figuur 6.3 is $s = 0$ hier.

Hoe zit het met de mate van spreiding in Figuur 6.9? Er is minder spreiding dan in Figuur 6.6, want de punten van de klasse ‘injectiespuit’ zijn duidelijk in de meerderheid. Er is zelfs minder spreiding dan in Figuur 6.5. De spreidingsparameter is dus kleiner dan die van Figuur 6.5, maar groter dan die van Figuur 6.7, waar er slechts één vreemde eend in de bijt is.



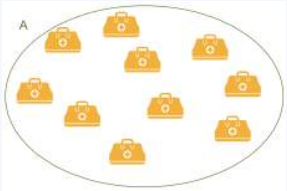
Figuur 6.9: Heterogeen. Er is spreiding.

Versie 1.0

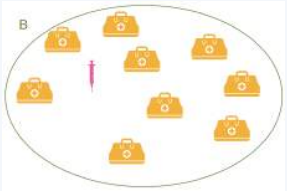
Oefening 6.3.1: spreiding over klassen

In de verzamelingen zitten elementen die behoren tot de klasse ‘dokterstas’ of ‘injectiespuit’. Elke verzameling vertoont een andere spreiding van de elementen over deze twee klassen.
s is een spreidingsparameter die een maat is voor de spreiding.

- Welke verzamelingen horen er op de lege plaatsen?

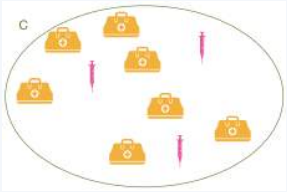


$s = 0$
$p =$
$q =$



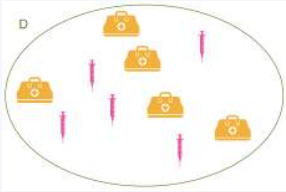
$s =$
$p =$
$q =$

$s =$
$p =$
$q =$



$s =$
$p =$
$q =$

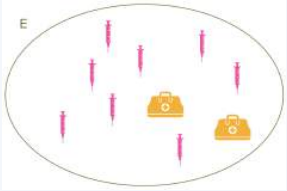
$s =$
$p =$
$q =$



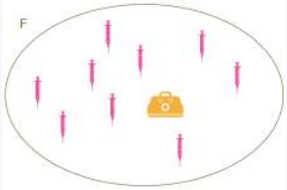
$s = 0{,}5$
$p =$
$q =$

$s =$
$p =$
$q =$

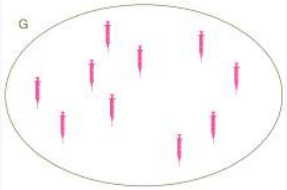
$s =$
$p =$
$q =$



$s =$
$p =$
$q =$



$s =$
$p =$
$q =$



$s =$
$p =$
$q =$

Vervolg oefening: spreiding over klassen

- Benoem de nieuwe verzamelingen.
- Stel dat p gelijk is aan het **aandeel van de elementen dat tot de klasse 'dokterstas' behoort**. Voor de eerste verzameling is dat $100\ \%$, dus daar geldt dat $p = 1$.
  q is het aandeel van de elementen dat tot de klasse 'injectiespuit' behoort. Voor de eerste verzameling is $q = 0$.
  Vul nu de waarden van p en q in op de vorige bladzijde.
- Bereken voor elke verzameling de som $p + q$. Wat merk je op?
- Vul de waarde van s in, daar waar je ze al kent.
- Als je de rij van verzamelingen bekijkt, wat merk je dan op:
  - Neemt p toe of af?
  - Neemt q toe of af?
  - Neemt s toe of af?
  - Wat is de hoogste en wat is de kleinste waarde die s aanneemt?

Hoe ziet de grafiek van s in functie van p er mogelijk uit?
Plaats p, het aandeel elementen dat tot de klasse 'dokterstas' behoort, op de x-as; op de y-as plaats je de spreidingsparameter s.
Stel de verzamelingen voor met een punt, bijvoorbeeld de verzameling A komt overeen met het punt met coördinaat $(0{,}0)$. De verzamelingen waarvoor je te weinig gegeven hebt, geef je een mogelijke plaats op de grafiek.
Schets een mogelijke grafiek.

Figuur 6.10: s in functie van p.

6.4 Een formule voor de spreidingsparameter

Figuur 6.10 doet voor een probleem met twee klassen het volgende besluiten: wil je zo'n spreidingsparameter s uitzetten in functie van p, dan moet je dus een functie gebruiken die vanuit 0 naar een maximum $\frac{1}{2}$ stijgt en dan terug daalt naar 0.

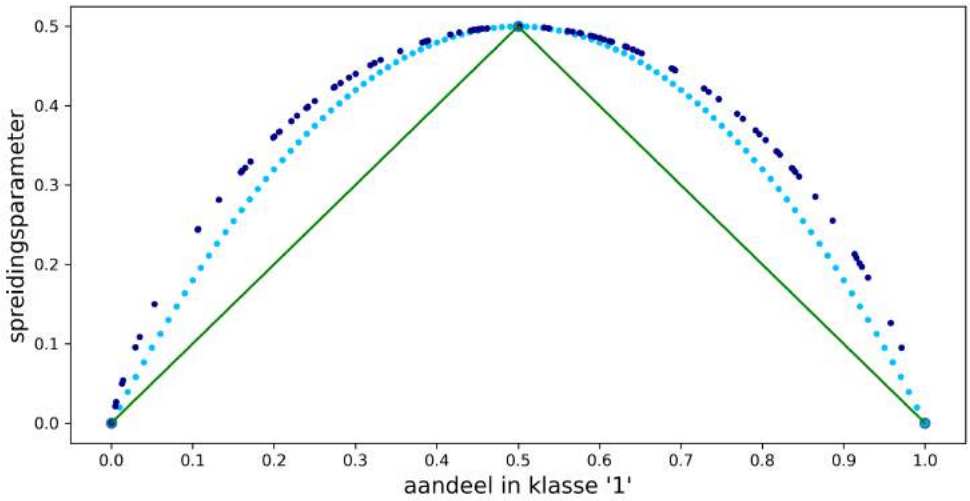
De grafiek van zo'n functie gaat door $(0, 0)$ en $(1, 0)$ en heeft een top $(\frac{1}{2}, \frac{1}{2})$. Tussen de oorsprong en de top stijgt de functie, na de top daalt de functie tot in $(1, 0)$. Het verloop van een geschikte functie wordt dus gegeven door Figuur 6.11.

Merk op dat er een van de klassen moet worden uitgekozen die dan overeenkomt met p. Algemeen kan je naar deze klasse verwijzen met klasse '1', de andere klasse is dan klasse '0'.

p	0		$\frac{1}{2}$		1
f(p)	0 min	↗	$\frac{1}{2}$ max	↘	0 min

Figuur 6.11: Waardetabel van de functie f, met $f(p) = s$.

Er zijn echtere meerdere functies mogelijk. Figuur 6.12 toont enkele voorbeelden, waarvan de lichtblauwe kromme een bergparabool is.



Figuur 6.12: Mogelijke spreidingsparameters.

Er zijn dus verschillende manieren om tot een spreidingsparameter te komen; deze cursus beperkt zich tot de '**gini-index**', die overeenkomt met deze bergparabool (zie Figuren 6.12 en 6.13).

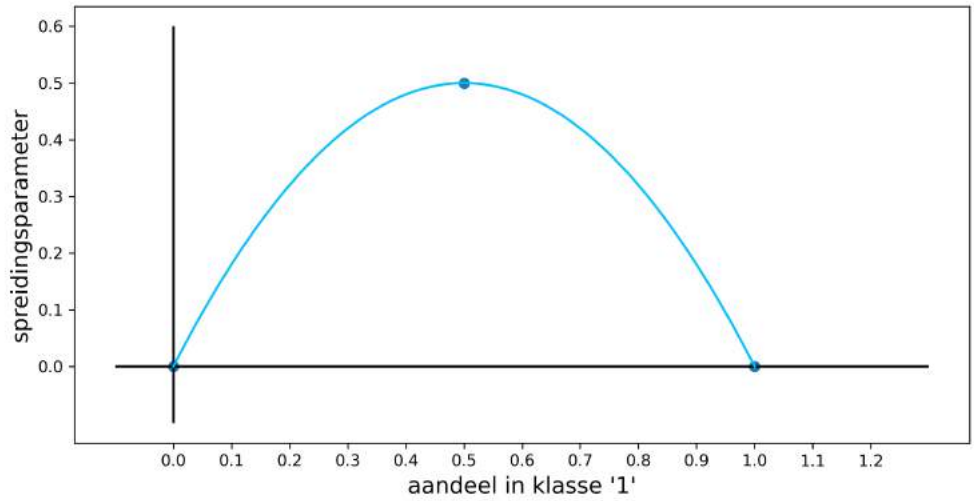
Gini-index

De **gini-index** is een van de mogelijke spreidingsparameters. De grafische voorstelling van de gini-index wordt gegeven door een bergparabool met top $(0{,}5; 0{,}5)$ en nulwaarden 0 en 1.

Nog eens samengevat wat er te zien is op deze bergparabool:

Vanaf nu wordt de spreidingsparameter gini-index genoemd.

- Alles in klasse '0' en niets in klasse '1': $q = 100$ % en $p = 0$ %, dus $p = 0$, $q = 1$ en $s = 0$.
- Meer in klasse '0' dan in klasse '1': $50\ \% < q < 100\ \%$ en $p < 50$ %, dus $0 < p < 1/2$, $1/2 < q < 1$, en $0 < s < 1/2$.
- De helft in klasse '1' en de helft in klasse '0': $p = 50$ % en $q = 50$ %, dus $p = \frac{1}{2}$ en $q = \frac{1}{2}$. Hier bereikt de grafiek een maximum: $s = \frac{1}{2}$.



Figuur 6.13: Bergparabool die overeenkomt met de gini-index.

- Meer in klasse '1' dan in klasse '0': $50\ \% < p < 100\ \%$ en $q < 50\ \%$, dus $\frac{1}{2} < p < 1$, $0 < q < \frac{1}{2}$, en $0 < s < 1/2$.
- Alles in klasse '0' en niets in klasse '1': $p = 100\ \%$ en $q = 0\ \%$, dus $p = 1$, $q = 0$, en $s = 0$.

Merk op dat de top $(\frac{1}{2}, \frac{1}{2})$ op de symmetrieas van de parabool ligt.

Opdracht 6.4.1: opstellen van het functievoorschrift dat de gini-index als maat voor de spreiding oplevert

Bepaal de vergelijking van een bergparabool door $(0,0)$ en $(1,0)$ die een maximum $\frac{1}{2}$ bereikt in $\frac{1}{2}$.

Formule gini-index

De **gini-index** is een spreidingsparameter die weergeeft in welke mate de elementen van een verzameling gespreid zijn over meerdere categorieën.

In het geval van twee categorieën wordt de gini-index s berekend met de volgende formule:

$$s = 2\,p\,q.$$

Hierbij is een van de klassen uitgekozen waarvan het percentage elementen uit die klasse dan overeenkomt met p. Het percentage elementen dat tot de andere klasse behoort, is q. $p + q = 1$.

Oefening 6.4.2: s berekenen

Je kan nu voor al de verzamelingen in oefening 6.3.1 de waarde van s berekenen en invullen.

Oefening 6.4.3: spreiding van 6 en 7 elementen over twee klassen

Maak oefening 6.3.1 opnieuw, maar nu met 6 elementen i.p.v. 10, en ook eens met 7 elementen.

6.5 De totale gini-index

De gini-index geeft de mate van spreiding binnen een verzameling. Bij een vertakking in een beslissingsboom verdeel je de elementen van een verzameling over twee verzamelingen. Elk van deze verzamelingen vertoont een eigen spreiding. Hoe breng je beide spreidingen in rekening? M.a.w.: wat is de maat van spreiding na de splitsing?

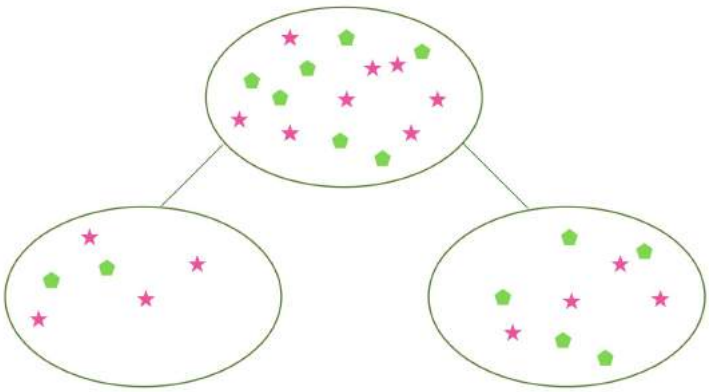
Je maakt een **gewogen som** van beide gini-indices (Sprague, 2021). Het gebied met het grootste aantal elementen weegt zo meer door. Deze gewogen som is de totale gini-index.

Voorbeeld:

In Figuur 6.14 zie je 15 punten: 8 sterren en 7 vijfhoeken. Om de spreidingsparameter te bepalen, bereken je de gini-index. Kies bv. p als het aandeel sterren, dan is $p = \frac{8}{15}$ voor de bovenste verzameling.

De gini-index $s = 2 \cdot \frac{8}{15} \cdot \frac{7}{15}$, dus $s = \frac{112}{225}$, dus $s = 0{,}497...$

Na de splitsing bekom je een gebied met gini-index $0{,}444...$ ($p = \frac{4}{6}; s = 2 \cdot \frac{4}{6} \cdot \frac{2}{6}$) en een met gini-index $0{,}493...$ ($p = \frac{4}{9}; s = 2 \cdot \frac{4}{9} \cdot \frac{5}{9}$).



Figuur 6.14: Een mogelijke splitsing.

Maar wat is de totale gini-index na de splitsing? Hoe breng je beide gebieden in rekening?

Je maakt een **gewogen som** van beide gini-indices (Sprague, 2021). Het gebied met de meeste elementen weegt zo meer door.

$$\text{gini}_{\text{totaal}} = a_1 \cdot s_1 + a_2 \cdot s_2$$

Hierbij is a_1 het aandeel elementen in de eerste verzameling en a_2 het aandeel elementen in de tweede verzameling.

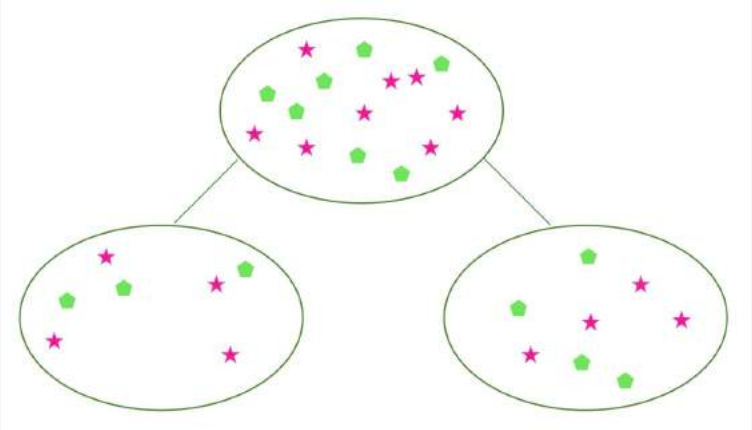
Voor dit voorbeeld betekent dit dat de totale gini-index na de splitsing gegeven wordt door:

$\text{gini}_{\text{totaal}} = \frac{6}{15} \cdot 0{,}444... + \frac{9}{15} \cdot 0{,}493...$, dus $\text{gini}_{\text{totaal}} = 0{,}474...$,

wat een kleinere waarde is dan voor de splitsing, dus wijst op minder spreiding over beide klassen.

Oefening 6.5.1: de totale gini-index - betere of slechtere splitsing?

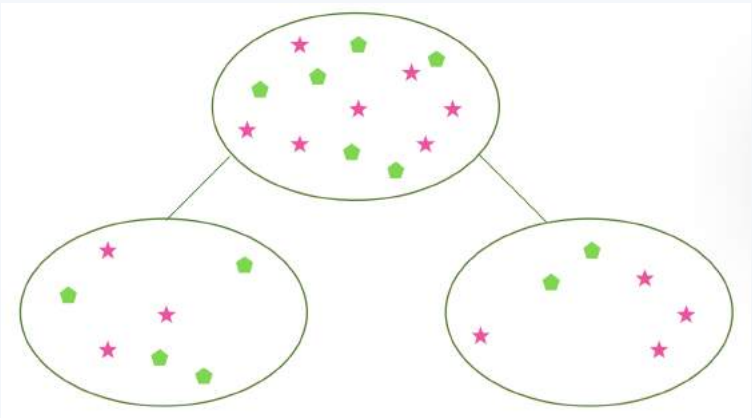
De verzameling uit het voorbeeld is nu anders gesplitst, zoals je ziet op de figuur.
Bereken $gini_{totaal}$ voor deze nieuwe splitsing.
Is dit een betere of slechtere splitsing dan die in het voorbeeld?



Nog een mogelijke splitsing.

Oefening 6.5.2: de totale gini-index

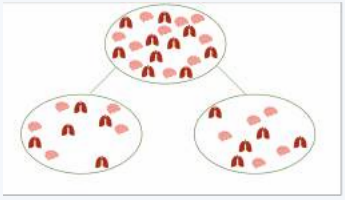
De gegeven verzameling bevat 13 elementen die behoren tot de klasse 'ster' of 'vijfhoek'.
Bereken $gini_{totaal}$ voor en na de gegeven splitsing.

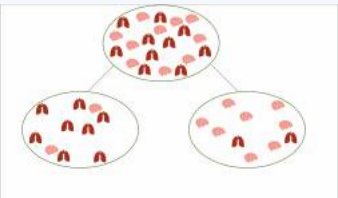


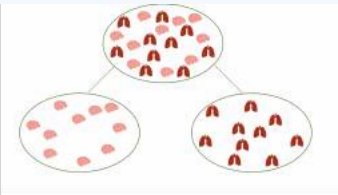
Een mogelijke opdeling van de 13 elementen.

Oefening 6.5.3: beste keuze aantonen

Herneem nu oefening 6.1.1 en bereken voor elke mogelijkheid de totale gini-index om aan te tonen dat je de beste keuze maakte.

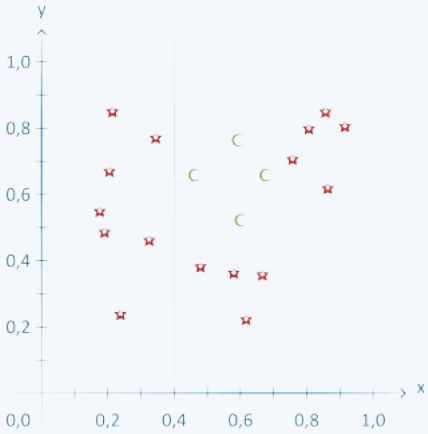




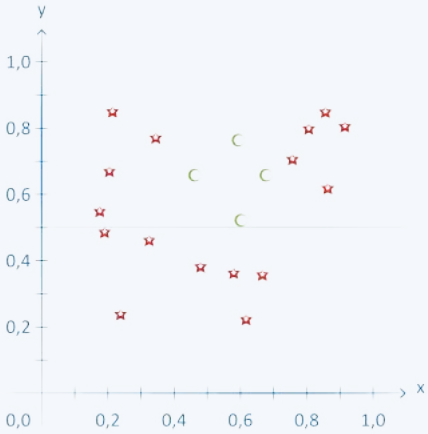


Wat is de beste keuze?

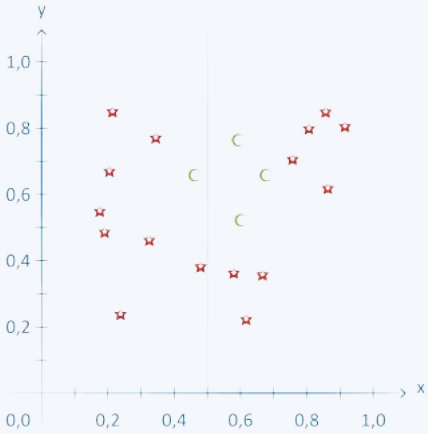
Oefening 6.5.4: splitsingen vergelijken



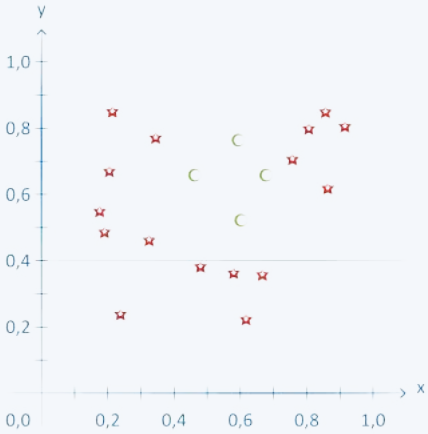
Figuur 6.15: eerste mogelijke splitsing



Figuur 6.16: tweede mogelijke splitsing



Figuur 6.17: derde mogelijke splitsing



Figuur 6.18: vierde mogelijke splitsing

Neem de scheiding uit Figuur 6.15 en toon aan dat de scheidingen uit Figuur 6.17 en 6.18 een slechtere keuze zijn dan die uit Figuur 6.15 en dat die uit Figuur 6.16 een even goede is als die uit Figuur 6.15.

6.6 Met de computer

Er kan al antwoord gegeven worden op de eerste vraag uit paragraaf 6.2:

- Hoe weet een computer wat 'de mate van spreiding over de verschillende klassen' is?
- Hoe maakt een computer ja-neevragen?

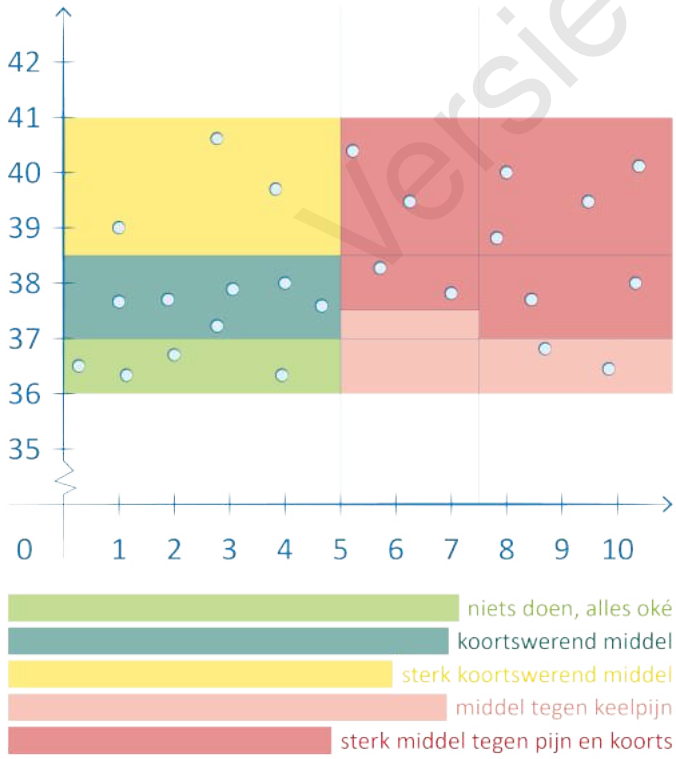
De (totale) gini-index geeft 'de mate van spreiding over de verschillende klassen' weer. Er is bovendien een formule beschikbaar om die gini-index te berekenen. De computer kan dus m.b.v. deze formule de totale gini-index berekenen voor elke mogelijke splitsing.

De ja-neevragen komen bij numerieke variabelen overeen met een voorwaarde die gebruikmaakt van een vergelijkingsoperator, bv. $>$ of $\leq$, zoals in Figuur 4.1. Eventuele kwalitatieve waarden zet je om naar een numerieke waarde zodat de computer ermee kan werken.

Een variabele kan bijvoorbeeld de waarden 'ziek' en 'gezond' aannemen. Dan laat je 'ziek' overeenkomen met de klasse '1' en 'gezond' met de klasse '0'. Dat betekent dat als de voorwaarde 'waarde < 0.5' waar is, dat leidt tot de klasse '0' (dus 'gezond'), maar als de voorwaarde niet waar is, dus als 'waarde $\geq$ 0.5' de uitkomst de klasse '1' (dus 'ziek') is.

6.7 Scheiding in klassen d.m.v. horizontale en verticale rechten

Versie 1.0



Figuur 6.19: Medicatie aangepast aan de ernst van de keelpijn (volgens een schaal van 0 t.e.m. 10; op de x-as) en de hoogte van de koorts (in graden Celsius; op de y-as). Voorbeeld van een scheiding d.m.v. rechten loodrecht op elkaar en loodrecht op de assen.

Fictief voorbeeld: Als een patiënt keelpijn en/of koorts heeft, neemt die persoon medicatie aangepast aan de hoogte van de koorts en hoe erg de keelpijn is: een koortswerend middel, een sterk koortswerend

middel, een middel tegen de keelpijn, of een sterk middel dat zowel tegen de koorts als de keelpijn helpt. Het spreekt voor zich dat iemand zonder koorts of keelpijn geen medicatie hoeft te nemen. Dit wordt grafisch voorgesteld in Figuur 6.19.

Hoe erg de keelpijn wordt ervaren, is weergeven door een getal op een schaal van 0 t.e.m. 10 (0 komt overeen met 'geen keelpijn' en 10 met 'ondraaglijke keelpijn'). De lichaamstemperatuur wordt weergegeven in graden Celsius.

De figuur is verdeeld in vijf gebieden die overeenkomen met de vijf verschillende categorieën van het nemen van medicatie.

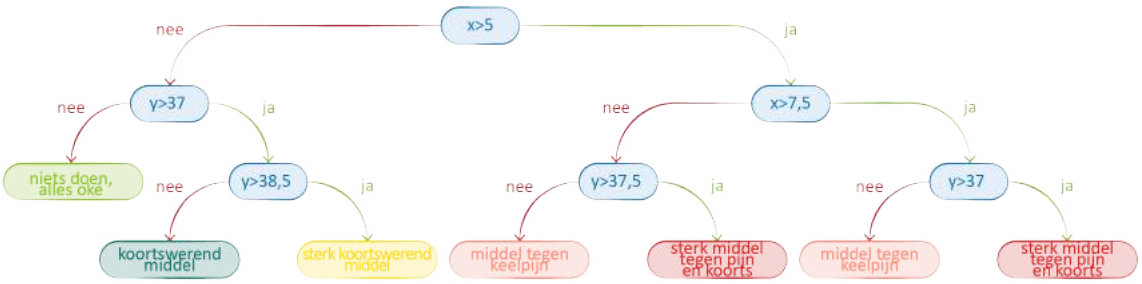
Bijvoorbeeld een patiënt met 38 graden koorts en keelpijn die aanvoelt als erge pijn (8 op een schaal van 0 t.e.m. 10), behoort tot het donkerrode gebied en zal een sterk middel tegen pijn én koorts moeten nemen. Zo'n patiënt kan op de figuur worden weergegeven door een punt met coördinaat $(8, 38)$.

Op de figuur zijn de klassen van elkaar gescheiden door verticale en horizontale lijnen.

Samengevat stelt Figuur 6.19 de scheiding van punten in vijf klassen voor, d.m.v. rechten die loodrecht staan op de x- en y-as én op elkaar. Meer specifiek gaat het om een classificatieprobleem met twee variabelen: 'de mate van keelpijn' en 'de hoogte van de koorts', respectievelijk te vinden op de x-as en op de y-as. Van een patiënt kan je op basis van de waarden van deze twee variabelen bepalen in welke categorie van medicatie die patiënt valt.

De voorstelling in Figuur 6.19 kan worden omgezet naar een beslissingsboom, zoals te zien is in Figuur 6.20. De opbouw van de beslissingsboom wordt bepaald door de plaats van de horizontale en verticale scheidingslijnen.

Versie 1.0



Figuur 6.20: Beslissingsboom bij het voorbeeld van de medicatie.

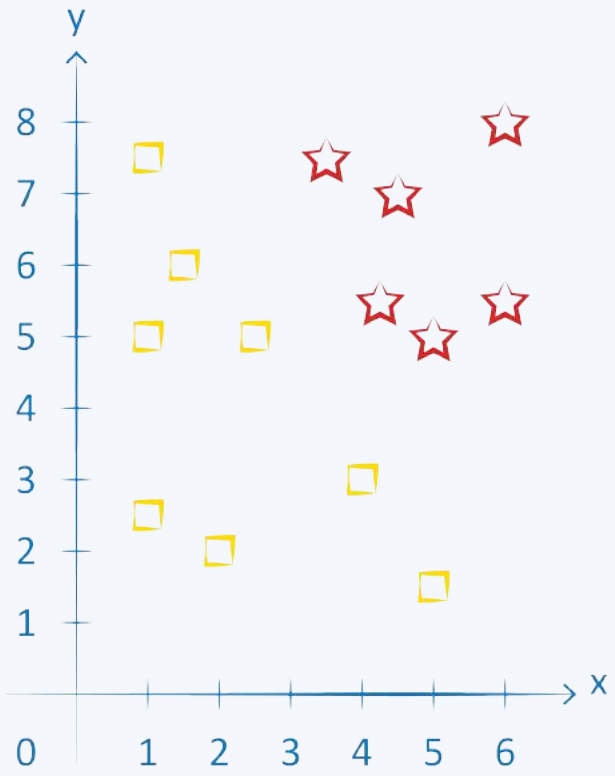
Werkwijze 6.7.1: van puntenwolk naar beslissingsboom

Deze kader doet uit de doeken hoe je twee klassen in een puntenwolk van elkaar scheidt m.b.v. horizontale en verticale lijnen en hoe je vervolgens de beslissingsboom opstelt.

Bekijk het spreidingsdiagram hieronder.
Het spreidingsdiagram bevat datapunten uit twee klassen, vierkanten en sterren. Om deze te scheiden plaats je achtereenvolgens:

- een verticale rechte die de punten verdeelt over twee gebieden en waarvan een van de gebieden enkel vierkanten bevat (noteer hoeveel vierkanten dat zijn);
- een horizontale halfrechte die het andere gebied nog eens in twee verdeelt en waarbij in elk gebied slechts één soort punten te vinden is.

Teken deze lijnen erbij op de figuur.



Puntenwolk met datapunten uit twee klassen.

Het spreidingsdiagram is nu verdeeld in drie gebieden. In elk gebied is er slechts één klasse te vinden; er is binnen een gebied **geen spreiding van de punten over de klassen**.

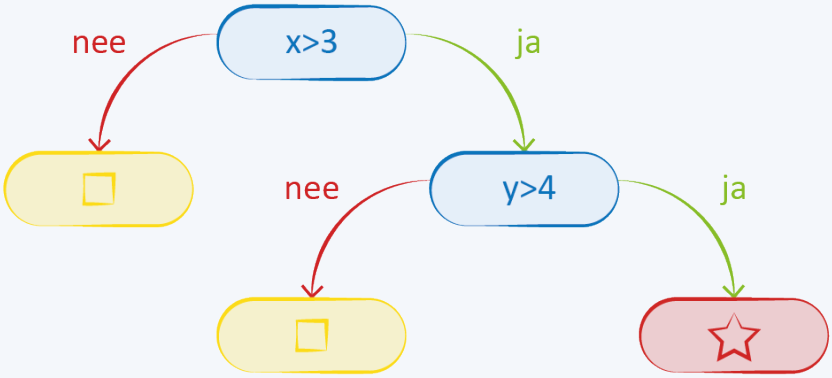
In principe had je ook kunnen starten met een horizontale lijn, i.p.v. een verticale, maar dan zitten er in het gebied met enkel vierkanten slechts 4 vierkanten i.p.v. 6. Starten met een horizontale scheiding zou overeenkomen met een grotere totale gini-index voor die stap in het scheidingsproces, en zoals gezegd kies je steeds voor een zo klein mogelijke totale gini-index.

Vervolg werkwijze: van puntenwolk naar beslissingsboom

Nu vertaal je de figuur naar een beslissingsboom:

- De wortel van de beslissingsboom komt overeen met de eerste lijn die je tekende. Die lijn verdeelde de puntenwolk in twee delen: een deel rechts van de scheidingslijn (waarvoor $x > 3$) en een gebied links van de scheidingslijn (daar is $x < 3$). Je plaatst in de wortel een van deze twee voorwaarden, bv. $x > 3$.
- Dan plaats je twee takken. De situatie komt overeen met een ja-neevraag: is de voorwaarde voldaan?
  Indien 'neen', dan betreft dat het linkergebied dat enkel vierkantjes bevat. Daar moet niets meer ondernomen worden. De beslissingsboom heeft daar een blad.
  Indien 'ja', dan betreft dat het rechtergebied waar wel nog spreiding van de inputs is (dat gebied bevat zowel vierkantjes als sterretjes). Daar moet nog een tweede scheiding gebeuren.
- Deze tweede scheiding vertaalt zich in een knoop die overeenkomt met de tweede lijn die je tekende. Die halfrechte verdeelde het rechtergebied in twee delen: een deel boven de scheidingslijn (waarvoor $y > 4$) en een gebied onder de scheidingslijn (daar is $y < 4$). In de knoop plaats je een van deze twee voorwaarden, bv. $y > 4$.
- Je plaatst opnieuw twee takken. Opnieuw gaat het over een ja-neevraag: is de voorwaarde voldaan?
  Indien 'neen', dan betreft het het onderste gebied dat enkel vierkantjes bevat. Daar moet niets meer ondernomen worden. De beslissingsboom heeft daar een blad.
  Indien 'ja', dan betreft het het bovenste deel dat enkel sterretjes bevat. Ook daar moet niets meer ondernomen worden en heeft de beslissingsboom een blad.

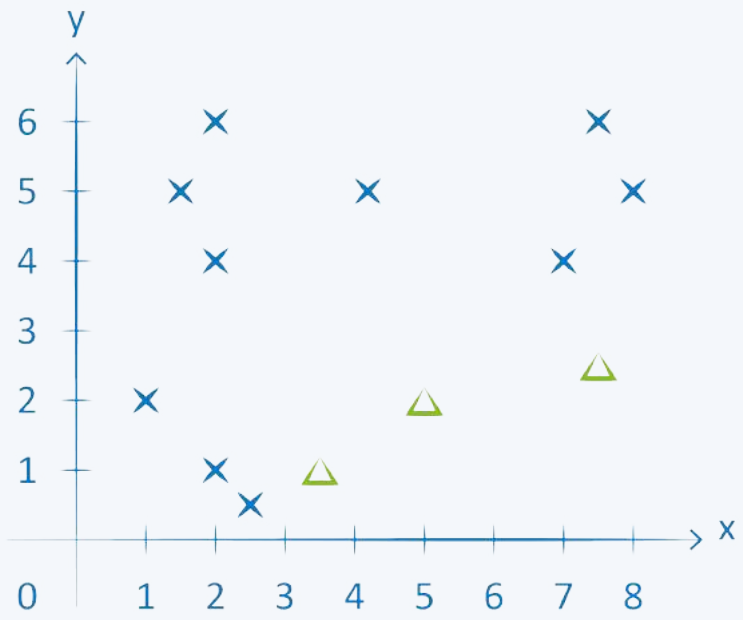
Je bekomt deze beslissingsboom:



Beslissingsboom.

Oefening 6.7.2

Het gegeven spreidingsdiagram bevat punten uit twee klassen, driehoeken en kruisjes.

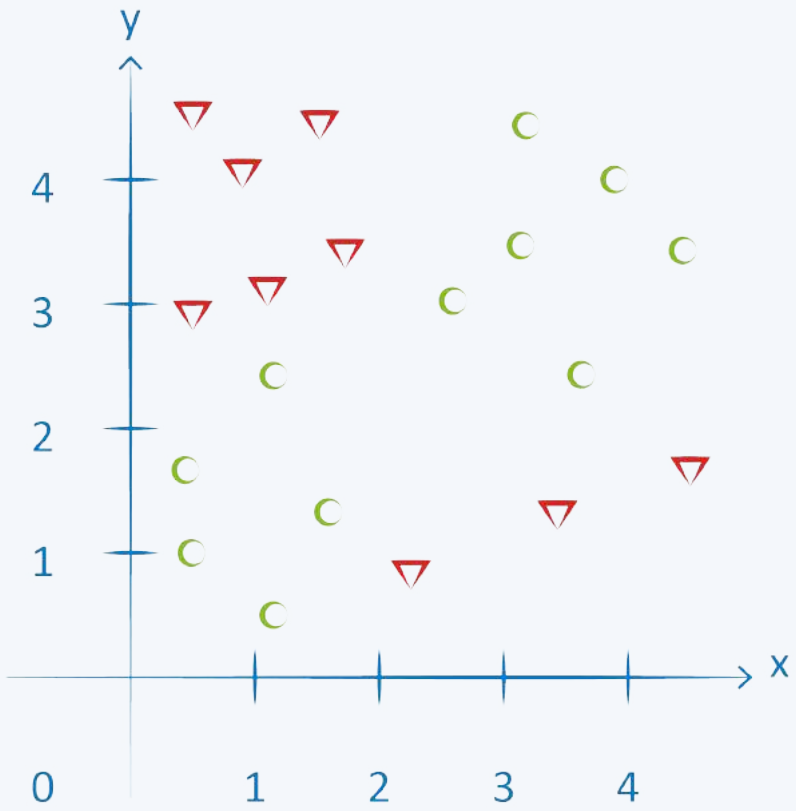


Puntenwolk met twee klassen van punten.

- Verdeel de puntenwolk in gebieden waarbinnen geen spreiding van inputs over de twee klassen meer is, d.w.z. dat de punten in een bepaald gebied tot dezelfde klasse behoren.
- Vervolgens bouw je de beslissingsboom op die overeenkomt met de gekozen scheidingen.

Oefening 6.7.3

Het gegeven spreidingsdiagram bevat punten uit twee klassen, driehoekjes en bolletjes.



Puntenwolk met twee klassen van punten.

- Verdeel de puntenwolk in gebieden waarbinnen geen spreiding van inputs is, d.w.z. dat de datapunten in een bepaald gebied tot dezelfde klasse behoren.
- Vervolgens bouw je de beslissingsboom op die overeenkomt met de gekozen scheidingen.

In wat volgt, wordt er gewerkt met 2 klassen.

6.8 Scheiding bepalen

Nu ken je een methode die je kan toepassen om een puntenwolk met datapunten uit twee klassen op te splitsen. A.d.h.v. de gini-index kunnen de beste horizontale en verticale scheidingslijnen bepaald worden. Je moet enkel berekenen in welk geval, dus bij welke mogelijke splitsing, de totale gini-index het kleinst is. En deze methode kan ook door een computer worden uitgevoerd.

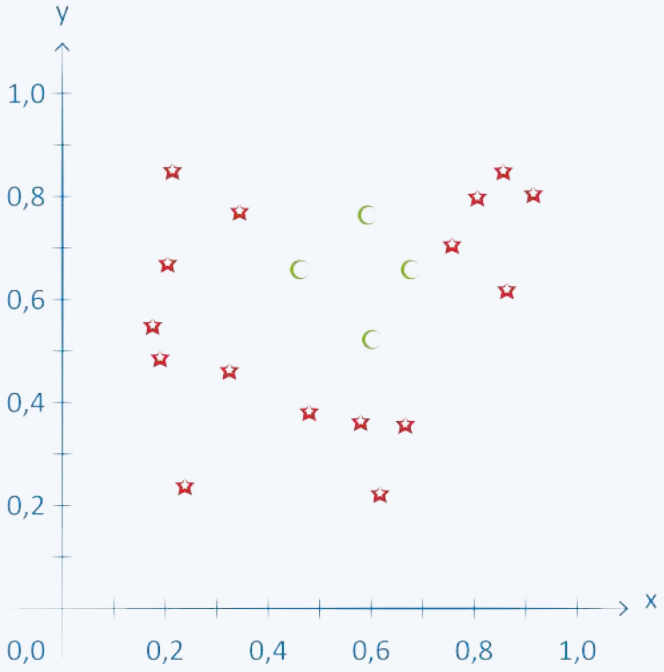
Uit paragraaf 6.7 weet je dat zo'n splitsing a.d.h.v. horizontale en verticale lijnen gemakkelijk omgezet kan worden naar een beslissingsboom.

In het volgende hoofdstuk zet je de computer aan het werk.

Oefening 6.8.1: beste scheiding narekenen

- Reken na dat starten met een horizontale scheiding in werkwijze 6.7.1 een grotere totale gini-index oplevert voor die eerste stap.
- Controleer of de scheidingen die je zelf uitvoerde in de oefeningen 6.7.2 en 6.7.3, de juiste waren. M.a.w. controleer a.d.h.v. een berekening van de gini-index of je steeds de juiste keuze maakte tussen horizontaal en verticaal.

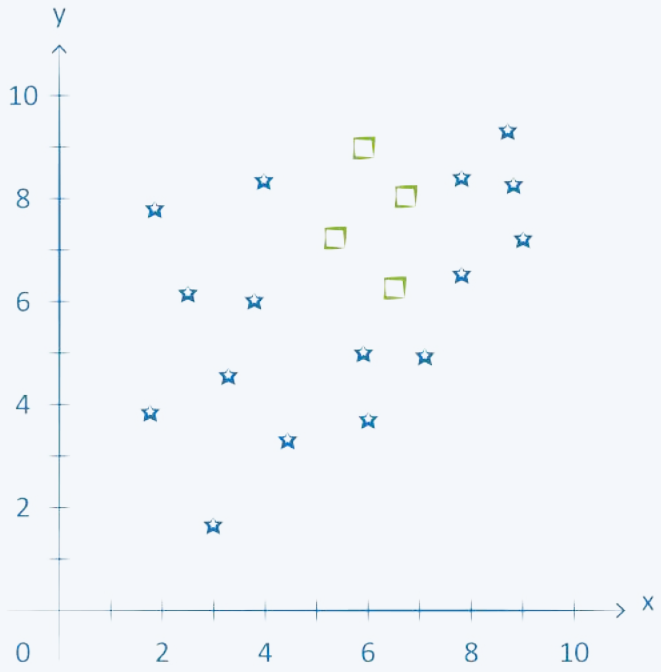
Oefening 6.8.2: van spreidingsdiagram naar beslissingsboom



Figuur 6.21: Splits op a.d.h.v. de gini-index. Gebruik met toestemming (Sprague, 2021).

- Bekijk de puntenwolk in Figuur 6.21.
  Verdeel de gegeven puntenwolk d.m.v. horizontale en verticale lijnen, rekening houdend met de best mogelijke keuze gestaafd met de gini-index.
- Zet de verdeling om naar een beslissingsboom.
- Classificeer de volgende punten a.d.h.v. deze beslissingsboom: $(0{,}4; 1)$, $(0{,}6; 1)$ en $(0{,}6; 0)$.

Oefening 6.8.3: classificatie a.d.h.v. een beslissingsboom



Splits op a.d.h.v. de gini-index. Gebruik met toestemming (Sprague, 2021).

- Bekijk de puntenwolk.
  Verdeel de gegeven puntenwolk d.m.v. horizontale en verticale lijnen, rekening houdend met de best mogelijke keuze. Staaf je keuze voor een verticale of horizontale scheidingslijn bij elke stap met de gini-index.
- Zet de verdeling om naar een beslissingsboom.
- Classificeer de volgende punten a.d.h.v. deze beslissingsboom: $(8;3)$, $(7;7)$ en $(8{,}2;11)$.

6.9 *Stappenplan om beslissingsboom te construeren*

Uit het voorgaande kan een stappenplan worden afgeleid om een beslissingsboom te kunnen opstellen. Het aanmaken van een beslissingsboom is een iteratief proces: om de wortel en de knopen te bepalen wordt telkens dezelfde techniek toegepast.

- Bekijk welke wortel de beste splitsing oplevert. Maak takken voor de mogelijke uitkomsten van de ja-neevraag.
- Bekijk voor elke tak welke knoop het meest geschikt is voor de volgende splitsing. Maak takken voor de mogelijke uitkomsten van de ja-neevragen.
- Herhaal de vorige stap telkens opnieuw voor de nieuw aangemaakte takken.
- Stop met de herhaling als de bekomen verzamelingen homogeen zijn, dus enkel elementen uit een en dezelfde klasse bevatten.

Zo'n stappenplan wordt ook een **algoritme** genoemd. Dit algoritme is een verdeel-en-heersalgoritme. Het is ook een gulzig algoritme.

Zie paragrafen 6.13.1 en 6.13.2 voor meer uitleg over gulzige en verdeel-en-heersalgoritmes.

6.10 Toepassen van het algoritme

Voorbeeld:
Je kan enkele parameters in rekening brengen om te proberen te voorspellen of een patiënt risico loopt op een hartaanval. Van gekende patiënten zijn bepaalde parameters terug te vinden in het patiëntendossier.

De volgende tabel toont zulke parameters voor zes patiënten waarvan geweten is of ze al dan niet een hartaanval kregen (Shoemaker et al., 2001).

Aan de hand van deze (te beperkte) dataset wordt getoond hoe je het verdeel-en-heersalgoritme in de praktijk toepast om een beslissingsboom aan te maken. Deze beslissingsboom kan dan gebruikt worden om voor nieuwe patiënten na te gaan of ze risico lopen op het krijgen van een hartaanval.

Patiëntnummer	Pijn in de borststreek	Man	Rookt	Voldoende lichaamsbeweging	Hartaanval
1	ja	ja	nee	ja	**ja**
2	ja	ja	ja	nee	**ja**
3	nee	nee	ja	nee	**ja**
4	nee	ja	nee	ja	**nee**
5	ja	nee	ja	ja	**ja**
6	nee	ja	ja	ja	**nee**

Tabel 6.1: Parameters van belang voor risico op hartaanval.

Versie 1.0

In deze tabel wordt een patiënt aangeduid met een 'patiëntnummer'. De parameters 'pijn in de borststreek', 'man', 'rookt' en 'voldoende lichaamsbeweging' in de tabel zijn de parameters die in aanmerking genomen worden om het risico op een hartaanval te bekijken.

De patiënt kan behoren tot de categorieën 'hartaanval' of 'geen hartaanval', dus tot de klasse 'ja' (wat hetzelfde is als 'hartaanval') of de klasse 'nee' (wat hetzelfde is als 'geen hartaanval').

Merk op dat de waarden van de variabelen hier categorisch of kwalitatief zijn, dus voor de computer zullen deze moeten worden omgezet naar numerieke of kwantitatieve variabelen.

Stap 1: op zoek naar de wortel

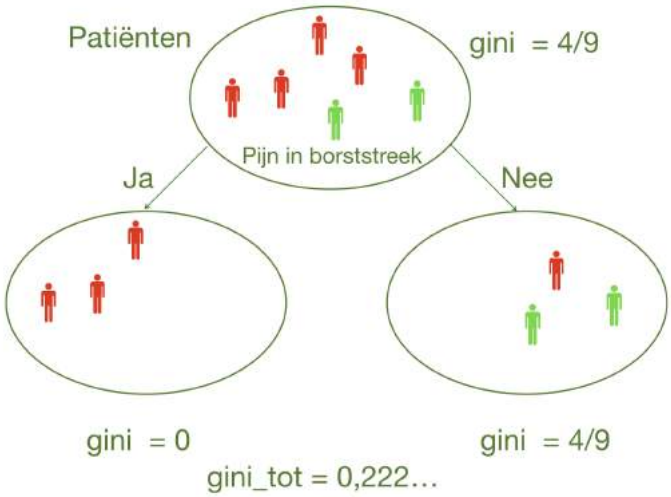
Voor elke parameter wordt bekeken wat de spreiding zou zijn, indien die parameter gebruikt zou worden voor de wortel van de beslissingsboom. De parameter waarvoor de spreiding binnen de bekomen verzamelingen het kleinst is, wordt dan gebruikt voor de wortel. Van elke parameter wordt het resultaat getoond d.m.v. de verzamelingen. Van elk element, dus van elke patiënt, is met een kleur aangeduid of die tot de klasse 'ja' of de klasse 'nee' behoort.

- *Pijn in de borststreek*. Er zijn drie patiënten met pijn in de borststreek en drie patiënten waarvoor dat niet geldt. Als je splitst op de parameter 'pijn in de borststreek', dan bekom je dus twee verzamelingen met elk drie elementen, zoals te zien op Figuur 6.22. Bereken de totale gini-index voor deze opsplitsing. De totale gini-index is $0{,}222....$

- *Man*. Er zijn vier mannelijke patiënten en twee vrouwen. Als je splitst op de parameter 'man', dan bekom je dus een verzameling met vier elementen en een met twee elementen, zoals te zien op Figuur 6.23. Bereken de totale gini-index voor deze opsplitsing. De totale gini-index is $0{,}333....$

- *Rookt*. Er zijn vier patiënten die roken en twee die niet roken. Als je splitst op de parameter 'rookt', dan bekom je een verzameling met vier elementen en een met twee elementen, zoals te zien op Figuur 6.24. Bereken de totale gini-index voor deze opsplitsing. De totale gini-index is $0{,}291....$

- *Voldoende lichaamsbeweging*. Er zijn vier patiënten die voldoende bewegen en twee die dat onvoldoende doen. Als je splitst op de parameter 'voldoende lichaamsbeweging', dan bekom je een verzameling met vier elementen en een met twee elementen, zoals te zien op Figuur 6.25. Bereken de totale gini-index voor deze opsplitsing. De totale gini-index is $0{,}333....$



Figuur 6.22: 'Pijn in borststreek' als mogelijke wortel.

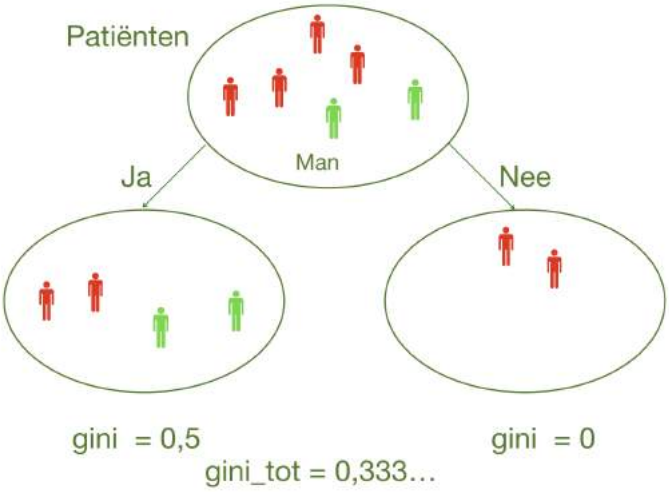
De parameter 'pijn in de borststreek' heeft de kleinste gini-index. Deze parameter wordt dus gekozen als wortel.

Stap 2: de eerste takken van de beslissingsboom

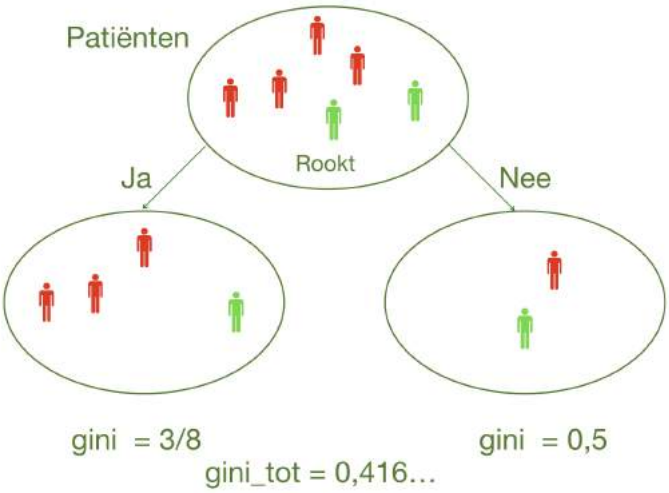
Na toepassing kom je tot de eerste takken van de beslissingsboom, zoals te zien in Figuur 6.26.

Stap 3: een volgende splitsing

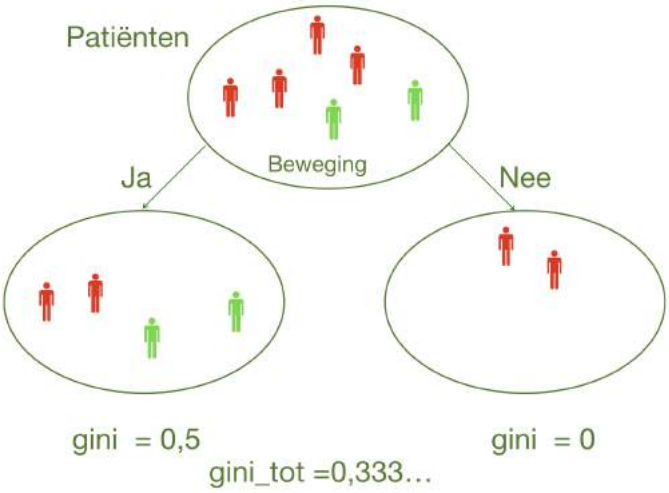
Voor elke van de nog niet gebruikte parameters wordt bekeken wat de spreiding zou zijn, indien die parameter gebruikt zou worden voor een knoop. De parameter waarvoor de spreiding binnen de bekomen verzamelingen het kleinst is, wordt dan gebruikt voor de knoop. De werkwijze is volledig dezelfde als voor de wortel.



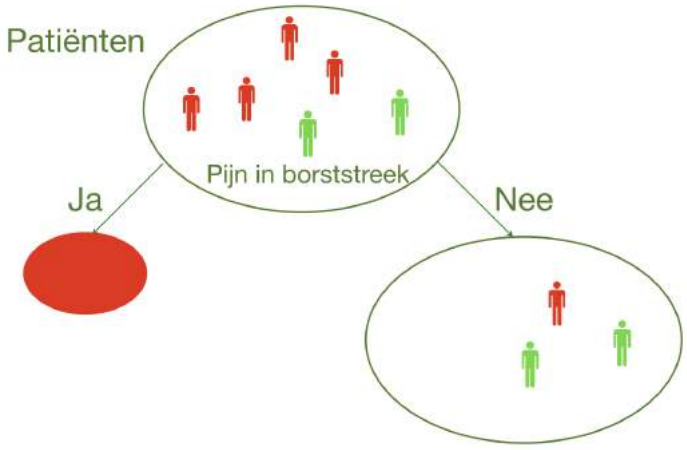
Figuur 6.23: ‘Man’ als mogelijke wortel.



Figuur 6.24: ‘Rookt’ als mogelijke wortel.



Figuur 6.25: ‘Lichaamsbeweging’ als mogelijke wortel.



Figuur 6.26: Eerste deel boom. Een rood blad betekent 'hartaanval'.

In de tabel zijn nu nog de gegevens van belang van de patiënten zonder pijn in de borststreek. Dat zijn de patiënten met patiëntnummer 3, 4 en 6.

Patiëntnummer	Man	Rookt	Voldoende lichaamsbeweging	Hartaanval
3	nee	ja	nee	**ja**
4	ja	nee	ja	**nee**
6	ja	ja	ja	**nee**

Tabel 6.2: Parameters van belang voor het krijgen van een hartaanval bij patiënten zonder pijn in de borststreek.

Versie 1.0

- *Man*. Er zijn twee mannelijke patiënten en een vrouw. Als je splitst op de parameter 'man', dan bekom je dus een verzameling met twee elementen en een met één element, zoals te zien op Figuur 6.27. Bereken de totale gini-index voor deze opsplitsing. De totale gini-index is 0.

Omdat de totale gini-index hier 0 is, heeft het geen zin om de gini-index ook nog voor de andere parameters te berekenen, ze kunnen toch niet kleiner zijn. De knoop zal dus gevormd worden op basis van de parameter 'Man'.

Stap 4: de volgende takken van de beslissingsboom

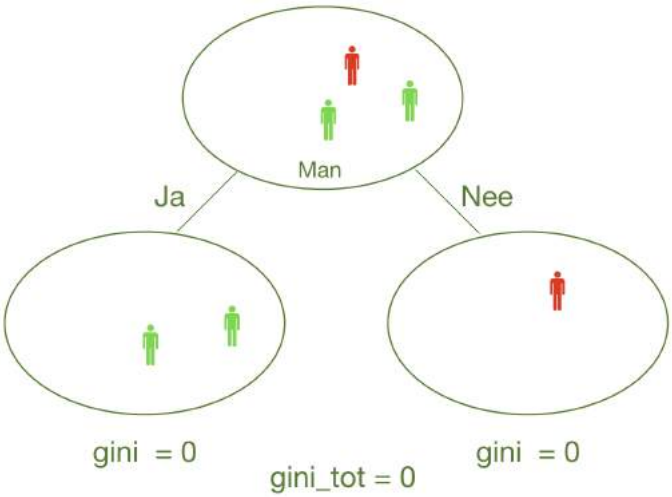
Na toepassing kom je tot de volgende takken van de beslissingsboom, zoals te zien in Figuur 6.28. Alle takken eindigen nu in een blad, dus de nodige opsplitsingen zijn voltooid.

Stap 5: de beslissingsboom

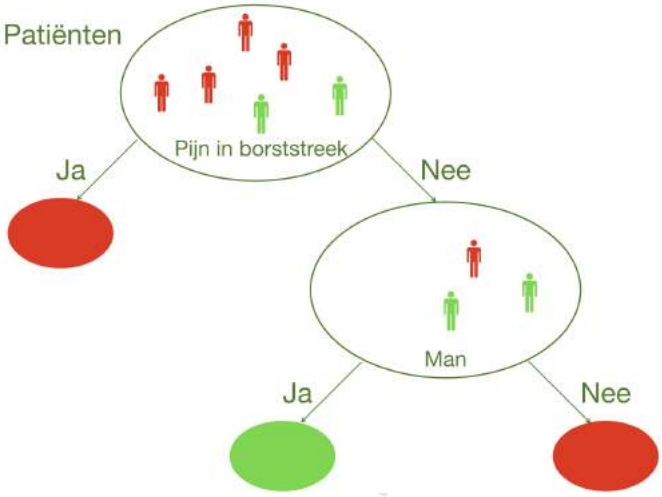
De bekomen beslissingsboom is te zien in Figuur 6.29.

In een volgend hoofdstuk wordt deze boom automatisch gegenereerd met de computer.

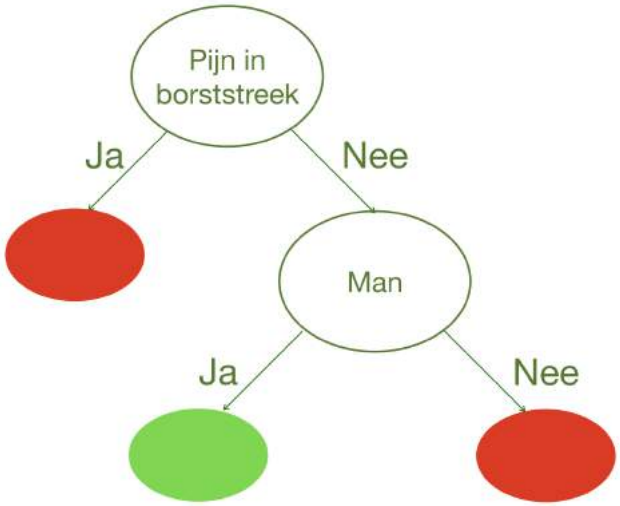
Zie hoofdstuk 8.



Figuur 6.27: 'Man' als mogelijke knoop.



Figuur 6.28: Volgende takken van de beslissingsboom. Een rood blad betekent 'hartaanval', een groen blad betekent 'geen hartaanval'.



Figuur 6.29: Beslissingsboom. Een rood blad betekent 'hartaanval', een groen blad betekent 'geen hartaanval'.

Oefening 6.10.1: logische uitdrukkingen

Elk pad in deze beslissingsboom kan geschreven worden met als-danregels. Vind je ze terug?

- ALS pijn in borststreek DAN hartaanval
- ALS geen pijn in borststreek EN mannelijk DAN geen hartaanval
- ALS geen pijn in borststreek EN niet mannelijk DAN hartaanval

Kritische noot 6.10.2

Deze beslissingsboom heeft wel wat beperkingen en is daarom niet bruikbaar in de praktijk.

- Welke beperkingen stel jij vast?
- Hoe kunnen deze beperkingen worden voorkomen?

6.11 Oefeningen

Pas nu zelf de methode toe die je aangereikt kreeg in het voorbeeld, om een beslissingsboom te maken voor de volgende oefeningen.

versie 1.0

Oefening 1
Van acht personen wordt nagegaan of ze bij het zonnebaden zijn verbrand. Ze zonden tegelijkertijd, even lang en op dezelfde plaats. Van elk van hen wordt de haarkleur, de lengte en gewicht genoteerd en of ze hun huid hebben beschermd door het smeren van een bepaalde zonnecrème (University of Birmingham, 2012).

Stel op basis van deze dataset een beslissingsboom op die voor een persoon die niet in de dataset voorkomt, aangeeft of die persoon (onder dezelfde omstandigheden) zou verbranden of niet.

Naam	Haarkleur	Lengte	Gewicht	Zonnemelk	Klasse
Laura	blond	gemiddeld	licht	nee	**verbrand**
Thomas	blond	groot	gemiddeld	ja	**niet verbrand**
Aline	bruin	klein	gemiddeld	ja	**niet verbrand**
Katrien	blond	klein	gemiddeld	nee	**verbrand**
Jarne	rood	gemiddeld	licht	nee	**verbrand**
Sofie	bruin	groot	zwaar	nee	**niet verbrand**
James	bruin	gemiddeld	zwaar	nee	**niet verbrand**
Emiel	blond	klein	licht	ja	**niet verbrand**

Tabel 6.3: Parameters van belang voor risico op verbranding door de zon.

Oefening 2
Een chirurg noteert, vanuit haar ervaring, vijf gevallen om te beoordelen of een patiënt de dag na een operatie normaal gezien al naar huis kan.
Om zo'n beoordeling vlot te laten verlopen voor nieuwe patiënten

wordt deze expertise omgezet naar een beslissingsboom.
Stel deze beslissingsboom op (Shoemaker et al., 2001).

Patiëntnummer	Ingrijpende operatie	Familie thuis	Oud	Naar huis
1	ja	nee	nee	**nee**
2	ja	nee	ja	**nee**
3	nee	nee	nee	**ja**
4	nee	nee	ja	**nee**
5	nee	ja	ja	**ja**

Tabel 6.4: Parameters om te voorspellen of een patiënt naar huis kan de dag na een operatie.

Oefening 3

Er is heel wat geweten over de passagiers op de Titanic, zoals langs welk dek ze aan boord gingen, of ze in eerste, tweede of derde klasse reisden, of ze broers, zussen, een echtgenoot(ote), kinderen of ouders aan boord hadden, en of ze het ongeluk met de Titanic overleefden of niet.

De tabel geeft enkele van deze gegevens voor acht van hen (Pratish, 2020). Stel op basis van deze tabel een beslissingsboom op die voorspelt wie overleeft en wie niet.

In hoofdstuk 8 kan je hetzelfde doen met de computer met een grote dataset.

Passagiersklasse	Geslacht	Broers, zussen, echtgenoot(ote)	Dek	Overleeft
1	man	2	S	**nee**
2	man	1	S	**nee**
1	vrouw	1	S	**ja**
3	vrouw	0	C	**ja**
1	vrouw	0	Q	**ja**
2	vrouw	2	C	**nee**
2	vrouw	0	C	**nee**
1	man	1	Q	**nee**

Tabel 6.5: Parameters om te zien of iemand de overtocht met de Titanic overleeft.

Versie 1.0

Oefening 4

Wie op zoek gaat naar een nieuwe woning, heeft meestal een verlanglijstje opgesteld over wat er in die woning allemaal moet aanwezig zijn en wat niet. De tabel geeft een voorbeeld van zo'n lijstje.
Stel op basis van deze tabel een beslissingsboom op die bepaalt of een woning die te koop staat, in aanmerking komt om te gaan bezichtigen of niet (University of Birmingham, 2012).

Gemeubeld	Aantal slaapkamers	Nieuwe keuken	Klasse
nee	3	ja	**komt in aanmerking**
ja	3	nee	**komt niet in aanmerking**
nee	4	nee	**komt in aanmerking**
nee	3	nee	**komt niet in aanmerking**
ja	4	nee	**komt in aanmerking**

Tabel 6.6: Parameters om te zien of een huis in aanmerking komt.

Oefening 5
Een leuke golfpartij kan niet onder gelijk welke weersomstandigheden plaatsvinden. Bijvoorbeeld teveel regen of teveel wind kan roet in het eten gooien.
In de tabel vind je een oplijsting van verschillende weersomstandigheden en of die toelaten om te gaan golfen (Serengil, 2018).
Zet de tabel om naar een beslissingsboom.

Weersvoorspelling	Temperatuur	Vochtigheid	Wind	Beslissing
zonnig	warm	hoog	zwak	**nee**
zonnig	warm	hoog	krachtig	**nee**
bewolkt	warm	hoog	zwak	**ja**
regen	mild	hoog	zwak	**ja**
regen	koud	normaal	zwak	**ja**
regen	koud	normaal	krachtig	**nee**
bewolkt	koud	normaal	krachtig	**ja**
zonnig	mild	hoog	zwak	**nee**
zonnig	koud	normaal	zwak	**ja**
regen	mild	normaal	zwak	**ja**
zonnig	mild	normaal	krachtig	**ja**
bewolkt	mild	hoog	krachtig	**ja**
bewolkt	warm	normaal	zwak	**ja**
regen	mild	hoog	krachtig	**nee**

Tabel 6.7: Parameters om te zien of het weer het toelaat om te gaan golfen.

6.12 En met de computer?

Het algoritme om een beslissingsboom te construeren, kan ook geprogrammeerd worden in een computer. Om een computer dit algoritme te laten uitvoeren, moet je de verschillende stappen op een ondubbelzinnige manier in een computerprogramma gieten.

Dat is reeds voor jou gedaan en bereikbaar via Python-modules. Je zal dat gebruiken in hoofdstuk 8. Je zal er dus a.d.h.v. een regelgebaseerd AI-systeem automatisch beslissingsbomen laten genereren.

Om een beslissingsboom aan te maken, berekent het **algoritme** voor elke mogelijke splitsing, m.a.w. voor elke mogelijke ja-neevraag, de spreidingsparameter, dus hoe de 'spreiding van de variabelen over de verschillende klassen' is. Hoe kleiner die spreidingsparameter, hoe beter de splitsing. Net zoals je zelf deed.

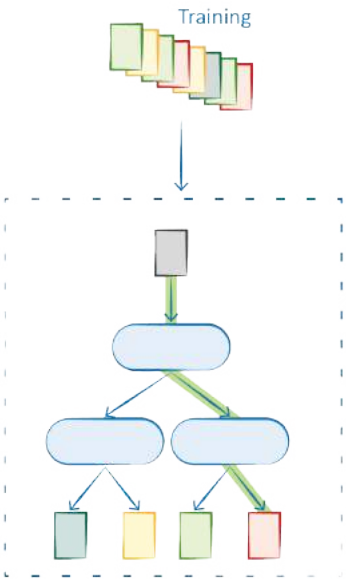
Als er veel parameters zijn en als er voldoende data beschikbaar zijn, dan kan je een beslissingsboom automatisch laten genereren, gebruikmakend van een lerend algoritme. Het algoritme moet leren om in elke knoop die splitsing te kiezen die het meest oplevert: de bedoeling is dat er na het leerproces, de *training*, een algoritme is dat telkens die splitsing kiest waarvoor de **spreiding van de trainingsinputs over de verschillende klassen zo klein mogelijk is**.

Figuur 6.30 toont een AI-systeem dat getraind wordt met gelabelde data; gelabelde data zijn data waarvan de computer weet tot welke klasse die behoren, omdat het in de data erbij vermeld wordt. Zo komt dan eigenlijk met een datagebaseerd AI-systeem een regelgebaseerd AI-systeem tot stand.

De groene, gele en rode kaarten kunnen bv. elektronische patiëntendossiers voorstellen. De gele en rode kaarten kunnen overeenkomen met een geschiedenis van hartfalen, waarbij de rode fataal afgelopen zijn. Het algoritme gaat dan in de patiëntendossiers op zoek naar factoren die bepalend zijn voor het hartfalen en de afloop

Versie 1.0

ervan. A.d.h.v. deze factoren construeert het systeem een beslissingsboom die dan voor niet eerder geziene dossiers kan worden gebruikt.



Figuur 6.30: Training van een beslissingsboom-systeem voor classificatie in drie klassen.

6.13 Het gebruikte algoritme

6.13.1 Verdeel-en-heersalgoritme

Het algoritme dat je in dit hoofdstuk gebruikte om een beslissingsboom te construeren is een verdeel-en-heersalgoritme. Het idee van verdeel-en-heers is dat je om een probleem op te lossen, je het probleem omvormt naar een **identiek probleem dat kleiner is**. Dan pak je dat kleinere probleem op dezelfde manier aan door het ook om te vormen naar een (bijna) identiek probleem dat nog kleiner is. Als je dit zo verder doet, wordt het probleem dat je effectief moet gaan oplossen zo klein dat de oplossing voor de hand ligt.

Met een verdeel-en-heersalgoritme maak je een probleem eenvoudiger bij elke stap.

Door de eeuwen heen heeft men effectieve algoritmes weten op te stellen die een antwoord bieden op specifieke problemen, zoals sorteer- en zoekalgoritmes. Deze algoritmes vertonen bepaalde kenmerken, bv. verdeel-en-heers en gretig.
Via patroonherkenning kan men achterhalen of een bestaand algoritme geschikt is voor een nieuw op te lossen probleem.

Voorbeeld: Stel dat je een medische term wilt opzoeken in een woordenboek, bv. meningitis. In een woordenboek staan de woorden alfabetisch geordend.

- Je opent het woordenboek in de helft.
- Staan de woorden die beginnen met een ‘m’ in het eerste of tweede deel? Neem het juiste deel.
- Verdeel dat deel weer in twee.
- Staan de woorden die beginnen met een ‘m’ in het eerste of tweede deel? Neem opnieuw het juiste deel.
- Herhaal deze methode tot je bij de woorden met een ‘m’ aankomt.

Versie 1.0

Er zijn ook wiskundige toepassingen. Het algoritme van Euclides is een verdeel-en-heersalgoritme om de grootste gemene deler van twee natuurlijke getallen te bepalen.

En er zijn ook praktische toepassingen. Om na het wassen dezelfde kousen weer bij elkaar te voegen, kan je ze eerst sorteren op kleur alvorens ze te matchen (Stock, 2017).

6.13.2 Gulzig algoritme

Het algoritme dat je in dit hoofdstuk gebruikte om een beslissingsboom te construeren heeft nog een bijzonder kenmerk. Het is een gulzig algoritme. Het zal bij elke stap voor het opbouwen van de beslissingsboom kiezen voor die splitsing die op dat moment zo optimaal mogelijk is, maar geen rekening houden met de stappen die erna nog komen. De gevoerde strategie is er een van 'neem wat je nu kunt krijgen' (Fack, 2007). Het algoritme is effectief, maar niet altijd het meest efficiënte.

Een gulzig algoritme wordt ook een gretig algoritme genoemd.

6.14 Het computationeeldenkenproces samengevat

Het schema in Figuur 6.31 op de volgende pagina vat het proces van computationeel denken samen dat je hebt afgelegd in dit hoofdstuk.

Versie 1.0

Stel gelabelde data uit de zorgsector voor op een manier die geschikt is om er beslissingen mee te nemen betreffende een diagnose of een behandeling.

Subtaken (**decompositie**):

1. In de voorbeelden op zoek gaan naar een patroon (een gerichte gewortelde graaf, een binaire beslissingsboom).
2. Uit welke elementen is een beslissingsboom opgebouwd?
3. Wat is de werking van een beslissingsboom?
4. Hoe de mate van spreiding weergeven door een getal.
5. Welke berekeningen leiden tot de juiste splitsing?
6. Hoe geef je ja-neevragen vorm a.d.h.v. een logische (wiskundige) uitdrukkingen?
7. Ontwerpen van een algoritme om een binaire beslissingsboom te construeren.
8. De implementatie van het algoritme in de computer.
9. De data voorverwerken tot het gewenste formaat.

De bestaande voorbeelden tonen dat zulke data vaak worden voorgesteld door een gerichte gewortelde graaf, meetal binair. M.a.w. een binaire beslissingsboom is geschikt om de data te representeren. (**patroonherkenning**)

Een binaire beslissingsboom vertrekt uit een wortel en een ja-neevraag die een scheiding van de data oplevert in twee verzamelingen die zo weinig mogelijk spreiding over de categorieën vertonen. Dat gaat op een analoge manier verder met volgende ja-neevragen. De boom wordt zo opgebouwd met takken en knopen. Tot slot geven de bladeren van de beslissingsboom de mogelijke beslissingen.(**patroonherkenning**)

De beslissingsboom is een abstracte voorstelling van de oplossing in de vorm van een graaf. Het is een model voor de oplossing.(**abstractie**)
Deze voorstelling is bovendien transparant.

Algoritme om binaire beslissingsboom op te stellen.

1. Bekijk welke wortel de beste splitsing oplevert. Maak takken voor de mogelijke uitkomsten van de ja-neevraag.
2. Bekijk voor elke tak welke knoop het meest geschikt is voor de volgende splitsing. Maak takken voor de mogelijke uitkomsten van de ja-neevragen.
3. Herhaal de vorige stap telkens opnieuw voor de nieuw aangemaakte takken.
4. Stop met de herhaling als de bekomen verzamelingen homogeen zijn, dus enkel elementen uit een en dezelfde klasse bevatten.

Dit is een iteratief proces: om de wortel en de knopen te bepalen wordt telkens dezelfde techniek toegepast.

Figuur 6.31: Schema basisconcepten van computationeel denken toegepast op de uitdaging van het project 'AI in de Zorg'.

Samengevat

In dit hoofdstuk staat het volgende probleem centraal: "Hoe kan je aan de hand van gelabelde data een beslissingsboom opstellen?".

Er wordt bepaald uit welke elementen een beslissingsboom wordt opgebouwd, en er wordt bekeken hoe zo'n beslissingsboom juist in zijn werk gaat. Vervolgens wordt nagegaan a.d.h.v. welke berekeningen je tot de juiste keuze van splitsing komt.
De nodige formule en een algoritme om een beslissingsboom te kunnen construeren, worden opgesteld. Tot slot wordt het algoritme gebruikt om een beslissingsboom op te stellen a.d.h.v. reële, soms medische, data.

Na dit hoofdstuk ...

- weet je wat spreiding van de elementen binnen een verzameling betekent;
- weet je dat je de mate van spreiding kunt voorstellen door een getal, nl. de gini-index;
- kan je de gini-index van een verzameling berekenen (bij een spreiding over twee categorieën);
- kan je de totale gini-index berekenen wanneer er gespreid wordt over meerdere verzamelingen (bij een spreiding over twee categorieën);
- ken je een algoritme om een beslissingsboom op te stellen a.d.h.v. de gini-index;
- kan je manueel een beslissingsboom construeren, a.d.h.v. een algoritme.

Vers

Versie 1.0

HOOFDSTUK
ZEVEN

DE NOTEBOOKS

7.1 Interactieve notebooks

Notebooks zijn **digitale documenten** die zowel uitvoerbare code bevatten als tekst, afbeeldingen, video, hyperlinks . . . Via de interactieve notebooks kan je dus nieuwe begrippen aanleren, opdrachten krijgen, tekst typen, foto's bekijken, bepaalde fundamenten van programmeren onder de knie krijgen, aanwezige code uitvoeren en zelf code opstellen.

Jupyter Notebook is een 'open source'-webapplicatie waarin de notebooks van dit project gecreëerd zijn.

De notebooks worden aangeboden via een webserver, toegankelijk via een webbrowser. Je moet op je computer geen extra software installeren om met de notebooks aan de slag te gaan.

In het leerpad 'AI in de Zorg' op `dwengo.org` kan je doorklikken naar de notebooks van het project. De webbrowser legt dan via het internet contact met een DNS-server om het IP-adres (een numeriek internetadres) op te zoeken van de server van AI Op School. Vervolgens wordt de boodschap van de gebruiker (de *client*, hier de webbrowser van de computer van de leerling of leerkracht) doorgegeven aan de webserver van AI Op School. Er wordt namelijk aan deze server gevraagd om toegang te geven tot de notebooks van 'AI in de Zorg'.

De notebooks van AI in de Zorg zijn onderverdeeld in enkele categorieën:

- automatisch genereren van beslissingsbomen;
- voorverwerken van de data;
- basisprincipes van grafen.

In de notebooks worden **groene, blauwe, rode en gele kaders** gebruikt. De groene vertellen wat de notebook behandelt en leggen linken naar de leerinhouden van het project. De blauwe kaders bevatten theorie, bv. begrippen uit de computerwetenschappen. De rode kaders geven tips. De gele kaders verwijzen naar andere notebooks.

De notebooks zijn opgebouwd uit cellen: **Markdown-cellen** voor tekst en beeld, en **code-cellen** om te programmeren.

Notebooks

- Surf naar de website `https://dwengo.org/zorg` waar je het leerpad 'AI in de Zorg' met de notebooks aantreft.

Versie 1.0

7.2 Python

Het programmeren gebeurt met Python 3. Python is een zeer toegankelijke programmeertaal, die vaak ook zeer intuïtief is in gebruik.

Python is een **objectgeoriënteerde taal**. Alles is er een object: elk getal, elk stuk tekst ... Een object neemt een bepaalde plaats in in het geheugen en heeft een bepaalde waarde. Elk object heeft ook een type, bv. integer, string en list. Alle objecten met eenzelfde **type** behoren tot dezelfde klasse. Binnen deze **klasse** zijn er **methodes** voorzien die je specifiek voor objecten van dit type kan oproepen. Naast deze methodes zijn er ook **functies** die je kan gebruiken, zoals `input()`, om iets op te vragen aan de gebruiker, en `print()`, om iets te laten verschijnen op het scherm.

Gaandeweg zal je leren wat de begrippen object, klasse, type, methode en functie inhouden.

Bovendien is Python populair omdat er heel wat **modules** voorhanden zijn die je vrij kan gebruiken. Python draagt zo ook bij tot de democratisering van machinaal leren.

In een module zitten heel wat functies vervat die ervaren informatici reeds hebben geprogrammeerd. Wie dat wilt, kan die modules en de bijbehorende functionaliteiten gebruiken. Daarvoor moet de gewenste module wel geïmporteerd worden in het Python-script. Om een beslissingsboom automatisch te laten genereren, zal je de module scikit-learn gebruiken, de modules pandas en NumPy om met de data te werken, en de module Matplotlib om de beslissingsboom te visualiseren. Je gebruikt functies en methodes om data in te lezen vanuit een zogenaamd csv-bestand, om de dataset voor te verwerken, om de beslissingsboom te genereren en om de boom te visualiseren.

Een programma in Python noem je een ***script***.

in de notebooks leer je ook wat een csv-bestand is en wat het voorverwerken van de data inhoudt.

Versie 1.0

7.3 Praktisch

Bij de notebooks worden de nodige modules altijd bij het begin van een notebook geïmporteerd.

Tekst wordt ingevoerd in een Markdown-cel. Code wordt ingevoerd en uitgevoerd in een code-cel. Cellen kunnen worden bewerkt, verwijderd, toegevoegd, gekopieerd, geknipt en geplakt.

Interessant om te weten is dat in een notebook alle code samen hoort. De notebook onthoudt als het ware welke code reeds werd uitgevoerd, ongeacht in welke volgorde die werd ingetikt in de notebook. **Het is de volgorde van uitvoeren die de volgorde van de instructies in het uiteindelijke script bepaalt.**

Besteed bij het invoeren van code voldoende aandacht aan een leesbare **programmeerstijl** en de nodige verduidelijkende **commentaar**.

Samengevat

Notebooks zijn digitale documenten die ook uitvoerbare code bevatten als tekst, afbeeldingen . . .

De programmeertaal in de notebooks is Python 3.

Voor Python zijn er heel wat modules voorhanden die je vrij kan gebruiken. Deze bevatten functionaliteiten die je kan gebruiken eens je de module geïmporteerd hebt.

De volgorde waarin je de cellen uitvoert, bepaalt de volgorde waarin de computer de instructies uitvoert.

Versie 1.0

Versie 1.0

HOOFDSTUK
ACHT

EEN BESLISSINGSBOOM IN PYTHON

8.1 Voorbeeld notebook 'Hartaanval'

A.d.h.v. patiëntengegevens stelde je in paragraaf 6.10 zelf een beslissingsboom op die voorspelt of een patiënt risico loopt op een hartaanval. Om te tonen hoe het opstellen van een beslissingsboom in zijn werk gaat met Python, gebruiken we hetzelfde voorbeeld.

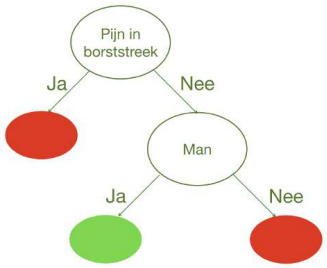
Vertrek opnieuw van dezelfde tabel met de parameters die worden gebruikt om te proberen voorspellen of een patiënt risico loopt op een hartaanval. De volgende tabel bevat gegevens van zes patiënten met de vermelding of ze al dan niet een hartaanval kregen.

Versie 1.0

Patiëntnummer	Pijn in de borststreek	Man	Rookt	Voldoende lichaamsbeweging	Hartaanval
1	ja	ja	nee	ja	**ja**
2	ja	ja	ja	nee	**ja**
3	nee	nee	ja	nee	**ja**
4	nee	ja	nee	ja	**nee**
5	ja	nee	ja	ja	**ja**
6	nee	ja	ja	ja	**nee**

Tabel 8.1: Parameters van belang voor risico op hartaanval.

Je bekwam de beslissingsboom uit Figuur 8.1.



Figuur 8.1: Beslissingsboom. Een rood blad betekent 'risico op een hartaanval', een groen blad betekent 'geen risico op een hartaanval'.

Nu zal je dit implementeren in Python. Je moet dus ook de beperkte dataset in Python ingeven. Dat gebeurt door een matrix in te voeren. Zo'n matrix is niet anders dan een tabel van getallen.

Zoals eerder vermeld dient opgemerkt te worden dat de waarden van de variabelen in de tabel categorisch zijn en dat deze voor de computer omgezet zullen moeten worden naar numerieke of kwantitatieve variabelen.

Notebook 'Hartaanval'

- Probeer dit zelf in de voorbeeldnotebook 'Beslissingsboom: voorbeeld hartaanval'. Je vindt de notebook in het leerpad 'Beslissingsboom' op `dwengo.org/zorg`, onderdeel 'Voorbeeldnotebook'. In deze notebook laat je Python een beslissingsboom aanmaken, gebruikmakend van de gini-index, en gebaseerd op de gegeven tabel.
- In de notebook worden de waarden van de variabelen ingegeven als numerieke waarden. Voor 'nee' wordt een '0' gebruikt en voor 'ja' een '1'.

Hoe wordt er concreet te werk gegaan?

Eerst worden de nodige Python-modules geïmporteerd, zodat de nodige functies en methodes om met beslissingsbomen aan de slag te gaan, beschikbaar zijn.

```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> from sklearn import tree
```

Het gaat hier om de module NumPy (om te kunnen werken met matrices), de submodule pyplot van Matplotlib (om een afbeelding te kunnen tonen) en de submodule tree van scikit-learn (om een beslissingsboom te kunnen genereren).

Een matrix ingeven met de gegevens over de patiënten, gebeurt met de **functie** `array()` van de module NumPy:

```
>>> data = np.array([[1, 1, 0, 1, 1],
                     [1, 1, 1, 0, 1],
                     [0, 0, 1, 0, 1],
                     [0, 1, 0, 1, 0],
                     [1, 0, 1, 1, 1],
                     [0, 1, 1, 1, 0]])
```

De rijen van de matrix worden een voor een ingegeven. `data` is een **variabele** die verwijst naar de ingegeven matrix.

De eerste vier kolommen van deze matrix bevatten de gezondheidsparameters die in aanmerking worden genomen; een parameter die aanwezig is, krijgt waarde '1', als de parameter niet aanwezig is, dan krijgt die waarde '0'.
De laatste kolom van de matrix bevat het feit of de patiënt een hartaanval gekregen heeft ('1') of niet ('0').

Om aan te geven waar de gezondheidsparameters en de klassen staan, gebruik je:

```
>>> gezondheidsparameters = data[:, :4]
>>> klasse = data[:, 4]
```

De plaats van elk element in de matrix wordt bepaald a.d.h.v. de rij en de kolom waarin dat element staat. Elke kolom van de matrix heeft een index; daarbij wordt een nummering gebruikt die start vanaf 0, d.w.z. dat de eerste kolom index 0 heeft, de tweede kolom index 1, de derde kolom index 2 en de vierde kolom index 3. De vijfde en laatste kolom heeft index 4.
Zo is bv. het element op de tweede rij en de vierde kolom een 0, en dat element kan in Python met de volgende instructie worden opgevraagd: `data[1, 3]`. De volledige tweede rij vraag je op met `data[1, :]`. De volledige vierde kolom vraag je op met `data[:, 3]`.
`gezondheidsparameters` en `klasse` zijn variabelen die refereren aan respectievelijk de eerste vier kolommen en de laatste kolom van de ingegeven matrix.

Extra uitleg:

```
data[:, 4]
```

betekent alle rijen van de kolom met **index** 4.

```
data[:, :4]
```

betekent alle rijen van alle kolommen t.e.m. die met index 3 (dus tot voor index 4).

Je vertelt Python dat je een beslissingsboom wilt om de patiënten te classificeren, dat die moet worden aangemaakt via de gini-index,

en dat je aan die beslissingsboom wilt refereren met de variabele `beslissingsboom`:

```
>>> beslissingsboom = tree.DecisionTreeClassifier(
    criterion="gini")
```

`DecisionTreeClassifier()` is een functie van de submodule tree van scikit-learn.

`beslissingsboom` is een variabele die verwijst naar de beslissingsboom die door Python wordt aangemaakt a.d.h.v. de gini-index.

Je vertelt dat de beslissingsboom ingevuld moet worden op basis van de gezondheidsparameters en de gegeven klassen:

```
>>> beslissingsboom.fit(gezondheidsparameters,
    klasse)
```

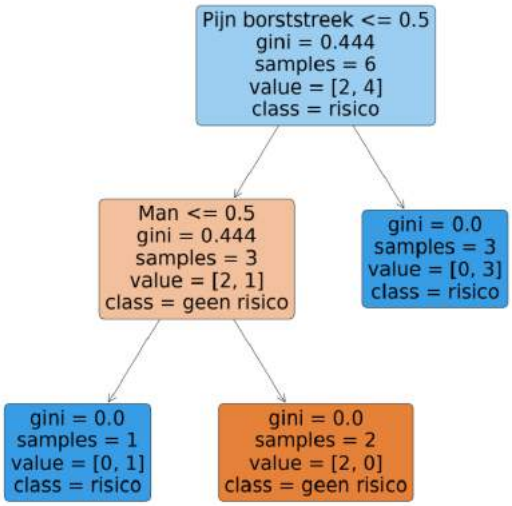
`fit()` is een methode die toegepast wordt op de aangemaakte beslissingsboom.

Tot slot wordt er een tekenscherm aangemaakt met de functie `figure()` en wordt er een afbeelding van de gegenereerde beslissingsboom getoond a.d.h.v. de functies `plot_tree()` en `show()`:

```
>>> plt.figure(figsize=(10,10))
>>> tree.plot_tree(beslissingsboom,
                   class_names=["geen risico", "
                       risico"],
                   feature_names=["Pijn borstreek",
                       "man", "Rookt", "Beweging"],
                   filled=True, rounded=True)
>>> plt.show()
```

`figure()` en `show()` zijn functies van de submodule pyplot van Matplotlib.
`plot_tree()` is een functie van de submodule tree van scikit-learn; de parameters `filled` en `rounded` betreffen de afgeronde hoeken van de knopen in de afbeelding van de beslissingsboom en dat deze knopen opgevuld zijn.

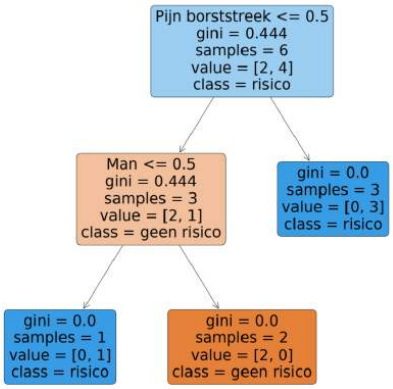
Python genereert de beslissingsboom uit Figuur 8.2.



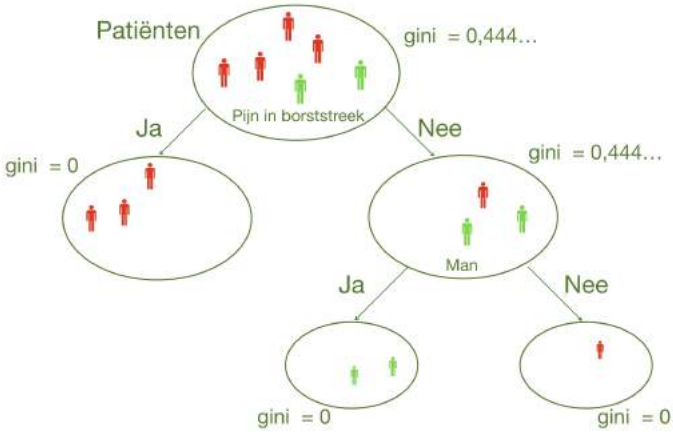
Figuur 8.2: Beslissingsboom. Een donkerblauw blad betekent 'risico op een hartaanval', een oranje blad betekent 'geen risico op een hartaanval'.
Bovenaan in de wortel en in de knopen (niet in de bladeren) staat een voorwaarde. Er wordt nagegaan of aan die voorwaarde is voldaan: de uitdrukking is waar of niet waar. De parameters kunnen slechts twee waarden aannemen, 0 of 1, dus om na te gaan welke van de twee het is, kijk je bv. of de waarde groter of kleiner is dan $0{,}5$. In een knoop leidt 'waar' steeds tot de linkertak en 'niet waar' tot de rechtertak. Voorbeeld: een patiënt met pijn in de borststreek, heeft voor de wortel een waarde '1', de uitdrukking 'Pijn borststreek $\leq 0{,}5$' is niet waar, dus de patiënt volgt de rechtertak.

8.2 De gegenereerde beslissingsboom in detail

Vergelijk deze uitvoer met wat je manueel deed in paragraaf 6.10. Figuur 8.4 toont de aangemaakte beslissingsboom met alle details erbij.



Figuur 8.3: Beslissingsboom gegenereerd met Python.



Figuur 8.4: Beslissingsboom manueel opgesteld. Rood betekent 'risico op een hartaanval', groen 'geen risico op een hartaanval'.

Versie 1.0

- Er werd vertrokken van een 'gelabelde' dataset van zes patiënten die terug te vinden is in de wortel van de boom: vier patiënten kregen een hartaanval (deze zijn in het rood aangeduid) en twee patiënten niet (deze zijn in het groen aangeduid). Deze dataset zie je in de wortel van de boom in Figuur 8.4.
  In de boom in Figuur 8.3 die gegenereerd werd met Python staat er `samples = 6`, wat slaat op de zes patiënten in de dataset, en `value = [2, 4]`, wat betekent dat er twee patiënten tot klasse '0' behoren ('geen risico') en vier tot klasse '1' ('risico'); omdat de meerderheid van deze patiënten tot klasse '1' behoren, staat er `class = risico`.

- Van alle patiënten hebben er drie het symptoom 'pijn in de borststreek', deze patiënten gaan bij de eerste vertakking van de zelf opgestelde boom naar de linkerkant (Figuur 8.4).
  In de met Python gegenereerde boom, zie Figuur 8.3, gaan die drie patiënten (`samples = 3`) bij de eerste vertakking naar rechts. Immers: het symptoom 'pijn in de borststreek' is aanwezig, dus de parameter heeft de waarde '1', m.a.w. de uitdrukking 'Pijn borst $\leq$ 0,5' is niet waar. Deze drie patiënten kregen een hartaanval en behoren m.a.w. allen tot klasse '1'. `value = [0, 3]` betekent 0 elementen in klasse '0' en drie in klasse '1'. Omdat de meerderheid (hier zelfs allemaal) van deze patiënten tot klasse '1' behoren, staat er `class = risico` vermeld.

- Terug naar de zelf opgestelde boom. De patiënten zonder het symptoom 'pijn in de borststreek' gaan bij de eerste vertakking dus naar rechts. Dat zijn er drie: twee zonder hartaanval en één met hartaanval. In de met Python gegenereerde boom gaan die naar

links, daar staat `samples = 3`, dus drie patiënten, en `value = [2, 1]`, dus twee in klasse '0' en één in klasse '1'; de meerderheid behoort tot klasse '0'. Er staat `class = geen risico`.

- In de zelf opgemaakte boom gaan de twee mannelijke patiënten naar de linkse vertakking. Beiden hebben het label 'geen hartaanval'; de overblijvende patiënt gaat naar rechts en heeft label 'hartaanval'.
  In de met Python gegenereerde boom wordt ook gekeken naar de parameter 'Man'. Voor mannelijke patiënten heeft de parameter 'Man' de waarde '1', voor hen is de uitdrukking 'Man $\leq$ 0,5' niet waar, dus deze twee patiënten gaan in de vertakking naar rechts; `samples = 2`. Niemand van hen heeft een hartaanval, m.a.w. ze behoren beiden tot klasse '0', wat weergegeven wordt met `value = [2, 0]` en `class = geen risico`. Voor patiënten die geen man zijn, heeft de parameter 'Man' de waarde '0', dus voor die ene patiënt is 'Man $\leq$ 0,5' waar; deze patiënt gaat in de vertakking naar links. Deze patiënt kreeg een hartaanval en behoort m.a.w. tot klasse '1', wat weergegeven wordt met `value = [0, 1]` en `class = risico`.

1.0

8.3 Oefeningen

Oefeningen: beslissingsbomen genereren met Python

Gebruik de notebook 'Beslissingsboom: de code' van het leerpad 'Beslissingsboom' op `dwengo.org/zorg`, onderdeel 'Voorbeeldnotebook', om voor de oefeningen uit paragraaf 6.11 een beslissingsboom te laten genereren met Python.

- Pas de code in de notebook aan in functie van het nieuwe probleem.
- Vergelijk telkens het resultaat met de beslissingsboom die je manueel bekomen hebt.
- Bewaar voor elke oefening een pdf van de ingevulde notebook onder een zinvolle naam.

8.4 Voorbeeld 'Hartaandoening' met echte data

In een volgende reeks oefeningen wordt er een beroep gedaan op 'echte' data. Deze gegevens zijn opgeslagen in een databestand, dat soms aanzienlijk is; er wordt gewerkt met csv-bestanden.

Hierbij staat csv voor *comma separated values.*

Deze bestanden zijn eigenlijk grote tabellen waarin elke kolom bv. de waarden van een specifieke gezondheidsparameter bevat van een groot aantal patiënten. Eén bepaalde kolom bevat een 'label', de klasse waartoe de patiënt behoort, bv. 'komt terecht op de intensieve zorg'.

Je zal de tabellen inlezen en voorverwerken met functionaliteiten van de Python-modules pandas en NumPy. Eens je de gewenste parameters en het label hebt bekomen in een NumPy-array, verloopt alles op dezelfde manier als in de vorige oefeningenreeks.

In de notebook 'Data frame' van het leerpad 'Beslissingsboom' op `dwengo.org/zorg` leer je de data voorverwerken.

In de notebook 'Beslissingsboom: hartaandoening' leer je hoe je met de csv-bestanden aan de slag gaat.

Notebook 'Hartaandoening'

- Open de notebook 'Beslissingsboom: hartaandoening' van het leerpad 'Beslissingsboom' op `dwengo.org/zorg`, onderdeel 'Oefennotebooks'. In deze notebook ga je aan de slag met een csv-bestand.
- Het is noodzakelijk om de data voor te verwerken.
- Laat Python een beslissingsboom aanmaken.

Hart- en vaatziekten vormen een belangrijke doodsoorzaak. Hartfalen bv. is een veelvoorkomende aandoening.

De dataset `heart.csv` bevat waarden van 11 parameters die gebruikt kunnen worden om een mogelijke hartziekte te voorspellen. Een hoge bloeddruk, diabetes, een verhoogde cholesterol zijn bv. gekende factoren die het risico op een hartaandoening verhogen. Deze dataset is een combinatie van 5 datasets uit de VS en Europa en bevat lichaamsparameters van 918 personen.

Om te kunnen beschikken over extra functionaliteiten voor het voorverwerken van de csv-bestanden, importeer je de module pandas:

```
>>> import pandas as pd
```

Het csv-bestand bevindt zich in de map data in de geopende notebookgroep op de server van AI in de Zorg. De computer leest het bestand in als hij de volgende instructie krijgt:

```
>>> hart = pd.read_csv("data/heart.csv")
```

De variabele `hart` verwijst naar dit ingelezen bestand. Met de volgende instructie vraag je om de tabel te laten zien:

`read_csv()` is een functie in de module pandas.

```
>>> hart
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina
0	40	M	ATA	140	289	0	Normal	172	N
1	49	F	NAP	160	180	0	Normal	156	N
2	37	M	ATA	130	283	0	ST	98	N
3	48	F	ASY	138	214	0	Normal	108	Y
4	54	M	NAP	150	195	0	Normal	122	N
...	...	...	...	...	...	...	...	...	...
913	45	M	TA	110	264	0	Normal	132	N
914	68	M	ASY	144	193	1	Normal	141	N
915	57	M	ASY	130	131	0	Normal	115	Y
916	57	F	ATA	130	236	0	LVH	174	N
917	38	M	NAP	138	175	0	Normal	173	N

Figuur 8.5: pandas tabel databestand 'heart'.

Versie 1.0

Elke kolom in de tabel heeft een 'titel', dat is de **sleutel** (de *key*) van de kolom. A.d.h.v. die sleutel kan je een kolom opvragen.

Test bv. het volgende eens uit:

```
>>> hart["ChestPainType"]
```

Merk op dat de tabel tamelijk wat categorische variabelen bevat. De tabel is bovendien omvangrijk. Je kan dus deze categorische variabelen dus best op een geautomatiseerde manier numeriek maken. Je geeft daartoe een instructie aan de computer. De methode `replace()` is daarvoor geschikt en kan je op elke (gewenste) kolom toepassen. Met deze methode kan je bv. een string vervangen door een getal.

Met `replace("F", 0)` wordt de letter F vervangen door een 0.

```
>>> hart["Sex"]= hart["Sex"].replace("F", 0)
>>> hart["Sex"]= hart["Sex"].replace("M", 1)
>>> hart["ChestPainType"]= hart["ChestPainType"].
    replace("ASY", 1)
>>> hart["ChestPainType"]= hart["ChestPainType"].
    replace("ATA", 2)
>>> hart["ChestPainType"]= hart["ChestPainType"].
    replace("NAP", 3)
>>> hart["ChestPainType"]= hart["ChestPainType"].
    replace("TA", 4)
>>> hart["RestingECG"]= hart["RestingECG"].replace("
    Normal", 0)
>>> hart["RestingECG"]= hart["RestingECG"].replace("
    ST", 1)
>>> hart["RestingECG"]= hart["RestingECG"].replace("
    LVH", 2)
>>> hart["ExerciseAngina"]= hart["ExerciseAngina"].
    replace("N", 0)
>>> hart["ExerciseAngina"]= hart["ExerciseAngina"].
    replace("Y", 1)
>>> hart["ST_Slope"]= hart["ST_Slope"].replace("Up",
     1)
>>> hart["ST_Slope"]= hart["ST_Slope"].replace("Flat
    ", 0)
>>> hart["ST_Slope"]= hart["ST_Slope"].replace("Down
    ", -1)
```

Je kan de aangepaste tabel eens bekijken via de instructie `hart`.

De laatste kolom van de tabel is de kolom met de labels: wel of geen hartaandoening. Deze kolom beschouw je apart:

```
>>> klasse = heart["HF"]
```

Omdat pijn in de borststreek, bloeddruk (in rust) en bloedsuikerspiegel (op nuchtere maag) gekend zijn als parameters die een aanwijzing geven over het al dan niet aanwezig zijn van een hartaandoening, haal je deze parameters eruit om te gebruiken als basis van de beslissingsboom.

```
>>> parameters = heart[["ChestPainType", "RestingBP
    ", "FastingBS"]]
```

Eens alles naar wens is, zet je de objecten om naar een NumPy array.

Let hier op de dubbele vierkante haakjes. De functies die je gebruikt, verwachten nl. een dergelijk formaat. Kijk eventueel eens terug naar de vorige oefeningenreeks. Daar zul je de dubbele vierkante haken ook zien staan.

```
>>> parameters = np.array(parameters)
>>> klasse = np.array(klasse)
```

Zo, nu ben je op het punt aanbeland waar je in de vorige oefeningenreeks de beslissingsboom liet genereren. De stappen die nu moeten volgen, zijn je dus bekend.

Je zal merken dat de gegenereerde beslissingsboom heel omvangrijk is. Om een gebruiksvriendelijke boom te bekomen wordt de **diepte** van de boom best beperkt. Je beperkt de diepte door bv. `max_depth=3` als een tweede parameter mee te geven aan de functie `DecisionTreeClassifier()`.

Er was al een eerste parameter expliciet meegegeven aan de functie `DecisionTreeClassifier()`: `criterion="gini"`.
Parameters die niet expliciet vermeld worden, krijgen de standaardwaarde toegekend.

8.5 Oefeningen met real-world data

Oefeningen: beslissingsbomen op basis van real-world data

- Via de server van AI Op School beschik je over enkele real-world datasets.

  Je vindt de databestanden `kyphosis.csv`, `postoperative.csv`, `borstkanker.csv`, `trainTitanic.csv` in de map data van de notebookgroep van het onderdeel 'Oefennotebooks' van het leerpad 'Beslissingsboom' (project 'AI in de Zorg').

- Gebruik een notebook om op basis van een van deze datasets een beslissingsboom te laten genereren met Python.

De data in `kyphosis.csv` werd gebruikt voor de beslissingsboom in Figuur 2 in de inleiding van deze cursus.

Na dit hoofdstuk, dus na de notebooks over beslissingsbomen, ...

- begrijp je waar de volgende begrippen voor staan: variabele, algoritme, sequentie, operator;
- ken je de types `string`, `int`, `float`;
- ken je de functie `print()`;
- leerde je dat een matrix in Python wordt voorgesteld door een NumPy `ndarray`;
- ken je de betekenis van een sleutel in een pandas-tabel en een index in een `ndarray`;
- kan je categorische variabelen omzetten naar numerieke, soms Booleaanse variabelen, en weet je waarom dat nodig is;
- kan je kolommen en rijen verwijderen in een pandas-tabel;
- kan je functies en methodes gebruiken om een beslissingsboom te genereren en te visualiseren.

Versie 1.0

Versie 1.0

HOOFDSTUK
NEGEN

EARLY WARNING SYSTEM

9.1 Homeostase

Levende wezens zijn “open systemen” die materie en energie uitwisselen met hun omgeving.

Zie ook de lessen Biologie.

Bij meercellige organismen spreken we van een **inwendig** en **uitwendig** milieu. Het inwendig milieu wordt gevormd door lichaamsvochten (bloed, weefselvocht ...). Het uitwendig milieu kan de buitenomgeving van het organisme zijn maar evengoed de holtes in contact met de buitenwereld, bv. de holtes van het spijsverteringskanaal, de baarmoederholte ...

In het uitwendig milieu van mensen en dieren zijn er voortdurend wijzigingen in fysische en chemische omstandigheden, zoals temperatuurschommelingen, schommelingen in zuurstofconcentratie, voedingsbronnen met verschillende suikerconcentraties ...

Heel wat organismen hebben in de loop van de evolutie geleerd om die schommelingen op te vangen zodat hun inwendig milieu zo stabiel mogelijk blijft. Dit is wat we noemen homeostase.

De term ‘homeostase’ is afgeleid uit het Grieks: *homeos* betekent gelijk en *stasis* betekent toestand.

Elk **homeostatisch regelsysteem** bestaat uit drie van elkaar afhankelijke niveaus: de **sensor**, het **controlecentrum** en de **effector** staan in nauw contact met elkaar en communiceren via het bloedvatenstelsel of via het zenuwstelsel.

Veranderingen in het lichaam, zoals bv. een verandering in bloeddruk of in lichaamstemperatuur, worden gedetecteerd door **sensoren**. De sensoren registreren en signaleren deze veranderingen aan het **controlecentrum** in de hersenen. Controlecentra ontvangen en verwerken die informatie. Zij werken op basis van *setpoints*; dit zijn grenswaarden waarbinnen een inwendig milieu stabiel moet gehouden worden. Vervolgens sturen de controlecentra een boodschap naar de **effectoren** om een geschikte actie in gang te zetten. Het resultaat van deze reactie wordt opnieuw door sensoren geregistreerd en doorgegeven naar het controlecentrum. Dit noemen we **terugkoppeling (feedback)**.

Je kan dit vergelijken met de werking van een thermostaat.

Door dit systeem van receptoren, hersenen en effectoren blijft het lichaam normaal gezien **in balans**. Als het lichaam van een persoon er niet in slaagt om weer tot de streefwaarden te komen via de na-

Versie 1.0

tuurlijke feedbackmechanismen, dan krijgt het lichaam “biologische stress”. Het meten van bepaalde waarden van de lichaamsparameters (lichaamstemperatuur, bloeddruk, bloedsuikerspiegel ...) geeft dus een beeld van de gezondheid van een organisme en de goede werking van homeostatische regelmechanismen.

9.2 EWS

De *Early Warning Score* (EWS) wordt internationaal gebruikt in ziekenhuizen om de gezondheid van een patiënt te kunnen inschatten. De EWS wordt berekend op basis van de waarden van enkele vitale parameters, lichaamsparameters die gelinkt zijn aan de vitale functies van ons lichaam: bloeddruk, lichaamstemperatuur, hartslag, ademhalingsfrequentie, alertheid en zuurstofsaturatie.

De EWS dient om een verslechtering in de toestand van een ernstig zieke patiënt snel te herkennen, zodat men tijdig kan ingrijpen. Vanaf een bepaalde EWS-waarde zal men bijvoorbeeld de vitale parameters frequenter gaan opmeten, wordt de hulp van een arts ingeroepen of zal een patiënt worden opgenomen op de intensieve zorg.

Berekening EWS

Het proces om de EWS te bekomen:

- data verzamelen: nauwkeurig opmeten van vitale parameters;
- het juiste gewicht toekennen aan een parameter, overeenkomstig met de mate waarin de meetwaarde afwijkt van de ‘normale’ waarde; deze gewichten staan in de bovenste rij van de tabel in figuur 9.1;
- alle gewichten optellen; de bekomen som is de EWS.

De zorgverleners beschikken over richtlijnen over welke actie ze dienen te ondernemen afhankelijk van de bekomen EWS (zie Figuur 9.2).

Een zo nauwkeurig mogelijk berekende EWS is belangrijk: bij een te hoge score wordt een arts misschien nodeloos opgeroepen, waardoor de EWS in ongenade kan vallen bij die persoon, en bij een te lage score krijgt een patiënt misschien niet de nodige zorg.

Voorbeeld 9.2.1: EWS berekenen

- Bereken de EWS bij een hartslagfrequentie van 86 slagen per minuut, een temperatuur van 38,8 °C, een systolische bloeddruk van 105 mmHg, een ademhalingsfrequentie van 20 per minuut, en een bloedsaturatie van 97 %. Het bewustzijn van de patiënt werd als ‘aanspreekbaar’ genoteerd.

  Welke actie wordt er ondernomen?

- Antwoord: HF -> 0; temp -> 1, bewustzijn -> 1, bloeddruk -> 0, ademhaling -> 1 dus EWS = 3.
  De arts wordt opgebeld.

MARIA MIDDELARES **Early Warning Score**

	3	2	1	0	1	2	3
Ademfrequentie		<9		9-14	15-20	21-30	>30
RR Systolisch	<70	70-80	81-100	101-180	181-200	>200	
HF		<40	40-50	51-100	101-110	111-130	>130
Temperatuur		<35,5		35,5-38	>38		
Bewustzijn			acute agitatie	W	A		P-A

Indien $satO^2$<92%: + 3 punten.
Indien RSP of neurologische uitval: bel arts.
Indien ongerustheid over de toestand van de patiënt: + 1.

Score <2: (her)beoordeel patiënt om de 8 uur.
Score = 2: (her)beoordeel patiënt om de 3 uur.
Score 3 - 4: Bel arts (SBAR). Herevalueer na 30'.
Score ≥ 5: Bel arts en RRT.

≥5
RRT: 7777

Figuur 9.1: Gewichten voor de EWS. Naargelang het interval waarbinnen een meetwaarde valt, wordt aan de meetwaarde een bepaald gewicht toegekend, zoals te zien op de eerste rij van deze tabel. Deze gewichten zijn overeengekomen door gespecialiseerde artsen.

Indien $satO^2$<92%: + 3 punten.
Indien RSP of neurologische uitval: bel arts.
Indien ongerustheid over de toestand van de patiënt: + 1.

Score <2: (her)beoordeel patiënt om de 8 uur.
Score = 2: (her)beoordeel patiënt om de 3 uur.
Score 3 - 4: Bel arts (SBAR). Herevalueer na 30'.
Score ≥ 5: Bel arts en RRT.

Figuur 9.2: Acties afhankelijk van de EWS. Merk op dat de expertise van de zorgverlener ook wordt meegenomen.

Oefeningen 9.2.2: EWS berekenen

Oefening 1

- Bereken de EWS van een 'wakkere' patiënt die 10 keer per minuut ademt, en een systolische bloeddruk van 88 mmHg en een hartslagfrequentie van 64 slagen per minuut heeft. Zijn lichaamstemperatuur is 34,9 °C en zijn bloedsaturatie is 95 %.
- Welke actie wordt er ondernomen?

Oefening 2

- Een verpleger die ongerust is over de toestand van een patiënt berekent de EWS. De systolische bloeddruk bedraagt 200 mmHg, de hartslagfrequentie is 105 slagen per minuut, de temperatuur is 37 °C en de ademhalingsfrequentie 31 per minuut, er is een bloedsaturatie van 89 %. De verpleger beoordeelt het bewustzijn van de patiënt als 'acute agitatie'.
- Welke actie onderneemt de verpleger?

9.3 Fouten bij bepalen van de EWS

Als de EWS niet correct is, dan is het mogelijk dat er een verkeerd gevolg aan wordt gegeven.

Omdat er nogal wat fouten kunnen optreden bij het bepalen van de EWS van een patiënt, wordt bekeken hoe deze door het proces (deels) te automatiseren voorkomen zouden kunnen worden.

Voorbeeld 9.3.1: mogelijke fouten bij de EWS

Bedenk welke fouten er kunnen optreden bij het bepalen van de EWS.

Opdracht 9.3.2: oplossing voor een foute EWS

- Bedenk bij elke fout een mogelijke oplossing voor het probleem dat optrad.
- Verzin zeker een technologische oplossing.

9.4 Automatisatie

Opdracht 9.4.1

Formuleer enkele voor- en nadelen van een (deels) geautomatiseerde EWS.

Niet enkel het doen van de nodige metingen en het berekenen van de EWS kan automatisch gebeuren. Automatisatie kan ook een rol spelen bij de **ontwikkeling** van een EWS-systeem.

De gewichten die aan bepaalde waarden van de vitale parameters worden toegekend om de EWS te bekomen, zijn vastgelegd door mensen met expertise in het domein, vooral d.m.v. trial-and-error, weliswaar aangevuld met hun klinisch inzicht. Dat proces is zeer tijdrovend.

Als men over voldoende gelabelde data beschikt, kan men de EWS bepalen met een beslissingsboom. Onderzoek heeft aangetoond dat dit vergelijkbare resultaten oplevert, maar veel sneller gaat (Badriyah et al., 2013; Xu et al., 2015).

Notebook: een beslissingsboom voor de EWS

- Open de notebook ‘EWS’ in het leerpad ‘Beslissingsboom’ op `dwengo.org/zorg`.
- Stel een beslissingsboom op die de EWS-score voorspelt.

Samengevat

In dit hoofdstuk wordt dieper ingegaan op een specifiek systeem in de zorg: de EWS-score. Er wordt gekeken in welke mate dit systeem geautomatiseerd kan worden. Daarbij mag de zorgverlener zelf niet 'out of the picture' zijn.

Tot slot wordt gekeken hoe AI een rol kan spelen bij deze toepassing in de zorg, bv. om te streven naar een betere versie van de EWS. Dat kan bv. gebeuren met een transparant systeem zoals een beslissingsboom.

Na dit hoofdstuk ...

- weet je wat EWS is;
- kan je manueel de EWS berekenen en de actie die uit de score volgt, afleiden;
- kan je voorbeelden geven van hoe het EWS-systeem kan geautomatiseerd worden;
- kan je uitleggen hoe AI een rol kan spelen bij die automatisatie;
- kan je illustreren waarom de zorgverleners best betrokken worden bij de invoering van een nieuwe technologische toepassing in de zorginstelling.

Versie

Versie 1.0

Versie 1.0

Versie 1.0

BIBLIOGRAFIE

Afonso, A., Ebell, M., Gonzales, R., Stein, J., Genton, B., & Senn, N. (2012). The use of classification and regression trees to predict the likelihood of seasonal influenza. *Family Practice*, *6*(29), 671–677.

Astudillo, P. & Dambre, J. (2020). Algoritme neemt huiswerk voor hartoperatie over. *EOS Wetenschap Special*, Technologie en gezondheid, 63.

AZ Delta (2021). *1000ste knierobot* [Videobestand]. Geraadpleegd op 7 oktober 2019 via `https://www.orthopedie-roeselare.be/nl/specialisaties/knie/knieprothese-robot-geassisteerd`.

Badriyah, T., Briggs, J., Meredith, P., Jarvis, S., Schmidt, P., Featherstone, P., Prytherch, D., & Smith, G. (2013). Decision-tree early warning score (DTEWS) validates the design of the National Early Warning Score (NEWS). *Resuscitation*, *85*.

Barents, E., Bilo, H., Bouma, M., Dankers, M., De Rooij, A., Hart, H., Houweling, S., IJzerman, R., Janssen, P., Kerssen, A., Oud, M., Palmen, J., Van den Brink-Muinen, A., Van den Donk, M., Verburg-Oorthuizen, A., & Wiersma, T. (2021). Diabetes mellitus type 2. Geraadpleegd op 13 december 2021 via `https://richtlijnen.nhg.org/standaarden/diabetes-mellitus-type-2`.

BBC (2019). Go master quits because AI 'cannot be defeated'. Geraadpleegd op 9 april 2021 via `https://www.bbc.com/news/technology-50573071`.

Berbers, Y., Debeuckelaere, W., De Herdt, P., Desmedt, Y., De Smet, F., Hildebrandt, M., Poels, K., Pierson, J., Preneel, B., & Vandewalle, J. (2017). *Privacy in tijden van internet, sociale netwerken en big data*. Brussel: KVAB Standpunten 49.

Berkers, E. (2020). Eliza. Geraadpleegd op 9 april 2021 via `https://www.eclecticenergies.com/nederlands/psyche/eliza`.

Bingli (2021). Bingli, het slimme medische interview. Geraadpleegd op 1 augustus 2021 via `https://www.mybingli.com`.

Boden, M. A. (2016). *AI. Its nature and future*. Oxford, Verenigd Koninkrijk: Oxford University Press.

Bromet, F. (2016). *Mobiel bellen in 1998* [Videobestand]. Geraadpleegd op 8 augustus 2020 via `https://youtu.be/TNwhIHqM60g`.

Cavallo, J. (2019). Confronting the Criticisms Facing Watson for Oncology. A Conversation With Nathan Levitan, MD, MBA. Geraadpleegd op 5 januari 2022 via `https://ascopost.com/issues/september-10-2019/confronting-the-criticisms-facing-watson-for-oncology/`.

Chouldechova, A., Benavides-Prado, D., Fialko, O., & Vaithianathan, R. (2018). A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. 81, 134–148.

Versie 1.0

Cruz, J. A. & Wishart, D. S. (2006). Applications of machine learning in cancer prediction and prognosis. *SAGE*, 2.

DataNews (2021). Machine learning verlost 112 van broekzakbellers. Geraadpleegd op 14 december 2021 via https://datanews.knack.be/ict/magazine/machine-learning-verlost-112-van-broekzakbellers/article-normal-1781301.html.

Denning, P. & Tedre, M. (2019). *Computational Thinking*. Cambridge, MA: MIT Press.

Dickson, B. (2019). What is Natural Language Processing (NLP)? Geraadpleegd op 31 maart 2021 via https://www.pcmag.com/opinions/what-is-natural-language-processing-nlp.

DWDD (2019). *Mobiel bellen: 1998 vs NU* [Videobestand]. Geraadpleegd op 8 augustus 2020 via youtu.be/mQVkpdzPGtQ.

Eckert, M. (2019). Nip-test screent ook moeder op genetische defecten. *De Standaard*. Geraadpleegd via Gopress op 9 september 2019 via https://www.standaard.be/cnt/dmf20190904_04592057.

ELIZA (2018). *ELIZA conversation* [Afbeelding]. Publiek domein. Geraadpleegd op 28 juli 2019 via https://commons.wikimedia.org/wiki/File:ELIZA_conversation.jpg.

Enix, S. (2019). *Shadow-of-the-Tomb-Raider-010419-002* [Afbeelding]. Publiek domein. Geraadpleegd op 6 september 2019 via https://www.flickr.com/photos/161227653@N02/47513423051/.

Fack, V. (2007). *Algoritmen en Datastructuren*. Acco.

Flynn, J. (2021). Helping actor Val Kilmer reclaim his voice [Blogpost]. Geraadpleegd op 19 juni 2022 via https://www.sonantic.io/blog/helping-actor-val-kilmer-reclaim-his-voice.

Futureism (2017). *This 'Racist soap dispenser' at Facebook office does not work for black people* [Videobestand]. Geraadpleegd op 8 augustus 2020 via https://youtu.be/YJjv_OeiHmo.

Garderen, F. V. (2019). Nog steeds klachten over gebrekkige begeleiding: veel zwangere vrouwen weten niet wat NIPT inhoudt. *De Morgen*. Geraadpleegd 15 september 2019 via https://www.demorgen.be/nieuws.

GRAY (2021). App moet hoofdpijn voorspellen. Geraadpleegd op 24 januari 2022 via https://gray.be/projecten/migraine-wearables.

Güzeldere, G. & Franchi, S. (1995). Dialogues with colorful "personalities" of early AI. *Stanford Hum. Rev.*, *4*(2), 161–169.

Haydon, J. (2010). *Visual Map Of Worldwide Facebook Network* [Afbeelding]. CC BY-SA 2.0. Geraadpleegd op 4 mei 2022 via https://www.flickr.com/photos/johnhaydon/5261841180.

Kabari, L. & Nwachukwu, E. (2012). *Neural Networks and Decision Trees For Eye Diseases Diagnosis*, (pp. 63–84).

Kaushal, A., Altman, R., & Langlotz, C. (2020). Politie positief over inzet AI bij cold cases. Geraadpleegd op 11 december 2021 via https://www.scientificamerican.com/article/health-care-ai-systems-are-biased/.

Knack (2022). Groep wetenschappers en opiniemakers schrijft manifest voor breder coronabeleid. Geraadpleegd op 19 januari 2022 via https://www.knack.be/nieuws/belgie/groep-wetenschappers-en-opiniemakers-schrijft-manifest-voor-breder-coronabeleid/article-news-1821895.html.

Versie 1.0

Koh, D. (2019). Ping An Good Doctor launches commercial operation of One-minute Clinics in China. Geraadpleegd op 31 maart 2021 via https://www.mobihealthnews.com/news/apac/ping-good-doctor-launches-commercial-operation-one-minute-clinics-china.

Lichtveld, R. & van der Werken, C. (1996). Eerste hulp bij verstikking. *Nederlands Tijdschrift Geneeskunde*, *21*(140), 1115–1118.

Lim, M. (2018). History of AI Winters. Geraadpleegd op 27 december 2018 via https://www.actuaries.digital/2018/09/05/history-of-ai-winters/.

Liu, B. (2015). *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

McKenna, M. (2019). Three notable examples of AI bias. Geraadpleegd op 11 december 2021 via https://aibusiness.com/document.asp?doc_id=761095.

Nardone, C. (2007). *Shakey the Robot* [Afbeelding]. CC BY-SA 2.0. Geraadpleegd op 28 juli 2019 via https://commons.wikimedia.org/wiki/File:Shakey_the_Robot_(developed_between_1966-1972_at_SRI_International)_-_Computer_History_Museum_(2007-11-10_23.16.01_by_Carlo_Nardone).jpg.

NJIT IST (2018). ELIZA: a very basic Rogerian psychotherapist chatbot. Geraadpleegd op 8 april 2021 via https://web.njit.edu/~ronkowit/eliza.html.

Norori, N., Hu, Q., Aellen, F. M., Faraci, F. D., & Tzovara, A. (2021). Addressing bias in big data and ai for health care: A call for open science. *Patterns*, *2*(10), 100347.

Ongenae, F. (2021). Project mBrain. Via e-mail en gesprekken.

Orthopedie Roeselare (2021). *Getuigenis na 2 robot geassisteerde knieprotheses via het Joint Care Zorgpad in AZ Delta Roeselare* [Videobestand]. Geraadpleegd op 7 oktober 2019 via https://www.orthopedie-roeselare.be/nl/specialisaties/knie/knieprothese-robot-geassisteerd.

Pratish (2020). Decision Trees for Dummies. Geraadpleegd op 11 december 2021 via https://towardsdatascience.com/decision-trees-for-dummies-37dbee6c7169.

PwC (2018). *Benchmark studie over artificiële intelligentie, uitgevoerd in opdracht van het Departement Economie, Wetenschap en Innovatie van de Vlaamse Overheid. Rapport PwC*. Geraadpleegd op 29 mei 2019 via https://www.ewi-vlaanderen.be/sites/default/files/bestanden/departement_economie_wetenschap_en_innovatie_-_benchmark_studie_over_artificiele_intelligentie.pdf.

Ramos, H. (2019). *David Bowie predicted in 1999 the impact of the Internet in BBC* [Videobestand]. Geraadpleegd op 14 december 2021 via https://youtu.be/8tCC9yxUIdw.

Sciensano (2021). Beslisbomen van de maatregelen. Geraadpleegd op 26 november 2021 via https://covid-19.sciensano.be/nl/procedures/beslisboom.

Serengil, S. (2018). A Step by Step CART Decision Tree Example. Geraadpleegd op 14 december 2021 via https://sefiks.com/2018/08/27/a-step-by-step-cart-decision-tree-example/.

Shin, T. (2020). Real-life Examples of Discriminating Artificial Intelligence. Geraadpleegd op 11 december 2021 via https://towardsdatascience.com/real-life-examples-of-discriminating-artificial-intelligence-cae395a90070.

Shoemaker, C., Chen, S., & Ruiz, C. (2001). CS4341 Introduction to Artificial Intelligence. Geraadpleegd op 11 december 2021 via http://web.cs.wpi.edu/~cs4341/D01.

Versie 1.0

Sprague, N. (2021). CS 444: Artificial Intelligence. Geraadpleegd op 22 november 2021 via https://w3.cs.jmu.edu/spragunr/CS444/activities/trees_bagging/.

Stock, M. (2017). Sorting laundry in linear time [Blogpost]. Geraadpleegd op 21 juni 2022 via https://michielstock.github.io/posts/2017/2017-12-20-Algorithms/#sorting_laundry_in_linear_time/.

Strong, K. (2021). Computer program aimed at predicting child abuse debated. Geraadpleegd op 11 december 2021 via https://apnews.com/article/child-abuse-b9e948bba66efe03b9a06a023f2e00b8.

Taira, B., Kreger, V., Orue, A., & Diamond, L. (2021). A Pragmatic Assessment of Google Translate for Emergency Department Instructions. *Journal of general internal medicine*, *36*(11), 3361–3365.

Tegmark, M. (2017). *LIFE 3.0*. New York, Verenigde Staten: Vintage Books, Penguin Random House.

University of Birmingham (2012). Worksheet for Decision Tree Learning. Geraadpleegd op 14 december 2021 via https://www.studocu.com/en-gb/document/university-of-birmingham/machine-learning-extended/worksheet-decision-tree-learning/643682.

Vaithianathan, R., Kulick, E., Putnam-Hornstein, E., & Prado, D. B. (2019). Allegheny Family Screening Tool: Methodology, Version 2. Geraadpleegd op 11 december 2021 via https://www.alleghenycountyanalytics.us/wp-content/uploads/2019/05/Methodology-V2-from-16-ACDHS-26_PredictiveRisk_Package_050119_FINAL-7.pdf.

Van Hoecke, S., Ongenae, F., Paemeleire, K., & Vandenbussche, N. (2020). App moet hoofdpijn voorspellen. *EOS Wetenschap Special*, Technologie en gezondheid, 25.

Vlaamse Wetenschappelijke Vereniging voor Jeugdgezondheidszorg (2021). Beslisboom COVID-19 voor leerlingen in het kleuter, lager en secundair onderwijs. mag deze leerling naar school? Geraadpleegd op 26 november 2021 via https://www.vwvj.be/sites/default/files/infectieziekten/infectieziekten_-_nieuws/beslisboom_covid-19_leerlingen_ko_lo_so_20210831r2.pdf.

VO-Raad (2020). *Beslisboom 12+* [Afbeelding]. Geraadpleegd op 18 maart 2022 via https://www.antoonschellenscollege.nl/Op-t-Antoon/Nieuwsarchief/art/693920.

Wikipedia (2019). Shakey the robot. Geraadpleegd op 26 juli 2019 via https://en.wikipedia.org/wiki/Shakey_the_robot.

Woebot Health (2021). Welcome to the future of mental health. Geraadpleegd op 9 april 2021 via https://woebothealth.com/.

Wolfram, S. (2015). Untangling the Tale of Ada Lovelace [Blogpost]. Geraadpleegd op 21 september 2019 via https://blog.stephenwolfram.com/2015/12/untangling-the-tale-of-ada-lovelace/.

Xu, M., Tam, B., Thabane, L., & Fox-Robichaud, A. (2015). A protocol for developing early warning score models from vital signs data in hospitals using ensembles of decision trees. *BMJ open*, *5*, e008699.

Zorgnet-Icuro (2021a). Een langdurig hoog plateau is absoluut te mijden. Geraadpleegd op 14 januari 2022 via https://www.zorgneticuro.be/nieuws/een-langdurig-hoog-plateau-absoluut-te-mijden.

Zorgnet-Icuro (2021b). Open brief: De zorg is zeer bezorgd. Geraadpleegd op 14 januari 2022 via https://www.zorgneticuro.be/nieuws/open-brief-de-zorg-zeer-bezorgd.

Versie 1.0

INDEX

112, 18

abstractie, 65
AI-winter, 19
algoritme, 22–24, 47, 65, 86, 93
automatisatie, 116

basisconcepten, 63, 95
beslissingsboom, 47, 79, 81
bias, 22
binaire boom, 42, 43, 45, 47, 68
Bingli, 19
blad, 40
boog, 29
boom, 39
buur, 29

categorisch, 86, 103
classificatie, 17, 47–49
computationeel denken, 63, 95
COVID-19, 26, 48, 49
cyborg, 26
cykel, 34, 39

datagebaseerd, 15, 16
decompositie, 65
definitie, 63
diepte, 43, 110
dilemma, 25

eerste hulp, 18, 49
effector, 113
ethiek, 25, 27
exoskelet, 26

Facebook, 33
feedback, 113
formaat, 14

geldbejag, 26
gericht, 36
gewortelde boom, 47
gini-index, 73, 74, 77, 80, 84, 85, 104
graad, 29
graaf, 29
gretig algoritme, 68, 86, 95
gulzig algoritme, 68, 86, 95

heterogeen, 67
homeostase, 113, 114
homogeen, 67
hypoglykemie, 49

input, 47
Instagram, 36

ja-neevraag, 48, 81, 85, 93

keelkanker, 19
kennisgebaseerd, 15, 16
kinderbescherming, 23
kindknoop, 42
klasse, 47, 48, 79, 80, 93
knieoperatie, 26
knoop, 29, 40
kyfose, 5, 110

lerend systeem, 15
Locked-insyndroom, 65
logica, 58

machinaal leren, 16
machine learning, 16
migraine, 49

NIPT, 25

oogziekte, 54
operatierobot, 17, 26
ouderknoop, 42

pad, 34, 47
parabool, 73, 74
patroonherkenning, 65
programma, 68

regelgebaseerd, 15
regressie, 17

samenhangend, 38
scheiding, 68, 77, 79, 80, 83, 86
sensor, 113
server, 99
sollicitatie, 23
soorten AI, 15
spreiding, 68, 80, 82, 83, 86, 93
spreidingsparameter, 73
Sprouts, 29
suikerziekte, 49

tak, 40
technologie, 21, 25, 26, 63
terugkoppeling, 113
transparant, 16
transparantie, 22

verdeel-en-heersalgoritme, 86, 94
verstikking, 49
vooringenomenheid, 22–24
voorspellen, 16

wandeling, 34
webbrowser, 99
Woebot, 19, 26
wortel, 40

Versie 1.0

In het project 'AI in de Zorg' maken leerlingen kennis met een beslissingsboom, een techniek uit machinaal leren, die veel gebruikt wordt in de zorgsector. De principes van zo'n beslissingsboom zijn al te begrijpen met de leerstof van de tweede graad van het secun+ dair onderwijs.

De Early Warning Score (EWS) is een richtlijn die internationaal wordt toegepast in ziekenhuizen om de gezondheid van een patiënt te kunnen inschatten. De EWS is gebaseerd op de vitale functies: bloeddruk, lichaamstemperatuur, hartslag, ademhalingsfrequen+ tie, alertheid en zuurstofsaturatie. In het project 'AI in de Zorg' bekijken leerlingen hoe de EWS gebruikt wordt in de praktijk. Bovendien gaan ze na hoe een computer op basis van kunstmatige intelligentie automatisch het risiconiveau van een patiënt kan inschatten.

Beslissingsbomen uit de zorgsector zijn maatschap+ pelijk relevant en vormen de ideale context voor een STEM-project dat leerlingen concepten van artificiële intelligentie bijbrengt.

9 789464 441024