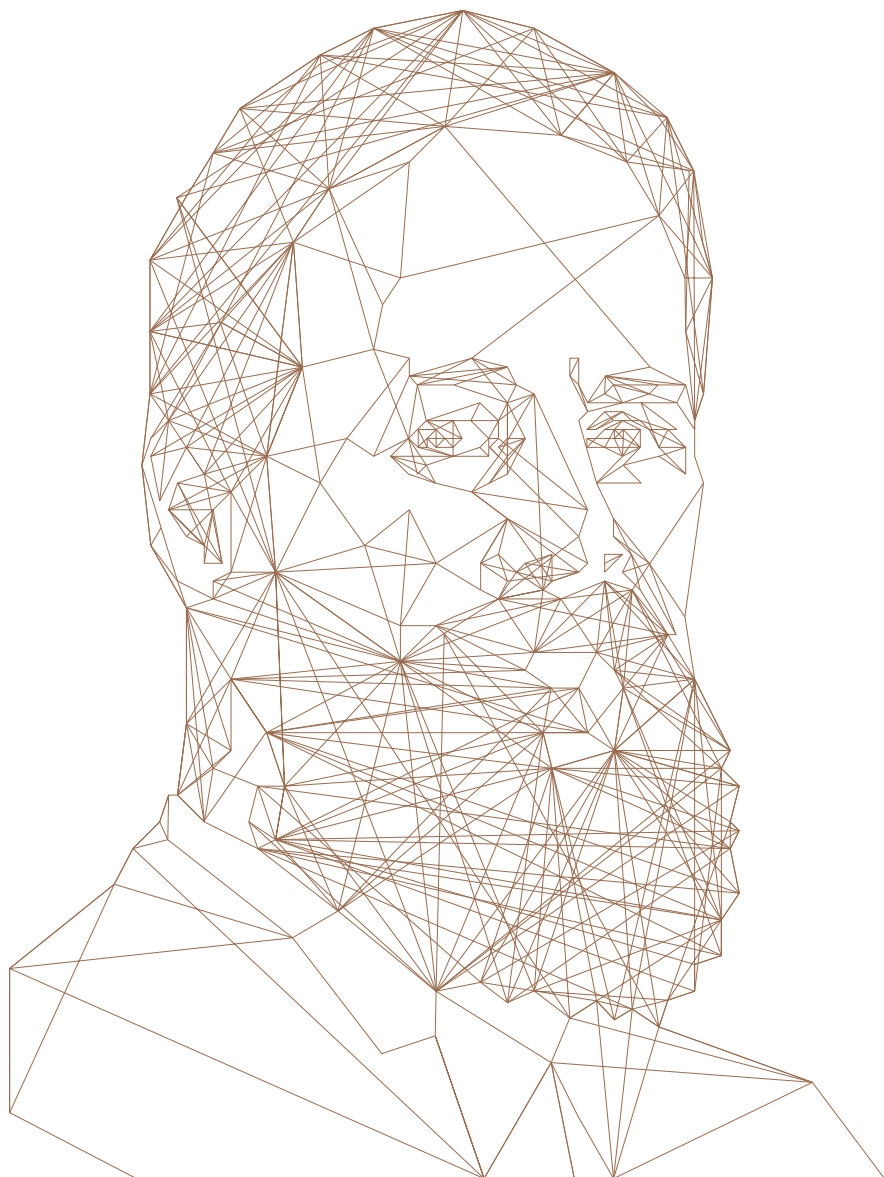


Algoritmisering, wen er maar aan!



Leven, werken en geld verdienen
met kunstmatige intelligentie

Jim Stolze

Boom

Algoritmisering, wen er maar aan!

**Leven, werken en geld verdienen met
kunstmatige intelligentie**

Jim Stolze

Boom

INHOUD

Voorwoord 7

Proloog 11

1 ALGORITMISERING 19

Domme machines 22

Slimme modellen 27

2 MACHINE LEARNING 33

Leren van data 34

Praktijk: Bij Netflix 43

Praktijk: Bij Google 52

3 VERMIJD DE BLACK BOX 57

Algorithmic bias 58

Vooroordelen 59

Responsible AI 62

Moraliteit 66

4 NEURALE NETWERKEN 73

Hoe werkt het? 74

Praktijk: Fraudepreventie 78

5 DEEP LEARNING 85

- Beeldherkenning 88
- Praktijk: In de zorg (1) 93
- Pas op voor de hype 95
- Praktijk: In de zorg (2) 99

6 GEZICHTSHERKENNING 105

- Camerabeelden 106
- Praktijk: Je gezicht spreekt boekdelen 108

7 SPRAAKHERKENNING 111

- Natural language processing 112
- Praktijk: De Helpdesk 116

8 AI IN BEDRIJF 125

- Deep learning op het spoor 126
- Deep learning op de veiling 128
- Algoritmen in de bouw 131

6

9 AI BIJ DE OVERHEID 135

- Ambtenaar 3.0 136
- Digitale speurneuzen 138
- Chinese toestanden 140

10 ROBOTANGST 149

- Autonome wapens 150
- Kan een robot zelf denken? 152

11 DE TOEKOMST 159

- Mist Nederland de boot? 160
- Waar zit het plafond? 165
- En de arbeidsmarkt? 166

BRONNEN 173

ALGORITMISERING

Blaricum, 16 september 2017

Mijn zoon Max (veertien jaar) komt opvallend vrolijk uit school. Voor het eerst loopt hij niet linea recta naar de Playstation op zijn kamer. Hij vertelt enthousiast over een masterclass die hij afgelopen week elk laatste uur heeft gevolgd. 'Pap, hebben wij hier een kubus liggen?' Dat laat ik me geen twee keer zeggen. Op zolder vind ik mijn oude Rubiks-kubus. Volledig door elkaar gehusseld. Het is mij in de jaren tachtig nooit gelukt om de kubus op te lossen. Verder dan twee hele vlakken ben ik nooit gekomen.

Max laat vol trots een diploma zien. Hij kan de kubus wél helemaal oplossen en was de snelste van zijn klas: 1 minuut en 6 seconden. Onder de indruk vraag ik hem hoe hij dat voor elkaar krijgt. Het blijkt dat de leraar in kwestie aan de klas had uitgelegd dat iedereen de kubus kan oplossen. Zolang je maar een bepaald stappenplan volgt.

1. Top Cross
2. Ri Di R D
3. UR Ui Ri Ui Fi U F Ui Fi U F U R Ui Ri
4. F R U Ri Ui Fi
5. R U Ri U R U U Ri
6. U R Ui Li U Ri Ui L
7. Ri Di R D

20

In feite had de docent de klas laten kennismaken met een algoritme. Max begon dat woord ook te gebruiken toen hij mij uitlegde dat je aan zeven algoritmen eigenlijk genoeg had om de kubus op te lossen. 'Als je bijvoorbeeld dit blokje op de rechterbovenhoek naar linksachter-onder wil krijgen, doe je gewoon U-U-R-R-M-B.' Max kon zonder kijken de stappen uitvoeren en toverde het vakje naar de andere kant, zonder de rest van zijn vlakken in de war te schoppen. Hij had alle zeven algoritmen uit zijn hoofd geleerd en was daardoor de snelste van de klas geworden.

Onthoud voor nu: een algoritme is niets ingewikkelds. Het is gewoon een stap voor stap procedure om iets voor elkaar te krijgen.

Een andere metafoor is koken. Iedereen heeft wel eens met een kookboek in zijn handen gestaan of op internet een recept opgezocht. Eerst wordt er gezegd welke ingrediënten er nodig zijn: 2 eieren, 2 tomaten en 1 plakje kaas. Dit zijn de data. Daarna volgt de bereidingswijze: Breek de eieren en gooi ze in een pan met boter. Snijd in de tussentijd de tomaat en rasp de kaas. Gooi de tomaat en de kaas in de pan en laat twee minuten meebakken. Dit is het algoritme voor een lekkere omelet.

Als je er zo over nadenkt, bestaat het hele leven uit algoritmen. Je hebt misschien wel een vast ritueel bij het opstaan of je rijdt altijd

via een bepaalde route naar je werk? Algoritmen helpen ons de dag door. We hebben algoritmen nodig om orde te scheppen in de chaos van onze wereld.

Het woord algoritme komt van de naam van een Perzische geleerde die rond 720 na Christus aan het hof van Bagdad de functie van hoofd-astronoom bekleedde. Zijn naam was Mohammed Al-Chwarizmi. In de twaalfde eeuw werd een aantal boeken van hem in het Latijn vertaald. Zo werd zijn naam verbasterd tot Al Goritmi. Een aantal van zijn boeken heeft ook de basis gelegd voor de moderne algebra. Het wiskundige principe van een algoritme heeft Al-Chwarizmi overigens niet uitgevonden. Al ver voor het jaar nul waren er geleerden die stapsgewijze formules hadden ontworpen.

Een van de oudste algoritmen stamt uit de derde eeuw voor Christus. De Griek Eratosthenes stelde toen een eenvoudig algoritme op voor het vinden van priemgetallen. Hij schreef alle getallen tussen 1 en 31 op een stuk papier. Daarna omcirkelde hij het cijfer 2 en vervolgens streepte hij alle veelvouden (4, 6, 8 et cetera) door. Daarna omcirkelde hij cijfer 3 en streepte daarvan ook de veelvouden van door (6, 9, 12 et cetera). Dit herhaalde hij steeds voor de opvolgende getallen die nog niet doorgestreept waren. Het snelle resultaat is een mooie lijst met priemgetallen. Dit principe is de wiskundeboeken ingegaan als de zeef van Eratosthenes.

In de computerwetenschap was het Ada Lovelace die in 1892 het eerste algoritme voor een computer ontwierp. De computer zelf was nog niet uitgevonden. Het was slechts een theoretisch concept waaraan Charles Babbage zijn leven wijdde. Onder de noemer *analytical machine* had hij alles uitgedacht waarover een moderne computer zou moeten beschikken; het ontbrak hem alleen aan de technologische en de financiële middelen om de machine ook daadwerkelijk te bouwen. Dat gebeurde uiteindelijk in 1921 (na zijn dood), toen een kleinzoon aantoonde dat de ideeën van zijn grootvader hun tijd *letterlijk* te ver vooruit waren. De oude Babbage realiseerde zich in die tijd dat de machine ook software nodig had. Door een speling van het lot ontmoette hij Ada Lovelace. De dochter van de beroemde dichter Byron had in haar kinderjaren al uitzonderlijke belangstelling getoond voor wiskunde en wetenschap. In plaats van zich te ontwikkelen in de kunsten koos zij een carrière als wetenschapper. Toen zij in 1888 een

publicatie van Babbage vertaalde, was hij zo onder de indruk van de door haar toegevoegde passages dat hij met haar wilde samenwerken. Lovelace doorgrondde de machine en schreef een algoritme om zogenoemde Bernoulli-getallen te laten berekenen. Een typische klus voor een computer en een opwindend begin van een nieuwe revolutie!

DOMME MACHINES

Ada Lovelace schreef een algoritme waarmee de machine van Babbage complexe berekeningen kon uitvoeren. Dat was een vaardigheid die voorheen was voorbehouden aan mensen. Tot dan toe was bijvoorbeeld het telraam slechts een hulpmiddel. De mens voerde de berekening uit. Net zoals een hamer niet zelf een spijker in het hout slaat. In de wereld van Babbage en Lovelace was het echter de machine die de berekening uitvoerde. Creëerden zij daarmee ook het eerste intelligente systeem?

22

Dergelijke vragen lokken meestal een oneindige discussie uit. Wetenschappers zijn het nog lang niet eens over wat intelligentie eigenlijk is. Laat staan over hoe je verschillende vormen daarvan met elkaar kunt vergelijken. Zo'n vraag is niet alleen heel subjectief, ze is ook zeer tijdgebonden. Wanneer je aan iemand die leeft in 1930 een TomTom-kastje uit 2005 zou demonstreren, zou hij vast en zeker onder de indruk zijn van de intelligentie die in dat zwarte doosje zat opgesloten. Je kinderen zullen dubbelklappen van het lachen. Vergelijken met de zelfrijdende Tesla's die zij kennen, waren die TomTom-kastjes maar domme dingen.

Terzijde: ik tik dit hoofdstuk op de nieuwste MacBook, met de nieuwste versie van Pages en voorzien van Siri. Ondanks deze wonderen van de techniek en alle kunstmatige intelligentie die Apple in deze computer heeft gestoken, corrigeert het apparaat mijn 'zelfrijdende Tesla's' heel eigenwijs naar 'zelfrijdende Tesla's'. Ondanks de belofte van intelligentie ben ik nog veel tijd kwijt met het corrigeren van de autocorrectie.

Voor nu hanteer ik de simpelste definitie van intelligentie: het vermogen om problemen op te lossen.

BRUTE KRACHT

Tijdens een vakantie in Turkije kreeg ik het een keer Spaans benauwd. Mijn andere zoon (Felix, 12 jaar) had uitgevogeld hoe hij het cijferslot van onze koffer kon aanpassen. Dat klinkt natuurlijk heel leuk, totdat je erachter komt dat hij zelf niet meer weet wat de nieuw ingestelde code is. Nadat ik op goed geluk een aantal codes (000, 123, 999) had geprobeerd, drong het tot mij door dat ik die koffer voorlopig niet zou kunnen openmaken.

Ik woog mijn twee opties op een schaal van tijdsinspanning:

OPLOSSING	CONSEQUENTIE	TIJD
Naar een winkel in de stad gaan waar ze koffers verkopen. Daar vervolgens vragen of ik van hen een TSA-sleutel (universeel) zou mogen lenen.	Van het vakantiehuisje naar de stad en weer terug zou veertig minuten reistijd kosten. De tijd die we in de winkel zouden doorbrengen niet meegerekend.	40 minuten
Met de koffer tussen mijn benen op een stoel gaan zitten en elke code tussen 001 en 999 uitproberen.	Ik moest aan een of twee radertjes per combinatie draaien en daarna op de knop van het slot drukken. Ik kon er drie in elke vijf seconden doen. 999 combinaties, zou dus 333 x 5 seconden = 1565 seconden = 27 minuten werk opleveren.	27 minuten

De conclusie was snel getrokken: het was efficiënter om handmatig alle codes af te gaan. Mijn mazzel was dat de code ongeveer halverwege (577) zat. De blaar op mijn rechterduim nam ik op de koop toe.

De benadering van dit probleem lijkt erg op een algoritme. Ik zou het in pseudocomputercode kunnen uitschrijven:

- Als (slot = geblokkeerd) probeer code = 000
- Als (slot = geblokkeerd) probeer code = 001
- Als (slot = geblokkeerd) probeer code = 002
- Als (slot = geblokkeerd) probeer code = 003
- ...
- Als (slot = geblokkeerd) probeer code = 999

Indien je een beetje ervaring met programmeren hebt, weet je dat bovenstaande code allesbehalve efficiënt is. Een programmeur zou nooit duizend softwareregels gebruiken voor een dergelijk probleem. Een simpele lus (in het Engels: *loop*) zou volstaan:

```
code = 000
```

```
Zolang (slot = geblokkeerd) probeer (code = code + 1)
```

Deze oplossing is een voorbeeld van *brute force*: domme maar overweldigende rekenkracht. Maar is het intelligent? Stel, ik ben de *viercijferige* code van mijn telefoon vergeten. En laten we voor het gemak aannemen dat er geen beveiliging op zit van een maximaal aantal foute pogingen. Uitgaande van het kofferprobleem kan ik inschatten dat het uitproberen van alle mogelijke combinaties (tien-duizend stuks) mij maximaal 270 minuten, oftewel 4,5 uur zou kosten. Daar heb ik geen zin in, dus zou ik graag gebruikmaken van een geautomatiseerde oplossing.

24

Een computer zou dit probleem kunnen oplossen met een algoritme dat gebruikmaakt van de randomfunctie (het genereren van een willekeurig getal) in een lus, net zolang totdat de combinatie is geraden. In pseudocomputercode:

```
Zolang (geblokkeerd) probeer {random code}
```

De computer zou dus starten met een willekeurige viercijferige code (bijvoorbeeld 2763). En als die niet de oplossing zou bieden, probeert hij een andere willekeurige code (bijvoorbeeld 8872). Hij blijft het proberen, net zolang totdat de code de blokkade opheft. In het Engels staat dit bekend als een *if-then-else loop*. Het is de basis van de meeste computerprogramma's. In het uitzonderlijke geval dat hij 9999 keer moet gokken, zou een simpele computer dit probleem alsnog in enkele seconden hebben opgelost. Bij mensen meten we de intelligentie vaak in IQ; bij computers gaat het dus eigenlijk om het aantal berekeningen per seconde.

Daarbij, een mens zal nooit 9999 verschillende combinaties gaan uitproberen. Daarvoor hebben we het geduld niet. Gebrek aan geduld is een cadeautje van de evolutie dat ervoor zorgt dat we onze hersencapaciteit zo nuttig mogelijk inzetten. Doordat wij de luxe niet heb-

ben om brute force te kunnen gebruiken, zullen we eerder intuïtief te werk gaan. Daar waar de computer koos voor een random aanpak, zou mijn zoontje Felix eerder de standaardcode van alle telefoons proberen (0000). Als dat niet lukt misschien mijn geboortjaar (1973), maar na een dozijn pogingen zal ie verveeld en gefrustreerd de handdoek in de ring gooien. Een computer niet! Die blijft pogingen wagen, net zolang totdat de code is gevonden. De enige beperking is de hoeveelheid tijd die beschikbaar is.

INTELLIGENTE MACHINES?

Het kraken van codes kan worden gezien als een belangrijke aanjager van kunstmatige intelligentie. Een belangrijke grondlegger van het vakgebied is dan ook Alan Turing. Dankzij zijn ideeën konden de Engelsen rond 1944 een antwoord vinden op de Duitse codeermachine (Enigma). Wat de Enigma deed, leek eigenlijk best wel op de manier waarop we nu onze e-mails of WhatsApp-berichten beschermen. Ze husselden de letters door elkaar, en alleen door een bepaalde sleutel kon de boodschap ontcijferd worden. Het lukte de beste codekrakers van Engeland maar niet om de Duitse boodschappen op tijd te ontcijferen. Denk nog even terug aan de Rubiks-kubus. Menselijke hersenen kunnen maar een beperkt aantal berekeningen per seconde maken; zo iets moet je overlaten aan een machine. Iedereen die de film *The Imitation Game* heeft gezien, weet dat Alan Turing precies zo'n machine wist te bouwen. Een machine die in korte tijd veel variaties kon uitproberen, net zolang totdat de juiste code was gevonden.

Deze overwinning luidde in de jaren vijftig het vakgebied van de kunstmatige intelligentie in. De term als zodanig bestond nog niet. Het was computerwetenschapper John McCarthy die samen met Marvin Minsky het initiatief nam om hier een wetenschappelijk zomerproject van te maken.

'Wij stellen een twee maanden durende studie naar kunstmatige intelligentie voor — uitgevoerd door tien onderzoekers — in de zomer van 1956 op Dartmouth College in Hanover, New Hampshire.'

~ John McCarthy (1955)

Met die woorden legde McCarthy de basis voor een onderzoeksveld dat tot op de dag van vandaag nog even boeiend, relevant als 'nog lang niet af' is. Het oorspronkelijke voorstel is te bewonderen in de archieven van Dartmouth College en bij de universiteit van Stanford. McCarthy en zijn collega's benoemden de volgende aspecten van kunstmatige intelligentie:

26

1. **Automatic computers.** Voortbouwend op het werk van Alan Turing, realiseerden de onderzoekers zich dat de huidige prestaties van computers nauwelijks in de buurt komen van die van het menselijke brein. De grootste beperking is echter niet de hardware, maar ons onvermogen om software te schrijven die het huidige potentieel benut.
2. **Natural language processing.** De manier waarop mensen denken en redeneren kan voor een groot deel toegerekend worden aan de wijze waarop wij woorden gebruiken, ordenen en betekenis toekennen. Computers zullen op termijn gewonemensentaal moeten kunnen begrijpen.
3. **Kunstmatige neuronen.** Is het mogelijk om inzichten uit de neurologie toe te passen in de computerwetenschap? Bijvoorbeeld door neuronen te simuleren in een machine?
4. **Efficiëntie.** Een probleem oplossen door alle mogelijke oplossingen achter elkaar uit te proberen (brute force) is verre van efficiënt. Intelligentie betekent de juiste benadering kunnen kiezen bij elk voorkomende complexe situatie.
5. **Machine learning.** Een intelligente machine zal het vermogen moeten hebben om zelf te leren, zichzelf te verbeteren.
6. **Abstraheren/ generaliseren.** Een intelligente machine zal conclusies moeten kunnen trekken op basis van nieuwe gepercepieerde informatie.
7. **Onvoorspelbaarheid.** Kunnen we een machine minder voorspelbaar maken? Wat is de relatie tussen willekeurige keuzes en creativiteit?

De beschrijving van wat een computer intelligent maakt, wijkt niet zoveel af van hoe we er nu naar kijken. Wat de onderzoekers toen nog niet konden voorzien was de enorme rekenkracht die we tot onze beschikking zouden krijgen. Anno 2018 gaat intelligentie bij computers vooral over het volgende: hoeveel voorbeelden kun je

een systeem laten zien, zodat het rekenkracht en statistische modellen kan gebruiken om het foutpercentage van voorspellingen zo laag mogelijk te krijgen?

Als er op die exponentiële schaal dus iets verdubbelt, is dat het aantal berekeningen per seconde, en daarmee de nauwkeurigheid van de voorspellingen die een computer doet. Over de ontwikkeling van emotionele intelligentie, intuïtie of 'begrip' zegt het (nog steeds) niets.

SLIMME MODELLEN

Vandaag de dag is er bijna geen bedrijf te vinden dat niet zegt in een 'digitale transformatie' te verkeren. De CEO brengt het vaak op een stoere, geruststellende manier. Aandeelhouders en deskundigen weten wel beter. Zodra een bedrijf zegt prioriteit te geven aan digitale transformatie, zegt het eigenlijk iets anders. Laat ik het eens vertalen.

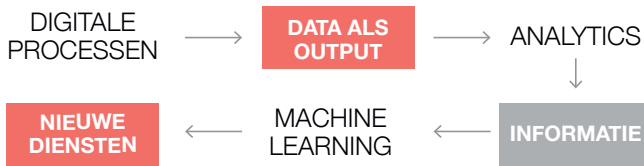
'Wij zijn ons ervan bewust dat de wereld buiten onze organisatie sneller draait dan bij ons binnen. En door te zeggen dat we in digitale transformatie verkeren, hopen we dat u niet aan ons vraagt wat we hieraan gaan doen. Want we hebben eigenlijk geen idee.'

Organisaties die alleen bezig zijn met digitalisering, jagen hun eigen staart achterna. In het Engels is het mooier: *they're spinning their wheels*.



Vaak zijn zulke bedrijven bezig om bestaande processen om te zetten naar een digitaal equivalent. 'Vroeger schreven onze klanten een brief als ze wilden opzeggen. Tegenwoordig kan dat met een formulier op onze website.' Wow ... de digitale transformatie is voltooid. Nee toch? Wat is er nu echt veranderd? Eigenlijk niet veel.

Het is hetzelfde dat Free Record Shop eerst lp's verkocht en daarna cd's. Dat heeft ze niet kunnen redden van de ondergang. Digitalisering is niet het eindpunt. Het is juist de start! Zodra iets digitaal is, verliest het de beperkingen van de fysieke wereld. Opeens kun je kosteloos distribueren. Je kunt kopieën delen met collega's. Hun vragen om hun data ook te delen. Je kunt analyses uitvoeren. Je geeft betekenis (labels) aan data. Dit is de start van algoritmisering.



28

In de wereld van algoritmisering had Free Record Shop ook prima kunnen bestaan. Er zijn immers nog steeds veel mensen die muziek willen luisteren en daar geldt voor overhebben. Laten we daarom eens kijken naar een bedrijf dat hiermee wél succesvol is. Spotify ziet digitalisering niet als eindpunt, maar als startpunt. Dus in plaats van mp3-files te verkopen in een online winkel, bieden ze de muziek streaming aan. De distributie is gratis. En door labels aan de data te geven, weet Spotify precies wie op welk moment naar welke muziek luistert en wat hij of zij daarvan vond. Op die manier vertaalt Spotify de data naar inzichten. Deze fase kunnen we simpelweg *analytics* noemen. Het is een bekend grapje onder datawetenschappers dat de meeste organisaties *data rich* maar *information poor* zijn.

Zodra je de eerste patronen in de datasets hebt gevonden, kun je misschien een *machine learning* algoritme inzetten. Wat dat precies is, komt in het volgende hoofdstuk aan bod. Voor nu zou je kunnen denken aan drie jaar historische data van Spotify-gebruikers. Is er écht een verschil tussen zo'n database met wie wat heeft geluisterd en een met wie welk DNA heeft? Als 23andMe erin slaagt om mijn vijftien neven en nichten te vinden, dan kan Spotify mij toch zeker wel voorstellen aan een aantal nieuwe liedjes?

En dat is precies wat hun dienst Discover Weekly doet. Als ik Spotify-gebruikers spreek en ik noem Discover ... verschijnt er een glimlach op hun gezicht. Op internet zijn zelfs mensen te vinden die zeggen: 'If Discover Weekly was a person, I would marry it.' Het is Spotify gelukt om iets te bouwen wat sommige mensen beter kent dan ze zichzelf kennen. Als je hun playlist aanzet, word je automatisch verrast. Je hoort muziek die je nog niet kende, maar die helemaal bij je past. Hoe ze dat doen? Jawel, door algoritmisering.

Het is de reden dat ik met plezier negentien euro per maand betaal voor Spotify Premium. Beste koop ooit! Mijn twee pubers hebben

toegang tot alle muziek van de wereld zonder dat ze daarmee mijn playlist in de war schoppen. Ik zit heus niet te wachten op Boef of Ronnie Flex in mijn Discover Weekly. Spotify laat zien dat data niet het bijproduct zijn (zoals bij digitalisering), maar dat ze hét product zijn. Ze bouwen een businessmodel op een algoritme. Net als Google dat heeft gedaan op zijn database met zoekalgoritmiëk.



Spotify, 23andMe en Google is het gelukt om hun data door middel van algoritmen om te zetten in een exponentieel businessmodel: de relatieve kosten van een nieuwe klant vallen in het niet bij de opbrengsten. Neem eens een moment voor jezelf. Wat ben je de afgelopen jaren aan het digitaliseren geweest? Heb je historische data opgebouwd? Welke patronen zouden daarin te vinden zijn? Is het mogelijk om de Discover Weekly van jouw branche op de markt te brengen?

NIEUWE KANSEN, NIEUWE BEDREIGINGEN

Daarnaast brengt algoritmisering natuurlijk niet alleen maar geld in het laatje. Nieuwe technologie brengt altijd nieuwe kansen én bedreigingen. Algoritmisering kan ook op een foute manier worden ingezet. Stel je voor dat Spotify zijn algoritmen verkeerd heeft afgesteld. Dat zou kunnen betekenen dat als je één keer *De vlieger* hebt aangeklikt, ze je tot in lengte van dagen blijven lastigvallen met het repertoire van André Hazes. Je bent dan gevangen in je eigen filterbubbel.

Of wat als de algoritmen worden ingezet voor snode plannen? Zou het mogelijk zijn om Spotify-achtige oplossingen in te zetten om een verkiezingsuitslag te beïnvloeden? Dat sommige bezoekers van Facebook compleet andere berichten te zien krijgen dan andere bezoekers? Daarom in dit boek niet alleen praktische voorbeelden van hoe algo-

ritmen in het bedrijfsleven en in de maatschappij worden ingezet, maar ook aandacht voor de ethische kanten van algoritmisering. Wat zijn de grootste zorgen en hoe kun je die wegnemen? Kun je systemen bouwen die niet bijdragen aan een filterbubbel? Kun je privacy inbouwen in de software voordat gebruikers deze gaan gebruiken?

Want het randje is snel opgezocht. Zoals gezegd ontwikkelt technologie zich sneller dan de mens. Zou ik het kunnen accepteren als 23andMe mij een mailtje zou sturen met 'Volgens onze informatie hebt u nog 23 maanden, 2 weken en 4 dagen te leven'?