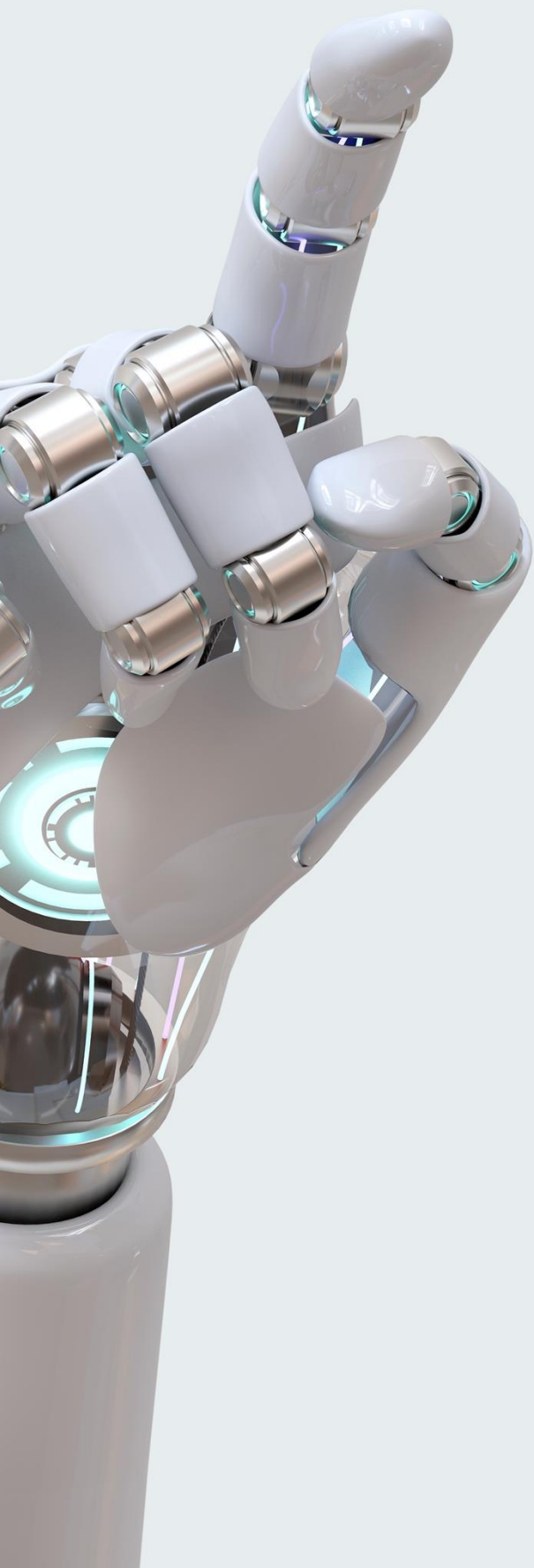


Handboek AI voor het MKB



30 juni 2022

Inhoud

1.	Inleiding.....	3
1.1	Terminologie vaststellen: Artificial Intelligence (AI) en Machine Learning (ML).....	4
2	Het ontwikkelen van een waardevolle AI-propositie.....	5
2.1	Checklist AI-oriëntatiefase.....	5
2.2	Scoping sjabloon.....	6
2.3	AI maturity scan.....	8
2.4	Uitgangspunten voor AI-ontwikkeling.....	9
2.5	Een innovatiemodel voor het ontwikkelen van AI-producten en diensten.....	11
3	Het verzamelen en beoordelen van data.....	23
3.1	Het verzamelen van data.....	24
3.2	Het combineren van gegevens uit verschillende bronnen.....	27
3.3	Het vaststellen van het type, het bereik, en de betrouwbaarheid van gegevens.....	30
3.4	Het verwerken van ontbrekende gegevens.....	32
3.5	Dubbele metingen verwijderen.....	35
3.6	Data verrijken.....	36
3.7	Exporteren data.....	38
4	Data-exploratie en -visualisatie.....	39
4.1	Een statistische analyse: methoden, inzichten en vervolgstappen.....	41
4.2	Data-visualisatie.....	45
5	Het organiseren van het project.....	48
5.1	Een backlog voor AI-projecten.....	48
5.2	Vaardighedenscan voor AI-ontwikkeling in het MKB.....	49
6	Ethische aspecten.....	50
6.1	Checklist ethische aspecten.....	50
6.2	De Ethische Data Assistent (DEDA).....	52
6.3	Ethics Inc., een agile spel om ethische belangen af te wegen.....	53
7	Machine learning modellen.....	54
7.1	Doel.....	54
7.2	Uitleg.....	54
7.3	Model ontwikkelen.....	55
7.4	Model keuze.....	56
7.5	Model validatie.....	57
7.6	Model toepassen.....	58
8	AI en applicatieontwikkeling.....	60
8.1	Software Engineering practices for Machine Learning.....	60
8.2	Kwaliteitskenmerken voor Machine Learning en AI.....	61



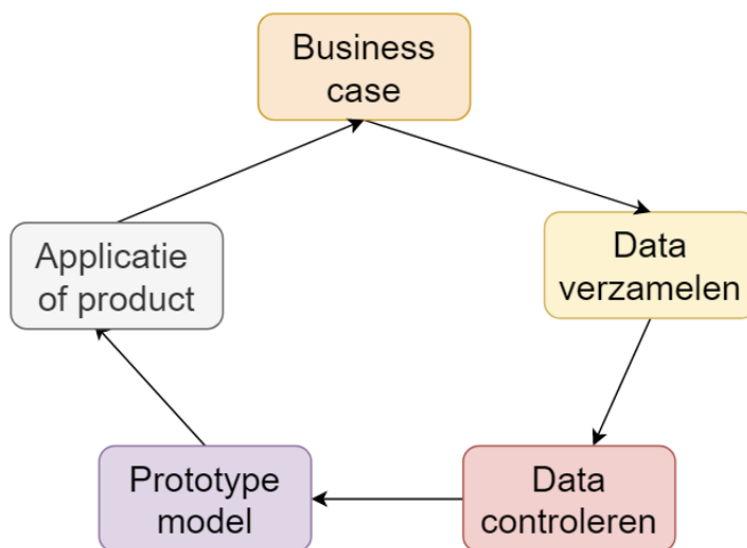
1. Inleiding

Veel bedrijven willen graag de stap maken naar een toepassing van kunstmatige intelligentie (i.e. artificial intelligence, **AI**) in hun bedrijfsprocessen. Dit kan zijn om de interne bedrijfsvoering te verbeteren, maar het kan ook zijn om nieuwe producten binnen het bedrijf te ontwikkelen of nieuwe applicaties voor klanten te maken. Het kan ook zijn dat een bedrijf nog niet kan inschatten welke toepassing van AI zinvol is en dus eerst kennis wil opbouwen door het uitvoeren van een pilotproject.

Het ontwikkelen van een AI-toepassing is niet eenvoudig. Een blik op de inhoudsopgave van dit handboek laat zien dat er veel verschillende disciplines en invalshoeken nodig zijn om tot een succesvolle toepassing te komen. Voor een MKB'er kan dit overweldigend zijn, waar te beginnen?

Het doel van dit handboek is om MKB-bedrijven een verzameling praktische hulpmiddelen aan te reiken waarmee ze naar behoefte aan de slag kunnen gaan bij hun eerste AI-ontwikkelprojecten. Het handboek is nadrukkelijk niet bedoeld als een stappenplan dat van voor tot achter moet worden gevolgd. In plaats daarvan is het handboek een bron van praktische kennis waar afhankelijk van de vragen uit geput kan worden.

AI-ontwikkeling is een cyclisch proces. Hoewel er een zekere volgorde in de hulpmiddelen zijn, moet je telkens schakelen tussen de stappen en weer terugkeren naar eerdere activiteiten. Dat is ook de reden waarom een iteratieve projectaanpak goed past bij dit soort projecten.



Figuur 1.1: AI-ontwikkelen is een cyclisch proces

Een deel van de hulpmiddelen uit dit handboek is binnen het project KI-Agil ontwikkeld. De overige hulpmiddelen hebben we met deskresearch geselecteerd uit andere bronnen. Daarbij is vooral gelet op de meerwaarde voor MKB AI-ontwikkeltrajecten.

1.1 Terminologie vaststellen: Artificial Intelligence (AI) en Machine Learning (ML)

Doel

Helder maken wat we in deze handleiding bedoelen met de termen Artificial Intelligence en Machine Learning.

Uitleg

Er zijn veel definities van Artificial Intelligence en Machine Learning (ML) te vinden. Machine Learning is tegenwoordig dusdanig populair dat het vrijwel synoniem is geworden met AI. In de praktijk worden deze termen daarom vaak door elkaar heen gebruikt.

In deze handleiding is het verschil in de meeste gevallen ook niet zo relevant. In feite komt het erop neer dat we computers regels laten ontdekken op basis van grote hoeveelheden data. Op basis van deze regels kunnen dan door mensen of door de computers zelf beslissingen genomen worden.

Tabel 1.1 geeft een nadere uitleg voor deze gevallen waarin het precieze verschil tussen AI en ML wel van belang is.

Begrip	Betekenis
Artificial Intelligence	Het vertonen van intelligent gedrag door een machine. Over het algemeen geldt dat een systeem dat over Artificial Intelligence (AI) beschikt onderdelen van zijn omgeving kan <i>waarnemen</i> , hierover kan <i>redeneren</i> om tot inzichten te komen, en daarop kan <i>acteren</i> . Het vakgebied omvat onder andere regel-gebaseerde systemen (Als A en niet B, dan C), statistische methoden (bijv. Bayesiaanse statistiek) en <i>Neural Networks</i> die bij Machine Learning worden toegepast. AI wordt in de praktijk onder andere toegepast bij plannen, waarnemen (computer vision), natuurlijke taalverwerking (automatisch vertalen), redeneren (klant-inzicht/aanbevelingen doen) en meer.
Machine Learning	Het automatisch kunnen structureren, reduceren, samenvatten of analyseren van gegevens op basis van (grote hoeveelheden) data. Over het algemeen leert een Machine Learning systeem door zelf verbanden te ontdekken die aanwezig zijn in de data. Op basis van het geleerde kan een machine intelligente beslissingen nemen. Machine Learning is daarom een onderdeel van het vakgebied Artificial Intelligence (AI).
Deep Learning	Een nog weer specifiekere onderdeel van Machine Learning waarbij gebruik gemaakt wordt van complexe kunstmatige neurale netwerken die (losjes) gebaseerd zijn op de manier waarop hersencellen werken. Over het algemeen is voor Deep Learning een grote hoeveelheid data nodig.

Tabel 1.1: Terminologie van AI, ML en Deep Learning

Meer informatie

Zie ook de blogpost '[Wat is Artificial Intelligence](#)' op de website van AI4MKB.

Een andere laagdrempelige bron om meer te leren over wat AI is, is de nationale AI-cursus: <https://app.ai-cursus.nl/>.

2 Het ontwikkelen van een waardevolle AI-propositie

2.1 Checklist AI-oriëntatiefase

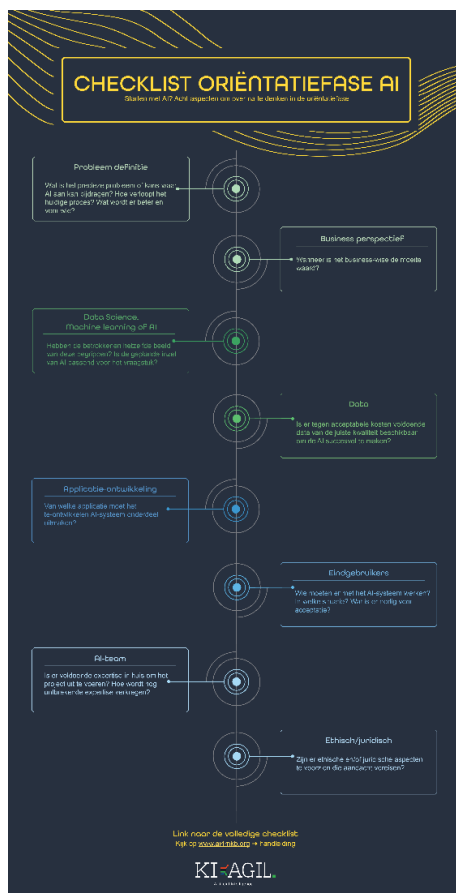
Doel

Het structureren van de eerste oriëntatie op AI door middel van een checklist.

Uitleg

Bij het oriënteren op de inzet van AI in het MKB is het belangrijk om een breed scala aan factoren mee te nemen. De oriëntatiefase kent diverse valkuilen, bijvoorbeeld dat er stappen worden overgeslagen, dat begrippen niet goed gedefinieerd worden of dat er te snel op techniek wordt gefocust. De checklist AI-oriëntatiefase kan behulpzaam zijn om geen onderwerpen missen.

Het is niet de bedoeling van de checklist om alle onderwerpen meteen volledig uit te werken. De verschillende onderwerpen kunnen in een later stadium van het project met behulp van andere hulpmiddelen en checklists verder uitgediept worden.



Figuur 2.1: Infographic checklist Oriëntatiefase

Bijlage

[Checklist AI-oriëntatiefase](#)

Voorbeeld

In de projecten van KI-Agil was de checklist AI-oriëntatiefase de leidraad. De checklist leverde zoveel relevante vragen dat er bij de partners vervolgesprekken nodig waren. Vooral de vragen over de business case bleken best lastig om te beantwoorden.

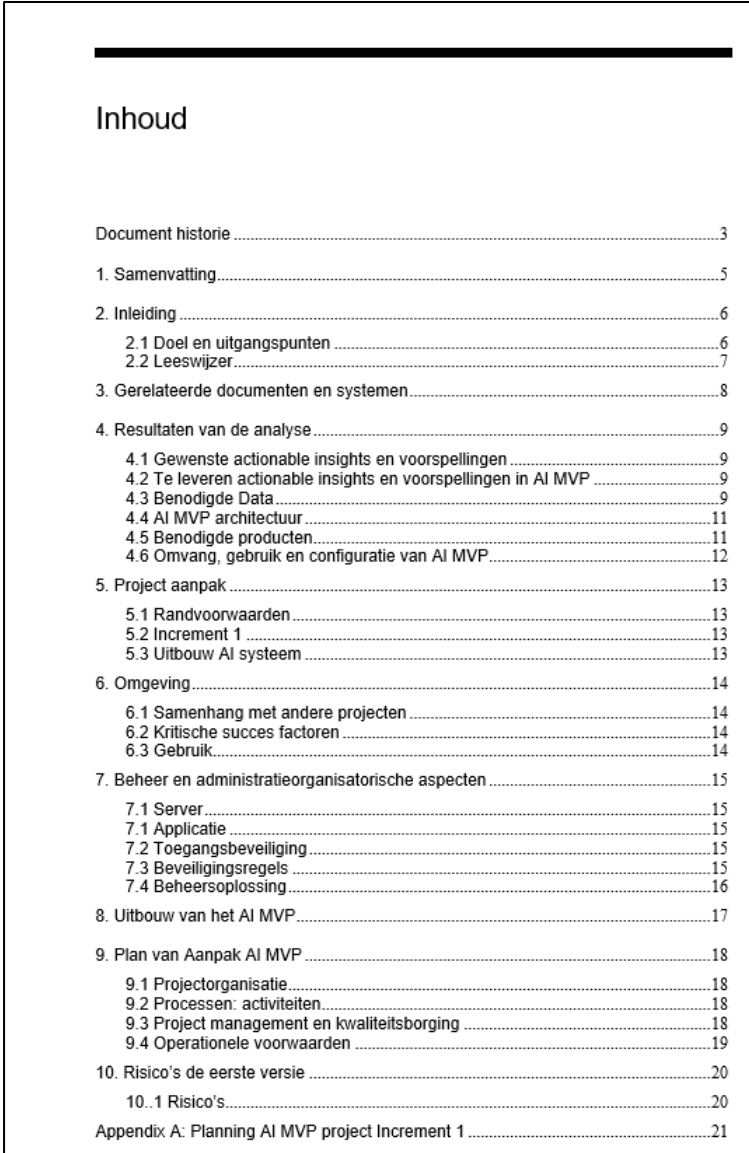
2.2 Scoping sjabloon

Doel

Het doel van dit scoping sjabloon is om een MKB-bedrijf dat wil starten met het ontwikkelen van een AI-toepassing te helpen om het doel en de scope van een eerste 'minimal viable product' (MVP) te bepalen.

Uitleg

Het AI Scoping document is een sjabloon ter ondersteuning van de ontwikkeling van een AI-toepassing. Het invullen van de scoping sjabloon is een eerste stap en helpt het bedrijf om na te denken over fundamentele vragen als "Welk probleem lossen we op?", "Wat zijn de gewenste inzichten die AI zou moeten leveren?", "Is het realistisch haalbaar?". Daarnaast zijn er hoofdstukken over projectaanpak en organisatie.



Inhoud	
Document historie	3
1. Samenvatting.....	5
2. Inleiding	6
2.1 Doel en uitgangspunten	6
2.2 Leeswijzer.....	7
3. Gerelateerde documenten en systemen.....	8
4. Resultaten van de analyse.....	9
4.1 Gewenste actionable insights en voorspellingen	9
4.2 Te leveren actionable insights en voorspellingen in AI MVP	9
4.3 Benodigde Data	9
4.4 AI MVP architectuur	11
4.5 Benodigde producten.....	11
4.6 Omvang, gebruik en configuratie van AI MVP.....	12
5. Project aanpak	13
5.1 Randvoorwaarden	13
5.2 Increment 1	13
5.3 Uitbouw AI systeem	13
6. Omgeving	14
6.1 Samenhang met andere projecten	14
6.2 Kritische succes factoren	14
6.3 Gebruik.....	14
7. Beheer en administratieorganisatorische aspecten	15
7.1 Server	15
7.1 Applicatie	15
7.2 Toegangsbeveiliging	15
7.3 Beveiligingsregels	15
7.4 Beheersoplossing	16
8. Uitbouw van het AI MVP.....	17
9. Plan van Aanpak AI MVP	18
9.1 Projectorganisatie.....	18
9.2 Processen: activiteiten.....	18
9.3 Project management en kwaliteitsborging	18
9.4 Operationele voorwaarden	19
10. Risico's de eerste versie	20
10.1 Risico's.....	20
Appendix A: Planning AI MVP project Increment 1	21

Figuur 2.2: Inhoud scoping sjabloon

Centraal in dit sjabloon staat de 'mapping' van de behoefte van de (eind)gebruikers aan voorspellingen en 'actionable insights' naar de hiervoor benodigde data. Dit staat of valt met het toetsen van de kwaliteit en beschikbaarheid van de data in relatie tot de behoefte van de gebruikers.

Op basis van deze bevindingen wordt gekeken of er knelpunten zijn die eerst opgelost moeten worden voordat er aan AI gedacht kan worden.

Verder worden de randvoorwaarden van de systeemspecificaties beschreven en vastgesteld welke informatieproducten c.q. AI-oplossingen tot de eerste versie van het systeem behoren.

De behoefte aan voorspellingen en actionable insights, welke als basis dient voor het AI-systeem, wordt globaal geïnventariseerd. Vervolgens wordt een keuze gemaakt welke voorspellingen en actionable insights toegevoegd worden aan het AI minimal viable product (AI MVP). Op basis van de resultaten van de scoping kan worden besloten over de voortzetting van het AI-project.

Bijlage

[Scoping sjabloon](#)

2.3 AI maturity scan

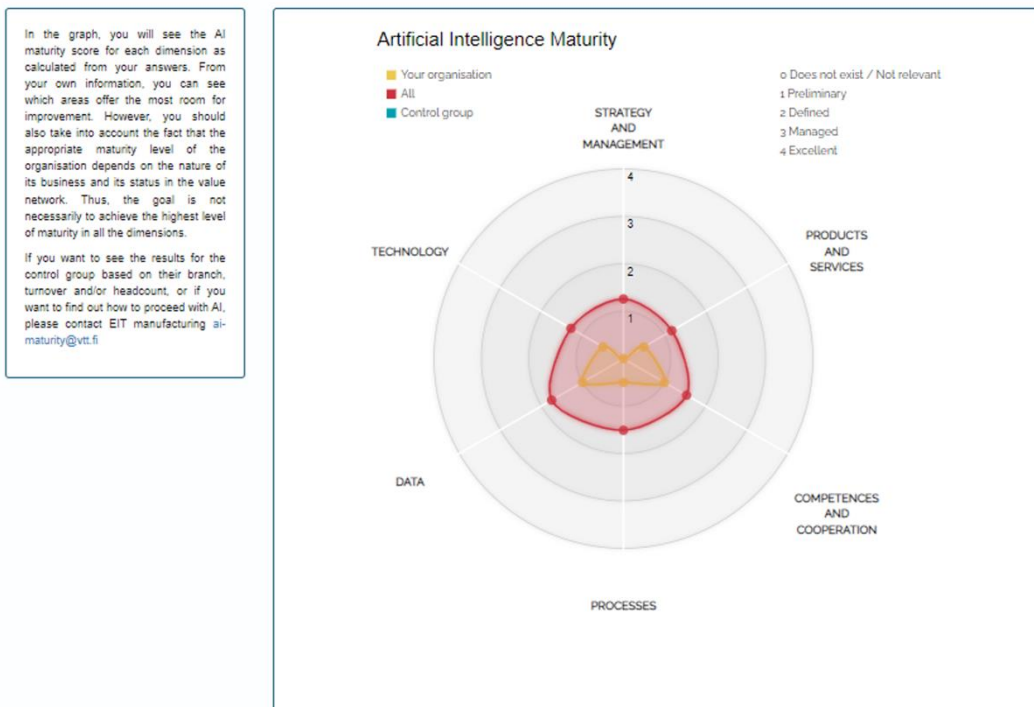
Doel

Een AI maturity scan geeft een beeld van de volwassenheid van de organisatie op verschillende aspecten die van belang zijn om succesvol AI toe te passen in een organisatie.

Uitleg

De AI maturity scan van het Finse onderzoeksinstituut VTT stelt vragen op zes verschillende gebieden (Strategy en Management, Producten en Diensten, Competenties en Samenwerking, Processen, Data en Technologie). De antwoorden zijn ingedeeld in de gebruikelijke 'maturity' levels van: 0 = niet aanwezig, tot 4 = excellent. Los van de specifieke antwoorden is het interessant om de verschillende 'maturity' levels te bekijken omdat ze inzicht geven in de verschillende mogelijke niveaus.

De resultaten van de scan worden weergegeven in een radarplot wat de AI-volwassenheid van de organisatie weergeeft en vergelijkt met de gehele groep van respondenten.



Figuur 2.3: Radar-plot van de AI-maturity van een organisatie

Meer informatie

De maturity scan is te vinden op <https://eit.aimaturity.vtt.fi/>. Voordat er gebruik gemaakt kan worden van de scan wordt er gevraagd om een account aan te maken en het emailadres te verifiëren.

2.4 Uitgangspunten voor AI-ontwikkeling

Doel

Bewustwording van de vier mogelijke uitgangspunten voor AI-ontwikkeling.

Uitleg

Wanneer je als MKB-bedrijf de stap wil maken naar het inzetten van AI, kan het innovatieproces vanaf een aantal uitgangspunten beginnen. Op basis van het uitgangspunt kan de volgorde van stappen binnen de eerste fase van het innovatieproces vorm worden gegeven.

Design is 'human centered'. Een ontwerpproces zou in het ideale geval altijd moeten beginnen met het valideren van de behoefte van de gebruiker. Het begrijpen van zijn/haar context is daarbij de belangrijkste stap. Het Desirable-Feasible-Viable diagram van IDEO's Human Centered Design Toolkit (2011) (zie Figuur 2.4) definieert innovatieve ideeën als "concepten die aansluiten bij de echte gebruikersbehoefte en die haalbaar zijn vanuit een technologie en business perspectief" (Grossman-kahn et al., 2012). In de praktijk kan echter ook de beschikbare data en techniek of het business idee het uitgangspunt vormen van het innovatieproces. Met welk uitgangspunt de organisatie ook begint, het concept dient in ieder geval vanuit drie perspectieven gevalideerd te worden;

- Desireability (Mens)
- Feasibility (Techniek en Data)
- Viability (Business)



Figuur 2.4: Het Desirable-Feasible-Viable diagram van IDEO's Human Centered Design Toolkit (2011)

Afhankelijk van het uitgangspunt waarmee de organisatie het proces begint, ziet de volgorde van activiteiten voor validatie er iets anders uit. De ideale route begint conform de theorie van 'User Centered Design' bij 'Desireability', maar in de praktijk zijn ook andere uitgangspunten mogelijk.

Er zijn 4 verschillende uitgangspunten (beginsituaties) denkbaar waarmee een organisatie de eerste fase zou kunnen beginnen;

- 1) Vanuit de behoefte van de mens (Desireability). Dit is de geprefereerde route. Er wordt begonnen vanuit een probleem of een kans die gesignaleerd wordt binnen (intern) of buiten (extern) de organisatie. Vanuit de Desireability ontstaat een concept dat gevalideerd wordt aan Feasability en Viability.
- 2) Vanuit de techniek en de data (Feasability). Dit is de klassieke innovatie route. Er wordt begonnen vanuit een technische mogelijkheid die ontstaat door aanwezige data of technologische ontwikkeling. Vanuit de Feasability ontstaat een concept dat gevalideerd wordt aan Desireability en Viability.
- 3) Vanuit een initieel concept. Dit is de route die vooral voorkomt bij de visionaire ondernemer of domeinexpert. Deze komt op basis van kennis en ervaring tot een initieel concept dat nog vanuit alle drie de perspectieven gevalideerd moet worden; Desireability/Feasability en Viability.
- 4) Vanuit de innovatiebehoefte. Dit is de route waarbij er nog geen concreet concept of idee is, maar wel een hele sterke behoefte om te innoveren. Vooral bij (semi)overheidsorganisaties komt dit voor. Vanuit Desireability/Feasability moet er eerst een concept ontstaan wat vervolgens gevalideerd moet worden; Desireability/Feasability en Viability.



Figuur 2.5: Uitgangspunten voor AI-ontwikkeling

Meer informatie

IDEO's Human Centered Design Toolkit (2011)

2.5 Een innovatiemodel voor het ontwikkelen van AI-producten en diensten

Doel

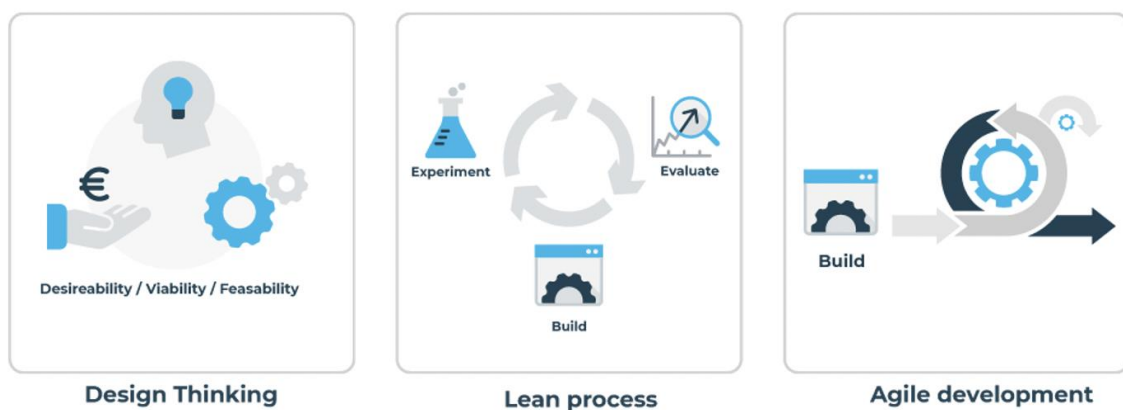
Het inzicht geven in de stappen die genomen moeten worden om een innovatief en commercieel succesvol AI-product of dienst te ontwikkelen.

Uitleg

Het innovatiemodel voor het ontwikkelen van AI-oplossingen binnen een MKB-bedrijf dat wij voorstellen, is gebaseerd op verschillende reeds bestaande en veelgebruikte modellen. Als hoofdindeling gebruikt het innovatiemodel het stage-gate model van Cooper (Cooper, 1990). Het stage-gate model verdeelt het innovatieproces in vijf afzonderlijke fasen ('stages') die gescheiden zijn door expliciete beslismomenten.

Per fase gebruikt het innovatiemodel elementen van Design Thinking, Agile en Lean (Grossman-Kahn, 2012) (zie **Error! Reference source not found.**2.6). Dankzij het gebruik van Design Thinking is er in elke fase van het innovatiemodel oog voor drie aspecten: business (viability), gebruikersbehoefte (desirability) en technische haalbaarheid (feasibility).

In de rest van deze paragraaf lichten we de belangrijkste onderdelen van het model toe. De sub paragrafen 2.5.1 t/m 2.5.5 gaan vervolgens meer in detail in op de verschillende fasen.



Figuur 2.6: Het innovatiemodel combineert elementen van Design Thinking, Lean en Agile (Grossman-Kahn, 2012)

Fasen

Voor de ontwikkeling van AI-oplossingen kunnen vijf verschillende fasen onderscheiden worden waar het product of de dienst doorheen gaat, vanaf het initiële idee tot aan lancering in de markt:

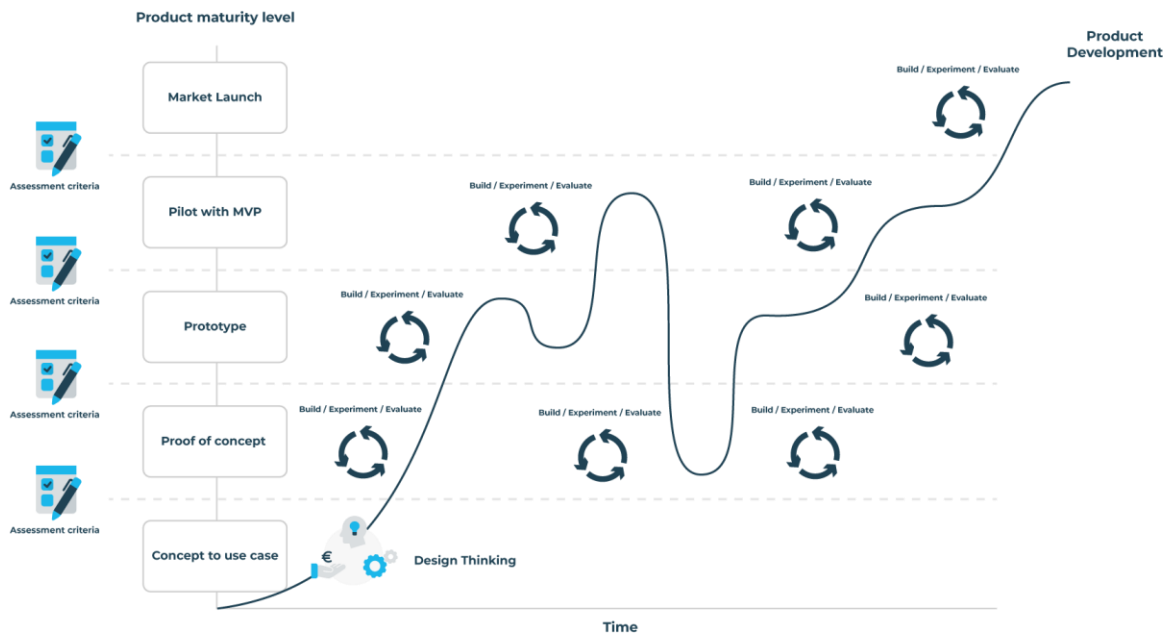
- 1) **Concept to use case** (werkt het op papier?)
- 2) **Proof of concept** (werkt het idee?)
- 3) **Prototype** (werkt het in een lab?)
- 4) **Pilot with MVP** (werkt het in de praktijk?)
- 5) **Market Launch** (werkt het in de markt?)



Figuur 2.7: Fasen van het Model

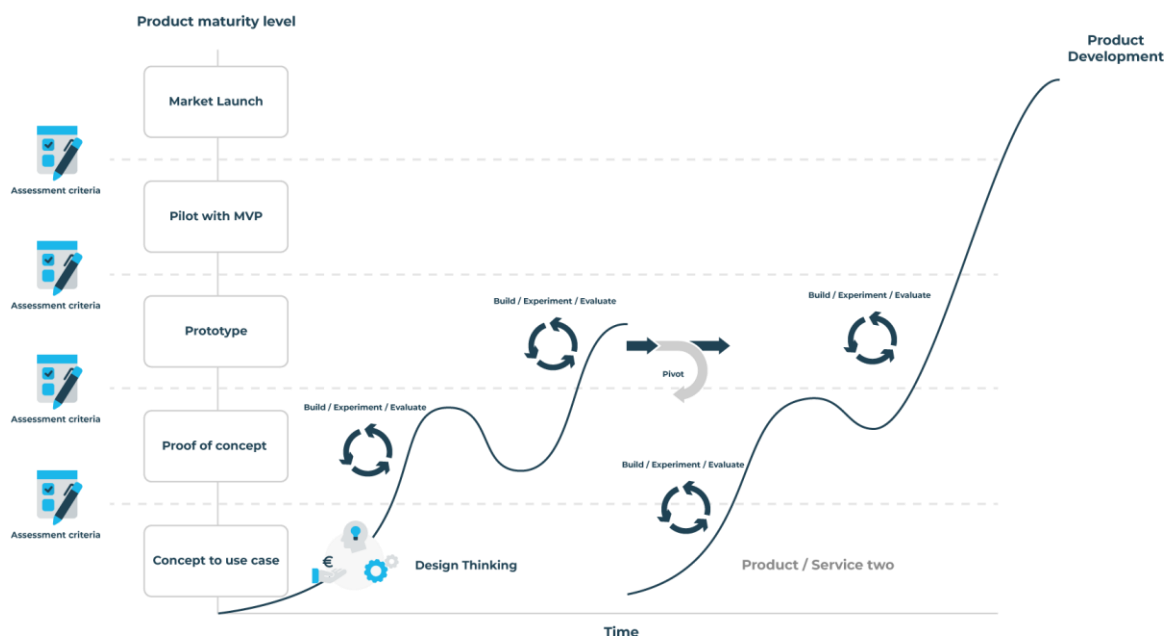
Een product of dienst doorloopt deze fasen zelden op een lineaire manier (zie Figuur 2.8) . Het ontwikkelen van succesvolle AI-producten en -diensten is een zeer iteratief ontwikkelingsproces. De fasen zijn enkel bedoeld om het ontwikkelteam te helpen begrijpen waar ze zich in de ontwikkelingscyclus bevinden en helpt hen vervolgens de ontwikkelingscyclus op microniveau te beheren.

In deze fasen verrichten de verschillende projectleden activiteiten om de AI-innovatie telkens een stap verder te ontwikkelen, met als uiteindelijk doel een product of dienst op de markt te zetten. Voor de verschillende betrokken rollen verwijzen we naar de vaardighedenscan in paragraaf 5.2.



Figuur 2.8: Fasen van het Model – niet lineair

Het is ook goed mogelijk dat er tijdens het proces nieuwe inzichten ontstaan op basis waarvan besloten wordt de huidige koers om te gooien en een pivot te maken naar een ander product of service. (zie Figuur 2.9).



Figuur 2.9: Fasen van het Model – met pivot

Tussenproducten en activiteiten

In elke fase van het project identificeert het projectteam de belangrijkste onbekenden, onzekerheden en meest kritische veronderstellingen voor die fase. Hierbij wordt gekeken naar business-, gebruikers- en technische aspecten. Bijvoorbeeld: “Voorziet de AI-innovatie daadwerkelijk in een gebruikersbehoefte?”, “Kunnen we op basis van de beschikbare trainingsdata voldoende nauwkeurigheid behalen?”, “Klopt de veronderstelling dat de totale markt een omvang heeft van X euro per jaar?”.

Vervolgens wordt er bepaald welke informatie nodig is om vragen te beantwoorden, dan wel deze veronderstellingen te valideren. Deze informatiebehoefte bepalen de tussenproducten die nodig zijn voor de evaluatie van de fase. Om de voorbeeldvragen uit de vorige alinea te beantwoorden kan er bijvoorbeeld gedacht worden aan tussenproducten als een gebruikersonderzoek bij een aantal typische gebruikers, een ‘quick-and-dirty’ test van een aantal algoritmes op een gedeelte van de data of een marktonderzoek. De tussenproducten bepalen op hun beurt weer de activiteiten die uitgevoerd moeten worden. Op deze manier brengt het projectteam zijn eigen set van tussenproducten en fase-activiteiten in kaart die specifiek zijn voor het project (Zie **Error! Reference source not found.**2.10).



Figuur 2.10: Elke fase kent activiteiten die leiden tot deliverables die bij de gates gecheckt worden tegen de assessment criteria

Gates

Gates zijn een essentieel onderdeel van het innovatieproces. Het zijn belangrijke momenten waarop de tussenproducten van elke fase kritisch worden beoordeeld en kritische beslissingen worden genomen.

Het is belangrijk om vooraf, dus voordat de fase begint, een duidelijke set beoordelingscriteria te hebben. Het doel is hier niet om star te zijn en vooraf een vaststaande productdefinitie te hebben, maar om zoveel mogelijk duidelijkheid te hebben over wat er aan het eind van de fase verwacht wordt om een succesvol product of een dienst te hebben.

Het beoordelen van de deliverables wordt gedaan door zogeheten 'gatekeepers'. Deze gatekeepers moeten zorgvuldig gekozen worden. Meestal worden hierbij relevante unitmanagers, domeinexperts en potentiële klanten betrokken. In veel gevallen heeft het MKB onvoldoende expertise aan boord om de voortgang van de ideeën kritisch te beoordelen, of is het moeilijk om voldoende objectief te zijn. In dat geval kan het verstandig zijn om te leunen op een extern netwerk van durfkapitalisten, domeinexperts, etc. Best practices met betrekking tot de te beoordelen aspecten vindt u in Cooper (2008).

Uiteraard kunnen de beoordelingscriteria verschillen, afhankelijk van de prioriteiten van het management en de industrie. De gezondheidszorgsector kan bijvoorbeeld strikte certificeringen en toestemming van de juiste autoriteiten vereisen. Er zijn echter enkele generieke elementen, zoals:

- Is er een geschikt klantensegment gedefinieerd?
- Heeft het team een reeks gevalideerde waardeproposities gedefinieerd?
- Vervult het idee van het product/de dienst een reële behoefte?

Deze en andere vragen worden in de volgende sub paragrafen per fase nader uitgewerkt.

Meer informatie

De volgende papers bieden meer informatie over de achtergrond van de verschillende ingrediënten van het innovatiemodel.

De Paula, D. F., & Araújo, C. C. (2016, July). Pet empires: combining design thinking, lean startup and agile to learn from failure and develop a successful game in an undergraduate environment. *In International Conference on Human-Computer Interaction* (pp. 30-34). Springer, Cham.

Grossman-Kahn, B., & Rosensweig, R. (2012). Skip the silver bullet: driving innovation through small bets and diverse practices. *Leading Through Design*, 18, 815.

Cooper, R. G. (2008). Perspective: The stage-gate idea-to-launch process—update, what's new, and nexgen systems. *Journal of product innovation management*, 25(3), 213-232.

Robert G. Cooper (2017) Idea-to-Launch Gating Systems: Better, Faster, and More Agile, *Research-Technology Management*, 60:1, 48-52, DOI:10.1080/08956308.2017.1255057

2.5.1 Fase1: Concept to use case

Doel

Het doel van fase 1 is om een initieel idee te vertalen naar een levensvatbare use case. Hierbij is de uitdaging om het markt-/bedrijfspotentieel te bepalen met zo min mogelijk middelen. Vanuit technisch perspectief is het belangrijk om te bepalen of AI daadwerkelijk van toegevoegde waarde zou kunnen zijn en om in deze fase de bruikbaarheid van de data te borgen.

Business en gebruikersperspectief

De focus ligt hier op het verkrijgen van een duidelijk begrip van het idee en het uitwerken van de details. Identificeer potentiële klanten en eindgebruikers van het product. We maken specifiek onderscheid tussen klanten en eindgebruikers omdat ze niet hetzelfde zijn. Een klant is degene die voor het product betaalt, terwijl de eindgebruiker degene is die het product gebruikt, en beide belanghebbenden zijn belangrijk om te overwegen. Verder moet de waarde propositie worden gedefinieerd als klant- en gebruikersgroep. De deliverable die bij deze activiteiten horen, kan een gevalideerd klant- en gebruikersprofiel en waarde propositie zijn. Bovendien helpt het meestal om de waarde proposities te vertalen naar use-cases en vervolgens naar functionele en technische vereisten. Tools zoals service blue printing helpen het idee verder uit te werken. Het is essentieel dat het team naar concurrerende producten en diensten zoekt en nagaat hoe ze zullen concurreren met gevestigde exploitanten.

In deze fase kan ook de totale adresseerbare markt worden geschat. Een ander belangrijk aspect waarmee in deze fase rekening moet worden gehouden, is het team dat aan het idee werkt. Het team moet over alle benodigde vaardigheden beschikken om het idee verder te ontwikkelen en uit te voeren. Ook in andere fasen moet dit continu worden gemonitord. Onderstaande tabellen onder elke fase geven een overzicht van de benodigde vaardigheden/rollen die nodig zijn om het innovatieproces uit te voeren.

Tip: We raden aan om het waarde propositie canvas te gebruiken om bovenstaande activiteiten uit te voeren. Voor meer informatie over het toepassen van de bovenstaande techniek verwijzen wij u naar: *A. Osterwalder, Y. Pigneur, G. Bernarda en A. Smith, Value proposition design: How to create products and services customers want. John Wiley & zonen, 2014.*

Techniek

In deze fase dient ook de initiële technische haalbaarheid te worden beoordeeld. In het geval van AI-enabled applicaties is het noodzakelijk om de beschikbaarheid van data te beoordelen. Wat cruciaal is in deze fase is het identificeren van de bruikbaarheid van de data. De bruikbaarheid is een optelsom van de beschikbaarheid, de betrouwbaarheid, en de bestendigheid waarbij de vraag beantwoord moet worden of de data de informatie bevat die nodig is om de gewenste actionable-insights en voorspellingen te kunnen realiseren.

Mocht de data niet bruikbaar zijn dan moeten acties uitgezet worden om enerzijds de data bruikbaar te maken, anderzijds de gewenste actionable-insights en voorspellingen nog een keer tegen het licht te houden. Hierbij is het ook van belang om niet opnieuw het wiel uit te vinden maar ook te kijken of er al oplossingen zijn gerealiseerd in de buitenwereld en of deze toepasbaar zijn in deze situatie.

Tabel 2.1 geeft een voorbeeld van activiteiten, deliverables en rollen in fase 1. Voor de volledige lijst van activiteiten en deliverables; zie bijlage Innovatiemodel tabellen

Activiteiten	Deliverables	Rol
Zoeken naar vergelijkbare technologische oplossingen (qua functionaliteit)	Marktanalyse	Business analist
In kaart brengen + oordelen over (kwaliteit/kwantiteit) van data + labels (indien van toepassing)	Beschrijving van de data (intern + extern) die benodigd is om het probleem op te kunnen lossen met AI	Data analist

Tabel 2.1: Fase 1 - Activiteiten, deliverables, en rollen (voorbeeld)

Assessment criteria

Bij de gate van fase 1 beoordelen de gatekeepers de deliverables van de fase aan de hand van een lijst met criteria. geeft een voorbeeld, Voor een meer gedetailleerde lijst van assessment criteria; zie bijlage Innovatiemodel tabellen

Human/Business	Tech
Hebben ze het juiste klantsegment geïdentificeerd?	Bepaal de geschiktheid van de technologie om het probleem op te lossen en waarom de technologie geschikt is voor de klantgroep.

Tabel 3: Fase 1 – Assessment criteria (voorbeeld)

2.5.2 Fase 2: Proof of concept fase

Doel

Het doel van deze fase is om te komen van 'een idee' tot een 'proof of concept' waarmee de waarde propositie gecommuniceerd en gevalideerd kan worden bij de relevante stakeholders. Ook zal de haalbaarheid op technisch gebied gevalideerd moeten worden. Het gaat er hier om om aan te tonen dat hetgeen bedacht is, ook daadwerkelijk mogelijk is.

Vanuit een business en gebruikersperspectief moeten de volgende vragen beantwoord worden: Levert het product of de service genoeg waarde voor de gebruiker? Kan er een rendabel businessmodel gedefinieerd worden?

Vanuit een technisch perspectief proberen we de volgende vraag te beantwoorden: Kunnen we een werkende basis oplossing voor het probleem creëren?

Business en gebruikersperspectief

De businessactiviteiten zijn gericht op het verzamelen van kosteninformatie met betrekking tot ontwikkeling/productie/sourcing. Bovendien moeten de activiteiten zich ook richten op het verzamelen van aanvullende eisen met betrekking tot regelgeving, certificeringen, etc., die nodig zijn om het product commercieel te exploiteren. De aanvullende vereisten kunnen worden geïdentificeerd door te praten met belangrijke partners zoals leveranciers, retailers, certificerende instanties etc. Alle relevante vereisten moeten aan het ontwikkelingsteam worden gecommuniceerd. Ook is het van belang om erachter te komen welke kernactiviteiten binnen het bedrijf worden uitgevoerd en welke worden uitbesteed aan derden. Als u dit doet, kunt u de belangrijkste partners op de shortlist zetten en de voorwaarden van partnerschappen verkennen.

Daarnaast is het ook nodig om erachter te komen welk type relaties met de klanten moeten worden aangegaan en onderhouden en via welke kanalen de klanten willen worden bereikt.

Let op: dit is geen theoretische oefening, maar al deze aspecten moeten worden uitgewerkt in interactie met relevante belanghebbenden en de echte wereld. Met betrekking tot de volgende fase (fase 3 pilot/prototype) moeten bepaalde voorbereidingen worden getroffen. Gebruik het canvas van het bedrijfsmodel om de bovenstaande elementen te documenteren. De belangrijkste stap in deze fase is onder andere het werven van een launching customer op basis van de proof of concept.

Tip: het ontwikkelen van *customer journey maps* helpt bij het identificeren van de contactpunten en het soort relaties dat met de klant moet worden aangegaan en onderhouden.

Techniek

De technologie gerelateerde activiteiten zullen gericht zijn op het vertalen van de waarde proposities naar een werkend proof of concept of een prototype. Dit helpt bij het beoordelen van de technische haalbaarheid. Het zal met name helpen bij het inschatten van de benodigde ontwikkelings-/sourcing-/productiecapaciteiten. Bovendien levert het, zodra een duidelijk beeld van de technische haalbaarheid is gevormd, input voor de bedrijf gerelateerde activiteiten.

Voor het AI-gedeelte geldt dat de dataset geïmplementeerd moet worden, middels visualisaties en statistiek gecontroleerd. Waarna de features gerealiseerd worden om te komen tot een eerste experiment met de AI met échte data. Om daarna een eerste model te bouwen op basis van een versimpelde versie van het probleem. De kunst is om het KISS principe toe te passen. Maar wel kijken naar de (on)mogelijkheden van de data en de performance van het model. Daarnaast moet er worden gekeken naar technische aspecten zoals de infrastructuur, het oplossen van gevonden omissies van de vorige fases en naar de ethische kant van de zaak.

Voor een meer gedetailleerde lijst van activiteiten en deliverables; zie bijlage Innovatiemodel tabellen

Assessment criteria

Bij deze gate beoordelen de gatekeepers de deliverables van de fase aan de hand van een lijst met criteria, zie bijlage Innovatiemodel tabellen

2.5.3 Fase 3: Prototype fase

Doel

Het doel van deze fase is om vanuit business en gebruikersperspectief de volgende vraag te beantwoorden: Kunnen we het product/de dienst maken of leveren tegen een redelijke prijs en schaal?

Vanuit technisch perspectief moet tijdens deze fase de volgende vraag beantwoord worden: Werkt de oplossing in een volledig gecontroleerde omgeving/setting?

Business en gebruikersperspectief

De prototypefase richt zich op het omzetten van de proof of concept naar een prototype. Dit gebeurt bij voorkeur in nauwe samenwerking met de launching customer. Het prototype bevat de kernfunctionaliteit. Daarnaast is het belangrijk om een pilot/demonstratieplan te ontwikkelen. Het plan dient het implementatieplan bij de organisatie van de opdrachtgever te specificeren. Het plan moet de technische specificaties bevatten die nodig zijn om de oplossing te implementeren, een trainingsplan voor zowel technische als gebruikers, noodzakelijke wijzigingen in bedrijfsprocessen, budget, klantenondersteuning en integratie met andere systemen.

Het plan moet ook duidelijk de hypothese voor een businessmodel specificeren die het team van plan is te testen tijdens de volgende fase. Bovendien is het in dit stadium noodzakelijk om de omzet- en prijsstrategie te bepalen. Zeker als de launching customer daar om vraagt voordat hij aan de volgende fase begint.

Als u begrijpt hoe de concurrerende producten en diensten worden geprijsd, kan dit helpen bij het bepalen van de prijsstrategie. Ook kan het testen van verschillende bundels van waarde proposities op verschillende prijspunten via onlineadvertenties helpen bij het bepalen van de winst maximaliserende prijs. Er worden verschillende tools en technieken gebruikt om de prijs te schatten, bijvoorbeeld een op kosten gebaseerde prijsstrategie versus op waarde gebaseerde prijsstrategie. Het schatten van de betalingsbereidheid is een veelgebruikte techniek in de op waarde gebaseerde prijsstrategie. De keuze van kanalen om het product of de dienst te verkopen, kan ook van invloed zijn op de prijsstrategie.

Voor meer informatie over prijsstelling, zie: *Ramanujam, M. en Tacke, G. (2016) Monetizing innovation: how smart companies design the product around the price. John Wiley & Sons.*

Techniek

In deze fase wordt het AI-model hertraint en het resultaat presentabel voor de gebruikers gemaakt, onder andere door het interactief maken van de oplossing en visualisaties toe te voegen. Van het grootse belang is dat de oplossing geëmbod kan worden in de omgeving van de organisatie. Hierbij wordt het model geïntegreerd in de technische en actuele context van de organisatie en wordt begonnen met het automatiseren van de datapijplijn om de handmatige data verzameling te vervangen.

Hierbij wordt de effectiviteit/ performance van model getoetst. Een belangrijk aspect is ook de trustworthiness van het model te onderbouwen om ongelukken in de praktijk te voorkomen. Zie ook hoofdstuk 1 van dit handboek.

Om inzicht te krijgen in de inspanningen van de volgende fase moeten er nagedacht worden over toekomstbestendigheid/onderhoudbaarheid van het model en over opleidingen en skills van betrokkenen.

Voor een meer gedetailleerde lijst van activiteiten en deliverables; zie bijlage Innovatiemodel tabellen

Assessment criteria

Bij deze gate beoordelen de gatekeepers de deliverables van de fase aan de hand van een lijst met criteria, zie bijlage Innovatiemodel tabellen

2.5.4 Fase 4: Pilot/MVP

Doel

Het doel van deze fase is om het product/de dienst te valideren in een real-world setting en het gereed maken van het product/de dienst en de bedrijfsinfrastructuur voor de marktintroductie. Verschillende aspecten van het product/de dienst worden in deze fase gevalideerd, zoals of het product/de dienst de beoogde waarde creëert, gebruiken gebruikers het product zoals bedoeld, heeft het team de nodige vaardigheden en capaciteiten om gebruikers te trainen en klantenondersteuning te bieden, etc. Het testen van het product/de dienst in een real-world omgeving kan leiden tot aanvullende eisen of tweaks die nodig zijn voor het eindproduct.

Business en gebruikersperspectief

De bedrijf-gerelateerde activiteiten zijn gericht op het bewaken en beoordelen van de aan de klant geleverde waarde. Het team richt zich ook op het beoordelen van de bedrijfsvoering die nodig is om het product/de dienst te verkopen, op te zetten en te ondersteunen. Deze activiteiten kunnen aanvullende informatie over de kostenstructuur opleveren. De kostenmodellen moeten dien overeenkomstig worden bijgewerkt en de prijsstrategie opnieuw bekijken. Waar nodig moeten de teams ook de nodige gegevens verzamelen voor certificeringen, het kwantificeren van de waarde propositie, etc.

Vanuit het business en gebruikersperspectief zijn de activiteiten gericht op het opzetten van de benodigde infrastructuur om een daadwerkelijke pilot te kunnen draaien. Hieronder vallen onder andere het verkrijgen van de juiste licenties, het opzetten van geschikte kanalen, klantenondersteuning, verkoopcontracten, het ontwikkelen van een marketing- en verkoopplan, enz.

Het uitvoeren van deze activiteiten kan aanvullende informatie onthullen met betrekking tot kosten die in de vorige fasen gemist hadden kunnen worden. Het zou verstandig zijn om de kostenstructuur bij te werken en de prijsstrategie opnieuw te bekijken. Het team richt zich ook op het uitvoeren van schijnverkoop/setup en klantenondersteuning om ervoor te zorgen dat alle systemen en processen operationeel zijn.

Techniek

De technisch gerelateerde activiteiten draaien om het product/klaar maken voor implementatie, het monitoren van de productprestaties, etc. Een ander belangrijk aspect dat in deze fase moet worden overwogen, is integratie met andere systemen. Producten en diensten functioneren zelden in een standalone modus, ze vereisen vaak integratie met andere systemen op de locatie van de klant. Daarom is het belangrijk dat het team, zo niet eerder, de integraties met andere systemen uitzoekt. Dit concept wordt ook wel het systeemgereedheidsniveau genoemd.

In praktische zin moet ervoor gezorgd worden voor dat een dashboard gerealiseerd is waarmee de AI-oplossing bediend kan worden. Bijvoorbeeld controle op drift van de uitkomsten van de AI-oplossing, mogelijke hertraining van de modellen of het afkoppelen van de oplossing. En dat de productieomgeving ingericht is met een datapijplijn en model waarvan de uitkomsten ingrijpen in andere systemen. Daarnaast is beheer van de oplossing geregeld bij systeembeheer waarbij logging en monitoring op de lange en de korte termijn ingericht en geautomatiseerd is. Ook is de beheerorganisatie ingericht voor support van de gebruikers. De organisatie wordt ingeregeld op het gebruik van de oplossing waarbij de gebruikers worden opgeleid en bewustgemaakt van de toegevoegde

waarde. Tijdens het gebruik van de AI-oplossing worden de verbeterpunten zowel in systeemtechnische zin als in organisatorische zin gemonitord en bijgehouden.

Ook andere activiteiten ter voorbereiding van de daadwerkelijke lancering dienen aan het einde van deze fase uitgevoerd te worden. Denk hierbij aan; het automatiseren van integraties, het oplossen van bugs, het bijwerken van gebruikershandleidingen en het gereedmaken van het product/de dienst om via de juiste kanalen live te gaan, zodat de onderliggende infra kan worden geschaald volgens de verkoopprognoses.

Voor een meer gedetailleerde lijst van activiteiten en deliverables; zie bijlage Innovatiemodel tabellen

Assessment criteria

Bij deze gate beoordelen de gatekeepers de deliverables van de fase aan de hand van een lijst met criteria, zie bijlage Innovatiemodel tabellen

2.5.5 Fase 5: Market Launch fase

Doel

Dit is de laatste fase van het innovatieproces en richt zich op de commerciële bedrijfsvoering en opschaling van de onderneming. Vanaf hier richt het ontwikkelteam zich op het verfijnen van het systeem en de volgende reeks functies die relevant is voor het opschalen van het bedrijf. We identificeren geen specifieke activiteiten of deliverables voor deze fase omdat het product/de dienst commerciële activiteiten betreedt en verder gaat dan het transformeren van een idee naar een commercieel levensvatbaar product.

Bijlage

[Innovatiemodel tabellen](#)

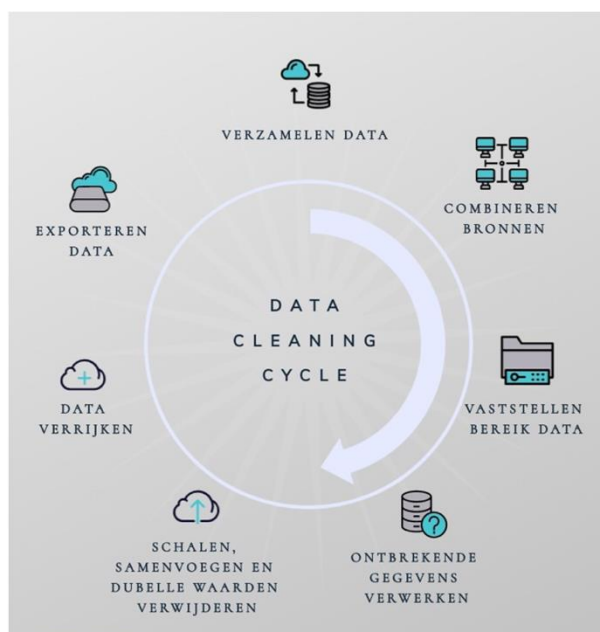
3 Het verzamelen en beoordelen van data

Elk AI-model staat of valt met de data waarmee het is gebouwd. Een bekend credo binnen AI luidt “garbage in = garbage out”. Zelfs het meest geavanceerde Machine Learning model zal tekortschieten zonder voldoende data van voldoende kwaliteit. Naast het feit dat data noodzakelijk is voor het toepassen van de meeste AI-modellen of -algoritmen kan een gestructureerde dataset met bijbehorende analyse ook helpen met het aanscherpen van de vraagstelling: past de data die wij hebben bij het probleem dat we op willen lossen?

Het is daarom van groot belang om voldoende aandacht te besteden aan de voorbereidende stappen die leiden tot een dataset van voldoende kwaliteit. In de praktijk kan het zijn dat deze fase een significant deel van de beschikbare tijd in beslag neemt; onderschat deze fase dus niet!

In dit hoofdstuk doorlopen we alle stappen die noodzakelijk zijn om een dataset te kunnen maken die geschikt is voor AI/ Machine Learning, zoals het verzamelen van gegevens, het combineren van data afkomstig van verschillende bronnen, het op peil krijgen van de datakwaliteit (ontbrekende waarden, detecteren van meetfouten) en het verrijken van de dataset.

Dit doen we aan de hand van de ‘data cleaning cycle’ (zie **Error! Reference source not found.** 3.1). Dit is een proces dat ondersteunt in het structureren en vormgeven van een dataset zodat die geschikt is voor AI- en Data Science doeleinden. Hoewel de *data cleaning cycle* vrij lineair wordt beschreven, kan het soms nodig zijn om een stap over te slaan, of terug te gaan naar een eerdere stap.



Figuur 3.1: De data cleaning cycle

Aangezien er een grote verscheidenheid aan soorten problemen en typen data bestaat zal er voornamelijk aandacht worden geschonken aan het aanreiken van verschillende mogelijkheden, valkuilen en methodes, zonder al te diep in te gaan op de specifieke materie. Afhankelijk van de precieze ontwikkelomgeving, probleemdefinitie en het soort data is elke aanpak en situatie net iets anders. Aan het einde van dit hoofdstuk geven we een aantal verwijzingen naar diepgaandere bronnen, en per stap van de data cleaning cycle wordt naar relevante materie verwezen.

Meer informatie

Zie ook deze [infographic](#) die het data cleaning proces beschrijft.

3.1 Het verzamelen van data

Doel

In deze stap reiken we een aantal handvatten aan voor het op een correcte wijze verzamelen van data uit beschikbare bronnen, ook wel 'data acquisitie' genoemd. Het op een correcte wijze verzamelen van gegevens vergroot de kans op een succesvolle AI-toepassing.

Uitleg

3.1.1 Bronnen

Een dataset voor Machine Learning doeleinden bestaat uit één of meerdere bronnen. Wellicht heb je intern al een (grote) verzameling aan data, of misschien moet je nog meer gegevens verzamelen n.a.v. de initiële analyse van de probleemcontext. Over het algemeen maken we onderscheid tussen twee soorten bronnen: interne bronnen en externe bronnen.

Interne bronnen beslaan gegevens die direct afkomstig zijn vanuit het bedrijf of de organisatie zelf. Hierbij kan je denken aan personeelsgegevens, onderhoudsgegevens, verkoopcijfers, de grootte van het bedrijf, sensorgegevens verzameld door de organisatie etc.

Het grote voordeel van interne bronnen is dat je niet afhankelijk bent van derden m.b.t. de verzameling en/of verwerking van de gegevens. Ook heb je als organisatie de volledige controle over (de mate en methode van) verwerking en opslag van de data.

In de praktijk zal je voor sommige gegevens afhankelijk zijn van externe bronnen. Voorbeelden zijn gegevens van social media platforms zoals facebook, publieke datasets zoals van het CBS, of niet-openbare datasets verzameld door een marketingbureau.

Voor externe bronnen geldt dat de variëteit van de inhoud, beschikbaarheid en kwaliteit sterk verschilt. Ook moet rekening gehouden worden met de betrouwbaarheid van externe bronnen: is de organisatie die de externe data heeft verzameld onafhankelijk, of hebben zij een eigen agenda? Heb je de rechten om (commercieel) gebruik te maken van de gegevens (welke licentie is van toepassing op het ontvangen of verwerken van de gegevens)?

Tabel 3.1 geeft een aantal bronnen voor openbare machine learning datasets.

Naam	Beschrijving	URL
CBS	Data over maatschappelijke onderwerpen zoals Arbeid, Economie en Onderwijs.	https://www.cbs.nl/nl-nl
KNMI	Weer- en klimaatdata over verschillende weerstations in Nederland.	https://www.knmi.nl/kennis-en-datacentrum
Overheid	Open datasets van allerlei overheidsinstanties op niveau van rijk, provincies, gemeentes, waterschappen etc.	https://data.overheid.nl
Kaggle	Kaggle is een platform waarop machine learning competities gehouden worden. Let op: minder betrouwbare data!	https://www.kaggle.com/datasets
Google-data search	Een google-zoekmachine specifiek voor datasets. Let op: afhankelijk van de bron kan het resultaat minder betrouwbaar zijn.	https://datasetsearch.research.google.com/

Visual data	Open source datasets voor computer vision doeleinden (beeldmateriaal).	https://visualdata.io/discovery
NLP-datasets	Datasets specifiek voor natuurlijke taalverwerking (NLP) - overzicht van meerdere bronnen.	https://github.com/niderhoff/nlp-datasets

Tabel 3.1: bronnen voor machine learning datasets

3.1.2 Hoeveel meet je?

Over het algemeen geldt: meer is beter. Data weggooien of een subset nemen van je data kan altijd nog, maar meer verzamelen kan achteraf vaak niet. Meer meten kan zowel in de tijd (bijvoorbeeld elke minuut in plaats van elk uur – zie 'Hoe frequent meet je?'), als in het variabelendomein (bijvoorbeeld naast het toerental van een motor ook de temperatuur – zie 'Wat meet je?').

3.1.3 Wat meet je?

Welke gegevens zijn eigenlijk relevant voor het probleem dat je op wilt lossen? Kennis van een domeinexpert kan snel duidelijk maken welke factoren in meer of mindere mate belangrijk zijn. Staat alle data die nodig is om het probleem op te (kunnen) lossen tot je beschikking? Zijn de meetpunten 'ruwe datapunten' (de exacte temperatuur op een bepaald punt in de tijd), of zijn het afgeleide waarden (de gemiddelde temperatuur over bijvoorbeeld 1 minuut)?

3.1.4 Hoe frequent meet je?

Wanneer of hoe vaak wordt een meetpunt opgeslagen? Verschilt dit per variabele? In hoeverre komt de meetfrequentie overeen met de resolutie die je nodig hebt? De gewenste meetfrequentie is afhankelijk van de toepassing die je voor ogen hebt, en de (natuurlijke) variatie in het signaal dat je wilt meten. Bijvoorbeeld: als de temperatuur honderd keer per seconde wordt gemeten, is dat in het ene domein wellicht overkill (bijv. buitentemperatuur voor een slim klimaatsysteem), terwijl het in een ander domein wellicht noodzakelijk is om zo'n hoge informatiedichtheid te hebben (bijv. interne temperatuur van computerchip-fabricatiemachine).

Daarnaast is de context waarin je het uiteindelijke systeem wilt gebruiken van belang: wanneer je bijv. een model wilt hebben dat elke 5 minuten een nieuwe voorspelling doet, dan is het aan te raden om dezelfde tijdsperiode in je data te hanteren (en dus bijvoorbeeld niet data aggregeren over 15 minuten). Over het algemeen geldt dat je ten minste zo frequent wilt meten, als dat je wilt voorspellen. Bijvoorbeeld: als ik wil dat mijn model elke 15 minuten een nieuwe voorspelling doet, dan is het aan te raden om ook (ten minste) eens per 15 minuten te meten – indien mogelijk. Als je dit niet doet, loop je het risico dat het model noodgedwongen met 'verouderde' data werkt.

3.1.5 Nieuwe data

Hoe ga je om met nieuwe data? Is het noodzakelijk om de bestaande dataset periodiek aan te vullen met nieuwe gegevens? En zo ja: hoe vaak moet de dataset geüpdatet worden? Is nieuwe data eenvoudig toe te voegen aan de huidige data? Moeten er voorbewerkingen uitgevoerd worden voordat deze aan een bestaande dataset toegevoegd kunnen worden? Zo ja: welke? Kost dit (veel) tijd, of kan dit redelijkerwijs geautomatiseerd worden?

Het antwoord op deze vragen geeft je een indicatie hoeveel moeite je kunt steken in het inrichten van het automatiseren van het proces om nieuwe data te verkrijgen en te integreren in de bestaande dataset.

Voorbeeld

Om de aansturing van een Smart Indoor Climate systeem te kunnen optimaliseren wordt gebruik gemaakt van verscheidene bronnen. Zo wordt er zowel gebruik gemaakt van interne data (huidige ventilatie-stand, temperatuur, beweging en CO₂-concentratie in verscheidene ruimtes) als van externe data (weer-data van het KNMI). De interne gegevens worden opgeslagen in een (lokale) database, en de externe gegevens worden middels een API-call binnengehaald.

3.2 Het combineren van gegevens uit verschillende bronnen

Doel

Nadat je data van verschillende bronnen hebt gemeten, is het waarschijnlijk wenselijk om de gegevens van alle verschillende plekken en formats op te slaan en samen te voegen op één centrale locatie. Om dit te bewerkstelligen is het belangrijk om stil te staan bij de verschillende mogelijkheden en mogelijke drempels. In dit hoofdstuk belichten we kort de mogelijkheden, uitdagingen en mogelijke valkuilen met betrekking tot het combineren van verscheidene bronnen.

Uitleg

We gaan in op verscheidene mogelijkheden en keuzes m.b.t. het combineren van data van verschillende bronnen. Zo weet je na het lezen van dit hoofdstuk wat het verschil is tussen 'Tall' en 'Wide' stijlen van dataset-structurering (en wat de voor- en nadelen zijn), en wat verschillende manieren zijn om twee of meer tabellen met informatie samen te voegen.

3.2.1 Tall en Wide: twee manieren om een dataset te structureren

Over het algemeen zijn er twee manieren om je data te structureren in een dataset: *Tall* of *Wide* (ook wel *Narrow/Wide* of *lang/breed* genoemd). **Error! Reference source not found.** 3.2 toont een voorbeeld voor beide representaties.

TIJDSTIP	VARIABELE	METING
2022-03-14 9:00	CO2	434
2022-03-14 9:00	Binnentemperatuur	20.4
2022-03-14 9:00	Buitentemperatuur	17.3
2022-03-14 9:03	Beweging	TRUE
2022-03-14 9:05	CO2	441
2022-03-14 9:05	Binnentemperatuur	20.2

pivot melt

TIJDSTIP	CO2	BINNENTEMPERATUUR	BUITENTEMPERATUUR	BEWEGING
2022-03-14 9:00	434	20.4	17.3	N/A
2022-03-14 9:03	N/A	N/A	N/A	TRUE
2022-03-14 9:05	441	20.2	N/A	N/A

Figuur 3.2: De Tall en Wide representaties van data

De *Tall* representatie heeft voor elke meting een aparte rij. Dit heeft als voordeel dat het eenvoudig uit te breiden is met nieuwe (soorten) metingen zonder grote aanpassingen aan de tabel. Ook is het eenvoudig om data met verschillende meetfrequenties in één dataset te stoppen. Aan de andere kant is het voor mensen lastiger om de data te interpreteren, doordat bijvoorbeeld missende waarden minder goed opvallen, en data die dezelfde informatie beschrijft (bijv. CO2 in bovenstaande tabellen) visueel (ver) uit elkaar ligt.

In een *Wide* representatie zitten de verschillende meetvariabelen elk in hun eigen kolom. Elke rij beschrijft op deze manier een *sample*: de metingen van de verscheidene meetvariabelen die bij elkaar horen. Wanneer op een gegeven moment een meting

ontbreekt (zoals BEWEGING om 9:00), wordt hier N/A (Not Available) ingevuld. Het voordeel van deze representatie is dat alle informatie over één tijdstip op één regel staat. Het nadeel is dat bij verschillende meetfrequenties veel ontbrekende waarden kunnen ontstaan.

Verschillende frameworks en programmeertalen bieden hun eigen manieren om van een *wide* representatie een *tall* representatie te maken en vice versa. Een aantal voorbeelden zijn de *tidyr* package voor **R**, of de **python** package *panda's* met de functies *pivot* en *melt*.

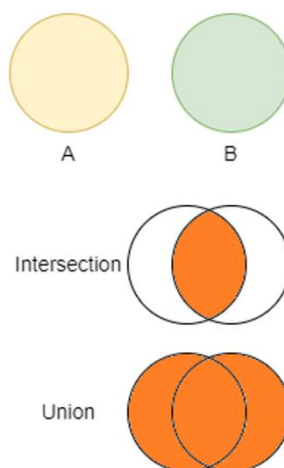
Veel Machine Learning/ AI modellen en algoritmen verwachten als input data die een *wide* representatie heeft. Elke kolom heeft dan namelijk een vaste 'betekenis' waar de algoritmes onder de motorkap gebruik van maken.

Het feit dat ontbrekende waarden in een *wide* representatie expliciet weergegeven worden is een zegen en een vloek. Aan de ene kant zul je in de praktijk vaak missende waarden moeten invullen of observaties moeten schrappen (zie 3.4), maar aan de andere kant dwingt het je ook om na te denken over de data die je het model als input geeft. Bovendien geeft het je een concreet beeld van hoe vaak welke metingen ontbreken (en of dit logischerwijs te verklaren is).

3.2.2 Samenvoegen van verschillende tabellen/ databases

Bij het samenvoegen van verschillende bronnen is het belangrijk om stil te staan bij wat je wel en niet kan samenvoegen. Het kan al misgaan bij iets simpels als de *timestamps* (tijdstip van meting): wanneer de ene databron wel gebruik maakt van zomer- en wintertijd en de andere niet, dan loopt je data al uit elkaar.

Over het algemeen combineer je data uit meerdere bronnen door in beide bronnen een zogenaamde *index* kolom te definiëren met gelijksoortige waarden (bijvoorbeeld een tijdstip). Afhankelijk van welk platform je gebruikt (bijv. een SQL-database, Excel tabel, CSV, panda's dataframe, etc.) zijn de mogelijkheden en exacte namen van functies net iets anders.



Figuur 3.3: Intersection en Union operaties op twee datasets

In eerste instantie volstaat het om simpelweg de Union te nemen van alle data (zie Figuur 3.3). Op deze manier heb je alle gegevens die in beide datasets voorkomen. Zoals gezegd: data weggooien kan altijd nog.

Ook kan het voorkomen dat dezelfde informatie in beide tabellen staat. Zo heeft het KNMI bijvoorbeeld een dataset met zon-gerelateerde data en een dataset met bewolingsdata. Beide datasets bevatten dezelfde breedte- en lengtegraden van de weerstations. In eerste instantie is het aan te raden om de 'overbodige' informatie te laten staan. In 3.5 gaan we in op het opschonen en verwijderen van dubbele data.

Zorg er tenslotte voor dat in de resulterende dataset de kolommen een duidelijke (= beschrijvende, leesbare) naam hebben. Dit maakt de rest van de analyse een stuk inzichtelijker.

Voorbeeld

Voor een Smart Indoor Climate systeem wordt gebruik gemaakt van verschillende bronnen, sommige intern en andere extern. Van de eigen sensoren wordt elke 5 minuten de huidige status/meting in een database geplaatst. De externe data van het KNMI wordt middels een API elke 10 minuten opgehaald en aan de interne database toegevoegd. De KNMI-data bevatten behalve het tijdstip geen overlap met de interne data, en kan dus simpelweg in zijn eigen kolom in de dataset terecht komen.

In dit geval heeft de externe data (weerdata van KNMI) een lagere tijdsresolutie dan de interne data (eigen sensoren). Dit heeft als consequentie dat voor de helft van de observaties die in de database terecht komen geen buitentemperatuur beschikbaar is. In paragraaf 3.4 gaan we in op het verwerken van ontbrekende gegevens.

Meer informatie

Tabellen/ dataframes combineren in [python](#), [SQL](#) en [MongoDB](#).

3.3 Het vaststellen van het type, het bereik, en de betrouwbaarheid van gegevens

Doel

In deze sectie stellen we het bereik en de mogelijke waarden vast van de verschillende meetwaarden die wij verwachten. Dit heeft als doel om de kwaliteit van de ruwe data vast te stellen. Waarden die (ver) buiten ons verwachte bereik vallen zijn wellicht verkeerd ingevoerd of het resultaat van een defecte sensor.

Uitleg

Door vast te stellen welke waarden de data mogelijkwijs aan zou kunnen nemen creëren we een referentiekader en 'sanity check' m.b.t. de metingen. Hiermee kunnen we eventuele foutieve of onbetrouwbare data opsporen en verwijderen of anderzijds vervangen.

3.3.1 Het data type en mogelijke bereik van meetwaarden

Bepaal voor elke meetvariabele wat het type data is dat je zou verwachten. Zo kan het voorkomen dat een kolom met temperaturen bestaat uit de volgende data: [16.3, 18.5, '19.3', 19.1, ..]. In dit geval hebben we meerdere datatypes in één kolom: floats en een string ('19.3'). We verwachten bij de temperatuurkolom alleen floats, en zullen dus alle elementen die nog geen float zijn naar een float moeten omzetten (zie Tabel 3.2). Ook kun je hier kijken of variabelen van een bepaald type eigenlijk een ander type beschrijven (zo zou 'Vier' misschien juist als 4 in de dataset terecht moeten komen).

Door van tevoren voor de verschillende variabelen een mogelijk bereik te definiëren kun je snel bepalen of er (significante) meetfouten zijn in de ruwe data.

Naam	Voorb.	Beschrijving
Int	1, 834, -69292	Gehele getallen
Float	0.33, 56.23, -82.0	Kommagetallen
String	'Hond', 'Sint Jansstraat'	Woorden of zinnen (tekst)
Boolean	TRUE, FALSE	Binair - waar/onwaar
Object	Dict(), class instances	Container van

Tabel 3.2: Een greep uit mogelijke data types

3.3.2 Betrouwbaarheid sensordata

Is een deel van de data afkomstig van sensoren (denk aan CO2, temperatuur, waterpeil etc.)? Verzeker jezelf er dan van dat de sensoren goed gekalibreerd zijn. Let ook op de locatie van sensoren: een verkeerd geplaatste sensor kan zorgen voor onbetrouwbare metingen! Zo zal bijvoorbeeld een temperatuursensor die de kamertemperatuur moet meten, maar als deze verkeerd geplaatst is (in de zon, naast een hittebron zoals een verwarming of PC), al snel onbetrouwbare data opleveren.

Voor een buitentemperatuur in Nederland is het bijvoorbeeld redelijk om te stellen dat waarden onder de -25 graden Celsius en boven de 45 graden Celsius waarschijnlijk het resultaat van een meetfout zijn.

3.3.3 Betrouwbaarheid handmatig ingevoerde data

Is er sprake van handmatig ingevoerde data (overgetypte vragenlijsten, handmatig tellen, labelen, classificeren etc.)? Controleer dan extra op fouten die voor kunnen komen zoals het invoeren van dubbele getallen (66 i.p.v. 6), extra nullen (1000 i.p.v. 100) en onmogelijke waarden (-3 i.p.v. 3). In het geval van meer subjectieve data (bevat een röntgenfoto wel of geen aandoening?) speelt ook de expertise van degene die de data invoert een (grote) rol. Een manier om hier mee om te kunnen gaan, is meerdere experts te vragen om hetzelfde meetpunt te beoordelen.

Hiervoor kun je kijken naar de inter-rater agreement, waarbij je kijkt hoe vaak verschillende beoordelaars het met elkaar eens zijn over hetzelfde meetpunt. Een andere mogelijkheid is gebruik maken van de intra-rater agreement, waarmee je hetzelfde meetpunt vaker (in willekeurige volgorde) langs laat komen bij dezelfde expert, en kijkt of ze het consistent beoordelen. Zelfs bij 'triviale' labeling-problemen ('bevat een foto een hond of een kat?') kunnen er nog foutjes in sluipen doordat de invoerder verkeerd klikt, of even niet oplet.

3.3.4 Het toetsen van de ruwe data

Nadat het bereik en de data typen voor je verschillende metingen zijn vastgesteld, kun je toetsen of de ruwe data aan de verwachtingen voldoet. Mocht dit niet het geval zijn, kijk dan in eerste instantie of er een logische verklaring kan zijn voor de observatie (misschien is een CO2 meting wel hoger dan verwacht omdat er twintig mensen in een kleine ruimte een borrel aan het houden zijn). Mocht er geen eenduidig antwoord zijn (omdat bijvoorbeeld de benodigde expertise ontbreekt), laat de meting dan in eerste instantie staan. Op een later moment in de data cycle gaan we nog in op het visualiseren en beschrijven van de data. Voor niet-numerieke data (bijv. tekstuele data) kan het vaak uit om alle unieke waarden weer te geven. Als één van de variabelen in je data bijvoorbeeld een dier beschrijft, dan geeft het weergeven van alle unieke waarden een snel overzicht van mogelijke typfouten en overlap.

Count	DIER
376	'Hond'
311	'Kat'
163	'kat'
150	'hond'
73	'Cat'
11	'Dog'
1	'Hondd'

Tabel 3.3: Alle unieke waarden van de kolom 'DIER' met aantal observaties

Tabel 3.3 geeft een voorbeeld van hoe handmatig ingevoerde data fouten kan bevatten. Er staan slechts twee verschillende dieren beschreven in de dataset (honden en katten), maar toch zijn er zeven verschillende (unieke) strings in de kolom 'DIER'. Een deel van de overlap kan weggehaald worden door eerst alle strings naar lowercase te veranderen (maakt van 'Hond' 'hond'). In Excel zijn de unieke waardes eenvoudig te zien met de optie Data -> Filter.

Om in de toekomst dergelijke ambiguïteiten te voorkomen, is het aan te raden om bij data-entry de kans op invoerfouten te verminderen door bijvoorbeeld een dropdown-menu met als opties Hond/ Kat te maken (of een Engelstalige versie die de antwoorden in de database alsnog als 'Kat' of 'Hond' opslaat).

Meer informatie

Een beschrijving hoe de OECD haar datakwaliteit nastreeft: [OECD-data kwaliteit](#)

Hoofdstuk 4 van het scoping document (zie paragraaf 2.2) helpt om bovenstaande toe te passen en vast te leggen.

3.4 Het verwerken van ontbrekende gegevens

Doel

In de praktijk is er vaak sprake van ontbrekende gegevens: van sommige variabelen is wellicht elke 5 minuten nieuwe informatie beschikbaar terwijl andere variabelen slechts één keer per uur gemeten worden, of bepaalde sensoren zijn niet altijd actief. Hierdoor ontstaan gaten in je dataset. Voor een initiële analyse is dat (nog) geen probleem, maar verderop in het proces kan dit problemen opleveren – in het bijzonder bij het bouwen van verscheidene soorten Machine Learning modellen.

Uitleg

We gaan in deze sectie in op hoe je om kan gaan met ontbrekende gegevens. Zo behandelen we het verwijderen van observaties en het vervangen van ontbrekende datapunten door gemiddelden, geïnterpoleerde punten en zogenaamde forward-en backfill-methoden.

3.4.1 Verwijderen van ontbrekende data

Wanneer ontbrekende waarden slechts sporadisch optreden kan een observatie in zijn geheel verwijderd worden. Bij 'sporadisch' kun je denken aan bijvoorbeeld 1 op de 100 observaties heeft 1 of meer ontbrekende waarden. Hier zijn geen harde regels voor, maar dit moet al naar gelang de situatie worden bepaald.

Wanneer waarden vaker ontbreken kan dit verschillende betekenissen hebben: sommige meetvariabelen hebben wellicht een *event-driven* karakter (bijvoorbeeld een bewegingssensor) en sturen alleen informatie wanneer iets gebeurt. Andere meetvariabelen hebben wellicht een (significant) andere meetfrequentie (bijvoorbeeld 1 meting per uur in plaats van 1 meting per 5 minuten).

Wanneer het onzeker is of ontbrekende waarden zinvol kunnen worden vervangen en er later niet met ontbrekende waarden om kan worden gegaan, heeft het verwijderen van observaties de voorkeur. Dit aangezien anders de geïnterpoleerde of vervangen waarden wellicht geen accurate afspiegeling van de werkelijkheid zijn.

3.4.2 Vervangen van ontbrekende data

In plaats van het verwijderen is het ook mogelijk om ontbrekende waarden te vervangen door andere waarden. Hiervoor is het belangrijk om bekend te zijn met de dynamiek van de data die je meet: wanneer de meetfrequentie (veel) hoger is dan enige significante veranderingen in je data, dan is er meer mogelijk op het gebied van interpoleren/vervangen van ontbrekende data.

Als concreet voorbeeld: stel dat elke minuut interne sensordata wordt opgeslagen in een database. Daarnaast wil je ook gebruik maken van de buitentemperatuur. Deze wordt door het KNMI eens in de 20 minuten gepubliceerd. Omdat de buitentemperatuur in een interval van 20 minuten niet heel veel zal fluctueren, is het te verdedigen om de ontbrekende buitentemperatuurmetingen (van de 19 minuten die niet samenvallen met de KNMI-metmomenten) te vervangen door een goede benadering hiervan.

3.4.3 Methoden voor vervangen van ontbrekende numerieke gegevens

Voor het vervangen van ontbrekende gegevens zijn er verscheidene methoden beschikbaar die we hier kort beschrijven. Alle methoden behalve imputation verwachten dat de data 'numeriek ordinaal' is, dat wil zeggen dat de data uit cijfers bestaat en er een natuurlijk volgorde in de data zit (bijv. van vroeg naar laat, of van klein naar groot).

Imputation - Constant (al dan niet tijdsafhankelijk)

Eén van de eenvoudigste manieren van het invullen van ontbrekende gegevens is het invullen van een constante waarde. Meestal wordt hier het gemiddelde (of de meest

voorkomende waarde) in gestopt. Het risico hiervan is dat deze waarde (sterk) kan afwijken van de daadwerkelijke meting.

Een slimme manier om hier mee om te gaan is door gebruik te maken van de context van de ontbrekende waarde: door bijvoorbeeld het gemiddelde te nemen van een uur of van een dagdeel heb je een 'gelokaliseerd' gemiddelde dat waarschijnlijk dicht op de daadwerkelijke meting ligt.

Ook kan gebruik gemaakt worden van zogenaamde *Hot-deck imputation*. In dit geval wordt voor een ontbrekende observatie gezocht naar een andere meting die (m.b.t. de andere variabelen) het meest lijkt op de huidige observatie waarin de ontbrekende waarde wél aanwezig is. Deze waarde wordt vervolgens gebruikt om de ontbrekende waarde mee te vullen.

Origineel	CONSTANT	Interpolated	Nearest	FFill	BFILL
1	1	1	1	1	1
N/A	6	2.5	1	1	8
N/A	6	4	1	1	8
N/A	6	6.5	8	1	8
N/A	6	8	8	1	8
8	8	8	8	8	8
9	9	9	9	9	9

Tabel 3.4: Een kolom met ontbrekende waarden en verscheidene methoden om de ontbrekende waarden in te vullen

Interpolation

Een veelgebruikte manier om met missende data om te gaan is interpolation of interpolatie. In dit geval wordt een lineair verband gebruikt om alle ontbrekende waarden in te vullen. Dit is met name geschikt voor tragere processen (die niet heel snel heel veel veranderen) of kleinere overbruggingen (niet te veel ontbrekende waarden achter elkaar).

Tabel 3.4 toont in de eerste kolom een voorbeeld van een dataset met ontbrekende waarden, en het resultaat na interpolatie. Het risico is dat door de interpolatie waarden voor zouden kunnen komen die in 'de echte wereld' nooit voor zouden komen: stel dat bovenstaande grafiek het aantal bezoekers in een restaurant beschrijft; in dit geval bevat de data na interpolatie ineens metingen die nooit voor zouden kunnen komen, zoals 2.5 bezoekers!

Nearest

In het geval van een nearest fill methode wordt gekeken naar welk daadwerkelijk gemeten punt het dichtst bij het ontbrekende punt ligt. Het ontbrekende datapunt wordt dan vervangen door dit dichtstbijzijnde punt.

Tabel 3.4 toont het resultaat van een nearest fill methode. Het voordeel van een nearest fill methode is dat alle waarden die middels deze methode toegevoegd worden ook voor (kunnen) komen in de ruwe data. Daarentegen is het resultaat vaak vrij 'schokkerig'; d.w.z. dat er een abrupte sprong kan zijn wanneer de waarde voor en na de ontbrekende waarde ver uit elkaar liggen.

Fill (backward- en forward-fill)

Tenslotte hebben we nog de forward fill en backward fill methodes. Deze kijken bij ontbrekende waarden respectievelijk naar de vorige niet ontbrekende meting (FFILL – forward fill; het naar voren invullen) en de eerstvolgende niet ontbrekende meting (BFILL – back fill; het terug-invullen vanaf een observatie).

Het meest belangrijke bij het bepalen hoe je omgaat met ontbrekende waarden is dat het recht doet aan de dynamiek van de data. Wanneer door de gekozen methode data punten ontstaan die in de ruwe data nooit voor zouden kunnen komen is dit waarschijnlijk een

indicatie dat de gekozen methode een minder goede benadering is van de daadwerkelijke data.

Meer informatie

Een [beschrijving](#) hoe in het panda's-framework met missende data gewerkt kan worden.

3.5 Dubbele metingen verwijderen

Doel

Op dit moment heb je een dataset die het product is van meerdere bronnen en metingen. Het kan natuurlijk voorkomen dat bepaalde data dubbel in de dataset terecht is gekomen. In deze sectie gaan we in op het verwijderen van dubbele metingen (duplicates).

Uitleg

Dubbele metingen verstoren een dataset in de zin dat de geduplicateerde observaties vaker in de dataset aanwezig zijn dan in het onderliggende systeem (de 'echte wereld'). In het geval van een exacte match is een dubbele meting eenvoudig te verwijderen, maar in andere gevallen – zoals wanneer de data hetzelfde beschrijft maar een net andere vorm heeft – is dit minder eenvoudig.

Tijdstip	Naam_station	Label_station	Longitude	Longitude
09:00	Testlocatie Aalsmeer A	Locatie Aalsmeer A	4.7	4.7
09:00	EII	EII	4.65	4.65
09:00	EII	EII	4.65	4.65
09:00	Waarneemterrein de Bilt	De Bilt	4.25	4.25

Tabel 3.5: Een dataset met duplicate waarden in rijen en kolommen

Tabel 3.5 toont een (klein deel van een) dataset met duplicate values: de kolommen NAAM_STATION en LABEL_STATION beschrijven dezelfde informatie, maar hebben net iets andere namen. De kolom LONGITUDE is in zijn volledigheid geduplicateerd en hiervan kan simpelweg een van beide worden verwijderd.

Voor de NAAM_STATION en LABEL_STATION kolommen moet eerst vast worden gesteld of de gegevens daadwerkelijk hetzelfde beschrijven. In dit geval doen we dat door in de volledige dataset te kijken of de latitude en longitude (breedtegraad en lengtegraad) voor beide labels overeenkomen. Is dit het geval, dan kunnen één van beide kolommen weggooien zonder informatie weg te gooien.

Ten slotte is er nog een rij met duplicate values: de tweede en derde rij zijn exact equivalent aan elkaar, en één van de twee kan zonder problemen verwijderd worden. Tabel 3.6 toont het resultaat na het verwijderen van de duplicate data.

Tijdstip	Naam_station	Longitude
09:00	Testlocatie Aalsmeer A	4.7
09:00	EII	4.65
09:00	Waarneemterrein de Bilt	4.25

Tabel 3.6: Dataset na verwijderen duplicate waarden

Meer informatie

Find en remove duplicates in [Excel](#), [SQL-server](#) en [Pandas](#)

3.6 Data verrijken

Doel

Op dit punt hebben we een opgeschoonde dataset zonder dubbele of ontbrekende waarden. Voor het analyseren van de dataset, kunnen we de dataset nog verrijken door nieuwe datapunten af te leiden uit de bestaande gegevens. Dit levert zogenaamde *created features*. Na het lezen van deze stap zul je weten wat *created features* zijn, en hoe deze een dataset potentieel kunnen verbeteren.

Uitleg

Afgeleide variabelen – *created features* – zijn variabelen die het resultaat zijn van een manipulatie of operatie op gemeten waarden. Een voorbeeld hiervan is het omvormen van een *event-driven* variabele. Ter herinnering: een *event-driven* variabele is er één die niet met een vaste regelmaat wordt gemeten (bijvoorbeeld: de temperatuur elke 5 minuten), maar één die afhankelijk is van een gebeurtenis (bijvoorbeeld: een bewegingssensor).

We laten nu een aantal voorbeelden zien hoe je van een ruwe meting (beweging op een bepaald moment) afgeleide meetwaarden kan maken.

Tijdstip	Buitemtemperatuur	Beweging
9:00	8.3	TRUE
9:05	8.4	N/A
9:06	N/A	TRUE
9:08	N/A	TRUE
9:10	8.4	N/A
9:15	8.5	N/A
9:20	8.5	N/A
9:25	8.6	N/A

Tabel 3.7: Periodieke en event-driven data

TIJDSTIP	Buitemtemp.	Min_sinds_BEW	bewogen_prev_5M	Aantal_Beweg_5m
9:00	8.3	0	TRUE	1
9:05	8.4	5	FALSE	0
9:10	8.4	2	TRUE	2
9:15	8.5	7	FALSE	0
9:20	8.5	12	FALSE	0
9:25	8.6	17	FALSE	0

Tabel 3.8: Dezelfde informatie als in Tabel , met created features

Tabel 3.7 en Tabel 3.8 tonen respectievelijk een dataset met *event-driven* informatie (wanneer is er beweging gedetecteerd?) met daardoor onregelmatige tijdsintervallen en een dataset met variabelen die afgeleid zijn van de BEWEGING variabele. In Tabel 3.8 zijn drie *derived features* – of afgeleide variabelen – gecreëerd. Elke afgeleide variabele beschrijft soortgelijke informatie als de ruwe BEWEGING data, maar op een net andere manier zodat het gecombineerd kan worden met de periodiek gemeten temperatuurdata.

Een andere situatie waarin afgeleide variabelen nuttig kunnen zijn, is wanneer vanuit expert- of domeinkennis bekend is dat een bepaalde relatie tussen beschikbare variabelen relevant is. Bijvoorbeeld: voor een toepassing worden prijs en verkochte aantallen vastgelegd in een dataset. Vanuit de domeinkennis is bekend dat voor het probleem dat we op willen lossen de omzet ook relevant is. In dit geval hebben we de benodigde informatie (prijs en aantal) en kunnen we zelf de afgeleide feature omzet toevoegen aan de data. Daarmee helpen we het model 'op weg' door de bestaande inzichten omtrent de relevante relatie tussen de variabelen expliciet te maken.

Tenslotte kunnen we ook variabelen toevoegen die meer informatie geven bij het aggregeren (of samenvoegen) van meerdere datapunten. Bijvoorbeeld: stel dat je 300 metingen hebt van een bepaalde variabele in een periode van 5 minuten. In je uiteindelijke dataset wil je elke wellicht 5 minuten een voorspelling doen. In dit geval zou je bijv. de meest recente of gemiddelde waarde van de 300 metingen kunnen gebruiken, maar je zou daarnaast ook de distributie kunnen beschrijven (gemiddelde + standaarddeviatie), of zaken als de minimale en maximale waarde van de 300 metingen.

Meer informatie

Relevante zoektermen: *Feature Engineering, Feature Creation, Feature Extraction, Derived Features.*

3.7 Exporteren data

Doel

De laatste stap van de data cleaning cycle is het exporteren oftewel opslaan van de dataset. We beschrijven een greep uit de meest gebruikte opties met het oog op schaalbaarheid, uitbreidbaarheid en toegankelijkheid.

Uitleg

Het exporteren van de opgeschoonde dataset rondt het proces van de data cleaning cycle af. In deze zin verschilt het opslaan van een AI-gerelateerde dataset niet van andersoortige data: het kan lokaal in een .xls of .csv bestand worden opgeslagen, of in een MongoDB, (MySQL, postgresql of andersoortige database worden geplaatst.

Uiteindelijk is het bij de keuze voor het een database of opslagmedium met name belangrijk om stil te staan bij welke functionaliteiten je verwacht nodig te hebben, welke functionaliteiten de verschillende opties bieden, en – misschien nog wel het belangrijkste – welk type database/ opslag al gebruikt wordt binnen de organisatie.

Sta stil bij hoe vaak en op welke manier je de gegevens van de opgeschoonde dataset aan zou willen vullen met nieuwe data. Moeten en kunnen updates geautomatiseerd gebeuren? Hoe (makkelijk) kun je gegevens uit de database halen als input voor een model, visualisatie of statistische analyse, en hoe makkelijk kun je bestaande waarden aanpassen, toevoegen of overschrijven?

Ten slotte nog een advies over de opslag van waardevolle of gevoelige gegevens (bijv. persoonsgegevens): zorg ervoor dat je je data op een centrale plek hebt opgeslagen, bij voorkeur met back-ups. Is alle data op één locatie beschikbaar, of bevindt deze zich op verschillende locaties (bijv. verschillende databases of computersystemen)?

Ook is het belangrijk om stil te staan bij wie toegang moet (kunnen) krijgen tot de gegevens. Voor gevoelige en/of waardevolle gegevens geldt over het algemeen het 3-2-2 systeem: 3 kopieën van de data op ten minste 2 fysieke locaties (bijvoorbeeld op een lokale server en op de Cloud) met tenminste 2 verschillende opslagmedia (Harde schijf, SSD).

4 Data-exploratie en -visualisatie

Een belangrijke stap in het ontwikkelen van een AI-systeem is het verkennen van de verschillende gegevens die aanwezig zijn in de dataset. Dit doen we door de gegevens te beschrijven, analyseren en visualiseren. Dit proces staat ook wel bekend als *Exploratory Data Analysis* (EDA). Door EDA toe te passen worden mogelijke verbanden tussen variabelen (*features*) duidelijk, en kunnen we onze dataset zodanig bijschaven dat we beter voorbereid aan het daadwerkelijke toepassen van modellen toe kunnen komen.

Door meer inzichten te krijgen in de dataset kunnen we de gegevens beter inzetten; soms leidt een analyse zelfs tot een *reframing* van het probleem dat opgelost wordt. Denk hierbij aan het omvormen van een regressieprobleem¹ (“hoeveel broden moeten er morgen worden gebakken?”) naar een binair classificatieprobleem² (“Moeten er meer broden worden gebakken dan op een normale maandag?”) of omgekeerd. Ook kunnen aan de hand van de inzichten die opgedaan zijn nieuwe vragen ontstaan.

Daarnaast kunnen mogelijk redundante (overbodige) of foutieve waarden worden ontdekt die tot nog toe niet zijn opgevallen. Hierbij geven we mogelijke vervolgstappen en consequenties voor bepaalde inzichten.

Conceptueel ligt dit hoofdstuk het dichtst bij het veld *Data Science*: het verkrijgen van inzichten over (grote hoeveelheden) data middels (automatische) analyses en visualisaties.

In de twee onderdelen van dit hoofdstuk (4.1.1 en 4.1.2) wordt respectievelijk ingegaan op de statistische analyse en visualisatie van de gegevens. In tegenstelling tot Hoofdstuk 3 is hier geen gebruikelijke volgorde: de visualisatie en statistische analyse kunnen als één geheel beschouwd worden, en door elkaar gebruikt worden. Zowel de statistische analyse als de visualisatie zullen waarschijnlijk vragen oproepen die de andere methode kan beantwoorden.

Voor beide typen analyse (statistisch en visueel) is het praktisch om scherp te hebben wat hetgeen is dat je wilt modelleren: wil je een getal voorspellen (*regressie*), aan een categorie of klasse toewijzen (*classificatie*), of simpelweg kijken welke verbanden er in de data aanwezig zijn (*unsupervised learning*)? Afhankelijk hiervan zal de analyse er anders uitzien; in de eerste twee gevallen heb je namelijk (als het goed is) een “*label*” of “*ground truth*” van wat je wilt classificeren of voorspellen (bijv.: foto’s zijn gelabeld met het object dat er op staat, of er is bekend hoeveel broden er de dag na een bepaalde dag verkocht zijn).

Dit zogenaamde label zal in de analyses centraal staan: het identificeren van mogelijke correlaties of sterke verbanden tussen je metingen en je label is zeer waardevol, en kan helpen bij het selecteren van een geschikt AI-model (zie Hoofdstuk 7).

¹ In een regressieprobleem probeer je een hoeveelheid of kwantiteit te modelleren, bijvoorbeeld hoeveel broden er op een dag verkocht worden, of hoe warm het morgen wordt.

² In een classificatieprobleem wordt aan de hand van de input 1 of meerdere klassen toegewezen – classificatie. Denk hierbij aan het herkennen van wat er op een foto staat (hond, kat, boot, etc.). Binaire classificatie is een specifiek geval waarin er slechts 2 klassen zijn – bijvoorbeeld wel of geen dier op een foto.

Als voorbeeld wordt in dit hoofdstuk de Iris³ dataset gebruikt. Dit is een dataset bestaande uit 150 *records* van drie typen bloemen met eigenschappen als afmetingen van de kroon- en kelkbladeren. De namen van de bloemen zijn in dit voorbeeld de *labels*, en het probleem dat we willen modelleren is het classificeren van een bloem gebaseerd op deze afmetingen.

NB: In dit hoofdstuk wordt een kleine laag-dimensionale classificatie dataset gebruikt. Voor andere typen data (regressie, time series, unsupervised) kunnen andere typen statistische analyse en visualisatie relevant of geschikt zijn. Bij unsupervised learning zal bijvoorbeeld clustering en dimensionality reduction relevant zijn, terwijl bij time series data autocorrelation, time-lagged cross correlation en seasonality relevant zijn.

³ <https://archive.ics.uci.edu/ml/datasets/iris>

4.1 Een statistische analyse: methoden, inzichten en vervolgstappen

Doel

Door een statistische analyse uit te voeren op de dataset die je tot je beschikking hebt, kan de probleemstelling scherper gedefinieerd worden, kunnen overbodige of redundante gegevens verwijderd worden, en kunnen mogelijke valkuilen m.b.t. modelkeuze geïdentificeerd worden.

Uitleg

We zullen hieronder de meest gangbare (statistische) analyses kort belichten. Dit doen we aan de hand van de Iris dataset. Zoals genoemd gaat bij een exploratieve data analyse de statistiek vaak hand in hand met een bijbehorende visualisatie. Daarom worden indien van toepassing ook figuren en andere visualisaties van de (statistische) resultaten getoond. Hierbij wordt doorverwezen naar de sectie in 4.2 die verdere uitleg over de totstandkoming en het doel van de betreffende visualisatie.

4.1.1 Descriptive statistics

Eén van de eerste stappen bij het analyseren van een dataset is kijken naar de descriptive statistics, oftewel de beschrijvende statistiek. Hierbij wordt voor elk van de features statistische informatie uitgerekend zoals het gemiddelde, standaard deviatie, minimum en maximum waarde, aantal records (of missende waarden), percentielen en meer (zie **Error! Reference source not found.**).

	lengte kelkblad (cm)	breedte kelkblad (cm)	lengte kroonblad (cm)	breedte kroonblad (cm)
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Figuur 4.1: Beschrijvende statistiek voor de Iris dataset

Beschrijvende statistiek wordt over het algemeen gebruikt om een eerste beeld te vormen van de (distributie van) de verschillende metingen of *features*. Het kan namelijk van belang zijn om te weten of een variabele normaal verdeeld is, of er *outliers* (sterk afwijkende waarden die zaken als het gemiddelde sterk beïnvloeden) zijn, etc.

Alle drie typen bloemen (3 *classes*) die in de Iris dataset voorkomen zijn hier samengevoegd bij het uitrekenen van de statistische (distributie) informatie. In dit geval kan het interessant zijn om de gegevens uit te splitsen per klasse – vaak is er in ieder geval een aanleiding om te denken dat de verschillende klassen zichzelf ook op verschillende manieren uiten in de features of metingen.

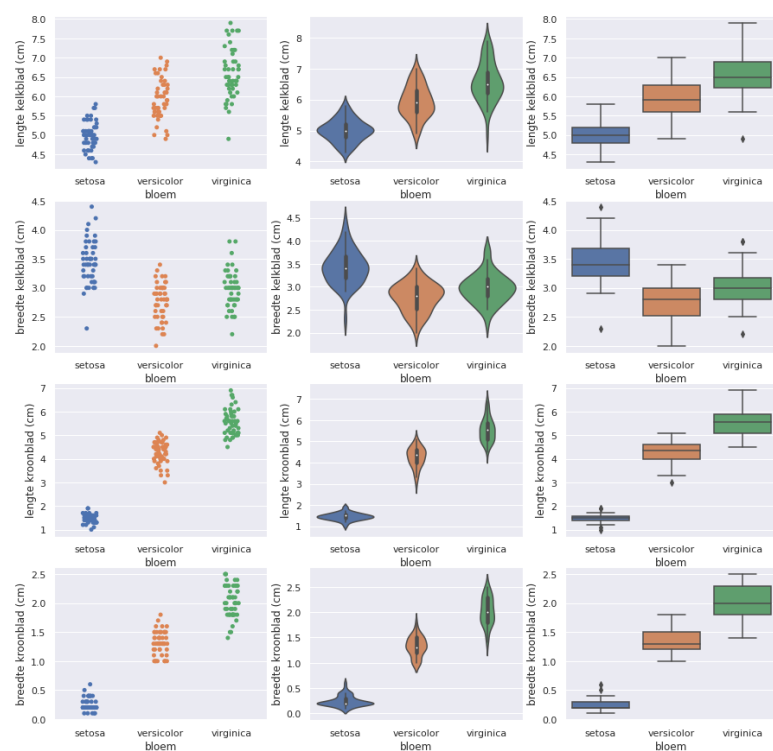
In **Error! Reference source not found.** is een voorbeeld geschetst van de lengte van het kelkblad voor elk van de drie typen bloemen. We kunnen nu in één oogopslag al verschillende conclusies trekken: de dataset is goed gebalanceerd – dat wil zeggen dat het aantal observaties of metingen voor de verschillende klassen (in de tabel aangegeven met

count⁴). De verdeling van de *setosa* bloem verschilt redelijk veel van de andere twee klassen, die dicht bij elkaar liggen.

lengte kelkblad (cm)								
bloem	count	mean	std	min	25%	50%	75%	max
<i>setosa</i>	50.0	5.006	0.352490	4.3	4.800	5.0	5.2	5.8
<i>versicolor</i>	50.0	5.936	0.516171	4.9	5.600	5.9	6.3	7.0
<i>virginica</i>	50.0	6.588	0.635880	4.9	6.225	6.5	6.9	7.9

Figuur 4.2: De statistieken van de lengte van het kelkblad voor de verschillende typen bloemen uit de Iris dataset

Ook hier geldt het gezegde ‘een plaatje zegt meer dan duizend woorden’; **Error! Reference source not found.** (hieronder) worden de distributies van de verschillende *features* op verschillende manieren gevisualiseerd voor onze 3 typen bloemen.



Figuur 4.3: Distributies van de features in de Iris dataset op drie verschillende manieren. Van links naar rechts: swarm plot, violin plot, boxplot. Op rij-niveau laten de verschillende grafieken dezelfde data op verschillende manieren zien.

Hier is in één oogopslag te zien dat de *setosa* bloem duidelijk verschilt van de andere typen bloemen. De andere twee klassen liggen dicht bij elkaar. Om ze van elkaar te kunnen scheiden zul je waarschijnlijk naar interactie-effecten (combinaties van features) moeten kijken. Hierover meer in de sectie over visualisatie.

⁴ Mochten de klassen niet goed gebalanceerd zijn – zoals in de praktijk vaak het geval zal zijn – dan kan dat bij de modelselectie of -toepassing meegenomen worden. Relevante zoektermen: imbalanced classes, SMOTE, class_weights

4.1.2 Correlaties

Ten slotte is het aan te raden om correlaties tussen de verschillende *features* uit te rekenen. Correlaties beschrijven de verbanden tussen paren van *features*. De meest gebruikte correlatie-methode is *Pearson's r correlation coefficient*.

Op het moment dat een stijging in de ene helft van het paar altijd een (proportioneel gelijke) stijging in de andere helft van het paar veroorzaakt, spreken we van een perfecte (positieve) correlatie – uitgedrukt met +1. Op het moment dat een **stijging in de ene** waarde samenvalt met een **daling in de andere** waarde, spreken we van een negatieve correlatie (-1). Wanneer er geen (of een slecht) verband is tussen de twee waarden, is er sprake van geen tot een zwakke correlatie (rond 0).



Figuur 4.4: Correlation matrix van de Iris dataset. De drie soorten bloemen (*Setosa*, *Versicolor*, *Virginica*) hebben duidelijk verschillende correlaties met de verscheidene features ten opzichte van elkaar.

Error! Reference source not found. laat de correlaties tussen de verschillende metingen én de klasse-labels zien. Hierbij is duidelijk dat er een sterke (negatieve) correlatie is tussen de *setosa* klasse, en de breedte en de lengte van het kroonblad. Dit betekent concreet dat wanneer een bloem tot de *setosa* klasse behoort, de lengte en breedte van het kroonblad lager is dan wanneer het niet tot de *setosa* klasse behoort. Dit kan ook worden bevestigd door te kijken naar de distributies in **Error! Reference source not found.**

Wanneer een paar *features* een correlatie van exact +1 of -1 heeft⁵ is dit indicatief voor een zogenaamde redundante *feature*. Dit houdt effectief in dat de twee metingen op data-niveau dezelfde informatie beschrijven. Zo is bijvoorbeeld de correlatie van een buitentemperatuur in Celsius en dezelfde buitentemperatuur in Fahrenheit exact +1: hoewel ze met verschillende hoeveelheden omhoog en omlaag gaan bij een verandering, is de verandering altijd proportioneel **exact** hetzelfde. In termen van informatie binnen een dataset biedt het aanbieden van een temperatuur in Celsius en dezelfde temperatuur in Fahrenheit geen extra informatie, en kan één van beide het best uit de dataset verwijderd worden.

Meer informatie:

Een 'all-purpose' EDA package binnen python is de **pandas-profiling** package (<https://pypi.org/project/pandas-profiling/>). Deze berekent automatisch beschrijvende statistieken, correlaties, missende waarden en interacties tussen *features* voor een python **pandas** dataframe.

Zoektermen: **Exploratory data analysis**, pandas profiling report, kurtosis / scwness, (pearson) correlation, feature redundancy

⁵ Met uitzondering van de correlatie van een feature met zichzelf; dit is per definitie 1.

4.2 Data-visualisatie

Doel:

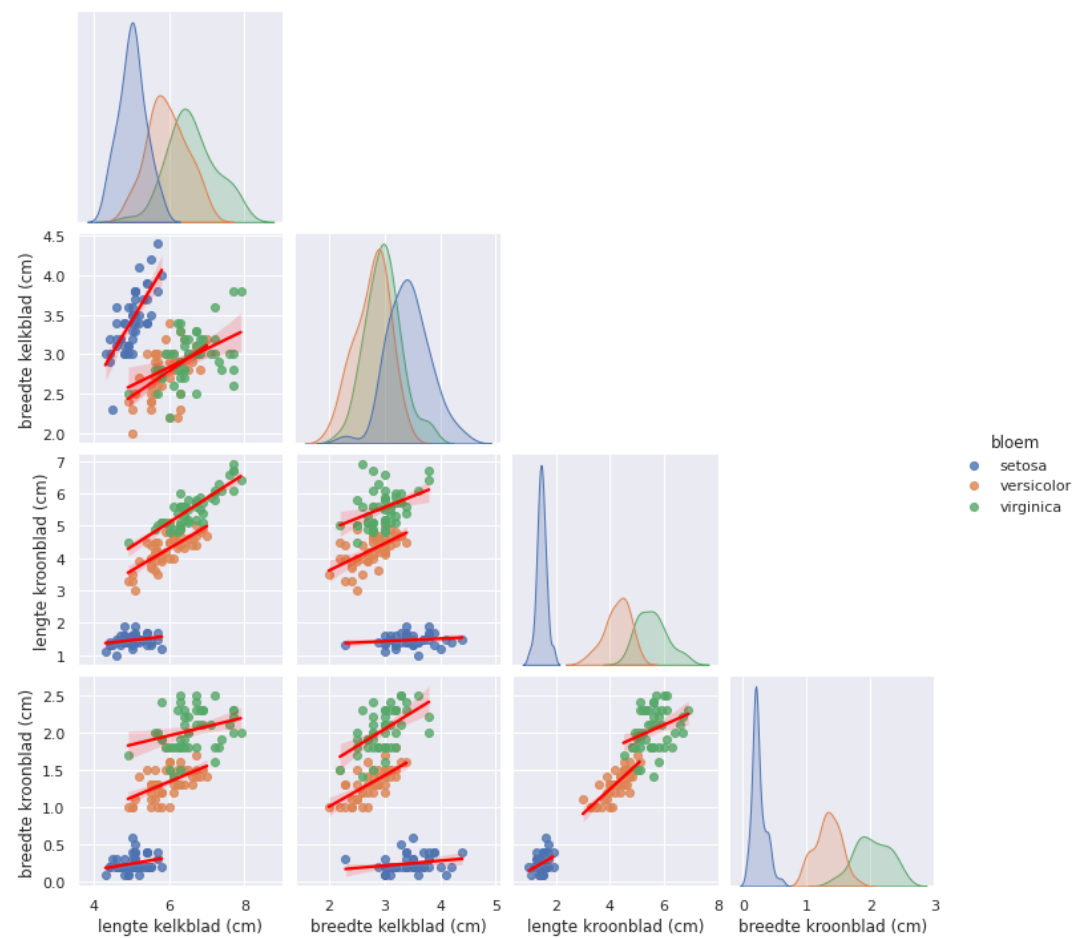
Het visualiseren van data is een belangrijke stap in het verkennen van een dataset. Door gegevens te visualiseren worden verschillen, afwijkingen en andere eigenaardigheden snel duidelijk. Ook geldt over het algemeen dat, wanneer wij als mens bijvoorbeeld bepaalde klassen in een visualisatie al goed van elkaar kunnen onderscheiden, een (goed) getraind AI model dit ook zou moeten kunnen doen

Uitleg:

In de meeste brede zin van het woord zijn er talloze manieren om dezelfde gegevens te visualiseren. Daarom is het in eerste instantie belangrijk om stil te staan bij de onderliggende vraag die beantwoord moet worden; In deze sectie gaan we in op de meest gebruikte visualisaties, en benoemen we kort hoe in dit geval omgegaan kan worden met hoog-dimensionale data (veel features).

4.2.1 Pair plots

Voor datasets met niet te veel features kan een *pair plot* worden opgesteld. Binnen een pair plot worden alle combinaties van *features* tegen elkaar afgezet, en worden de distributies van de individuele *features* gevisualiseerd. Zie **Error! Reference source not found.**



Figuur 4.5: Pairplot van de features van de Iris dataset. Hier worden telkens twee variabelen tegen elkaar afgezet. De rode lijnen zijn een lineaire regressie gefit op de verschillende bloemen met confidence interval (lichtrode achtergrond).

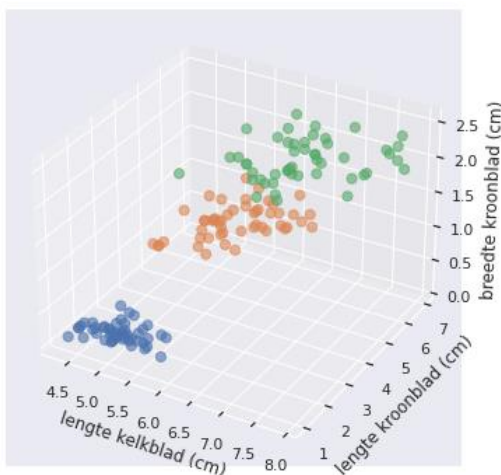
Doordat alle combinaties van variabelen tegen elkaar uit worden gezet, kan dit met een hoger aantal variabelen snel uit de hand lopen; met 4 variabelen gaat het nog om 6 combinaties, bij 20 variabelen zijn dit er al 190. In dit geval is het aan te raden om een selectie te maken van de variabelen. Andere manieren om met hoog-dimensionale datavisualisatie om te gaan worden hieronder besproken.

4.2.2 Hoog-dimensionale data

Wanneer je dataset uit een klein aantal (< 5) *features* of variabelen bestaat, is het nog goed te doen om met bovenstaande aanpakken een visualisatie te maken. Hierboven loop je al snel tegen problemen aan: een *pair plot* wordt veel te groot, line plots overlappen (te) veel met elkaar, en standaard (2D) plots zijn niet uitgerust om hoog-dimensionale gegevens weer te geven.

3, 4, en 5D visualisaties

Naast de fysieke dimensies (x- en y-as), kan een data-dimensie ook nog op andere manieren worden weergegeven. In het voorbeeld van de Iris dataset, kunnen we vrij eenvoudig 4 *features* tegelijkertijd visualiseren door gebruik te maken van de z-as in aanvulling op de x- en y-assen, en door kleur te gebruiken om het type bloem te visualiseren (zie **Error! Reference source not found.**). Andere manieren om extra *features* te visualiseren zijn door middel van grootte (hoe hoger een waarde, des te groter het te visualiseren object), vorm (bloemen zijn rondjes, bomen zijn kruisjes, dieren zijn driehoekjes etc.), doorzichtigheid (ook wel *alpha* genoemd) en meer.



Figuur 4.6: 3D scatter plot van de Iris dataset. Er worden 4 *features* tegelijkertijd weergegeven: het type bloem (de klasse) met kleur, de lengte en breedte van het kroonblad, en de lengte van het kelkblad op respectievelijk de y-, z-, en x-as.

Een potentieel nadeel van deze manier van visualiseren is dat het lastiger is om in één oogopslag te zien wat er aan de hand is – elk individueel object in je plot bevat erg veel informatie (positie, vorm, kleur, inhoud, doorzichtigheid en meer zijn allemaal betekenisvol).

Dimensionaliteitsreductie

Om dit probleem aan te kaarten kan worden gewerkt met dimensionaliteitsreductie. Bij dimensionaliteitsreductie wordt het aantal dimensies (of *features* of variabelen) gereduceerd. Dit kan door een selectie te maken (*feature selection*), of door een hoger dimensionale representatie te *mappen* op minder dimensies. Eén van de meest gebruikte methoden hiervoor is PCA, of *Principal Component Analysis*.

Meer informatie:

https://matplotlib.org/cheatsheets/_images/handout-beginner.png

https://matplotlib.org/cheatsheets/_images/handout-intermediate.png

https://matplotlib.org/cheatsheets/_images/handout-tips.png

Zoektermen: PCA, feature selection, Time Series Visualization, high-dimensional data visualization. EDA visualization

5 Het organiseren van het project

5.1 Een backlog voor AI-projecten

Doel

Het aanreiken van een voorbeeldbacklog met taken die passen bij de ontwikkeling van AI-systemen in het MKB.

Uitleg

Ontwikkelpromen worden tegenwoordig vaak volgens een agile werkwijze gedaan. Een centraal element bij agile is de backlog, de lijst met taken die op volgorde van prioriteit worden uitgevoerd. Bij het ontwikkelen van AI-systemen zijn er bepaalde taken die waarschijnlijk op de backlog thuishoren. Met deze standaard backlog krijg je als MKB deze taken overzichtelijk aangereikt.

Uiteraard is deze backlog een hulpmiddel en kunnen taken die niet van toepassing zijn of waar al in voorzien is, verwijderd worden.

Categorie	Naam	Omschrijving	Discussie	Hoe te demonstreren
Probleem Definitie/ Business Case	Bepaal waardepropositie	Bepaal de waardepropositie van het product (business canvas, waardepropositie canvas)		Business document
Probleem Definitie/ Business Case	Formuleer business hypothesen	Precieze aannames formuleren waarop waardepropositie is gebaseerd		Business document
Probleem Definitie/ Business Case	Test business hypothesen	Testen van aannames om te zien of ze gelden.	Op volgorde van prioriteit	Gedocumenteerde business experiments
Probleem Definitie/ Business Case	Opstellen business case v1	Is er een positieve business case voor het project?		Gedocumenteerde business case (evt. onderdeel van business canvas)
Probleem Definitie/ Business Case	Versie 2.0 van business canvas	Volgende iteratie van het business canvas op basis van de bijgestelde business hypothesen		
Data	Vereiste data specificeren	Zo gedetailleerd mogelijk bepalen welke data nodig is om het probleem op te kunnen lossen. Is deze data beschikbaar of hoeveel moeite gaat het kosten om de data te verzamelen?	Redeneer je van probleem->vereiste data of van beschikbare data->mogelijkheden?	Specificatie in Data document
Data	Controleer kwaliteit van de data	Check: is de data waarmee we willen gaan werken inderdaad van voldoende kwaliteit?	Vaak blijkt data minder bruikbaar dan vooraf ingeschat.	Idealer een basaal werkend ML-model. Anders een inschatting van de ML-expert.
Data	Kiezen van wijze van data-opslag	Op basis van het soort en de hoeveelheid data een geschikte opslagvorm bepalen.	Bijvoorbeeld als onderdeel van een datamanagementplan of een software architecture document.	Keuze gedocumenteerd in Data document en goedgekeurd door projectleider (i/vm budget).
Data	Nadenken over reproduceerbaarheid en integratie van data	In hoeverre moeten de data reproduceerbaar en integreerbaar zijn, bijvoorbeeld als er meerdere dataverzamelingen zijn of verwacht worden in de toekomst. Wat is daarvoor nodig?	Sterk situatieafhankelijk. Is dit één AI project of zijn er meer mogelijkheden.	Ideen gedocumenteerd in Data document of Software Architectuur Document.
Data	Kiezen van tools voor verwerking/annotatie van data			Keuze gedocumenteerd in Data document en goedgekeurd door projectleider (i/vm budget).
Data	Inrichten van data-opslag			Demo van data-opslag (Create Read Update Delete)

Figuur 5.1: Screenshot voorbeeldbacklog

Bijlage

[Voorbeeldbacklog AI4MKB](#)

Voorbeeld

De projectpartners in het KI-Agil project werkten agile en gebruikten daarbij de tool Jira. Op basis van de voorbeeldbacklog werd de projectbacklog aangemaakt. Het ene project werkte volgens Scrum, in het andere project werd Kanban gehanteerd. Na de start speelde de voorbeeldbacklog geen rol meer.

Meer informatie

Microsoft biedt binnen haar Team Data Science Process een [projectplanning template](#) die deels te vergelijken is met bovenstaande backlog.

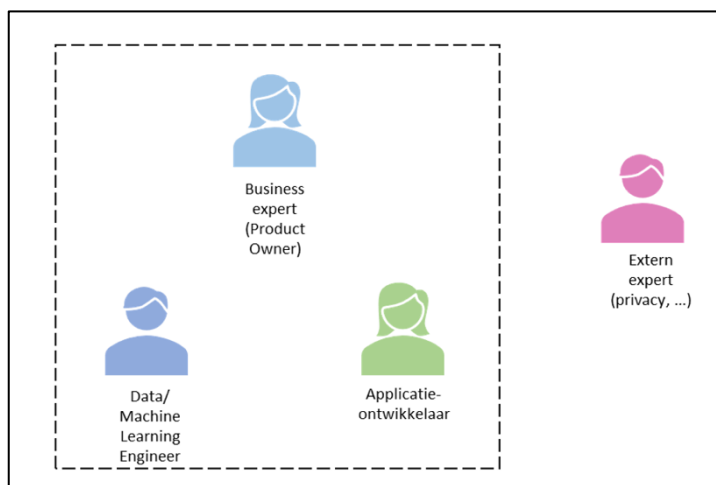
5.2 Vaardighedenscan voor AI-ontwikkeling in het MKB

Doel

Het doel van de vaardighedenscan is het geven van een overzicht van de vaardigheden die nodig zijn voor ontwikkeling van een AI-systeem. Met dit overzicht kan vervolgens gekeken worden of die vaardigheden beschikbaar zijn in het eigen project, dan wel ontwikkeld of ingehuurd moeten worden.

Uitleg

Er zijn veel verschillende rollen met bijbehorende vaardigheden die betrokken kunnen of moeten worden bij het ontwikkelen van AI. Uit de literatuur hebben we 17 verschillende rollen geïdentificeerd. Het is uiteraard goed mogelijk om meerdere rollen in één persoon te combineren. Figuur 5.2 geeft bijvoorbeeld een minimaal team, bestaande uit vier personen.



Figuur 5.2: Minimaal team voor AI-ontwikkeling

Doorloop de volgende stappen om te inventariseren of de juiste rollen en vaardigheden aanwezig zijn voor het AI-project:

1. Download de vaardighedenscan voor AI-ontwikkeling.
2. Neem de checklist door en vul die met een aantal personen in. Bij voorkeur hebben deze personen verschillende functies of achtergrond.
3. Bespreek de gezamenlijke resultaten. Kijk of er consensus is over welke vaardigheden de organisatie meent te beschikken, welke vaardigheden er nog bij moeten en welke minder van toepassing zijn voor dit project.
4. Bedenk wat te doen met de vaardigheden die belangrijk zijn maar nog ontbreken. Mogelijkheden zijn: opleiden eigen personeel (bv. online opleiding bij instellingen als het LOI, opleiding bij IT Academy Noord Nederland, online MOOCs), ontwikkelen 'on-the-job', of inhuren van specifieke expertise.

Bijlage

[Vaardighedenscan voor AI-ontwikkeling](#)

Meer informatie

De vaardighedenscan is gebaseerd op literatuuronderzoek naar data science/machine learning rollen en frameworks. Referenties zijn te vinden bij vaardighedenscan.

6 Ethische aspecten

6.1 Checklist ethische aspecten

Omschrijving

De Assessment List for Trustworthy AI (ALTAI) van de EU is een vragenlijst om te reflecteren of bij de ontwikkeling en implementatie van een AI-toepassing voldoende rekening is gehouden met ethische aspecten, risico's en ongewenste effecten op gebruikers of de samenleving. Door de vragenlijst te doorlopen aan het begin van het ontwikkeltraject kun je de ethische knelpunten eerder achterhalen en kritiek voor zijn.

Voor het MKB is de vragenlijst echter onpraktisch lang (117 vragen). We hebben daarom een beknoptere variant gemaakt ("ALTAI-4MKB") die de essentiële vragen voor MKB'ers stelt. De oorspronkelijke vragenlijst voorziet ook in dergelijke aanpassingen: "*This Assessment List for Trustworthy AI (ALTAI) is intended for flexible use: organizations can draw on elements relevant to the particular AI system from this [list], taking into consideration the sector they operate in.*"

Uitleg

De vragenlijst bestaat uit een serie vragen over de essentiële ethische aandachtsgebieden voor het ontwikkelen, gebruiken en beheren van AI-systemen.

1. Technische robuustheid en veiligheid
2. Privacy en databeheer
3. Diversiteit, non-discriminatie en rechtvaardigheid
4. Menselijke zelfbeschikking (autonomie) en toezicht
5. Transparantie
6. Maatschappelijk welzijn en duurzaamheid

Het invullen van de vragenlijst is 'self reporting', er is geen controle en data wordt niet gedeeld.

Voor elke vraag kan men kiezen uit de antwoorden "Goed genoeg", "Nog niet duidelijk", "Vereist nog actie" en "Niet van toepassing". De antwoorden worden voor het overzicht gekleurd volgens het 'stoplichtsysteem' (groen/oranje/rood). Niet elke vraag zal in elke situatie of ontwikkelingsfase van toepassing zijn. Gebruik in dat geval de optie "Niet van toepassing".

Het is optioneel mogelijk om per vraag een toelichting en te nemen acties te noteren. Dit helpt bij uitleg of verantwoording in de richting van collega's, gebruikers of klanten.

We raden aan om de vragenlijst en de antwoorden door te nemen met meerdere personen die betrokken zijn bij het ontwikkelen of gebruik van het systeem. De verschillende perspectieven kunnen een waardevolle bijdrage leveren aan de kijk op risico's en gewenste maatregelen. Het betrekken van een potentiële gebruikers is zeker aan te bevelen.

We raden aan om de vragenlijst zo vroeg mogelijk in het ontwikkeltraject te doorlopen en regelmatig te controleren of de bedachte maatregelen goed uitgewerkt zijn. Dat kan bijvoorbeeld door die maatregelen mee te nemen in de back log.

Checklist ethische aspecten voor AI-ontwikkeling bij MKB			
Technische robuustheid en veiligheid			
Score	Zelf check	Toelichting	Te nemen actie
Vereist nog actie	Zijn er binnen het AI-systeem beveiligingsmaatregelen getroffen om de integriteit van data en het systeem te beschermen?	Voldoet het systeem aan de wettelijke vereisten op het gebied van veiligheid? Is er tijdens het ontwikkelen, testen en in productie nemen van het systeem een penetration test uitgevoerd?	
Vereist nog actie	Zijn de verschillende risico's (kans & impact) voor het AI-systeem in kaart gebracht en wordt er gewerkt om deze risico's te managen?	Mogelijke risico's voor het systeem zijn het wegvallen, verkeerd werken of misbruiken van het systeem. Mogelijk risico management: Avoid, Accept, Transfer, Reduce. ISO 27001 en ISO27002 bieden handvatten om het risico in kaart te brengen en deze te verkleinen	
Vereist nog actie	Is de nauwkeurigheid van het AI-systeem op een goede manier geborgd?	Is er bepaald hoe nauwkeurig het AI-systeem voorspellingen kan maken en/of advies kan geven aan de eindgebruiker? Wordt er ook wat gedaan om de nauwkeurigheid van het AI systeem te verhogen?	
Vereist nog actie	Is de betrouwbaarheid van het AI-systeem op een goede manier geborgd?	Is er een strategie aanwezig om te managen of het AI systeem na verloop van tijd nog steeds zijn doelen behaalt en correct wordt toegepast?	
Privacy en Data beheer			
Score	Zelf check	Toelichting	Te nemen actie

Figuur 6.1: Screenshot ALTAI-4MKB

Bijlage

[Checklist ethische aspecten voor AI-ontwikkeling bij MBK](#)

Meer informatie

Assessment List for Trustworthy AI (ALTAI),

https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=68342

6.2 De Ethische Data Assistent (DEDA)

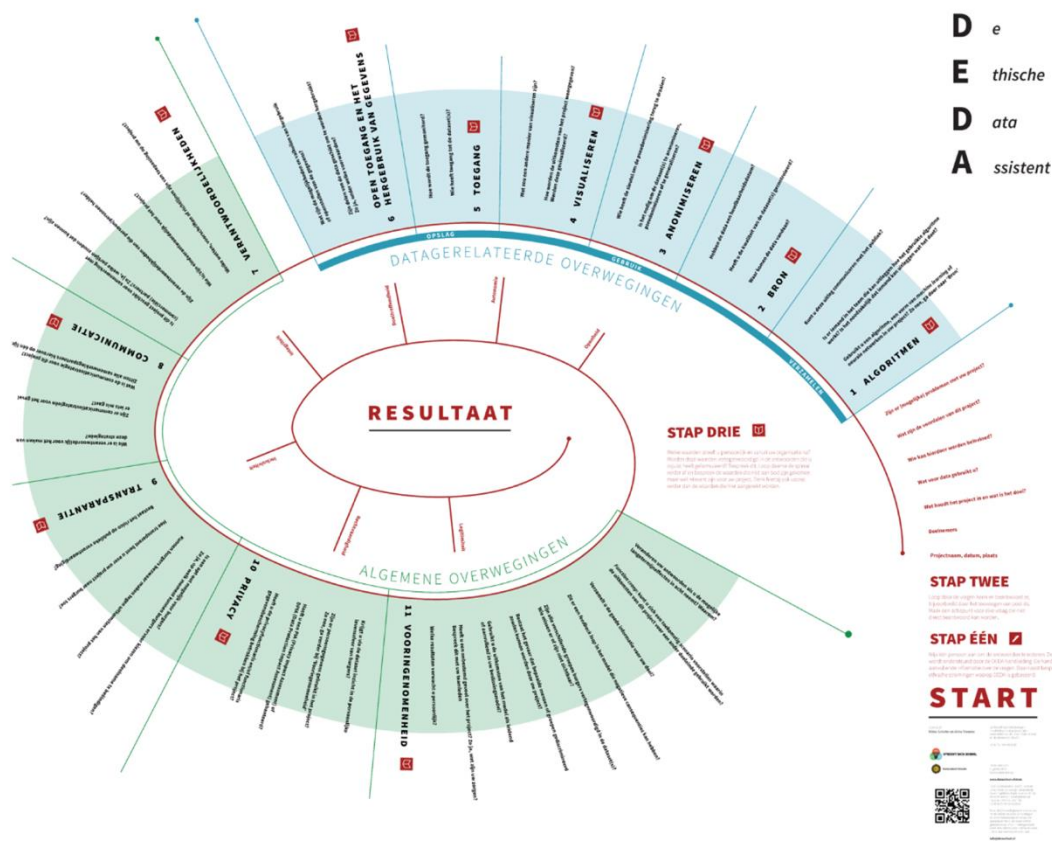
Omschrijving

De Ethische Data Assistent is een toolkit ontwikkeld door de Utrecht Data school van de Universiteit Utrecht die data-analisten, projectmanagers en beleidsmakers helpt om samen ethische problemen in dataprojecten te herkennen.

Uitleg

De basis van DEDA is een uitgebreide set aan vragen over ethische aspecten van big data/machine learning projecten. Deze vragen zijn afgebeeld op een grote poster die bedoeld zijn om workshops met meerdere belanghebbenden te faciliteren. Het doel is om in de workshop vanuit verschillende invalshoeken te bepalen wat de ethische aandachtspunten zijn van een beoogd systeem. Deze workshops kunnen zelfstandig uitgevoerd worden met behulp van de bijgeleverde handleiding of met ondersteuning van een externe facilitator.

De meerwaarde van DEDA ten opzichte van andere checklists zit in het bijbehorende proces samen met de andere belanghebbenden. Wanneer je bijvoorbeeld als Mkb'er een systeem wil ontwikkelen voor toepassing in de zorg of bij de overheid dan kan DEDA goed helpen om draagvlak te creëren en vroegtijdig potentiële risico's te adresseren.



Figuur 6.2: Screenshot DEDA

Meer informatie

De Ethische Data Assistent is te downloaden op <https://dataschool.nl/deda/>.

6.3 Ethics Inc., een agile spel om ethische belangen af te wegen

Omschrijving

Ethics Inc. is een spel ontwikkeld door het lectoraat Artificial Intelligence aan de Hogeschool Utrecht in samenwerking met de Stichting Toekomstbeeld der Techniek en het NEN-instituut. Het doel van het spel is om voor een te ontwerpen AI-toepassing duidelijkheid te krijgen wie de belanghebbenden zijn, welke onderdelen van 'trustworthy AI' voor hun van belang zijn en waar er mogelijk spanning tussen die belangen zit. Dit inzicht wordt vervolgens gebruikt voor het formuleren van ontwerpprincipes voor de te ontwikkelen toepassing.

Ethics Inc. is vormgegeven als een verzameling kaarten en sluit aan bij agile werkwijzen zoals user story's en planning poker. Het is daarom goed bruikbaar voor agile ontwikkelteams samen met hun opdrachtgever ('product owner').

Uitleg

Op hoofdlijnen bestaat het spel uit de volgende stappen:

1. Selecteer een casus. Er zijn vijf standaard casussen aanwezig (bv. Zelfrijdende auto of Robotrecht) of je gebruikt een eigen casus.
2. Selecteer de meest relevante stakeholders.
3. Match de ethische belangen bij de stakeholders en bepaal hun mate van belangrijkheid met behulp van de waarderingskaarten. Discussieer over de verschillende meningen en probeer consensus te bereiken.
4. Weeg vervolgens de ethische belangen van de verschillende stakeholders tegen elkaar.
5. Probeer de ideale AI-toepassing te formuleren, die zoveel mogelijk recht doet aan de verschillende ethische belangen van de stakeholders. Welke ontwerpprincipes kunnen er geformuleerd worden? Hoe ziet de Minimal Viable Product eruit?



Figuur 6.3: Ethics Inc. kaartspel

Meer informatie

Voor meer informatie, zie dit [nieuwsbericht](#).

Het spel kan fysiek of als pdf. worden aangevraagd door contact op te nemen met het [lectorat Artificial Intelligence](#) aan de Hogeschool Utrecht.

7 Machine learning modellen

7.1 Doel

Machine learning is een onderdeel van kunstmatige intelligentie dat zich bezighoudt met hoe een model zijn intelligentie verkrijgt.

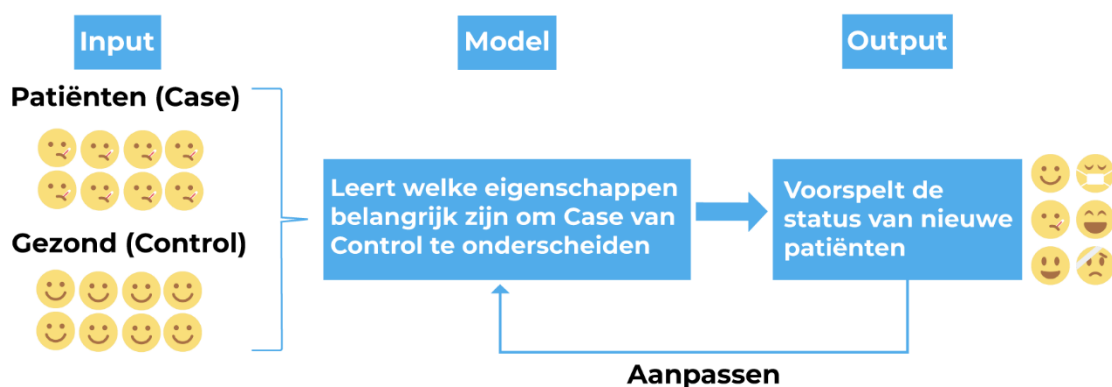
7.2 Uitleg

Net als mensen, heeft een model voorbeelden nodig op basis waarvan hij algemene regels deduceert en toepast. Deze beslissingen kunnen op van alles betrekking hebben, maar in het algemeen maakt het model een beslissing over het wel of niet voorkomen van een bepaalde situatie. Bijvoorbeeld: is een plaatje een afbeelding van een kat of van een hond, is een persoon wel of niet besmet met een virus, is een bepaalde machine aan vervanging toe of niet, is deze verkeersituatie gevaarlijk of niet etc.. Naast deze zogenoemde binaire keuzes (ja of nee) kan een model ook een keuze uit meerdere opties voorspellen. Bijvoorbeeld "van welk dier is dit een afbeelding?". Het model dat we willen maken bestaat uit één of meerdere wiskundige formules met parameters die op basis van input data een bepaalde berekening uitvoeren en daarmee een voorspelling doen. Het ontdekken van deze formules en de bijbehorende waarden van de parameters noemen we het trainen van het model.

In de volgende pagina's gaan we van een relatief simpel voorbeeldmodel uit (Figuur 7.1). Een medische situatie waarbij we met een model willen voorspellen of een persoon die we testen een infectie heeft of dat deze persoon gezond is.

We refereren in dit geval aan een geïnfecteerd persoon als een **case** en aan een persoon waar geen infectie wordt gemeten als **control**. Dit is ook bij het ontwikkelen van niet medisch georiënteerde modellen een gebruikelijke terminologie. De data die we willen gebruiken kunnen in dit voorbeeld bestaan uit waarden van een laboratoriumonderzoek aan het bloed of de urine van een persoon, gecombineerd met een uitslag van diagnostische PCR tests en algemene gezondheidsparameters zoals bloeddruk, BMI, temperatuur, vetgehalte, hartslag etc.

Het proces van machine learning kan in het algemeen in verschillende fases beschreven worden, die hieronder verder worden uitgelegd.



Figuur 7.1: Voorbeeldmodel

7.3 Model ontwikkelen

7.3.1 Doel

In deze fase wordt een model getraind op basis van een bestaande data set. Het is in deze fase belangrijk om goed vast te stellen wat het model moet kunnen, wat het moet voorspellen en welke data hier het beste voor gebruikt kunnen worden.

7.3.2 Uitleg

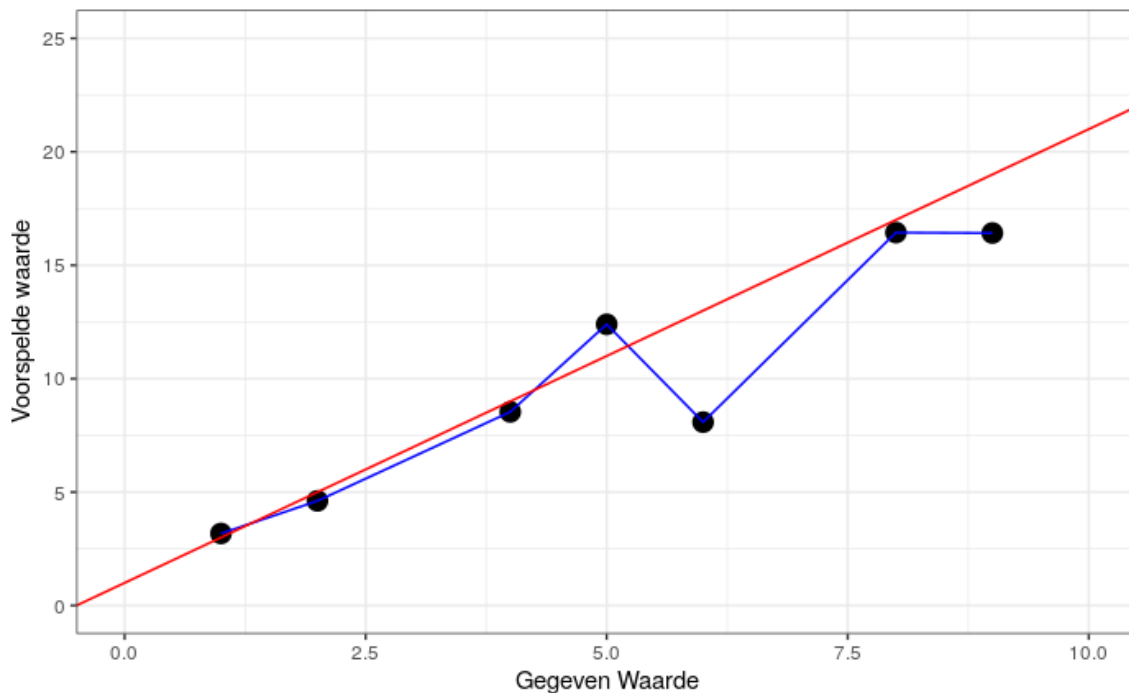
Het machine learning proces lijkt erg veel op hoe wij als mensen regels ontdekken over hoe de wereld in elkaar zit. Door heel veel voorbeelden van situaties met elkaar te vergelijken stellen we onbewust regels op die we later op nieuwe situaties toepassen.

In het algemeen wordt een model dus ook getraind door het model een aantal voorbeelden te geven van situaties waarvan de uitkomst bekend is. Bijvoorbeeld, een data set van duizenden plaatjes van honden en katten. Op basis van zo'n trainingsdata set leert het model juist die kenmerken van de plaatjes die op de meest efficiënte manier, de afbeelding van honden en katten van elkaar te onderscheiden. In ons specifieke voorbeeld wordt geleerd op basis van welke parameters het meest efficiënt geïnfecteerde personen van gezonde personen kunnen worden onderscheiden.

Het trainen van het model is een iteratief proces, waarbij steeds een deel van de data gebruikt wordt om de regels te leren (een aantal cases en controles, de training dataset) en een deel van de data gebruikt wordt om te testen hoe goed de regels zijn (de overige cases en controles, de validatie dataset).

Ook worden tijdens het trainen steeds een deel van de kenmerken van de cases en controles gebruikt voor de voorspelling. Zo leert het model welke kenmerken het belangrijkst zijn om goede voorspellingen te doen. In ons specifieke voorbeeld kan het dus zijn dat het model leert dat bloeddruk en BMI niet belangrijk zijn voor het vaststellen van een aanwezigheid van een infectie, maar dat hartslag en vetgehalte juist heel veel voorspellende waarde hebben.

Het grootste gevaar bij het trainen van een model is overfitting. Dit houdt in dat het model erg goed in staat is om de training data set te voorspellen en ook de validatie set goed voorspelt, maar dat een model later niet goed presteert op nieuwe data sets. Dit is te voorkomen door de validatie set nooit te gebruiken voor de training van het model, en daarbij ook het model te dwingen om niet te veel parameters te gebruiken om de data te fitten. Een goede vuistregel is om voor elke parameter die je aan het model toevoegt, tenminste 5 nieuwe cases en controles toe te voegen. Een model dat 10 parameters gebruikt om de case/control status van 10 personen te voorspellen is dus zeer waarschijnlijk een overfit model, dat niet goed zal presteren op nieuw data.



Figuur 7.2: Twee modellen (blauw en rood) die beide aan een aantal data punten gefit zijn. Het blauwe model is duidelijk onderhevig aan overfitting. Het fit alle data punten perfect, maar is niet door te trekken naar nieuwe data punten. Het rode model fit de datapunten niet perfect, maar is wel te generaliseren naar nieuwe data.

7.4 Model keuze

7.4.1 Doel

In deze fase moet een keuze voor een specifiek type algoritme gemaakt worden. Deze keuze is in de praktijk van meerdere factoren afhankelijk.

7.4.2 Uitleg

Er zijn een heel aantal modellen die getraind kunnen worden op de manier die hierboven generiek is beschreven. De keuze van het juiste model, met het juiste onderliggende algoritme is belangrijk, maar kan soms ook als een lastige keuze ervaren worden. Er zijn namelijk veel modellen die allemaal specifieke onderliggende wiskundige aannames en uitwerkingen hebben. Voor een deel werken deze modellen op dezelfde principes, maar voor een deel zijn modellen soms specifiek geschikt voor bepaalde type data. In de praktijk is het echter zo dat veel modellen met een grote verscheidenheid aan data uit de voeten kunnen en out-of-the box redelijk goed presteren.

Een bekend en veel gebruikt model is het RandomForest-model. De naam is ontleend aan het feit dat het model een groot aantal random gegenereerde beslisbomen gebruikt om op basis van data een voorspelling te doen. Een RandomForest-model kan gebruikt worden voor numerieke voorspellingen, binaire of Multi class voorspellingen en is redelijk ongevoelig voor uitbijters en missende datapunten in de data set. Dit is in de praktijk dan ook een goed model om mee te beginnen.

Goede suggesties voor modellen kunnen ook uit de literatuur gehaald worden door te kijken naar hoe mensen vergelijkbare problemen aangepakt en opgelost hebben en wat voor (type) model ze daarvoor hebben gebruikt.

Een andere overweging die gemaakt kan worden is de ervaring die modelontwikkelaars hebben met eerdere modellen en hoe makkelijk de modellen geïmplementeerd kunnen

worden in een bepaalde programmeertaal en hoe makkelijk de modellen in een uiteindelijke softwareapplicatie ingebouwd kunnen worden voor de eindgebruiker.

Het tegelijkertijd toepassen van meerdere modellen, ook wel de “ensemble” aanpak genoemd, waarbij de voorspellingen van meerdere modellen tegelijkertijd gebruikt en geëvalueerd worden is ook een efficiënte methode. Deze ensemble methodes zijn over het algemeen erg robuust omdat tekortkomingen van één model op een specifieke data set gecompenseerd worden door een model dat juist wel met dat specifieke stukje data om kan gaan.

De keuze voor een specifiek model kan ook gedreven worden door hoe goed het uiteindelijke model door de eindgebruiker te begrijpen moet zijn. Mag dit een black-box model zijn, waarbij het model wel goede voorspellingen doet maar de onderliggende formules niet duidelijk zijn? Of moet het model ook goed te interpreteren zijn waarbij het duidelijk is hoe het model de beslissingen neemt (en op basis van welke kenmerken)? Dit laatste geval wordt ook wel explainable AI genoemd.

Wellicht de beste reden om juist wel of niet voor een model te kiezen is te kijken hoe het in de praktijk werkt. Dit kan aan de ene kant geëvalueerd worden door te kijken naar de statistische parameters over hoe goed het model nieuwe data voorspelt (zie 7.5 Model validatie), maar ook hoe gebruikers het model in de praktijk ervaren en hoe goed het model bijdraagt aan het versnellen en verbeteren van beslissingen.

Kortom, de keuze van een model is erg context afhankelijk en moet op basis van veel criteria gemaakt worden. Het is daarom belangrijk om in een vroege fase een proof-of-concept te hebben waarin verschillende modellen in een realistische toepassingsomgeving getest kunnen worden.

7.5 Model validatie

7.5.1 Doel

In deze fase wordt het model gevalideerd door te kijken hoe goed het model presteert op nieuwe data, die niet gebruikt zijn om het model te trainen.

7.5.2 Uitleg

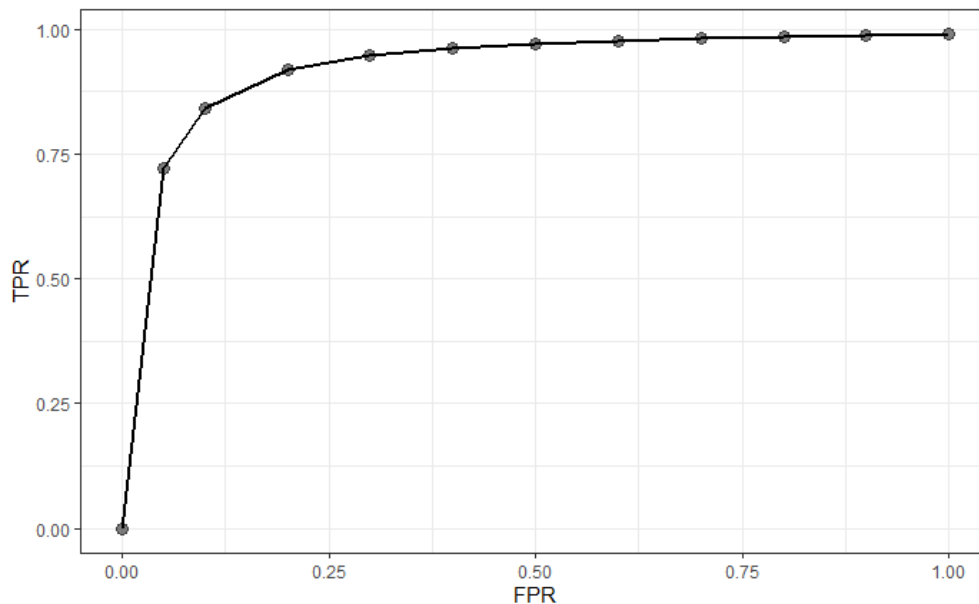
Er zijn veel verschillende manieren om de voorspellende waarde en daarmee de bruikbaarheid van een model te evalueren.

De meest gebruikte manier om het voorspellende karakter van een model te valideren is om te analyseren hoe goed het model presteert op een nieuwe data set, waarvan de labels (case of control) bekend zijn. In het voorbeeld, waarin je geïnteresseerd bent in het voorspellen van het voorkomen van een infectie (**case**) of de afwezigheid van een infectie (**control**) bij een aantal test personen zijn de belangrijkste parameters voor de evaluatie van het model de volgende:

- False Positive rate (FPR): Deze parameter geeft aan hoeveel personen ten onrechte als **case** aangemerkt worden (vals alarm).
- False negative rate (FNR): Deze parameter geeft aan hoeveel personen ten onrechte als **control** aangemerkt worden (valse veiligheid).
- True Positive Rate (TPR): Deze parameter geeft aan hoeveel van de werkelijke cases ook echt als **case** worden aangemerkt

In het ideale model is zowel de FPR als de FNR erg laag en de TPR erg hoog, maar in de praktijk is elk model een afweging tussen de FPR, TPR en de FNR. In het algemeen is het zo dat een model dat erop gericht is om een lage FNR te halen automatisch een wat hogere FPR heeft en andersom. Tijdens optimalisatie van het model kunnen door het

instellen van de model parameters verschillende FPRs en FNRs en TPRs behaald worden. De trade-off tussen FPR en TPR kan gevisualiseerd worden in een Receiver-Operator Curve (ROC-curve). In deze curve wordt bij elke parametersetting van een model, de FPR en TPR uitgerekend en tegen elkaar uitgezet.



Figuur 7.3 ROC-curve. Elke punt op deze curve vertegenwoordigt een specifieke versie van het model met een specifieke parameter setting. Aanpassing van het model om een hogere TPR te krijgen (uitgezet op de y-as) leidt (bijna) altijd ook tot een hogere FPR (uitgezet op de x-as).

Omdat er vaak meerdere combinaties van TPR en FPR mogelijk zijn afhankelijk van de parametersetting van het model is het dus belangrijk om te bekijken hoe erg het is om een vals alarm te hebben (een persoon die dus ten onrechte behandeld wordt voor een infectie) ten opzichte van hoe erg het is om een valse veiligheid (een persoon met een infectie die dus niet behandeld wordt).

7.6 Model toepassen

7.6.1 Doel

In deze fase wordt het model dat in eerdere fases getraind is, toegepast op nieuwe data met het doel om beslissingen in de praktijk sneller en beter gefundeerd te nemen.

7.6.2 Uitleg

In principe wordt net als in de validatiefase het model toegepast op nieuwe data. Maar deze keer zijn de uiteindelijke labels (case of control) niet bekend, het gaat om nieuwe personen die getest worden en op basis waarvan een beslissing genomen moet worden over een behandeling. Hierbij zijn een aantal dingen belangrijk.

7.6.3 Applicatie

In deze fase is het belangrijk dat het model in zo'n formaat gepresenteerd wordt dat ze door de eindegebruiker goed gebruikt en geïnterpreteerd kunnen worden. Hierbij kun je denken aan een model dat in de vorm van een spreadsheet in Excel geïmplementeerd wordt. Het voordeel van een implementatie in Excel is dat het door veel mensen gebruikt kan worden en de resultaten daarvan ook makkelijk naar andere programma's doorgezet

kunnen worden. Een beperking is dat niet alle AI-gebaseerde modellen, zoals de modellen die gebaseerd zijn op neurale netwerken, omgezet kunnen worden naar een Excel format.

Een andere vorm is het aanbieden van een model via het internet in een webbrowser, waarbij de eindgebruiker de nieuwe data doorgeeft in een formulier en de uitslag direct in de browser kan zien. Het voordeel van zo'n systeem is dat de gebruiker zelf weinig hoeft te weten van de interne structuur en formules van het model en geen additionele software hoeft te installeren.

Een model kan uiteraard ook ingebouwd worden in een specifiek softwareapplicatie die door de eindgebruiker in zijn eigen computeromgeving gebruikt kan worden. Ook deze vorm heeft voor- en nadelen. De software moet güpdatet worden als de modellen veranderen en installatie kan in sommige IT-omgevingen wellicht niet gemakkelijk geïmplementeerd worden.

Kortom, de keuze voor de juiste applicatie is erg afhankelijk van de specifieke situatie. Het is erg belangrijk om al in een vroeg stadium op hoofdlijnen vast te stellen hoe de applicatie eruit gaat zien, omdat dit ook consequenties voor de modelontwikkeling kan hebben.

7.6.4 Visualisatie

Het is in de toepassing ook belangrijk om goed duidelijk te maken hoe het model tot een bepaalde voorspelling is gekomen. Welke data zijn hiervoor gebruikt, en wat zijn de onzekerheden in de voorspelling etc. Dit is belangrijk om de acceptatie en daarmee het gebruik van een model te borgen.

8 AI en applicatieontwikkeling

8.1 Software Engineering practices for Machine Learning

Omschrijving

SE4ML (Software Engineering for Machine Learning) is een uitgebreide verzameling best practices voor het ontwikkelen van softwaresystemen met machine learning componenten.

Uitleg

SE4ML is verzameld door het Leiden Institute of Advanced Computer Science (LIACS) onder machine learning beoefenaars. De catalogus bevat op het moment van schrijven 45 best practices voor het ontwikkelen van machine learning systemen. De practices zijn onderverdeeld in de categorieën Data, Training, Coding, Deployment, Team en Governance. Voorbeelden zijn 'Write Reusable Scripts for Data Cleaning and Merging' en 'Automate Hyper-Parameter Optimisation'. Bij elke practice is beschreven: de moeilijkheid van toepassing, de effecten van toepassing (wat levert het op?) en of ze bijdragen aan de EU-eisen voor betrouwbare AI.

De survey waarmee het LIACS de practices verzamelt staat permanent open. Dus wanneer je als MKB zelf enige ervaring hebt opgedaan is het waardevol om die informatie weer te delen met de onderzoekers.



Figuur 8.1: SE4ML indeling van practices

Meer informatie

[SE4ML homepage](#), [catalogus](#) en [jaarlijks onderzoeksrapport](#)
[Leiden Institute of Advanced Computer Science \(LIACS\)](#)

8.2 Kwaliteitskenmerken voor Machine Learning en AI

Omschrijving

Toelichten welke kwaliteitskenmerken specifiek van belang zijn voor machine learning software.

Uitleg

Kwaliteitskenmerken zijn een belangrijk aandachtspunt bij softwareontwikkeling en dus ook van de ontwikkeling van AI-algoritmes en Machine Learning modellen. Ze worden gebruikt om de kwaliteit van de werking van de software te specificeren. De eisen die gesteld worden aan de kwaliteitskenmerken noemen we niet-functionele eisen ('non-functional requirements').

Waar functionele eisen beschrijven welke functies een stuk software moet leveren ('wat'), beschrijven de niet-functionele eisen aan welke kwaliteitscriteria de software moet voldoen terwijl het die functies levert ('hoe'). Een bekend voorbeeld is performance. De functionele eis is dat een afbeelding herkend moet worden, de niet-functionele eis is hoe snel dat moet. Moet een afbeelding van een zieke aardappelplant binnen 1 seconde herkend worden, of binnen 1 milliseconde?



Figuur 8.2: ISO 25010:2011 Software product quality

ISO-standaard 25010:2011 (Figuur 8.2) is een bekende standaard die een overzicht geeft van de verschillende kwaliteitskenmerken van software. In Tabel 8.1 geven we aan op welke manier deze kwaliteitskenmerken een rol spelen bij het ontwikkelen van software met AI en Machine Learning. We hanteren daarbij de terminologie uit de [Nederlandstalige versie van de standaard](#).

Tabel 8.1: Kwaliteitskenmerken toegepast op AI en Machine Learning

Kwaliteitskenmerk	Toelichting
Functionele geschiktheid	Onderdeel van Functionele geschiktheid is de <i>Functionele correctheid</i> . Dit betreft de nauwkeurigheid waarmee de software de resultaten geeft. In het geval van Machine Learning krijgt dit kwaliteitskenmerk vaak veel aandacht en wordt gemeten met metriecken zoals accuracy en precision . Ook het onderdeel <i>Functionele toepasselijkheid</i> is een aandachtspunt bij AI. Komt de situatie waarin het model wordt gebruikt overeen met de data waarop het model is getraind?

Prestatie-efficiëntie	<p>Prestatie-efficiëntie is voor de meeste softwaresystemen een belangrijk aspect en dat is voor AI-toepassingen niet anders. <i>Snelheid</i> betreft of de AI-toepassing in productie ('in deployment') voldoende snel is in het uitvoeren van de taak, zoals classificeren, detecteren of voorspellen.</p> <p><i>Middelenbeslag</i> en <i>capaciteit</i> zijn met name belangrijk tijdens de ontwikkel fase van het model. Het trainen van modellen met een grote hoeveelheid data of afbeeldingen doet een beroep op opslagcapaciteit en rekenkracht. Voor veel AI-modellen zijn middelenbeslag en capaciteit in de productieomgeving minder kritisch. Dit hangt dan voornamelijk af van de hoeveelheid data die gestreamd en/of opgeslagen wordt en de verdeling daarvan over de tijd (uniform of met pieken).</p>
Uitwisselbaarheid	<p>Het belangrijkste aandachtspunt bij uitwisselbaarheid is de <i>koppelbaarheid</i> van de getrainde AI-modellen met de omliggende software. Hiervoor bestaan meerdere aanpakken (integratie in de applicatiesoftware, het aanbieden van een API, het gebruikmaken van commerciële tooling voor Machine Learning Pipelines). Dit betreft een architectuurkeuze die in de architectuurfase al geadresseerd dient te worden.</p>
Bruikbaarheid	<p><i>Herkenbaarheid</i> is voor sommige AI-toepassingen relevant. Het moet voor de gebruiker duidelijk zijn dat de AI-toepassing werkt en wat de status is. Er zijn voorbeelden van artsen die systemen niet durfden te gebruiken omdat er geen duidelijk feedback was over het goed functioneren ervan.</p> <p><i>Fairness</i> is geen onderdeel van het ISO25010 raamwerk maar voor de toepassing van AI is het echter een belangrijk kwaliteitskenmerk dat we in deze categorie plaatsen. Fairness gaat over de vraag of de AI-toepassing bepaalde deelpopulaties wellicht onevenredig benadeelt, bijvoorbeeld omdat er de dataset bias bevat waardoor de onderliggende populatie niet goed gerepresenteerd is.</p>
Betrouwbaarheid	<p>De kwaliteitskenmerken in deze categorie (te weten: volwassenheid, beschikbaarheid, foutbestendigheid en herstelbaarheid) zijn voor AI-toepassingen in grote lijnen niet veel anders dan voor reguliere software.</p> <p>Een aandachtspunt is dat <i>volwassenheid</i> is geformuleerd in termen van 'voldoet onder normale omstandigheden'. Is de AI getraind met data uit de zomerperiode, dan is gebruik van de AI op data uit de winterperiode dus niet betrouwbaar. De eindgebruiker van de software kan dat echter lastig weten.</p> <p>Voor AI-toepassingen die gebruikt worden voor meet en regelsystemen is <i>foutbestendigheid</i> een aandachtspunt. Kan de software zelfstandig uit een foutsituatie komen? Er zijn filmpjes van zelfrijdende auto's die verlamd langs de weg staan omdat ze ten onrechte een belemmering menen te detecteren.</p>

Beveiligbaarheid	<p><i>Beveiligbaarheid</i> is een belangrijk kwaliteitskenmerk, dat samenhangt met de ethische aspecten van het project (zie hoofdstuk Ethische en Juridische aspecten). Gegevens die je om ethische redenen niet opneemt in het systeem, hoeven ook niet beveiligd te worden.</p> <p>Bij beveiliging is het belangrijk om naar het zogenaamde 'attack Surface' te kijken. Welke mogelijke aanvalspunten bevat het AI-systeem, denk aan API-calls. Verder wordt er veel onderzoek gedaan naar manieren om AI-algoritmen te misleiden. Bijvoorbeeld door het invoeren van afbeeldingen die het algoritme op het verkeerde been zetten. Dat is op het moment echter vooral een wetenschappelijk onderzoeksterrein en niet bijzonder relevant voor MKB-toepassingen.</p> <p><i>Verantwoording</i> wordt steeds belangrijker voor AI-systemen, juist omdat veel algoritmen als een zwarte doos fungeren. Er is daarom steeds meer aandacht voor zaken als uitlegbaarheid ('explainable AI') en transparantie.</p>
Onderhoudbaarheid	<p>Bij onderhoudbaarheid is <i>wijzigbaarheid</i> een belangrijk kenmerk. Veel AI-modellen moeten periodiek geüpdatet worden, vanwege betere trainingsdata, uitbreidingen of omdat de input data gaandeweg afwijkt van data waarop getraind is ('data drift'). Vanwege die periodieke hertraining is dan ook belangrijk dat deze updates efficiënt kunnen worden uitgevoerd.</p> <p><i>Analyseerbaarheid</i> betreft de mate waarin de impact van een wijziging bepaald kan worden, of de oorzaak van een fout gevonden kan worden. Een aspect dat hier makkelijk over het hoofd wordt gezien is de administratie van de verschillende versie van het model en de bijbehorende (hyper)parameters, de gebruikte trainingsdata, etc. Zeker wanneer er met verloop van tijd meerdere varianten van het model getraind zijn, wordt het al snel onoverzichtelijk. Er komt steeds meer commerciële software die helpt met die administratie.</p> <p><i>Testbaarheid</i> ligt dicht tegen verantwoording en uitlegbaarheid aan. Hoe kan een AI-systeem efficiënt worden getest om vast te stellen of aan de testcriteria is voldaan? Tijdens de trainingsfase wordt een model vaak automatisch getest. Is dat ook mogelijk met de applicatie in zijn geheel? En in hoeverre zijn periodieke evaluaties mogelijk wanneer de applicatie in productie draait?</p>
Overdraagbaarheid	<p>Hier spelen geen andere eisen dan bij reguliere applicaties. Het aanbieden van het AI-model via een API draagt bij aan de modulariteit en platformafhankelijkheid van de applicatie als geheel.</p>

Meer informatie

- [ISO/IEC 25010:2011 System and software quality models](#)
- Habibullah, K. M., & Horkoff, J. (2021, September). Non-functional requirements for machine learning: understanding current use and challenges in industry. In *2021 IEEE 29th International Requirements Engineering Conference (RE)* (pp. 13-23). IEEE.
- Lakshmanan, V., Robinson, S., & Munn, M. (2020). *Machine learning design patterns*. O'Reilly Media.