DATA SCIENCE INITIATIVE

AI-handleiding en stappenplan voor de casus bijstand van gemeente Den Haag

dr. Seyit Höcük

drs. Marcia den Uijl

dr. Patricia Prüfer

Pradeep Kumar, MSc.

AUTEURS

Dr. Seyit Höcük
Data Scientist

Dr. Höcük is gepromoveerd in astrofysica. Hij is ervaren in het analyseren van data en in het toepassen van voorspellende modellen met behulp van machine learning en AI-technieken.

Dr. Patricia Prüfer
Hoofd Data Science Unit

Dr. Prüfer is gepromoveerd in economie en hoofd van de Data Science Unit bij CentERdata. Ze heeft ruime ervaring in toegepast (data science) onderzoek en beleidsexperimenten.

Drs. Marcia den Uijl
Senior Data Scientist

Marcia is afgestudeerd in econometrie. Ze heeft kennis en ervaring in het ontwikkelen van software voor business intelligence rekenmodellen en het modelleren van optimale beroepsovergangen.

Pradeep Kumar, MSc
Data Engineer/Data Scientist

Pradeep is afgestudeerd in electrical engineering. Hij heeft ruime ervaring in de vakgebieden tekst analyse, tekst mining, NLP, en signaalanalyse.

Over CentERdata

CentERdata is een een academisch ingesteld onderzoeksinstituut voor dataverzameling en toegepast (wetenschappelijk) onderzoek, gevestigd op de campus van Tilburg University (TiU). Het instituut heeft als missie het beantwoorden van onderzoeksvragen op het gebied van mens en maatschappij; onderzoeksvragen die leven bij zowel wetenschappers als beleidsmakers. Bij de activiteiten die worden ontplooid staan drie kernwaarden centraal: wetenschappelijk, innovatief en maatschappelijk relevant.
Het instituut kent vijf kernactiviteiten: Online survey onderzoek,Beleidsanalyse en modelontwikkeling, Consumentenonderzoek, Software ontwikkeling op maat en Data science.
Op het terrein van data science is een breed palet aan expertise, kennis en ervaring aanwezig bij CentERdata: van praktische besluitvormingsmodellen tot kunstmatige intelligentie. De data scientists houden zich bezig met technieken en toepassingen voor big data, het gebruik van machine learning, data mining en visualisatie technieken voor het creëren van nieuwe inzichten uit gestructureerde en ongestructureerde data.

Meer informatie over CentERdata is te vinden op:
www.centerdata.nl

INHOUDSOPGAVE

1. Inleiding

Dit is een handleiding en een stappenplan met als doel om te helpen bij het toepassen van Artificial Intelligence (AI) en Machine Learning (ML) op datagedreven onderzoek en beleid. De handleiding is gericht op mensen met basiskennis in data science, maar ook voor de gevorderde data-analisten.

In deze handleiding worden meerdere ML- en data-analysemodellen beschreven. Daarbij komen de voor- en nadelen van elk model aan bod. Met de beschreven modellen kunnen onder andere voorspellingen worden gedaan waarmee het gewenste doel kan worden bereikt. Ook kunnen de belangrijkste indicatoren worden achterhaald die nodig zijn voor een voorspelling. Daarnaast wordt beschreven hoe elementen geclusterd of gescheiden kunnen worden op basis van gegevenskenmerken om onzichtbare patronen in de data te ontdekken. Er kan dus op verschillende manieren waarde uit data gehaald worden.

Er worden ook stappen beschreven die moeten worden doorlopen om een juiste keuze te kunnen maken bij het selecteren van het optimale model. Is een transparanter, makkelijker interpreteerbaar model belangrijker of toch de beste, krachtigste voorspelling? Moet je een voorspelling doen, kan je beter de indicatoren van je voorspellingsmodel achterhalen of is het toch verstandiger om je data te clusteren? Op deze vragen wordt ingehaakt door de benodigde vervolgstappen nader toe te lichten. Uiteindelijk zijn deze stappen nodig voor een goede interpretatie van de analyseresultaten.

Uiteraard vereist een succesvolle toepassing van data science binnen een organisatie een bewustzijn van de waarde die uit de data gehaald kan worden en van de vraagstukken die met behulp van data science technieken opgelost kunnen worden. De toegevoegde waarde van data science wordt door vele bedrijven onderkend, maar de mogelijkheden die het biedt zijn nog onvoldoende verweven in de mindset van individuele werknemers. Om datagedreven inzet binnen de organisatie een boost te geven, helpt een praktische AI-handleiding om doelgericht en datagedreven aan de slag te gaan.

1.1 Casus Bijstand

Deze handleiding is op maat gemaakt voor de casus bijstand van de gemeente Den Haag. Daarom worden er tot slot een op maat gemaakte richtlijn en adviezen gegeven met betrekking tot de beste modellen en aanpakken die kunnen worden ingezet.

De casus betreft het sneller helpen van mensen in de bijstand. De gemeente Den Haag wil graag de uitstroom voorspellen van mensen in de bijstand naar werk via datagedreven onderzoek. De gemeente Den Haag heeft veel data van bijstandsgerechtigden en wil hieruit waarde halen. Het is met name erg belangrijk om te achterhalen wie er als eerste uit de bijstand komt en wat de redenen daarvoor zijn. Daarnaast wil men graag weten welk instrument naar verwachting het meest effectief is voor een bepaalde persoon. Hiermee kunnen gerichte interventies opgezet worden op specifieke groepen waardoor een groter aantal mensen in de bijstand sneller aan werk kan worden geholpen.

2. Achtergrond, context en terminologie

Data is niet meer weg te denken uit de huidige samenleving. Allemaal laten we online gegevens achter. Denk maar aan een berichtje dat je net op Facebook plaatste of je laatste online aankoop. Ook als je werk zoekt, je cv plaatst en zoekopdrachten uitvoert, laat je weer een stukje van je persoonlijke gegevens, je wensen of je persoonlijke voorkeuren achter. AI is het niet gepersonaliseerd, met genoeg data kan groepsgedrag, beweegredenen en groepsdynamiek begrepen en zelfs voorspeld worden.

Data kom je dus overal tegen. Sommige bedrijven verzamelen doelgericht data om hun klanten beter te leren kennen en daardoor betere service te bieden. Andere organisaties beheren data die nodig (of verplicht) zijn voor het realiseren van een dienstverlening.

Ook overheden gebruiken data om betere service te bieden. Zo helpt data bijvoorbeeld om te kijken hoe publiek geld het beste ingezet kan worden. Of, door het combineren van technologie en data kunnen dingen gemakkelijker gemaakt worden. Denk bijvoorbeeld aan het vooraf ingevulde belastingaangifte formulier. Gemeenten zijn continu bezig om gemeentelijke dienstverlening te verbeteren. Mensen sneller en aan passender werk te helpen is een belangrijk onderdeel daarvan. Datagedreven kennis kan daarom ingezet worden om tot betere oplossingen te komen.

2.1 Data Science

Data science is een relatief nieuwe discipline die tot nieuwe ontwikkelingen leidt. Met data science wordt bedoeld: het analyseren en interpreteren van grote hoeveelheden data die, vaak complex en ongestructureerd, en tegenwoordig dagelijks gegenereerd worden. De kunst van data science is het transformeren van data in waardevolle acties. Dit vakgebied beslaat veel meer dan het conventionele gebruik van data en statistiek.

Conventioneel statistisch onderzoek is over het algemeen meer gericht op het verleden. In statistische analyses worden hypothesen opgesteld en met data worden deze getoetst om aan te tonen hoe bepaalde processen en verbanden hebben plaatsgevonden. Data science daarentegen is er veel meer op gericht om patronen te herkennen en op basis hiervan voorspellingen te doen voor de toekomst. Data science onderscheidt zich dus door de holistische benadering. Op die manier kunnen er aanbevelingen worden gedaan voor belangrijke beslissingen die genomen moeten worden.

Daarnaast zijn data science methoden geschikt om ook uit ongestructureerde data (denk aan vrije teksten) informatie te halen, met Natural Language Processing (NLP) technieken. Zo kunnen alle databronnen die betrekking hebben op natuurlijke taal worden gebruikt, zoals open antwoorden, tekstbestanden, aantekeningen van klantcontacten, rapporten of e-mails. Technieken zoals part of speech tagging, bi-/trigrams, lemmatisering en reguliere expressies kunnen binnen NLP, als onderdeel van data science worden ingezet.

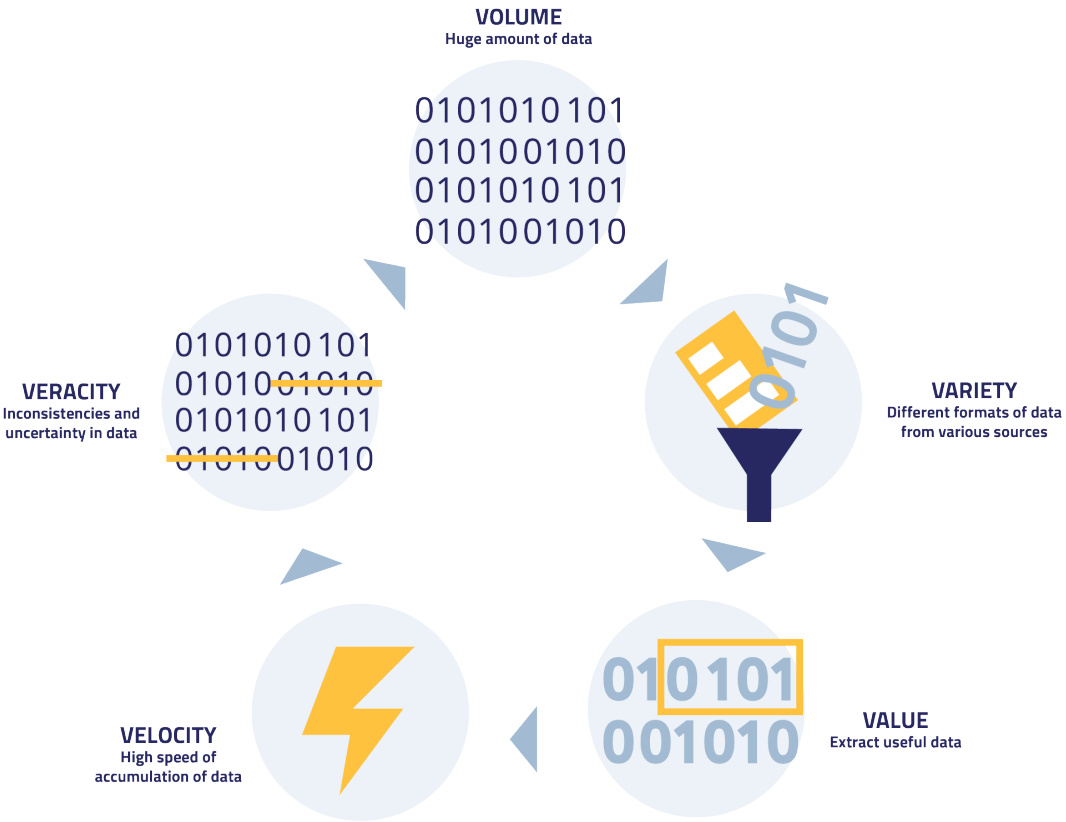
2.2 Big Data

Er komt steeds meer data beschikbaar. Het verzamelen, het opslaan, het delen en het verwerken van data wordt steeds makkelijker en goedkoper. Hierdoor groeit de hoeveelheid data maar door en door. Databestanden worden dus steeds groter en rijker. Ook worden we er tegenwoordig steeds beter in om allerlei data te koppelen. Zo wordt bijvoorbeeld de data die het Centraal Bureau voor de Statistiek (CBS) tot haar beschikking heeft vaak gekoppeld met andere datasets (in een beveiligde omgeving), zoals vragenlijsten, politie data en gemeentelijke data tot grote en vaak complexe datasets.

We spreken van big data als er sprake is van meerdere kenmerken van de data, met name de hoeveelheid data, de snelheid waarmee de data binnenkomen en/of opgevraagd worden, en de diversiteit van de data. Er wordt daarom ook vaak over de drie V's gesproken; Volume, Velocity en Variety.

Er zijn echter meer zaken die een rol spelen. De data is vaak ongestructureerd en kan niet in een traditionele database opgeslagen worden. Andere mogelijke factoren zijn de kwaliteit van de data (de ene bron is minder betrouwbaar dan de andere), de complexiteit van de data (soms is het lastig om ongestructureerde data van verschillende bronnen met elkaar te combineren) en de waarde van data (data op zich hebben geen nut tenzij ze worden omgezet in iets waardevols). Naast de drie V's komen er dus nog minimaal twee extra V's bij; Veracity en Value.

In Figuur 1 worden de vijf V's grafisch weergegeven.



Figuur 1: De vijf V's van Big Data.

2.3 Machine Learning

Machine Learning (ML) is een geavanceerd onderzoeksveld binnen data science dat zich bezighoudt met de technieken waarmee computers kunnen leren, zonder dat ze er speciaal voor geprogrammeerd zijn. Met de technieken van ML kunnen uit grote databestanden verbanden in de data worden geïdentificeerd. Machine Learning wordt doorgaans onderverdeeld in drie brede categorieën: Supervised ML, Unsupervised ML en Reinforcement Learning.

Supervised ML

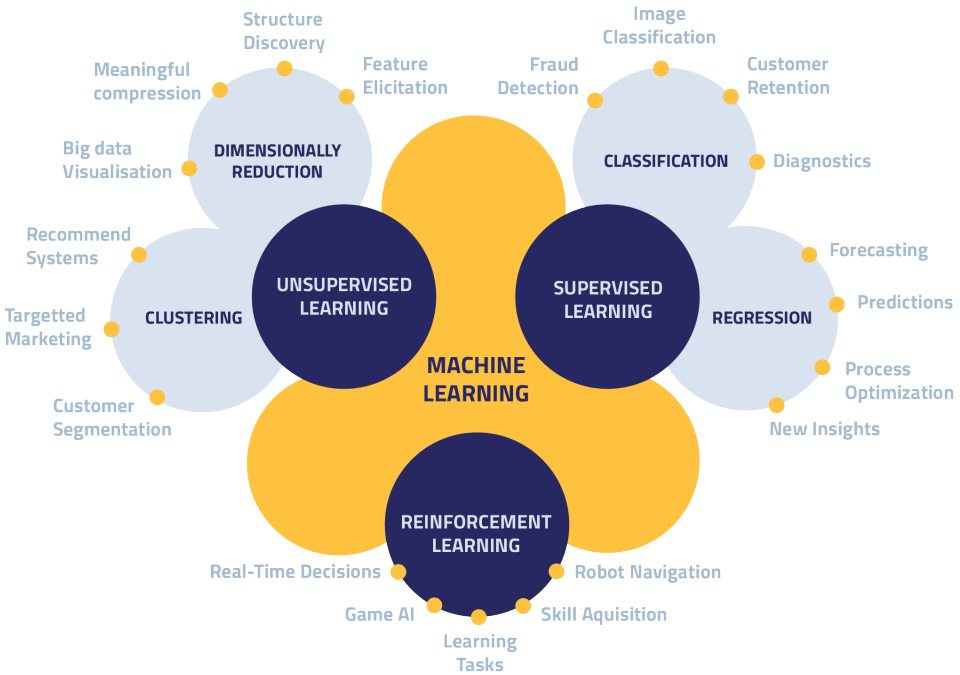
Supervised ML is de benaming van een set van geavanceerde algoritmen die gebruik maken van de informatie van bekende uitslagen, de zogenaamde labels, om de voorspellingen te kunnen maximaliseren. Gelabelde data is het deel van de data waarvan de uitslag bekend en dus getagd is; bijvoorbeeld bij het voorspellen van uithuisplaatsingen beschikt men over de data van kinderen die al uit huis zijn geplaatst. Het ML model kan hierop worden getraind.[1] Hiermee worden ook inzichten verkregen in de features (de variabelen die gebruikt worden voor een voorspelling) door iteratieve multidimensionele dataverkenning. Door middel van validatie bij elke iteratiestap, en bij voorkeur een teststap aan het eind, worden de algoritmen getraind om te worden geoptimaliseerd tegen "onbekende" positieve en negatieve patronen uit toekomstige data. Dat is de kracht van supervised ML.

Unsupervised ML

Machine Learning kan ook unsupervised worden uitgevoerd en worden gebruikt om baseline gedragsprofielen voor verschillende entiteiten te leren en vast te stellen om die vervolgens te gebruiken om zinvolle anomalieën te vinden. Indien er geen labels (geen bekende uitkomsten) beschikbaar of moeilijk te genereren zijn, wordt de voorkeur gegeven aan unsupervised learning. Een vorm van unsupervised ML kan bijvoorbeeld ingezet worden om informatie uit teksten te extraheren die op het eerste gezicht lastig uit de data te halen is.

Reinforcement Learning

Reinforcement Learning, geïnspireerd door gedragspsychologie, is een manier van versterkend leren via het idee van cumulatieve beloning door algoritmes in een omgeving acties te laten ondernemen. Op deze manier krijgt het algoritme directe feedback, iteratief via beloning (of straf), en maakt daardoor steeds betere keuzes. Reinforcement Learning is meer gericht op online planning en vereist een balans tussen exploratie (in het onbekende) en compliance (bestaande kennis). Figuur 2 geeft een grafische weergave van deze drie brede categorieën.



Figuur 2: De drie brede categorieën van Machine Learning, supervised, unsupervised en Reinforcement Learning, en de verschillende toepassingen.

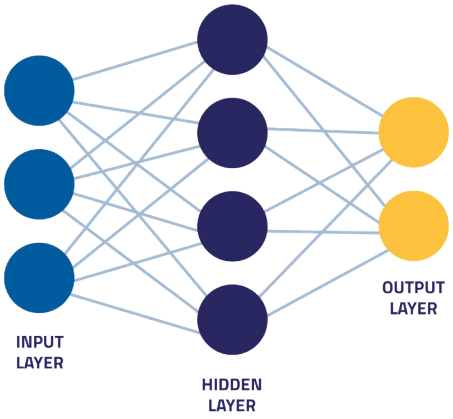
[1] Dit betekent dat het algoritme steeds beter wordt in het vinden van oplossingen, net zoals mensen leren aan de hand van ervaringen.

2.4 Deep Learning

Deep Learning (DL) is een onderdeel van ML. Het houdt zich bezig met algoritmen die zijn geïnspireerd op de structuur en functie van de hersenen. Deze algoritmen worden daarom kunstmatige neurale netwerken (Artificial Neural Networks, ANN) genoemd. DL is in feite een groot neuraal netwerk bestaande uit vele lagen, soms wel honderden. Deze gespecialiseerde techniek is tegenwoordig heel populair onder hard core techneuten.

De roots van DL gaan terug tot begin jaren zestig, maar pas vanaf 2010 wordt het structureel ingezet. Afgelopen tien jaar heeft DL ook veel successen geboekt door meerdere challenges te winnen, zoals de ImageNet challenge.[2] Ook de Turing Award voor excellentie in onderzoek naar kunstmatige intelligentie van 2019 werd toegekend aan drie van de meest invloedrijke architecten van DL: Yann LeCun van Facebook, Geoffrey Hinton van Google en Yoshua Bengio van de Universiteit van Montreal. Dit trio heeft, samen met vele anderen in het afgelopen decennium, de algoritmen, systemen en technieken ontwikkeld die verantwoordelijk zijn voor de enorme groei van de door AI aangedreven producten en diensten.

DL is een heel krachtige techniek die reguliere ML vaak overtreft in het juist voorspellen en in het herkennen van patronen. Het heeft echter veel data als input nodig om te 'leren' en is een black box-model. Dit zijn meteen ook de twee grootste nadelen van DL. In Figuur 3 is structuur van de simpele neurale netwerk afgebeeld.

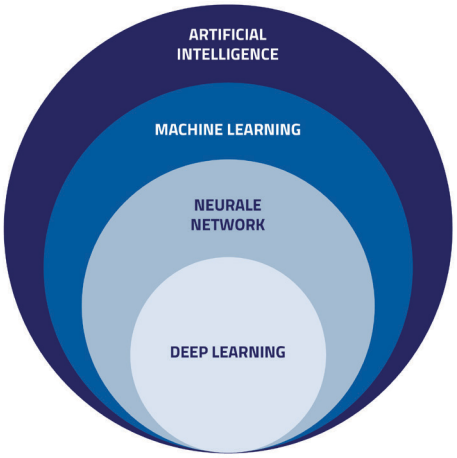


Figuur 3: De opbouw van een simpele neurale netwerk met één 'hidden' laag. DL heeft soms honderden lagen. Hoe meer lagen, des te meer details herkend kunnen worden. Te veel lagen kan ook leiden tot overfitting. De optimale balans tussen het aantal lagen en het aantal neuronen (4 in dit voorbeeld) moet gevonden worden.

Enkele voorbeelden van DL toepasingen zijn de ontgrendelingsfunctie van je telefoon en beeldherkenning, en is de reden waarom Alexa en Siri je stem begrijpen. Het zorgt ervoor dat Microsoft Translator en Google Maps goed werken. Commerciële bedrijven zoals Spotify en Netflix zetten DL in om erachter te komen wat je hierna wilt luisteren of wilt bekijken.

2.5 AI, ML en DL hiërarchie

Over het algemeen worden ML, ANN en DL gezien als onderdeel van artificiële intelligentie, zoals Figuur 4 hieronder weergeeft. De gespecialiseerde en tegenwoordig heel populaire techniek DL valt in deze hiërarchie onder ML en dus onder AI.



Figuur 4: De verhouding tussen Artificial Intelligence, Machine Learning en Deep Learning.

[2] http://www.image-net.org/challenges/LSVRC/

3. Tekstanalyse

In dit hoofdstuk lichten we text mining en text analytics nader toe, omdat dit vakgebied tegenwoordig een enorme groei doormaakt en steeds belangrijker wordt. De groei wordt voortgestuwd door de snelle ontwikkeling van de analysetechnieken in de afgelopen zes jaar. Maar ook door de grote hoeveelheden aan beschikbare tekstuele data in de wereld is de vraag ernaar groot. Waar gestructureerde data (doorgaans numeriek) niet zo lang bestaat, bestaat ongestructureerde data (met name teksten) al veel langer. De gedachtegang is dat ongestructureerde data het overgrote deel van de gegevens in een organisatie omvatten, sommige schattingen lopen op tot 80%.[3] Denk bijvoorbeeld aan alle juridische teksten en aan alle verslagen en rapporten van bedrijven geschreven in de afgelopen decennia, de potentie is enorm.

3.1 Text mining en text analytics

Text mining is het proces van het verkennen en analyseren van grote hoeveelheden tekstdata. Met behulp van software kunnen concepten, entiteiten, onderwerpen, trefwoorden en andere attributen in de data worden geïdentificeerd. Het proces van het verwerken en analyseren van tekst wordt ook wel 'text analytics' genoemd. Hoewel sommigen geen onderscheid maken tussen de twee termen is er wel degelijk een verschil. Text analytics is een toepassing die in feite volgt op de door text mining-technieken doorzochte en bewerkte datasets.

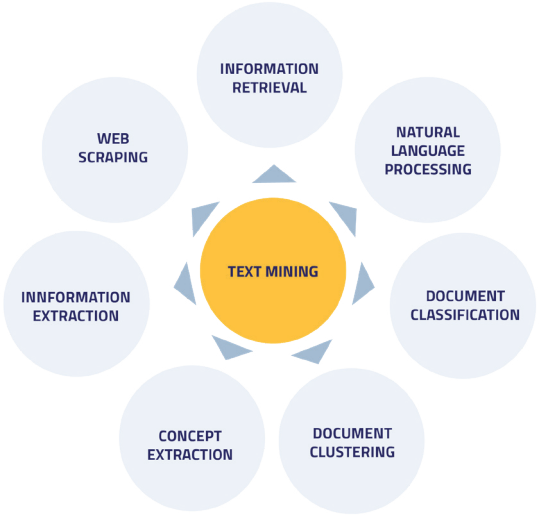
Text mining is praktischer en toegankelijker geworden voor data scientists en andere gebruikers vanwege de ontwikkeling van big data-platformen en deep learning-algoritmen die enorme sets ongestructureerde data kunnen analyseren.

Door de teksten te analyseren is het mogelijk om waardevolle inzichten te krijgen in bijvoorbeeld bedrijfsdocumenten, e-mails van klanten, logboeken van callcenters, ingevulde vragenlijsten, berichten in sociale netwerken, medische dossiers en andere bronnen van op tekst gebaseerde gegevens. In toenemende mate worden de mogelijkheden van text mining en text analytics ook opgenomen in AI-chatbots en gebruikt bij virtuele agenten die door bedrijven worden ingezet om klanten geautomatiseerde antwoorden te sturen als onderdeel van hun verkoop-, marketing- en klantenservice-activiteiten.

Hoe werkt text mining?

Eén van de eerste stappen in het text mining-proces is om de gegevens op een bepaalde manier te ordenen en te structureren, zodat ze zowel aan kwalitatieve als kwantitatieve analyses kunnen worden onderworpen. Daarbij wordt vaak gebruik gemaakt van NLP-technologie, die de principes van computationele taalkunde toepast op het analyseren en interpreteren van datasets.

Het voorbereidende werk bestaat uit het categoriseren, clusteren en taggen van tekst, het samenvatten van datasets, taxonomieën creëren, het modelleren van onderwerpen, en het extraheren van informatie van bijvoorbeeld woordfrequenties en relaties tussen gegevensentiteiten. In Figuur 5 wordt text mining grafisch weergegeven.



Figuur 5: De toepassingen van text mining. Ook zijn er overlappen met vakgebieden zoals data mining, computational linguistics, ML, information sciences en databases.

[3] Shilakes, Christopher C. en Tylman, Julie (16 nov. 1998). "Enterprise Information Portals". Merrill Lynch.

Hoe werkt text analytics?

Na het voorbewerken van de tekst en het voorbereidende werk en de informatie-extractie kunnen analytische modellen worden uitgevoerd om bevindingen te genereren. Deze bevindingen kunnen vervolgens helpen bij het sturen van bedrijfsstrategieën en operationele acties. Onder text analytics vallen analytische technieken zoals het vinden van associaties, voorspellende analyses, patroonherkenning, annotatie en lexicale analyse.[4] Deze bevindingen kunnen daarna statistisch geëvalueerd en visueel weergegeven worden.

3.2 Natuurlijke taalverwerking (NLP)

Alles wat we uitdrukken (mondeling of schriftelijk) bevat enorme hoeveelheden informatie. Het onderwerp dat we kiezen, onze toon, onze woordkeuze, alles voegt een soort informatie toe die kan worden geïnterpreteerd en waaraan waarde kan worden ontleend. In theorie kunnen we met die informatie menselijk gedrag begrijpen en zelfs voorspellen.

Natuurlijke taalverwerking ofwel Natural Language Processing (NLP) is de toepassing van computationele technieken op de analyse en synthese van natuurlijke taal en spraak. Het geeft machines de mogelijkheid om menselijke talen te begrijpen en te verwerken. In praktisch opzicht gaat het met name om hoe computers geprogrammeerd kunnen worden voor het verwerken en analyseren van grote hoeveelheden natuurlijke taal data.

Het ontginnen (mining), clusteren en analyseren van ongestructureerde gegevensbronnen vereist daarom het gebruik van analytische technieken voor natuurlijke taal die subjectieve informatie uit taal kan extraheren.

NLP omvat technieken zoals Named Entity Recognition (NER) en Part-of-Speech (POS) tagging. NER herkent entiteiten zoals organisaties, mensen, locaties, data, tijd en valuta. POS herkent woordsoorten zoals werkwoorden, zelfstandige naamwoorden etc.

3.3 Topic modeling

Waar praten mensen eigenlijk over? Is een vraag die we vaak willen beantwoorden als het gaat om social media data. Maar ook bij gesprekken met een huisarts of een zorghulpverlener of zelfs bij een aandeelhoudersvergadering is het soms belangrijk om te achterhalen wat het onderwerp van het gesprek was. Een van de belangrijke taken in tekstanalyse is om de kwaliteitszinnen van een tekstcorpus te identificeren. Aan de hand van deze zinnen kan vervolgens belangrijke informatie onttrokken worden uit documenten door toepassing van verschillende zinsdeeltechnieken, bijvoorbeeld topic modeling.

Een topic model is een soort statistisch model voor het ontdekken van de abstracte onderwerpen (“topics”) die voorkomen in een verzameling documenten. Modellering van onderwerpen is een veelgebruikte tool voor text mining als het gaat om het ontdekken van verborgen semantische structuren in een tekstcorpus.

Voor het ophalen van informatie worden vaak specialistische technieken zoals termfrequentie en inverse frequentie (tf-idf) ingezet. Via deze en soortgelijke technieken kunnen topics uit documenten worden geëxtraheerd.

Voordat topic modeling kan worden toegepast, moet eerst de tekst worden opgeschoond, door stopwoorden te verwijderen, te lemmatiseren en synoniemen te bepalen op basis van een lexicale database. Het goed opschonen van teksten is van groot belang voor topic modeling, maar ook in het algemeen voor tekstanalyse, omdat je alleen dan een grote verbetering in de uitkomsten ziet.

Een uitdaging is het vaststellen van het optimale aantal topics in een document. Voor de implementatie van een model, moet vooraf het aantal onderwerpen bepaald worden. Voor sommige documenten, zoals bij notulen, kan het vrij moeilijk zijn om dit vooraf te doen.

[4] Lexicale analyse is een term uit de informatica waarmee het omzetten van een reeks karakters in een reeks symbolen wordt aangeduid, zie ook https://nl.wikipedia.org/wiki/Lexicale_analyse.

3.4 Sentimentanalyse

Sentimentanalyse is de interpretatie en classificatie van emoties (positief, negatief en neutraal) binnen teksten met behulp van tekstanalysetechnieken. Sentimentanalyse stelt bedrijven in staat het sentiment van klanten ten opzichte van producten, merken of diensten te identificeren in online gesprekken en in feedback.

De kwantificatie van tekst op het onderliggende sentiment kan worden toegepast op documentniveau of op zinsniveau. Een uitdaging hier is de taal van het document; taalspecifieke oplossingen vergen meer inspanning, terwijl er bij een vertaling naar het Engels informatie verloren kan gaan.

Een van de pijnpunten van sentimentanalyse is dat het niet goed overweg kan met ironie en sarcasme, wat ook voor mensen soms moeilijk te herkennen is.

4. Typen modellen

Er zijn vele verschillende typen modellen beschikbaar waarmee voorspellingen gedaan kunnen worden of gelijkenissen en patronen uit data geëxtraheerd kunnen worden. Sommige modellen zijn meer beschrijvend en verkennend van aard en andere zijn meer voorspellend. Vaak hangt de voorspellende kracht van een model samen met hoe transparant en inzichtelijk een model is: hoe inzichtelijker het model des te zwakker de voorspelling.

Toch is er software beschikbaar en zijn er methodes toepasbaar die niet-transparante modellen, ofwel black box-modellen, inzichtelijker kunnen maken. Explainable Artificial Intelligence (XAI) is een specialistisch vakgebied dat specifiek hierop ingaat. Dit onderzoeksgebied pakt het belangrijke probleem aan dat complexe machines en algoritmen vaak geen inzicht kunnen geven in hun gedrag en denkprocessen. Twee goed leesbare review papers die zich hierin verdiepen en aan te raden zijn Adadi & Berrada, 2018[5] en Gilpin et al., 2019.[6]

In dit hoofdstuk worden meerdere modellen kort toegelicht en de voor- en nadelen opgesomd.
De modellen zijn ingedeeld in drie verschillende groepen: regressiemodellen, classificatiemodellen en clusteringmodellen. De focus ligt met name op de laatste twee, omdat deze meer van toepassing zijn op de casus van de gemeente Den Haag.

4.1 Regressiemodellen

4.1.1 Lineaire regressie

Een lineaire regressie is een statistisch model dat de relatie analyseert tussen een afhankelijke variabele (vaak de Y-variabele genoemd) en een of meer onafhankelijke variabelen (vaak X, verklarende variabelen, of features genoemd). Over het algemeen gesproken gaat het altijd om continue variabelen. Indien er een enkel onafhankelijke variabele beschouwd wordt, dan spreekt men vaak over een simpele lineaire regressie.

Een lineair model drukt de afhankelijke variabele (Y) uit in een som van gewogen inputvariabelen (X_1, X_2, … X_n), zoals bijvoorbeeld $Y = \alpha + \beta X_1 + \gamma X_2$. De veelgebruikte methode voor het schatten van de coëfficiënten (α, β, γ) voor lineaire regressieproblemen wordt kleinste kwadraten lineaire regressie genoemd, simpelweg ook wel kleinste kwadraten, of least-squares in het Engels. Lineaire modellen maken vooraf een sterke aanname over de relatie tussen de invoer X en de uitvoer Y. In dit geval zijn er geen parameters om de complexiteit van modellen te beheersen. Het lineaire model gebruikt alle inputvariabelen en wordt weergegeven door een rechte lijn.

Er is een andere manier om de coëfficiënten van het regressiemodel te schatten, en dat is met behulp van regularisatie. Regularisatie is het proces waarbij informatie wordt toegevoegd om een slecht gesteld probleem op te lossen of om overfitting[7] te voorkomen.

Regularisatie is een uiterst belangrijk concept in ML. Door het toevoegen van een zogenaamde strafterm (een extra factor dat ook bekend staat als penalty term) aan de objectieve functie[8] helpt het bij het voorkomen van overfitting en verbetert zo de algemene prestaties van het model. De parameterinstellingen van het model worden in feite door de strafterm ingeperkt. Het effect van deze beperking is dat de complexiteit van het uiteindelijke geschatte model vermindert en daardoor eleganter wordt.

4.1.2 Lasso- en Ridge-regressie

Het is belangrijk om in gedachten te houden dat dit bij het trainen van een regressiemodel op een groot aantal variabelen gemakkelijk kan leiden tot overfitting. In dit geval generaliseert het model niet goed. Regularisatie kan het risico van overfitting voorkomen. Door een penalty toe te voegen, kan de complexiteit van het model worden beheerst.

[5] Adadi, Amina end Berrada, Mohammed (2018), Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), IEEE Access.
[6] Gilpin, L.H., Bau D., en Yuan, B.Z. et al. (2019), Explaining Explanations: An Overview of Interpretability of Machine Learning. ArXiv180600069 Cs Stat. https://arxiv.org/abs/1806.00069.
[7] Overfitting is de populaire term voor het 'te goed' fitten van de gebruikte data waardoor het model niet goed presteert, oftewel generaliseert, voor andere data.
[8] De objectieve functie, of de cost function, is de mathematische formule voor het optimaliseren van het model.

Een van de populaire methoden die toegepast wordt is Lasso-regressie. Lasso-regressie (of L1 regularization) voegt een regularisatie-strafterm toe aan de som van de absolute waarden van coëfficiënten waardoor de modelparameters (coëfficiënten) naar nul krimpen. Lasso-regressie dwingt een subset van de coëfficiënten precies nul te zijn. In feite is dit een manier van automatische selectie van variabelen (features). De variabelen met nulcoëfficiënten worden in wezen volledig genegeerd in het model. Het is over het algemeen handig als er een groot aantal features is. De gereduceerde oplossing biedt een subset van de belangrijkste kenmerken (met coëfficiënten groter dan nul) en verhoogt daarmee de interpreteerbaarheid van het model.

Een andere techniek om het risico van overfitting te voorkomen is Ridge-regressie (of L2 regularization). Ridge-regressietechniek kwadrateert de coëfficiënten in de strafterm om de objectieve functie te minimaliseren. Dit heeft de neiging coëfficiënten van de minder belangrijke features te sturen naar bijna nul, maar niet helemaal tot nul (in tegenstelling tot Lasso, L1). Dit helpt bij het voorkomen van multicollineariteit en vermindert de complexiteit van het model door de krimp in coëfficiënten.

De hoeveelheid toe te passen regularisatie wordt bepaald door een (hyper)parameter, λ. Een grotere waarde voor de parameter betekent meer regularisatie en eenvoudigere lineaire modellen met gewichten die dichter bij nul liggen. Te veel regularisatie kan dus leiden tot een underfitting van de data. Meer informatie over hyperparameters is te vinden in sectie 5.7.

4.1.3 Elastic Net regressie

Een derde veelgebruikt regressiemodel is Elastic Net. Elastic Net maakt gebruik van zowel L1- als L2-regularisatie.

Elastic Net ontstond als gevolg van kritiek op Lasso, omdat de Lasso-methode te afhankelijk kan zijn van de onderliggende data en is dus onstabiel. De oplossing was om het beste van beide technieken te gebruiken door de straftermen van Ridge en Lasso te combineren. Naast de hyperparameter λ is er een tweede hyperparameter α. Deze alfa parameter term wordt toegevoegd aan de formule (objectieve functie) waarmee er bepaald wordt hoeveel L1-regularisatie ten opzichte van L2-regularisatie wordt toegepast. Zodoende probeert Elastic Net de objectieve functie nog beter te minimaliseren.

4.2 Classificatiemodellen

4.2.1 Decision Tree

Een eenvoudig interpreteerbare classificatiemethode is de beslissingsboom, in het Engels ook wel Decision Tree genoemd. Een Decision Tree is daarentegen een van de vaker toegepaste classificatiemethodes vanwege de hoge inzichtelijkheid dat het verschaft. Decision Trees worden vaak als leuk en managementvriendelijk beschouwd. Een grafische weergave is snel gemaakt en het geeft een duidelijk overzicht van de route naar de voorspelling.

De training observaties worden zo heterogeen mogelijk opgesplitst in twee subgroepen. Bij elk knooppunt wordt bekeken welke variabele de data het beste op kan splitsen in twee nieuwe knooppunten. Op deze manier wordt de data steeds verder opgesplitst, net zolang totdat aan een stopcriterium is voldaan (bijvoorbeeld minder dan n training observaties per knooppunt). Op basis van de waarden voor de variabelen, valt elke observatie uiteindelijk in één klasse (dus in één leaf).

Een voordeel van een Decision Tree is dat het begrijpelijk is, dat de resultaten goed te interpreteren zijn en dat de resultaten grafisch kunnen worden weergegeven. Het nadeel van Decision Trees is echter dat ze instabiel kunnen zijn en dat een relatief kleine verandering in de data kan resulteren in een andere boom. Met andere woorden: een Decision Tree kan knopen bevatten die gecreëerd zijn door specifieke gevallen in de trainingsdataset, waardoor het model zich slecht laat generaliseren naar andere data. Het model heeft dan een grote ‘generalization error’, een fenomeen dat ook wel ‘overfitting van de data’ genoemd wordt. Een manier om dit te voorkomen is het gebruik van ensemble

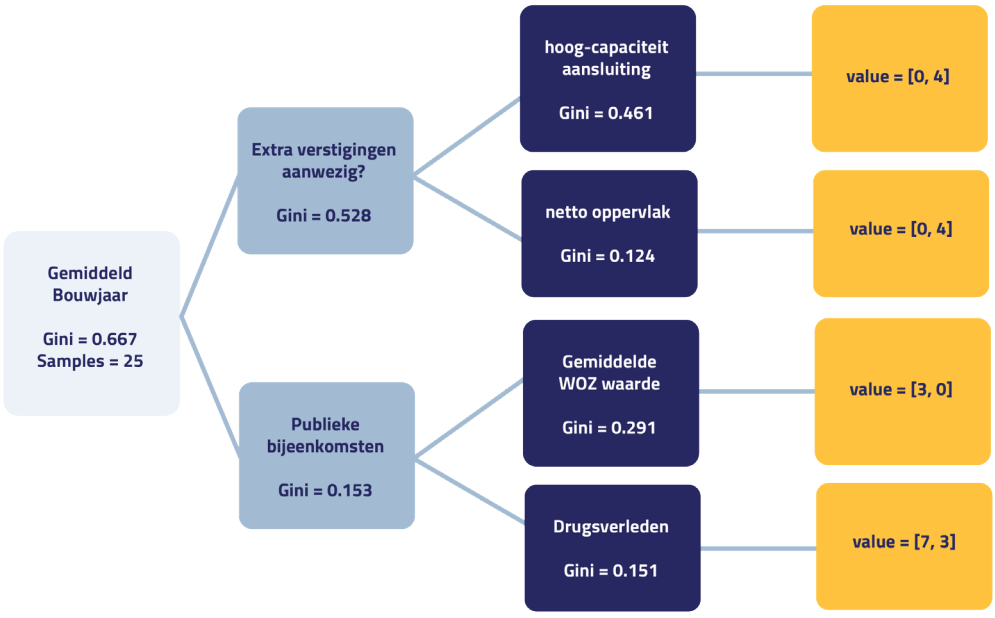
methoden, zoals bijvoorbeeld een Random Forest (zie sectie 4.2.2).

Een goed begrip van de voor- en nadelen van een Decision Tree-analyse laat zien dat de nadelen van de Decision Tree veel van de voordelen tenietdoen. Dit is vooral het geval bij grote en complexe Decision Trees, waardoor de wijdverbreide toepassing van Decision Trees als besluitvormingsinstrument wordt belemmerd.

Er zijn veel specifieke Decision Tree-algoritmen. De meest bekende en gebruikte Decision Trees vallen onder de zogenaamde CART-algoritmes. CART staat voor Classification and Regression Tree wat een overkoepelende term is voor Decision Trees die voor regressie en voor classificatie worden gebruikt. De term CART werd geïntroduceerd door Breiman et al. in 1984.[9] Bij de CART-type Decision Trees wordt de Gini impurity[10] (niet te verwarren met Gini-coëfficiënt) gebruikt om de kwaliteit van de splitsing te bepalen. Figuur 6 laat een voorbeeld van een Decision Tree-output zien.
Andere bekende Decision Tree algoritmen zijn onder meer: ID3, C4.5, CHAID en MARS. Er zijn overeenkomsten, maar ook verschillen tussen de typen Decision Trees, zoals de procedure die wordt gebruikt om te bepalen waar de boom moet worden gesplitst.

Voordelen:
- Makkelijk te begrijpen.
- Snel in opzet en in runtime.
- Kan goed omgaan met missings.
- Overzichtelijk en goed visualiseerbaar.
- Kan zowel numerieke als categorische variabelen aan.
- Onbelangrijke features en uitschieters worden automatisch genegeerd.

Nadelen:
- Scheiding is lineair.
- Leidt snel tot overfitting.
- Greedy (kijkt niet verder dan het moment).
- Voorspellend vermogen is zwak en onstabiel.
- Alleen binaire splitsing mogelijk (behalve bij CHAID).



Figuur 6: Voorbeeld van de opbouw van een Decision Tree.

[9] Breiman, L., Friedman, J., Olshen, R., en Stone, C. (1984), Classification and regression trees. Wadsworth Books, p. 358.
[10] https://en.wikipedia.org/wiki/Decision_tree_learning#Gini_impurity

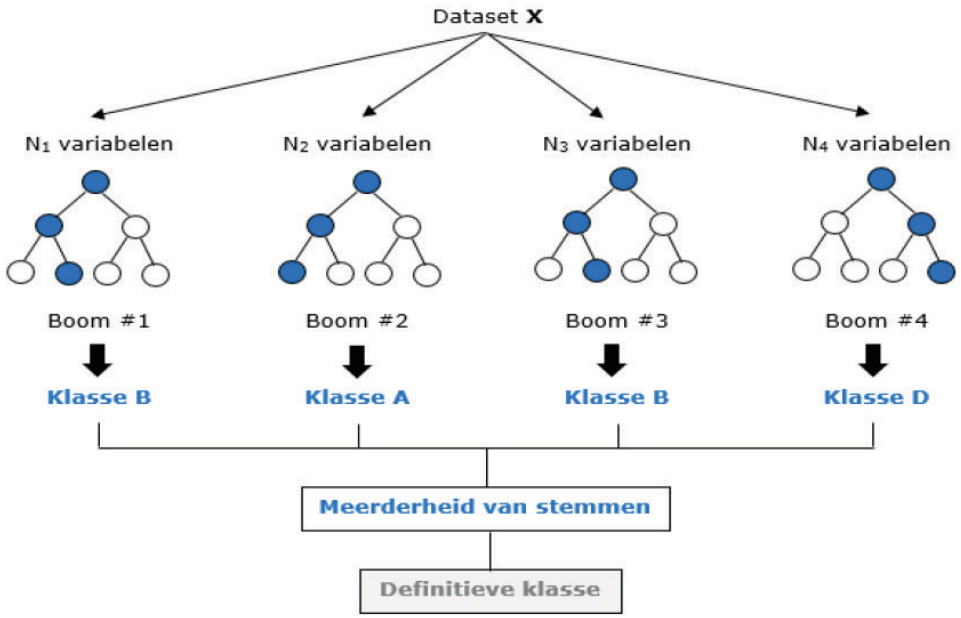
4.2.2. Random Forest

Random Forest is een populaire 'tree-based supervised learning' techniek, waarbij een groot aantal Decision Trees gecombineerd wordt om tot de uiteindelijke voorspelling te komen. We spreken over een ensemble methode als meerdere modellen gecombineerd worden in een groot model. Decision Trees zijn dus de bouwstenen van een Random Forest. Het combineren van vele losse Decision Trees tot een ensemble model resulteert in een hogere precisie en ook in stabielere voorspellingen. Een Random Forest geeft over het algemeen veel betere voorspellingen dan een Decision Tree.

De verzameling bomen wordt random genoemd, omdat elke boom getraind wordt op een willekeurige selectie van variabelen en observaties (met terugleggen). Elke individuele boom spuwt een voorspelling uit. Als de doelvariabele die we willen voorspellen categorisch is, wordt de uiteindelijke uitkomst bepaald door middel van 'majority voting'. Met andere woorden: de uitkomst van de meeste bomen geldt als einduitkomst (Figuur 7).

Het fundamentele concept achter Random Forest is eenvoudig maar krachtig - de wijsheid van menigten. De voorspellingen (en dus de fouten) van de individuele bomen moeten echter wel een lage correlatie met elkaar hebben.

De lage correlatie tussen modellen is de sleutel. Net zoals investeringen met lage correlaties (zoals aandelen en obligaties) samenkomen om een portefeuille te vormen die groter is dan de som van de delen, kunnen niet-gecorreleerde modellen ensemble-voorspellingen produceren die nauwkeuriger zijn dan alle individuele voorspellingen. De reden voor dit geweldige effect is dat de bomen elkaar beschermen tegen hun individuele fouten (zolang ze niet constant allemaal in dezelfde richting dwalen). Sommige bomen hebben misschien ongelijk, maar veel andere bomen hebben gelijk.



Figuur 7: Schematische weergave Random Forest algoritme.

Voordelen:
- Redelijk simpel en makkelijk inzetbaar model.
- Veel betere voorspellingen dan een Decision Tree.
- Geeft een betrouwbare lijst met belangrijkste features.
- Relatief goed interpreteerbaar.
- Weinig kans op overfitting.
- Kan overweg met missing data.

Nadelen:
- Veel hyperparameters.
- Minder goed bij underfitting van data.
- Computationeel duur en geheugen intensief.

4.2.3 Gradient Boosting

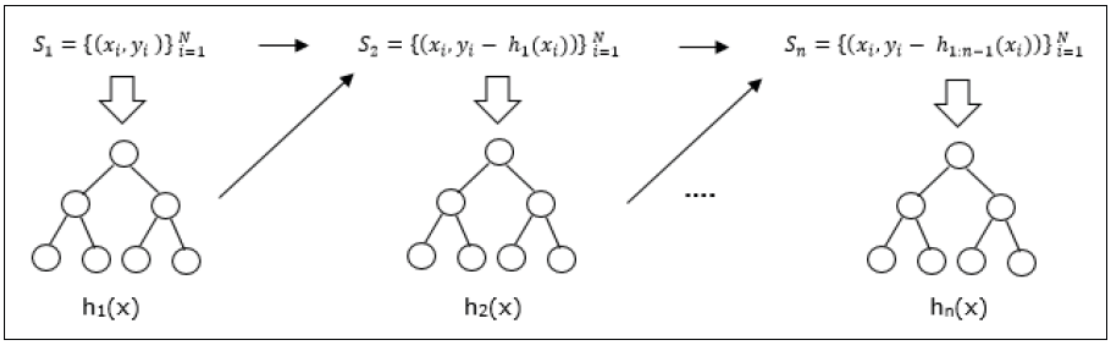
Gradient Boosted Tree, doorgaans ook gewoon Gradient Boosting genoemd, is eveneens een ensemble methode om Decision Trees te verbeteren. Waar Random Forest de individuele bomen parallel samenvoegt (bagging), werkt Gradient Boosting op een sequentiële manier (boosting).

Bij Gradient Boosting wordt gebruik gemaakt van boosting technieken om de voorspellende waarde van classificatie en regressie te optimaliseren. Gradient Boosting is een van de krachtigste ML technieken voor het maken van voorspellende modellen en wordt vaak ingezet bij ML competities, zoals Kaggle.[11] Het is zelfs een meme geworden dat XGBoost vaak het standaardrecept is voor het winnen van ML-wedstrijden. Bij Gradient Boosting worden er in een stapsgewijs proces meerdere modellen gecreëerd, waarbij elke nieuw model leert van het voorgaande model. Gradient Boosting generaliseert de modellen door de optimalisatie van een willekeurige differentieerbare verliesfunctie toe te staan (zie Figuur 8). Van een weak learner (Decision Tree) wordt op deze manier een strong learner (Gradient Boosting) gemaakt. Boosting is vaak een belangrijk onderdeel in veel aanbevelingssystemen.

Enkele bekende boosting-algoritmen zijn:
- AdaBoost
- XGBoost
- GBM
- CatBoost

Voordelen:
- Heel krachtig voorspellend model, wint vaak competities.
- Relatief goed interpreteerbaar.
- Helpt bij het underfitting van data.
- Veel boosting technieken en veel literatuur beschikbaar.

Nadelen:
- Langzaam om te trainen en te optimaliseren.
- Computationeel duur, moeilijk paralleliseerbaar.
- Veel hyperparameters.
- Gevoeliger op uitschieters dan Random Forest.



Figuur 8: Stapsgewijs optimalisatie via Gradient Boosting.

[11] https://www.kaggle.com/

4.2.4 Logistic Regression

Alhoewel de naam een regressietechniek suggereert, is Logistic Regression in feite een classificatietechniek. Logistic Regression is een algoritme dat wordt gebruikt om waarnemingen toe te wijzen aan een afzonderlijke set van twee of meer klassen.

Over het algemeen maakt de Logistic Regression classificatie gebruik van een lineaire combinatie van meerdere verklarende variabelen (X_1, X_2, … etc), net als bij lineaire regressie (dus $Y = \alpha + \beta X_1 + \gamma X_2 + \ldots$ etc, zie sectie 4.1.1). Maar nu wordt de lineaire combinatie van variabelen als argument van de sigmoïdefunctie gebruikt.

De sigmoïdefunctie is = $\frac{1}{1+e^{-Y}}$

De bijbehorende uitvoer van de sigmoïdfunctie is een getal tussen 0 en 1. Een input die een uitkomst van 0,5 of meer oplevert wordt tot klasse 1 gerekend, terwijl een output van 0,5 of minder tot klasse 0 wordt gerekend. In tegenstelling tot lineaire regressieanalyse, waarbij de afhankelijke variabele een continue variabele is, is de afhankelijke variabele bij Logistic Regression een categorische (discrete) variabele.

Ook bij Logistic Regression worden regularisaties ingezet. Regularisties zijn een belangrijke techniek die vaak wordt gebruikt om overfitting van het model te voorkomen. Deze technieken leggen een penalty (straf) op aan de coëfficiënten van het model om hun invloed op het model te reguleren. Ridge-techniek (L2) kwadrateert de coëfficiënten in de strafterm en heeft de neiging om de coëfficiënten van minder belangrijke variabelen te sturen tot bijna nul, maar niet helemaal tot nul. Lasso (L1) daarentegen stuurt een aantal coëfficiënten helemaal naar nul en helpt ook bij het selecteren van de variabelen.

Logistic Regression voor meer dan twee klassen heet multiclass-classification. Via het one-vs-rest-schema (OVR), waarin elke klasse alsnog als een binair classificatieprobleem beschouwd wordt, kunnen meerdere klassen behandeld worden. Voor meer informatie hierover kan verwezen worden naar de Scikit-Learn webpagina.[12]

Voordelen:

- Transparant en makkelijk interpreteerbaar model.
- Presteert heel goed wanneer de dataset lineair te scheiden is.
- Regularisatie (L1 en L2) technieken zijn makkelijk toepasbaar.
- Geeft de richting van variabele associatie (positief of negatief).
- Makkelijk te implementeren en zeer efficiënt te trainen.
- Multicollineariteit is geen probleem.

Nadelen:

- Minder voorspellende kracht dan andere ML-modellen (beter dan Decision Tree).
- Gaat uit van lineaire afhankelijkheid.
- Presteert niet goed als er veel features zijn.
- Kan een groot aantal categoriëen per variabele niet goed verwerken.

[12] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

4.2.5 K-Nearest Neighbours (k-NN)

k-Nearest Neighbours (k-NN) is een relatief eenvoudig algoritme dat alle beschikbare gevallen opslaat en nieuwe gevallen classificeert op basis van een afstandsfunctie. Een geval wordt geclassificeerd aan de hand van een meerderheid aan stemmen van zijn buren, waarbij het geval wordt toegewezen aan de klasse die het meest voorkomt bij zijn k naaste buren. k-NN is een niet-parametrische methode voor classificatie én regressie en wordt al sinds begin jaren zeventig gebruikt.

Het kiezen van (de optimale) k is niet triviaal. De beste keuze voor k hangt af van de data; in het algemeen verminderen grotere waarden van k het effect van de ruis op de classificatie, maar maken ze de grenzen tussen klassen minder duidelijk. Een goede k kan worden geselecteerd door middel van verschillende heuristische technieken. Ook kan k als een hyperparameter beschouwd worden. Bij een binaire classificatieproblemen is het handig om k oneven te kiezen, omdat dit het probleem van gelijke stemmen oplost.

Er bestaan ook varianten van k-NN, zoals "Centroid k-NN"[13] en verschillende afstandsfuncties, zoals Euclidean, Manhattan, Hamming, Minkowski en meer.[14]

Voordelen:

- Simpele algoritme, makkelijk in gebruik en interpretatie.
- Veelzijdig, kan zowel voor classificatie als regressie gebruikt worden.
- Non-parametric, maakt geen aannames van onderliggende data.
- Snel in training, veel sneller dan de meeste andere ML algoritmen.

Nadelen:

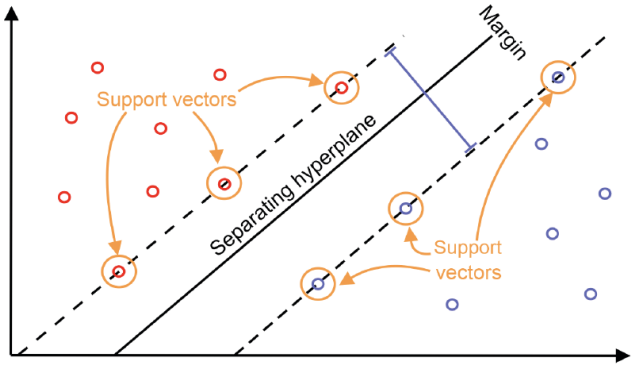
- Vereist veel geheugen, omdat alle datapunten worden opgeslagen.
- Lazy algoritme, voorspellingsfase is computationeel duur.
- Gevoelig voor uitschieters.

4.2.6 Support Vector Machine

Support Vector Machine is een supervised learning algoritme dat kan worden gebruikt voor classificatie- en regressieproblemen als ondersteunende vectorclassificatie (SVC) en ondersteunende vectorregressie (SVR).

Support Vector Machine is gebaseerd op het concept van marge. Het is gebaseerd op het idee om een hypervlak[15] te vinden dat de functies het beste in verschillende domeinen scheidt. De punten die zich het dichtst bij het hypervlak bevinden, worden de ondersteunende vectorpunten genoemd en de afstand van de vectoren tot het hypervlak worden de marges genoemd. Marge kwantificeert de mate waarin de classificator de klassen kan opsplitsen in twee regio's met een zekere mate van scheiding tussen de klassen. Deze classificator voor maximale marges wordt de Linear Support Vector Machine genoemd, ook bekend als LSVM of een Support Vector Machine met een lineaire kernel.

Figuur 9 geeft een beeld van de ondersteunende vectoren bij een Support Vector Machine.



Figuur 9: Support Vector Machine.

[13] Zhang Q. en Sun S. (2010), A Centroid k-Nearest Neighbor Method. In: Cao L., Feng Y., en Zhong J. (eds) Advanced Data Mining and Applications. ADMA 2010. Lecture Notes in Computer Science, vol. 6440. Springer, Berlin, Heidelberg

[14] Zie bv. https://medium.com/datadriveninvestor/k-nearest-neighbors-knn-7b4bd0128da7

[15] Een hypervlak is het equivalent van een vlak mat meer dan drie dimensies. In een k-dimensionale ruimte heeft een hypervlak k-1 dimensies, net zoals een gewone vlak tweedimensionaal is in de 3D-ruimte.

Voordelen:
- Werkt effectief in hogere dimensies.
- Geschikt als het aantal features meer is dan observaties.
- Beste algoritme wanneer klassen kunnen worden gescheiden
- Uitschieters hebben minder impact door de ondersteunende vectoren.
- Support Vector Machine is geschikt voor binaire classificatie in extreme gevallen.

Nadelen:
- Voor grotere datasets kost het veel tijd om te verwerken.
- Presteert niet goed bij overlappende klassen.
- Hyperparameter selectie is moeilijk.
- Het selecteren van de juiste kernelfunctie kan lastig zijn.
- Snelheid neemt drastisch af bij grote datasets.
- Black box-voorspellingsmodel.

4.2.7 Andere classificatiemodellen

Er zijn nog enkele andere classificatiemodellen die niet behandeld zijn. Enkele bekende classificatiemodellen zijn:
- Bayesiaanse modellen (o.a. Naive Bayes)
- Quadratic classifiers (o.a. QDA)
- Neurale Netwerken (o.a. LSTM, RNN, CNN)

4.3 Clusteringmodellen

Clustering is een van de meest toegepaste verkennende data-analysetechnieken die wordt gebruikt om intuïtie te krijgen voor de structuur van je data. Bij clusteranalyse worden 'natuurlijke' groepen van observaties gemaakt. Observaties zijn hierbij vergelijkbaar met elkaar of, anders gezegd, liggen dicht in de buurt van elkaar in de parameterruimte.

Daardoor kunnen trends en patronen goed in kaart gebracht worden. Met andere woorden, het doel is om homogene subgroepen binnen de data te vinden.
Om de datapunten binnen elk cluster zo vergelijkbaar mogelijk te maken en om te zorgen dat de datapunten tussen clusters zo ver mogelijk van elkaar zijn, wordt er volgens met een distance function (of metric) gewerkt, zoals bijvoorbeeld de Euclidische afstand.16 De keuze voor welke distance function gebruikt moet worden is toepassingsspecifiek.

Belangrijke datavoorbewerking

Voordat de data concreet geanalyseerd gaat worden, moet de data geschikt worden gemaakt voor de analyses. Categorische variabelen moeten eventueel gecodeerd worden en waarden moeten waar nodig samengevoegd worden. Bij numerieke variabelen dient naar uitschieters gekeken te worden en is het verstandig numerieke variabelen te schalen, standaardiseren en eventueel andere transformaties toe te passen, zoals waarden groeperen en nieuwe variabelen maken.

Fictieve persona's

Aan gevonden clusters kunnen fictieve persona's worden gegeven, zodat clusters een gezicht krijgen. Zo vertegenwoordigt elk persona een subgroep mensen met vergelijkbare achterliggende kenmerken, dus met een vergelijkbaar profiel. Ook kunnen de persona's verder uitgewerkt worden door per segment op basis van de aanwezige informatie (gedrags) wetenschappelijke inzichten te ontwikkelen.

4.3.1 K-means

De naam doet vermoeden dat K-means mogelijk vergelijkbaar is met k-NN, maar niets is minder waar. De een is een clusteringalgoritme (unsupervised), terwijl de ander onderdeel is van supervised learning. De gelijkenis houdt in feite op bij de letter 'k'.

K-means clustering is een iteratieve algoritme voor het partitionering van de data in K subgroepen (clusters). Door de data in K niet-overlappende clusters te plaatsen, waarbij elk datapunt tot slechts één groep behoort, probeert het algoritme de intercluster datapunten zo gelijk mogelijk te trekken en de intracluster datapunten zo ver mogelijk van elkaar te houden.

De werkwijze van het algoritme is dat het eerst de

centroïden van de K clusters willekeurig positioneert. Vervolgens wijst het elk datapunt toe aan één van de clusters op basis van de features van de datapunten. De Euclidische afstand[16] wordt hiervoor gebruikt (dit deel is een beetje vergelijkbaar met k-NN). Hierna worden de centroïden opnieuw berekend aan de hand van alle datapunten die behoren tot de clusters en aansluitend daarop worden de datapunten opnieuw toegewezen aan één van de clusters. Iteratief bereikt het uiteindelijk de optimale centroïde en creëert hiermee de optimale clusters.

Voordelen:
- Simpel en eenvoudig te implementeren.
- Is relatief snel.
- Flexibel, verandert mee met de data.
- Kan grote datasets aan.
- Garandeert convergentie (kan óók een nadeel zijn).
- Produceert vaak strakke clusters.

Nadelen:
- Het aantal clusters moet handmatig gekozen worden (optie: Elbow method).
- Inconsistent, uitkomsten kunnen variëren bij reruns.
- Waarden moeten genormaliseerd worden.
- Is gevoelig voor uitschieters.
- Kan grote dimensies niet goed aan.
- Ongeschikt voor categorische variabelen.

4.3.2 K-modes en K-prototypes

K-modes[17] is een uitbreiding van K-means en wordt gebruikt voor het clusteren van categorische variabelen. Het is uit noodzaak ontstaan, omdat K-means alleen geschikt is voor numerieke variabelen.

Om verschillen tussen categorische variabelen te kwantificeren maakt K-modes gebruik van dissimilariteit in plaats van de Euclidische afstand (zoals bij K-means, in sectie 4.3.1). De dissimilariteit is de kwantificering van de totale mismatches tussen twee objecten; hoe kleiner dit getal, des te meer de twee objecten op elkaar lijken. En in plaats van te middelen gebruikt het modi. Een modus is een vector van elementen die de verschillen tussen de vector zelf en elk object van de data minimaliseert. De modi fungeren in feite als centroïden.

De voor- en nadelen van K-modes zijn grotendeels vergelijkbaar met K-means. Waar K-means alleen toegepast kan worden voor numerieke (en continue) variabelen, is K-modes alleen geschikt voor categorische variabelen. Een klein extra voordeel is dat alle continue variabelen getransformeerd kunnen worden tot categorische variabelen. Zo kan lengte getransformeerd worden tot ‘kort’ en ‘lang’ door het kiezen van een grenswaarde. Een bijkomend nadeel is echter dat K-modes niet wijdverspreid beschikbaar is.

Voor gemengde data bestaat er een andere uitbreiding van deze algoritmen, in feite een combinatie van K-means en K-modes. Het heet K-prototypes. K-prototypes is geschikt voor gemengde variabelen, dus zowel numeriek als categorisch.

K-modes en K-prototypes zijn ontworpen door Huang (1997)[18] en verder ontwikkeld door anderen[19,20]. Beide modellen zijn beschikbaar in Scikit-Learn.

4.3.3 Density-based clustering

De op dichtheid gebaseerde clustering (Density-based clustering) werkt door gebieden in de data te detecteren waar punten zijn geconcentreerd en waar ze worden gescheiden door gebieden die leeg of schaars zijn. Punten die geen deel uitmaken van een cluster, worden aangeduid als ruis.

16 De Euclidische afstand (of metriek) is de afstand in rechte lijn tussen twee punten in de Euclidische ruimte, bv. $\sqrt{[(Ax-Bx)^2+(Ay-By)^2+(Az-Bz)^2]}$. In oudere literatuur wordt de metriek ook wel de Pythagoras-metriek genoemd.

17 https://github.com/nicodv/kmodes

18 Huang, Z. (1997), Clustering large data sets with mixed numeric and categorical values, Proceedings of the First Pacific Asia Knowledge Discovery and Data Mining Conference, Singapore, pp. 21-34.

19 Huang, Z.(1998), Extensions to the k-modes algorithm for clustering large data sets with categorical values, Data Mining and Knowledge Discovery 2(3), pp. 283-304.

20 Cao, F., Liang, J, en Bai, L.(2009), A new initialization method for categorical data clustering, Expert Systems with Applications 36(7), pp. 10223-10228.

Density-based clustering algoritmen lossen enkele van de problemen op die aanwezig zijn bij K-means clustering. K-means (en vergelijkbare aloritmen) zijn bijvoorbeeld gebaseerd op bolvormige oplossingen, terwijl Density-based clustering niet gebonden is aan een bepaalde vorm. Ook wijst K-means alle punten toe aan een cluster, zelfs de uitschieters, terwijl Density-based clustering ruis kan filteren. Een derde belangrijk voordeel is dat bij Density-based clustering het aantal clusters niet van tevoren hoeft te worden bepaald, in tegenstelling tot K-means clustering.

Er zijn enkele populaire Density-based clustering-technieken, zoals DBSCAN, HDBSCAN en OPTICS. Elk heeft zo zijn voordelen, we lichten ze hieronder toe. Gedefinieerde afstand (DBSCAN) - Gebruikt een gespecificeerde afstand om dichte clusters te scheiden van minder dichte gebieden en ruis. Het DBSCAN-algoritme is de snelste van de clustermethoden, maar is alleen geschikt als er een zeer duidelijke zoekafstand is die moet worden gebruikt. Dit vereist dat alle betekenisvolle clusters vergelijkbare dichtheden hebben.

Zelfinstellend (HDBSCAN) - Gebruikt een reeks afstanden om clusters van verschillende dichtheden te scheiden van minder dichte gebieden en ruis. Het HDBSCAN-algoritme is het meest datagedreven van de clustermethoden en vereist dus de minste gebruikersinvoer.

Multi-scale (OPTICS) —Gebruikt de afstand tussen naburige functies om een bereikbaarheidscurve te creëren die vervolgens wordt gebruikt om clusters van verschillende dichtheden van ruis te scheiden. Het OPTICS-algoritme biedt de meeste flexibiliteit bij het nauwkeurig afstemmen van de gedetecteerde clusters, hoewel het rekenintensief is, vooral met een grote zoekafstand.

4.3.4 Andere clusteringtechnieken

Er zijn nog veel meer clusteringtechnieken die niet behandeld zijn, maar in de eerste instantie minder interessant lijken voor de casus van de gemeente Den Haag. Enkele populaire daarvan zijn:

- Partitioning clustering (o.a. K-medoids, PAM)
- Hierarchichal clustering (o.a. Agglomerative, Divisive)
- Model-based clustering (o.a. Gaussian Mixture Model)
- Grid-based clustering
- Fuzzy clustering
- Hybride clustering methoden

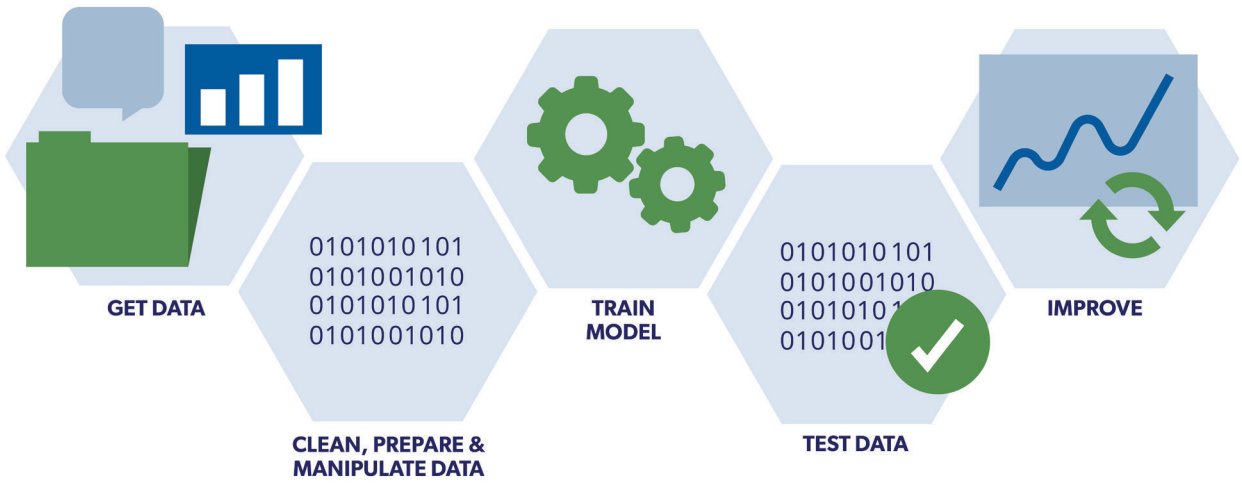
5. Stappenplan

Wanneer een datagedreven onderzoek opgestart wordt, is het handig om een uitgestippeld plan te volgen. In algemene zin zijn er vijf stappen die doorlopen moeten worden om goed onderzoek te kunnen uitvoeren, zie ook Figuur 10.

De stappen zijn:
1. Data verzamelen en klaarmaken
2. Data voorbewerken (o.a. imputeren)
3. Model trainen
4. Testen en evalueren
5. Optimaliseren

De volgorde van het stappenplan kan afwijken en is afhankelijk van het onderzoeksdoel of het type model. Het stappenplan is vaak ook een iteratief proces, bijvoorbeeld na het testen, evalueren en optimaliseren gaat men vaak terug naar het (her)trainen van het model, of zelfs terug naar het (beter) voorbewerken van de data om de modelresultaten te verbeteren.

Verkennende data-analyse is geen aparte stap, maar is wel een belangrijke tussenstap (ergens tussen stappen 1 en 3) die we gaan behandelen. In de volgende secties gaan we dieper in op de belangrijkste stappen en lichten deze nader toe.



Figuur 10: De vijf stappen van een datagedreven onderzoek.

De volgorde van het stappenplan kan afwijken en is afhankelijk van het onderzoeksdoel of het type model. Het stappenplan is vaak ook een iteratief proces, bijvoorbeeld na het testen, evalueren en optimaliseren gaat men vaak terug naar het (her)trainen van het model, of zelfs terug naar het (beter) voorbewerken van de data om de modelresultaten te verbeteren.

Verkennende data-analyse is geen aparte stap, maar is wel een belangrijke tussenstap (ergens tussen stappen 1 en 3) die we gaan behandelen. In de volgende secties gaan we dieper in op de belangrijkste stappen en lichten deze nader toe.

5.1 Data verzamelen

Ten eerste moet de data bemachtigd worden. Soms wordt deze verkregen van een externe partij en soms wordt deze gegenereerd met eigen middelen. In het geval van de gemeente Den Haag is het een reeds bestaande databron van mensen in de bijstand.
De data kan uit verschillende bronnen bestaan, maar ook aangevuld worden door additionele bronnen zoals bijvoorbeeld CBS Maatwerk en Microdata[21], BAG data[22], Wijk- en buurtstatistieken[23] of surveydata (bv. het LISS panel[24]). In dat geval moeten de databronnen eerst aan elkaar gekoppeld worden. Een koppelingsvariabele is dan van belang, zoals bijvoorbeeld een BSN-nummer als het onderzoek op individueel niveau gaat, of een postcode als het op PC6 niveau is.

Indien er geen koppelingsvariabele bestaat, zou er een manier gevonden moeten worden om de databronnen toch aan elkaar te koppelen. Soms zijn geavanceerdere data science-technieken hiervoor nodig. Volgens velen valt het koppelen van verschillende databronnen daarom onder datavoorbewerking en is een niet triviaal proces.

5.2 Verkennende data-analyse

Als eerste stap nadat de data bemachtigd is, is het altijd aan te raden om een verkennende data-analyse uit te voeren. Het populaire term in het Engels hiervoor is Exploratory Data Analysis (EDA). Dit geeft een goed overzicht van de data en hiermee bepaal je in sommige gevallen wat de (primaire) vervolgstappen moeten zijn.

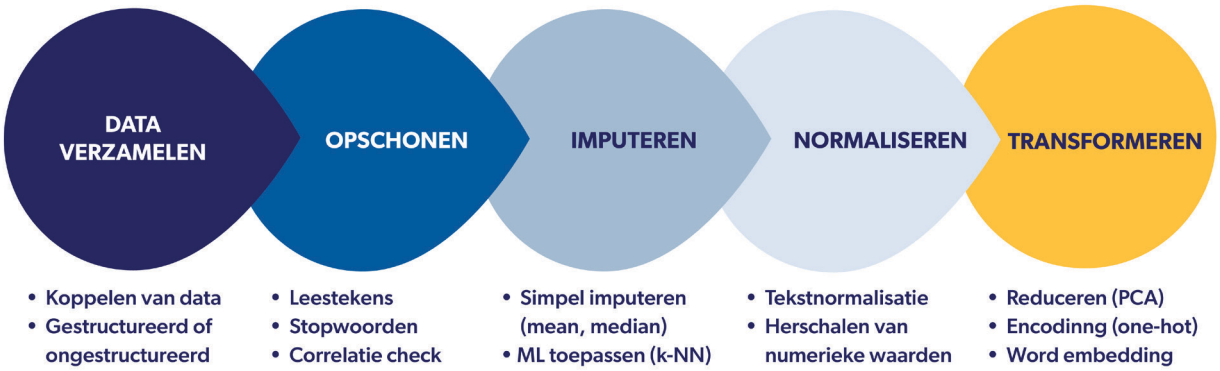
Tijdens EDA worden de individuele kenmerken verkend met behulp van statistieken en met grafieken en histogrammen. Wanneer de gegevens gevisualiseerd worden, worden de voor de hand liggende patronen direct zichtbaar. Een verkennende data-analyse leidt daarmee al snel tot een statistische representatie van de onderliggende data. Goede en duidelijke, soms statistische weergaven worden zelfs in wetenschappelijke artikelen gebruikt.
Vervolgens kan er verder worden gegaan met het verkennen van kenmerkrelaties (kenmerkpaar en kenmerkgroep) met behulp van scatter- en correlatieplots. Vragen kunnen worden gesteld zoals: wat is de betekenis van die patronen? Waarom zien we ze? Een verkennende analyse leidt al gauw tot het inzicht in hoe de patronen gebruikt kunnen worden om een goed ML-model te bouwen. Een direct voorbeeld hiervan is om sterk gecorreleerde variabelen te identificeren zodat deze uit de data verwijderd kunnen worden in de datavoorbewerkingstap. Op deze manier worden de potentiële 'ruisbronnen' vooraf gefilterd. Een EDA stap sluit hierdoor naadloos aan op de datavoorbewerkingsstap.

In het geval van tekstuele data is het niet zinvol om een EDA uit te voeren voordat de data opgeschoond wordt in de datavoorbewerkingsstap. Speciale (lees)tekens en stopwoorden moeten eerst worden verwijderd, de tekst moet worden genormaliseerd en zo mogelijk moeten spelfouten eerst worden verbeterd. Pas daarna kan er een EDA uitgevoerd worden.

5.3 Datavoorbewerking

Voordat modellen op de data losgelaten kunnen worden en nog voordat data-analyse kan plaatsvinden, is het belangrijk om ruwe data voor te bewerken. Ruwe data zijn vaak onvolledig, inconsistent, eenzijdig en bevatten waarschijnlijk veel fouten. Verschillende fasen zijn er nodig om de data gereed te maken voor gebruik. Afhankelijk van hoe schoon de data zijn, kost datavoorbewerking vaak 50% tot 80% van de tijd over het gehele proces. Onder datavoorbewerking valt het verzamelen, opschonen, imputeren, normaliseren en transformeren van data, zie Figuur 11.

[21] https://www.cbs.nl/nl-nl/onze-diensten/maatwerk-en-microdata
[22] https://zakelijk.kadaster.nl/bag-producten
[23] https://www.cbs.nl/nl-nl/dossier/nederland-regionaal/wijk-en-buurtstatistieken
[24] LISS = Longitudinal Internet studies for the Social Sciences, https://www.lissdata.nl/.



Figuur 11: Stappen van datavoorbewerking met voorbeelden van typische toepassingen.

Voorbeeld datavoorbewerking bij tekstanalyse

Bij het opschonen van tekstuele data wordt niet-relevante tekst verwijderd, zoals speciale leestekens (o.a. punten, komma's, aanhalingstekens) en stopwoorden. Stopwoorden zijn woorden die vaak voorkomen in de taal en geen informatie bevatten, zoals 'de' en 'en'. Maar stopwoorden zijn ook woorden die vaak voorkomen in betreffende teksten zoals 'als' of 'a.u.b.' Online zijn er vaak (taalafhankelijke) lijsten beschikbaar die je kunt downloaden om stopwoorden te filteren. Maar ook sommige libraries hebben stopwoordenlijsten, bijvoorbeeld NLTK (zie https://www.nltk.org/).

Vervolgens wordt de tekst genormaliseerd. Normalisatie van tekstdata houdt het omzetten van woorden naar gestandaardiseerde vorm in, zoals informele afkortingen (blk naar bulk, abt naar about, €200 naar tweehonderd euro). Tekstnormalisatie vereist kennis over welk type tekst moet worden genormaliseerd en hoe het daarna moet worden verwerkt; er is geen universele normalisatieprocedure.

Voordat analyses worden gedaan, worden woorden in een tekst vaak gerepresenteerd als vectoren. Dit gebeurt bij het transformeren van de data (van tekst naar numeriek). Deze vectoren kunnen in verdere analyses (vooral bij ML) gebruikt worden.

De gehele voorbewerkingsslag wordt idealiter dusdanig opgezet dat het later herhaald kan worden voor nieuwe data en kan worden ingebouwd in een Application Interface (API).

5.3.1 Feature engineering

Feature engineering, ofwel het genereren van nieuwe, afhankelijke variabelen van bestaande variabelen is een tussenstap tussen datavoorbewerking en modellering. Feature engineering is het proces waarbij domeinkennis wordt gebruikt om features (variabelen) uit ruwe data te extraheren via dataminingtechnieken. Deze variabelen kunnen worden ingezet om de prestaties van ML-algoritmen te verbeteren. Doorgaans rekent men feature engineering ook onder datavoorbewerking.

Feature engineering is vaak niet nodig als men gebruik maakt van neurale netwerken. Neurale netwerken maken automatisch niet-lineaire combinaties van alle inputvariabelen, soms vele miljoenen combinaties, en bereiken zo een hele hoge nauwkeurigheid. Voor alle andere typen modellen kan feature engineering wel veel waarde toevoegen.

Voorbeelden van feature engineering

De meeste ML-modellen nemen de variabelen aan zoals ze ook ingevoerd zijn, terwijl combinaties van variabelen ook interessant kunnen zijn. Bijvoorbeeld bij een onderzoek naar het voorspellen van kindermishandeling zijn de variabelen lengte en gewicht van kinderen beschikbaar en blijken achteraf gezien belangrijke indicatoren te zijn. De achterliggende factor is echter de Body Mass Index (BMI). Wanneer de BMI vooraf aangemaakt en in het model ingevoerd wordt, blijkt dat de BMI nog belangrijker te zijn dan lengte en gewicht afzonderlijk. Het voorspellende model wordt hierdoor krachtiger.

Een ander voorbeeld dat gegeven kan worden is bij het opsporen van criminaliteit en ondermijning op bedrijventerreinen. Het bedrijvenregister registreert jaarlijks het aantal (actieve en inactieve) bedrijven op een bedrijventerrein. Een variabele als het aantal bedrijven op een bedrijventerrein heeft weinig impact op een voorspellend model. Als er een variabele van het verschil in het aantal bedrijven ten opzichte van de voorgaande jaar aangemaakt wordt, bijvoorbeeld het aantal in 2019 minus het aantal in 2018, blijkt dat de groei of krimp van een bedrijventerrein wel een voorspellende waarde heeft.

Ook kunnen minder goed gevulde variabelen gecombineerd worden tot een enkele goed gevulde variabele. Stel er is een jaarlijkse registratie van het aantal bijstandsaanvragen per buurt. Een onderzoek op buurtniveau laat al snel zien dat de jaarlijkse variabelen veel lege cellen of datapunten (missings) kent, en uiteindelijk niet gebruikt worden. Een aggregatie op meerdere jaren kan daarentegen wel goed gebruikt worden.

Uit tekstdata kunnen ook features gemaakt worden om te gebruiken in ML-modellen. Van simpele features zoals het aantal woorden in een notitie of een e-mail tot numerieke vectoren van woorden met behoud van context, een fiets en een brommer hebben dus vergelijkbare vectoren. Een tegenwoordig populair algoritme is de geavanceerde techniek BERT (Bidirectional Encoder Representations from Transformers, Devlin et al. 2018[25]).

5.4 Imputatie

Imputatie is het proces waarbij ontbrekende gegevens ("missings") worden aangevuld door vervangende, relevante gegevens op basis van beschikbare informatie. Het vervangen van lege cellen wordt in data science alom toegepast. Het vervangen van ontbrekende waarden is bijna altijd beter dan het compleet weglaten van deze datapunten, want een ontbrekende datapunt leidt ertoe dat vaak ook andere (waardevolle) datapunten uit je data moeten worden verwijderd. Imputeren is goed te doen voor numerieke data, terwijl het minder effectief is voor tekstuele data.

Ontbrekende gegevens kunnen drie belangrijke problemen veroorzaken: ze kunnen een vertekend beeld van de data geven, de verwerking en analyse van de gegevens moeilijker maken, en de efficiëntie van de analyse of voorspelling verminderen. Om deze redenen is imputatie een belangrijk middel waarmee de analyses geoptimaliseerd kunnen worden en waarmee er een juist beeld van de data kan worden weergegeven.

De meeste statistische en data science pakketten gooien standaard de ontbrekende waarden weg, wat de resultaten kan beïnvloeden. Echter, deze pakketten hebben ook vaak de mogelijkheid om imputatie toe te passen. De keuze is vaak om in te zien wanneer imputatie gedaan moet worden en welke methode het beste gebruikt kan worden.

[25] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018) Bert: Pretraining of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Kiezen om data weg te gooien (marginalization)

Wanneer kies je ervoor om datapunten weg te gooien in plaats van te imputeren? We kunnen daarop ingaan door te kijken hoeveel datapunten er missen. Indien er te veel datapunten missen in een variabele of een observatie, wordt het moeilijker om een datapunt realistisch te imputeren. Stel dat een dataset tien variabelen heeft met één daarvan "gewicht" en dat deze voor 70% niet ingevuld/leeg gelaten is. In dit geval zou een imputatie van het gemiddelde of mediaan gewicht, of een andere directe methodologie, niet leiden tot een verbetering van de analyses. In dit geval is het soms beter om deze variabele in zijn geheel weg te gooien. Hetzelfde geldt voor een observatie.

Soms kan een variabele echter toch nog voorspeld worden aan de hand van geavanceerde technieken. In het geval van het voorbeeld van gewicht, als men ook variabelen heeft zoals leeftijd en geslacht of de BMI, kan het gewicht via een voorspellende of clusteringalgoritme alsnog redelijk goed afgeleid en geïmputeerd worden. Daarentegen, als gewicht een compleet onafhankelijke variabele is, bijvoorbeeld het gewicht van een F1-courreur terwijl alle andere variabelen over de auto gaan, dan helpt een voorspelling of clustering ook niet meer. Domeinkennis is hierbij essentieel.

Er is dan ook geen simpele vuistregel die helpt bij het beslissen wanneer een variabele weggegooid kan worden. Sommigen hanteren een 50%-regel, anderen laten variabelen al weg bij 5% missings. Zo'n beslissing hangt af van verschillende factoren, zoals hoeveel data er beschikbaar is; als een variabele relatief gezien veel missings bevat, is de keuze sneller gemaakt om deze compleet weg te laten als er duizenden andere variabelen zijn.

Kiezen om datapunten leeg te laten

Je kan ervoor kiezen om de lege datapunten te laten zoals die zijn. Veel clusteringpakketten hebben bijvoorbeeld een intern standaard wat te doen bij missings. Als ervoor gekozen wordt om de datapunten leeg te laten, is het van groot belang om precies te weten wat de datapakketten ermee doen.

Een clusteringalgoritme kan ervoor kiezen om de lege datapunten als een apart categorie te beschouwen. Dit kan ertoe leiden dat missings een bepalend factor kunnen zijn in de opbouw van de clusters, met name als dit een grote groep is. Als een categorische variabele missings kent, en deze missings onder één groep gebracht kunnen worden, bijvoorbeeld inkomen: laag, gemiddeld, hoog, en onbekend (=missing), kan het juist een bewuste keuze zijn om missings toe te laten. Het is echter altijd aan te raden om zelfs in dat geval de missings te imputeren door een aparte, zelfgekozen categorie om fouten te vermijden; het algoritme kan immers een ander keuze maken voor missings dan je denkt.

Kiezen om data te imputeren

Als je ervoor kiest om data te imputeren kan je twee richtingen op:

- Je kiest ervoor om simpele imputaties te doen.
- Je kiest ervoor om geavanceerdere of zelfs ML-technieken toe te passen.

Onder simpele technieken vallen de methodieken zoals de gemiddelde of mediaan waarde kiezen van de variabele. Deze zijn meteen ook de meest toegepaste manieren. Andere imputaties zijn bijvoorbeeld de vaakst voorkomende (waarde of tekst), het kopiëren van de voorgaande of de eerstvolgende, of het vervangen van een datapunt door een random of een nieuwe observatie.

De wat geavanceerdere imputatietechnieken zijn regressies, inter- en extrapolaties, clusteringalgoritmes en voorspellingen via Machine of zelfs Deep Learning. Soms is het ook handig als je meerdere imputatiemethoden kiest voor je data. Voor meer informatie zie bijvoorbeeld Wagstaff, 2004[26], Bertsimas et al., 2018[27] en andere online bronnen[28,29].

[26] Wagstaff, K. (2004) Clustering with Missing Values: No Imputation Required. In: Banks, D., McMorris, F.R., Arabie, P., en Gaul, W. (eds) Classification, Clustering, and Data Mining Applications. Studies in Classification, Data Analysis, and Knowledge Organisation. Springer, Berlin, Heidelberg

[27] Bertsimas, D., Pawlowski, C., en Zhuo, Y. (2018), From predictive methodsto missing data imputation: An optimization approach.Journal of MachineLearning Research, Journal of Machine Learning Research 18, 1-39.

[28] https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779

[29] https://www.theanalysisfactor.com/seven-ways-to-make-up-data-common-methods-to-imputing-missing-data/

Geïmputeerde waarden wegen even zwaar als waargenomen gegevens, maar ze zijn slechts zo goed als de aannames die zijn gebruikt om ze te creëren. Het is daarom van groot belang om deze stap zo goed mogelijk te behandelen door bewuste keuzes te maken (ook al is de keuze om geen imputatie te doen). Zodra alle ontbrekende waarden zijn toegerekend, kan de dataset worden geanalyseerd met standaardtechnieken voor volledige data.

5.5. Model trainen

5.5.1 Besluit 1: Transparantie of nauwkeurigheid

Een van de belangrijkste vragen die bij data science naar voren komen is: Hoe kies je je model? Vaak niet, maar soms is een eenduidig antwoord hierop geven mogelijk. We focussen daarom op het beslissingsproces dat ertoe leidt om uit te komen bij het meest passende model. Maar eerder is het zo dat je sommige andere modellen elimineert dan dat je het ideale model vindt.

De eerste vraag die men vaak stelt is of transparantie belangrijker is of juist de nauwkeurigheid van de uitkomsten, zoals het voorspellend vermogen of de accuraatheid van patroonherkenning. De reden is dat sommige modellen heel transparant en inzichtelijk zijn in de uitkomsten, zoals Logistic Regression of Decision Tree, terwijl andere modellen gebouwd zijn op nauwkeurigheid en precisie, zoals neurale netwerken.

5.5.2 Besluit 2: Model selecteren

Nadat de beslissing voor transparantie of nauwkeurigheid is gemaakt, komt de modelselectie (op een beperktere lijst). Welk model past het beste bij mijn onderzoek? Dit is vaak een van de moeilijkere vragen bij het oplossen van een probleem. Het is namelijk zo dat verschillende modellen beter geschikt zijn voor verschillende soorten data en verschillende soorten problemen.

We kunnen dit deels beantwoorden door de Scikit-Learn cheat-sheet te volgen. Deze is weergegeven in Figuur 12. Het schema is bedoeld om inzicht te verschaffen in welke modellen geschikt zijn in welke situaties. Het schema is geen uitputtende lijst en het kan zijn dat je niet exact vindt wat je zoekt. Toch kan het handig zijn om in de buurt te komen van de typen modellen die toegepast kunnen worden voor het soort onderzoek dat je uitvoert.

Hieronder een globaal stappenplan om in de buurt van een passend model uit te komen:

1. De eerste vuistregel is dat er indien minder dan 50 observaties in je data zijn (na opschonen), er simpelweg niet genoeg data zijn om AI-technieken toe te passen. Dan wordt er aangeraden om meer data te verzamelen.

2. Gaat het om het voorspellen van een klasse/ categorie?
a. Ja - kies uit classificatie of clustering
b. Nee - kies uit regressie of dimensionaliteitsreductie

3. Indien 2a-ja - beschik je over gelabelde data?
a. Ja - kies een supervised ML (classificatie) model
b. Nee - kies een unsupervised ML (clustering) model

4. Indien 2b-nee - wil je de data manipuleren?
a. Ja - dimensionaliteitsreductie is mogelijk je einddoel
b. Nee - stop… of ga terug naar stap 2

Zoals te zien is, kom je op deze manier in de juiste groep modellen terecht. Vervolgens is het de kunst om het beste model voor je taak uit te kiezen. Ook hier kun je een beetje op weg worden geholpen door kennis te hebben in de sterke punten en de beperkingen van verschillende modellen.

Zoals te zien is, kom je op deze manier in de juiste groep modellen terecht. Vervolgens is het de kunst om het beste model voor je taak uit te kiezen. Ook hier kun je een beetje op weg worden geholpen door kennis te hebben in de sterke punten en de beperkingen van verschillende modellen.

Enkele tips zijn:

- Indien transparantie van groot belang is, is het beter om neurale netwerken en dus DL te vermijden. Zelfs Support Vector Machine kan beter vermeden worden.
  DL en Support Vector Machine zijn black box-modellen. Ook al kunnen ze transparanter gemaakt worden met behulp van verschillende tools, toch zijn ze moeilijker interpreteerbaar dan transparante modellen. Ook vergt het de nodige ervaring en (diepte in) kennis van technieken zoals LIME.
- Voor Logistic Regression hanteert men vaak een één-op-tien vuistregel.
  Dit gaat over het aantal voorspellende parameters dat kan worden geschat op basis van de data bij het uitvoeren van de analyse, zodat het risico op overfitting laag blijft. De regel stelt dat voor elke tien gebeurtenissen één voorspellende variabele kan worden bestudeerd. Bijvoorbeeld als er van een dataset van 1000 observaties 150 tot de eerste klasse en 850 tot de tweede klasse behoren, kan er maximaal 15 (een tiende van de minderheidsklasse) voorspellende variabelen worden gebruikt. Indien er meer voorspellende variabelen ten opzichte van de minderheidsklasse observaties zijn, dan is een ander model aan te raden.
- Als er meerdere modellen zijn die passen bij de taak, is het handig en aan te raden om alle passende modellen te proberen.
  Tijdens de modelevaluatie en optimalisatiestappen kan dan het best presterende model uitgekozen worden. Dit maakt de uitslag uiteindelijk ook robuuster als meerdere modellen dezelfde voorspellingen afgeven, in de globale zin. In detail bekeken zullen er namelijk altijd wel enige verschillen tussen modellen zijn.

5.5.3 Data balancing

Data balancing is niet aan bod gekomen bij de voorbewerkingsstappen, omdat het in feite een onderdeel is van model building. Data balancing speelt een belangrijke rol bij data die sterk ongebalanceerd (dus de klassen niet evenredig verdeeld) zijn.

Wanneer de gelabelde categorieën (de klassen van de doelvariabelen) van een dataset niet gelijk zijn gedistribueerd, is een dataset ongebalanceerd. Dit zijn bijvoorbeeld data waarbij een klasse veel vaker of veel minder vaak voorkomen dan de andere klasse(n). Denk hierbij aan het positief testen van zeldzame ziektes bij patiënten of aan gevallen van fraude.

Ongebalanceerde data leiden snel tot een bias (voorkeur) of tot een overfitting van de meerderheidsklasse. De prestaties van een voorspellend model worden daarom sterk gereduceerd. Om een gebalanceerde dataset te verkrijgen kunnen de gelabelde data[30] van de meerderheidsklasse(n) willekeurig undersampled worden totdat er een balans is met de minderheidsklasse(n). Echter, net zoals bij het voorbeeld van zeldzame ziekten, is het vaak niet verstandig om data weg te gooien als er toch al weinig data beschikbaar is.

Ook kunnen oversampling technieken worden toegepast. Een voorbeeld hiervan is de Synthetic Minority Oversampling Technique (SMOTE, Chawla et al., 2002)[31]. SMOTE is een oversampling benadering waarbij de minderheidsklasse wordt verhoogd door het creëren van "synthetische" punten in de dataset. De minderheidsklasse wordt bemonsterd door middel van het nemen van een steekproef onder elk van de minderheidsklassen en het introduceren van synthetische datapunten langs de lijnsegmenten die aansluiten bij één of alle van de k-NN van de minderheidsklassen. De nearest neighbours zijn willekeurig gekozen, afhankelijk van de benodigde hoeveelheid oversampling.

[30] Gelabelde data is het deel van de data waarvan de uitslag bekend en dus getagd is; bijvoorbeeld bij het voorspellen van uithuisplaatsingen beschikt men over de data van kinderen die al uit huis zijn geplaatst. Het ML model kan hierop getraind worden.

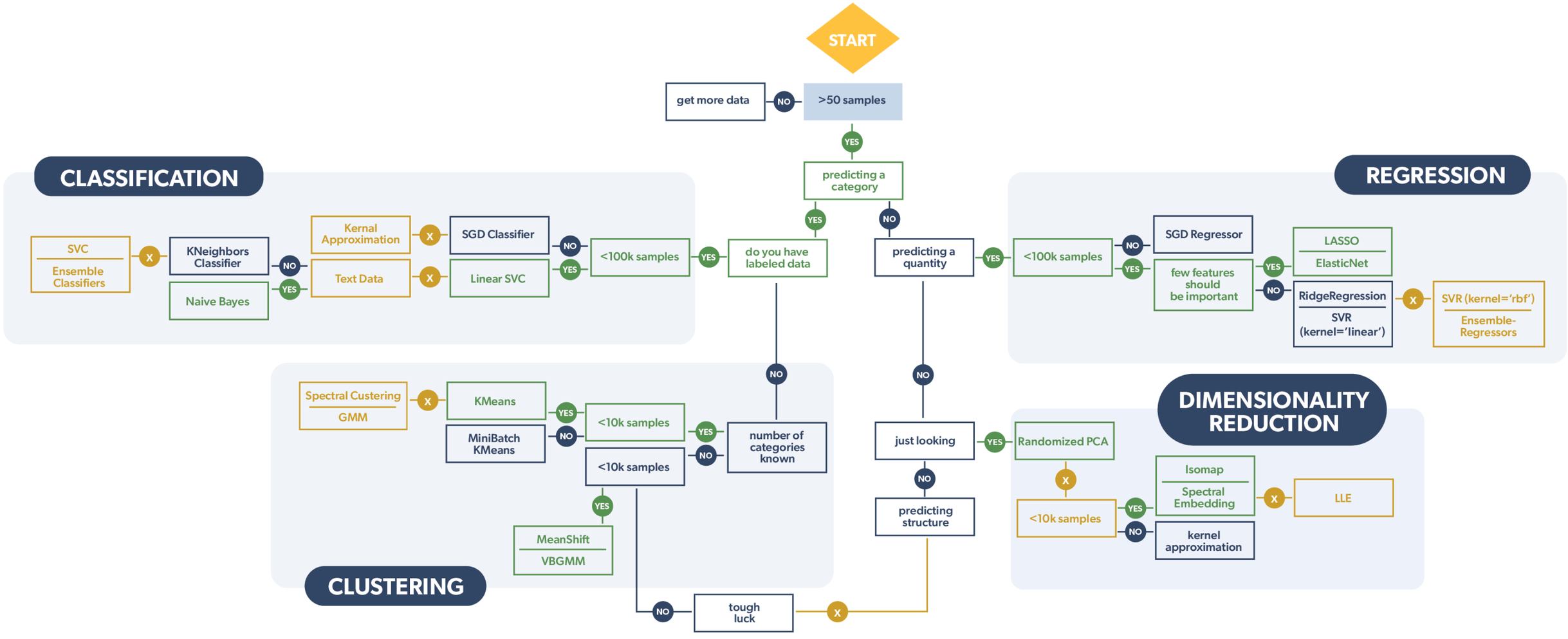
[31] Chawla, Bowyer, Hall, and Kegelmeyer (2002). "SMOTE: Synthetic Minority Over-Sampling Technique." Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

5.5.4 Overfitting

Een belangrijk argument bij het selecteren van een model is de prestatie van het model bij nieuwe data. Dat is dus de data die niet gebruikt is om het model te trainen. Een model kan goed scoren op de getrainde data, maar slecht op alle andere data. Dit heet overfitting (van de training). Overfitting leidt ertoe dat een model niet goed generaliseert.

Om dit te voorkomen wordt er doorgaans een (willekeurig) deel van de data achter de hand gehouden als een test-set, bijvoorbeeld 20%. Daarnaast kan nog een deel van de data als validatieset achter de hand worden gehouden om een model te optimaliseren, bijvoorbeeld ook 20%. Het nadeel hiervan is echter dat als er te veel data achter de hand worden gehouden, er weinig obesrvaties overblijven voor de training. En in geval van kleinere datasets (weinig observaties) lijdt dan de modeltraining eronder. Ook bij ongebalanceerde datasets is het niet erg wenselijk om data achter te houden.

Scikit-learn Algorithm Cheat-sheet



Figuur 12: Scikit-Learn cheat-sheet voor het kiezen van het juiste model. Image credit Scikit-Learn

5.6 Testen en evalueren

Zodra je model uitkomsten genereert, wil je weten hoe goed je model presteert. Om de prestatie van een model te beoordelen en om verschillende modellen te vergelijken, kan gebruik gemaakt worden van verschillende prestatiestatistieken. Er zijn meerdere manieren om modellen te evalueren en met elkaar te vergelijken. Voordat dit gedaan kan worden, moet de data eerst opgesplitst worden in een deel dat wordt gebruikt voor het trainen van het algoritme en in een deel voor modelevaluatie. Het opsplitsen van de data dient in feite al vóór het trainen van het model gedaan te worden.

5.6.1 Train-Test splitsing van de data

Supervised ML-algoritmen zoals classificatie en regressie vormen de meerderheid van ML-toepassingen. Deze modellen worden typisch getoetst op test data. Dit is een apart gezet stukje van de data dat niet wordt gebruikt tijdens het trainen van het model. De gehele data wordt dan opgesplitst in een training set en een test set, en vaak eigenlijk ook in een validatie set. Een dataset wordt bijvoorbeeld willekeurig opgesplitst in 80-20 (80% train en 20% test), of in 60-20-20 (60% train, 20% validatie en 20% test). Belangrijk is dat er bij de splitsing geen bias (systematische voorkeur) optreedt.

De validatie set wordt gebruikt om het model te optimaliseren. Het algoritme ziet normaal gesproken deze dataset ook meerdere keren tijdens het trainen van een model. Het biedt een testplatform voor het verfijnen van de parameters van een model en het selecteren van het best presterende model. De test set daarentegen wordt totaal niet gebruikt bij het trainen en het optimaliseren van een model. Deze wordt alleen ingezet om het model te evalueren aan het eind. Zodra je het model opnieuw traint mét de informatie die je hebt van de evaluatie van de uitkomsten, verliest de test zijn waarde en is in feite een validatie set geworden. Niet alle modellen hebben een validatie set nodig om ze te optimaliseren.

Afhankelijk van de grootte van de dataset (bij heel weinig data is het soms verstandiger om geen test set achter te houden), keuze die gemaakt wordt met betrekking tot de evaluatie van het model (wil je vergelijken met andere modellen of is de optimalisatie belangrijker?) en de evaluatietechnieken die je wenst toe te passen (wil je liever een cross-validation doen of toch een F1-soce berekenen?) moet er besloten worden hoe de data op te splitsen. Percentages van 80%-20% zijn gebruikelijk, maar in feite is de splitsing heel erg afhankelijk van de hoeveelheid beschikbare data.

Het is handig om te kijken wat anderen in een vergelijkbaar onderzoek hebben toegepast. Ook kan de train-test (of train-validatie) splitsing met terugwerkende kracht bepaald worden, door via een evaluatiecriterium te optimaliseren; je wilt bijvoorbeeld de positive predictive value (PPV[32]) maximaliseren en kiest ervoor om de train-test splitting als een van de hyperparameters te beschouwen.

5.6.2 Prestatiestatistieken

Modelprestatiestatistieken zijn vereist om modelprestaties te kwantificeren. De keuze van prestatiestatistieken hangt af van een bepaalde ML-taak (zoals onder andere classificatie, regressie, ranking, clustering, of topic modeling). Sommige statistieken, zoals Precision en Recall zijn nuttig voor meerdere taken. De meest toegepaste prestatiestatistieken zijn:

- Classificatie Nauwkeurigheid (Accuracy)
- Precisie (Precision)
- Gevoeligheid (Recall/Sensitivity)
- Confusion matrix
- Logaritmisch verlies (Log Loss)
- Oppervlakte onder curve (Area Under Curve, AUC)
- F-score (met name F1-score)

[32] https://en.wikipedia.org/wiki/Precision_and_recall

Accuracy (nauwkeurigheid) is het deel van de observaties dat correct voorspeld wordt.

$$Accuracy = \frac{aantal\ correct\ voorspelde\ observaties}{totaal\ aantal\ observaties} = \frac{TP + TN}{TP + TN + FP + FN}$$

Sensitivity/Recall (gevoeligheid) is het deel van de observaties dat voor de desbetreffende klasse correct was voorspeld. Dit kan per klasse of gemiddeld over alle klassen. Voor klasse 1 bij een dichotome variabele zou de formule bijvoorbeeld als volgt zijn:

$$Recall = \frac{aantal\ 'true\ positives'}{aantal\ werkelijke\ positives} = \frac{TP}{TP + FN}$$

Precision (precisie) is het deel van de voorspellingen dat voor de desbetreffende klasse correct geclassificeerd is. Dit kan per klasse of gemiddeld over alle klassen. Voor klasse 1 bij een dichotome variabele zou de formule bijvoorbeeld als volgt zijn:

$$Precision = \frac{aantal\ 'true\ positives'}{aantal\ voorspelde\ positives} = \frac{TP}{TP + FP}$$

De F1-score meet de prestaties van een model als het harmonische gemiddelde tussen precision en sensitivity (Provost & Fawcett, 2013)[33]. Het resultaat is een waarde tussen nul en één, waarbij een waarde van één een perfecte precisie en gevoeligheid zou impliceren. De F1-score wordt berekend door de volgende formule:

$$F1\ score = 2 * \frac{Precision\ *\ Sensitivity}{Precision\ +\ Sensitivity}$$

In de bovenstaande formules staan TP voor true positive, TN voor true negative, FP voor false positive en FN voor false negative. Deze termen worden duidelijker aan de hand van een confusion matrix.

Confusion matrix

In een confusion matrix wordt voor een binaire uitkomstvariabele (dat wil zeggen een uitkomstvariabele met twee categorieën) de voorspelde klasse vergeleken met de werkelijke klasse (zie Figuur 13).

		Voorspelde klasse	
		0	**1**
Werkelijke klasse	**0**	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

Figuur 13: Confusion matrix.

Bij aanwezigheid van meerdere klassen (meer dan twee) kan er per klasse gekeken worden naar de prestatie, bijvoorbeeld klasse 1 versus de rest. Het is gebruikelijk om vervolgens te kijken naar het gewogen gemiddelde. Het middelen kan op twee manieren gedaan worden: het micro- of macrogemiddelde.[34] Micro- en macrogemiddelden (voor welke maatstaf dan ook) zullen enigszins verschillende dingen berekenen en hun interpretatie verschilt. Het microgemiddelde wordt doorgaans iets vaker gebruikt, omdat het dichter bij de prestaties van het model voor de grotere klasse ligt.

Cross-validation

Kruisvalidatie, cross-validation in het Engels, is een procedure die wordt toegepast om ML-modellen te evalueren die te maken hebben met een beperkte dataset. De procedure heeft een enkele parameter 'k' die verwijst naar het aantal groepen waarin een gegeven data sample moet worden opgesplitst. De procedure wordt vaak k-fold cross-validation genoemd.

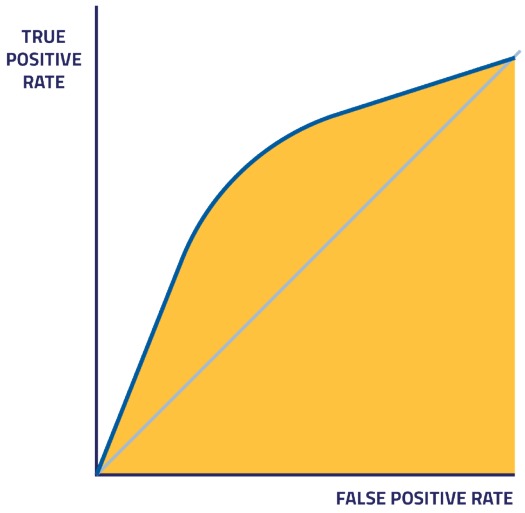
ROC-curve

Naast bovenstaande prestatiemaatstaven kan er gekeken worden naar de Receiver Operating Characteristic (ROC) curve (zie Figuur 14 voor een voorbeeld). De ROC-curve is een grafische weergave van de modelprestatie die gemaakt wordt door de true positive rate en de false positive rate te plotten (Provost & Fawcett, 2013)[26]. Hierin staat de diagonale grijze lijn voor de verwachting bij een volledig willekeurige classificatie. Hoe groter de afstand tussen de curve en de diagonaal richting de linker bovenhoek, oftewel hoe verder de curve van de diagonaal af staat, hoe beter de modelprestatie. Bij de best presterende modellen zal de curve dus door het gebied linksboven in de grafiek gaan. Een veelgebruikte maatstaf om de modelprestatie te beoordelen is de

[33] Provost & Fawcett (2013). Data Science and its Relationship to Big Data and Data-Driven Decision Making. https://doi.org/10.1089/big.2013.1508

[34] Een voorbeeld van een micro- vs. macrogemiddelde voor de maatstaf 'precision' is als volgt: Macrogemiddelde precision = 0.5 x (Precision1 + Precision2) Microgemiddelde precision = TP1 + TP2 / (TP1 + TP2 + FP1 + FP2)

"Area Under Curve" (AUC), het gebied onder de ROC-curve (in Figuur 14 het oranje/roze gekleurde gebied). De AUC ligt altijd tussen de nul en één, waar een AUC van 1 een perfecte voorspelling en een AUC van 0.5 een totaal willekeurige voorspelling betreft.



Figuur 14: Receiver Operating Characteristic (ROC) en de Area Under Curve (AUC).

5.7 Model optimalisatie en hyperparameter selectie

Bijna alle ML-modellen hebben enkele of meerdere parameters waarmee het model verfijnd kan worden. Deze parameters worden hyperparameters genoemd. Om precies te zijn is een hyperparameter een parameter waarvan de waarde wordt ingesteld voordat het ML-proces begint. Hyperparameters hebben vaak veel impact op de modelresultaten.

Standaard en typische waarden voor hyperparameters worden als default bij ML-modellen aangenomen. Indien modellen in een onderzoek vergeleken worden met een ander onderzoek, kunnen hyperparameters vastgelegd worden. In de meeste gevallen moeten hyperparameters echter handmatig ingesteld worden, waarmee de uitkomsten geoptimaliseerd kunnen worden (afgaande van een gekozen evaluatiecriterium).

De zoektocht naar de optimale hyperparameters kan ook gautomatiseerd worden. Hiervoor zijn verreweg de twee meest populaire manieren: Grid Search (of parameter sweep) en Random Search.

Grid search is simpelweg een uitputtende zoektocht door een handmatig gespecificeerde subset van de hyperparameterruimte van een leeralgoritme. Een grid search-algoritme moet worden geleid door een prestatiestatistiek, doorgaans gemeten door cross-validation op de training set of evaluatie op een uitgestelde validatie set. Grid search is computationeel gezien zeer duur als er veel hyperparameters zijn of het ML-model al een zwaar model is om mee te beginnen. Een vaak toegepaste alternatief is daarom ook een random search. Een random search doorzoekt niet alle mogelijke combinaties van de gegeven hyperparameters, maar kiest er willekeurig een aantal. Dit leidt al snel tot een significante verbetering van de modelprestaties.

6. Adviezen casus 'Bijstand'

6.1 Transparantie is belangrijk

Transparantie in AI-systemen is een van de belangrijkste aspecten in met name beleidsmatige, maatschappelijk relevante en juridische onderzoeken of toepassingen. Maar wat houdt transparantie precies in? Transparantie betekent niet alleen dat wat gedaan wordt helder van opzet en inzichtelijk is, bijvoorbeeld dat de code openbaar gemaakt wordt. Transparantie betekent ook traceability (traceerbaarheid), explainability (uitlegbaarheid) en communication (communicatie).

Om uitleg hieraan te geven heeft de Europese High-Level Expert Group on Artificial Intelligence (AI HLEG) een "Ethics Guidelines for Trustworthy AI" opgesteld, waarin transparantie ook als een van de kernonderdelen aan bod komt.[35] De eis voor transparante modellen is nauw verbonden met "the principle of explicability" (wat in feite neerkomt op traceability, explainability en communication) en omvat het transparantie van elementen die relevant zijn voor een AI-systeem: de data, het systeem en de businessmodellen.

Voor de gemeente Den Haag is de keuze voor transparantie tweeledig. Enerzijds worden de modellen inzichtelijker en beter interpreteerbaar (interpretability), wat helpt bij een analyse van de resultaten. De redenering hoe een uitkomst tot stand is gekomen is even belangrijk als de uitkomst zelf. Anderzijds zijn de uitlegbaarheid (explainability) en de communicatie (communication) richting de mensen erg belangrijk. Hoe leg je mensen de beweegredenen uit? En als er een probleem is, wie is verantwoordelijk (accountability) of hoe traceer (traceability) en identificeer je het issue?

Om deze redenen weegt transparantie vele malen zwaarder dan een krachtiger model dat een (iets) hoger percentage van de mensen in de bijstand aan werk kan helpen.

Advies

Ons advies is om een transparant model te verkiezen boven een krachtiger model met meer voorspellend vermogen. Hierdoor kunnen we de populaire black box ML-modellen, zoals Neurale Netwerk en Support Vector Machine uitsluiten. Maar men moet niet te veel opofferen door meteen het meest transparante model te verkiezen boven elk ander model, zodat een voorspelling geen waarde meer heeft. Door tools in te zetten die minder transparante ML-modellen transparanter maken, kan er een goede balans gevonden worden tussen interpreteerbaarheid en voorspellend vermogen. Naast begrip en interpreteerbaarheid in de einduitkomsten moet bij transparantie ook gedacht worden aan de uitlegbaarheid en communicatie naar mensen in de bijstand toe.

[35] https://ec.europa.eu/futurium/en/ai-alliance-consultation.

6.2 Keuze classificatiemodel: Ga verder dan een Decision Tree

De gemeente Den Haag heeft een Decision Tree ingezet voor de classificatie van klantsegmenten om de doorslaggevende factoren te vinden. Via deze aanpak heeft de gemeente 2500 cliënten met soortgelijke kenmerken in het huidige bestand gevonden.

Decision Trees behoren tot de meest transparante AI-modellen die er zijn. Echter, het is van groot belang dat bij het kiezen van een transparant model een weloverwogen keuze gemaakt wordt zodat de eindresultaten niet onder de nadelen van het model lijden (dat wil zeggen: hoeveel kwaliteit offer je op voor iets meer transparantie). Decision Trees zijn bijvoorbeeld onstabiel. Door kleine veranderingen in de onderliggende data kunnen de uitkomsten veranderen. Een ensemble methode, zoals een Random Forest, maakt een Decision Tree veel stabieler door honderden of zelfs duizenden Decision Trees te draaien en de uitkomsten te middelen.

Ensemble methoden zijn minder transparant dan Decision Trees, maar kunnen nog steeds redelijk inzichtelijk zijn. De combinatie van een ensemble methode en een transparante methode kan uitkomst bieden (zie het voorbeeld hieronder). Hiermee heb je het beste van twee werelden. Ook kunnen speciaal ontworpen tools om minder transparante modellen inzichtelijk te maken ingezet worden. LIME[36] (Local Interpretable Model-agnostic Explanations) is een voorbeeld van zo'n tool.

Voorbeeld van een Decision Tree als interpretatietool

Stel, bij een onderzoek om bijstandsgerechtigden sneller aan werk te helpen komen de volgende drie variabelen als topindicatoren van een Random Forest naar voren: leeftijd, geslacht en opleidingsniveau. Het is echter uit het Random Forest-model niet duidelijk of opleidingsniveau hoog of laag een hoger kans aanduidt bij het sneller vinden van werk. Met domeinkennis is deze bij opleidingsniveau voor de hand liggend, maar bij geslacht of leeftijd misschien niet. Het draaien van een Decision Tree op de topindicatoren biedt dan uitkomst. De Decision Tree laat bijvoorbeeld zien dat jonge vrouwen het snelst aan werk worden geholpen. De volgorde van de variabelen bij de Decision Tree of zelfs de splitsing van de variabelen (zoals de grens die gevonden wordt bij leeftijd) is niet meer van belang, alleen de richting. Zo is een Decision Tree niet als een voorspellend model gebruikt, maar als een interpretatie van de uitkomsten van de Random Forest.

Advies

Ons advies is om Random Forest of Gradient Boosting te verkiezen boven een Decision Tree. Het meest ideale is om beide modellen te runnen en de uitkomsten met elkaar te vergelijken. In dat geval kunnen de globale uitkomsten die beide modellen aangeven met grotere zekerheid aangenomen worden, wat de eindresultaten veel betrouwbaarder maakt. De uitkomsten kunnen verder worden geïnterpreteerd door Decision Trees te maken op de indicatoren die de ensemble methoden geven. Hierdoor wordt een Decision Tree als een verkennende analyse methode toegepast in plaats van als een voorspellende. En zo heb je de kracht van beide methoden gebruikt.

[36] Ribeiro, Marco Tulio, Singh, Sameer,en Guestrin, Carlos (2016). "Why should I trust you?: Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM.

6.3 Keuze clustering: Gebruik van gemengde variabelen

Na het samenvoegen van verschillende databronnen, zijn er vaak zowel numerieke (continue) als categorische (discrete) variabelen in de dataset aanwezig. Sommige clusteringalgoritmen kunnen rechtstreeks met categorische data overweg, maar hebben vaak beperkingen. De meeste algoritmen voor ML kunnen daarentegen geen categorische variabelen aan. Deze vereisen dat alle invoer- en uitvoervariabelen numeriek, of minimaal van dezelfde type, zijn. Daarom moeten de categorische variabelen omgezet worden in numerieke waarden, of andersom.

Voor het segmenteren van klanten heeft de gemeente Den Haag het model K-means clustering ingezet. K-means is een populair clusteringalgoritme, maar heeft een belangrijke beperking. Het is alleen geschikt voor numerieke (let op: continue) waarden en het gebruikt standaard de Euclidische afstand voor de afstandsbepaling tussen datapunten. Zelfs al transformeer je de categorische variabelen tot numerieke waarden door ze te coderen, bijvoorbeeld one-hot encoding[37], K-means kan er niet goed mee overweg, omdat ze nog steeds geen continue variabelen zijn.

De reden daarvoor is dat ook al accepteert K-means de data als numeriek, de afstandsberekening zal sterk worden beïnvloed. Uit discrete waarden, zoals een nul of een één (bv. door middel van one-hot encoding), kunnen geen realistische afstanden berekend worden. Denk hierbij bijvoorbeeld aan de Euclidische afstandsfunctie met invoerwaarden van $\sqrt{(0^2+1^2+0^2+1^2+\ldots}$etc). De afstanden zullen dus ook discreet zijn. Doordat de afstanden niet meer te vergelijken zijn met de afstanden van continue variabelen, zullen de uitkomsten van K-means een sterk vertekend beeld geven richting een van de typen variabelen (numeriek of categorisch).

Er zijn andere clusteringtechnieken die wel goed overweg kunnen met categorische variabelen, bijvoorbeeld K-modes. Dit is een variant van K-means die een ander afstandsbepaling toepast die specifiek is ontworpen voor categorische variabelen.
Echter, bij gemengde variabelen kan ook K-modes niet toegepast worden, zonder transformaties te doen (bv. door numerieke waarden te transformeren naar categorisch). Voor gemengde variabelen is er de package K-prototypes beschikbaar. Deze methode herkent de verschillende typen variabelen en gebruikt voor elk type een ander afstandsbepaling.

Andere tools voor het clusteren van categorische of gemengde variabelen zijn Gower[38], FactoMineR[39], clustMixType[40], PCAmix[41] of je voert een multiple correspondence analysis (MCA) uit waarmee je principal component analysis (PCA) kan doen. Deze alternatieven zijn echter veel technischer en vereisen diepere kennis in clusteringtechnieken om ze in te zetten. We raden ze daarom in de eerste instantie niet aan.

[37] One-hot encoding is het proces waarbij categorische variabelen worden omgezet in numerieke (wel discrete) variabelen. Elk categorie wordt afzonderlijk omgezet. Een categorische variabele met drie categorieën krijgt dus drie niewe variabelen met een 1 of een 0 als inputwaarde.
[38] https://www.rdocumentation.org/packages/gower/versions/0.2.1. Zie ook: J.C. Gower. A general coefficient of similarity and some of its properties Biometrics, 27 (1971), pp. 857-871
[39] http://factominer.free.fr/. Zie ook: Lê, S., Josse, J. & Husson, F. (2008). FactoMineR: An R Package for Multivariate Analysis. Journal of Statistical Software. 25(1). pp. 1-18.
[40] https://www.rdocumentation.org/packages/clustMixType/versions/0.2-2
[41] https://www.rdocumentation.org/packages/PCAmixdata/versions/3.1/topics/PCAmix

Advies

Ons advies is om K-modes voor categorische, of indien beschikbaar, K-prototypes voor gemengde data te gebruiken om data te clusteren. Het normaliseren van de numerieke waarden naar een waarde tussen 0 en 1 (bv. minmax scaling of mean normalization) wordt ten zeerste aangeraden. Voor K-modes, moet men eerst de numerieke waarden transformeren naar categorische waarden door een grenswaarde te kiezen. Meer dan twee categorieën zijn goed mogelijk. Om diepgang aan het onderzoek te geven kunnen de clusteringuitkomsten voor een gemengde dataset vergeleken worden met het clusteren van de dataset waarin alleen de numerieke, dan wel de categorische variabelen zitten. Een eerste alternatief voor K-modes of K-prototypes voor de wat gevorderden is FactoMineR.

6.4 Interpretatie van de analyseresultaten

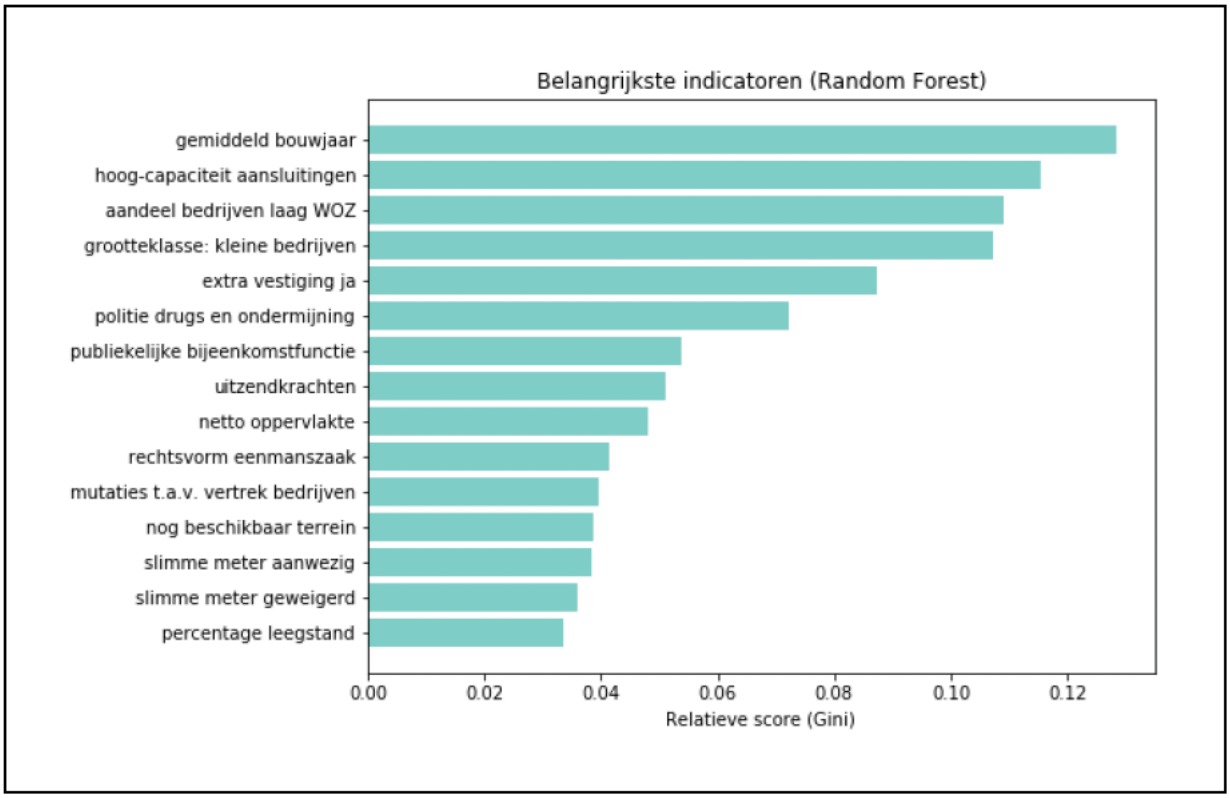
Indien er een Random Forest wordt toegepast op alle variabelen (features), is de uitkomst een gesorteerde lijst met de belangrijkste indicatoren. De uitkomsten kunnen visueel weergegeven worden, maar zijn minder inzichtelijk dan een Decision Tree. De lijst geeft echter wel duidelijk inzage in de variabelen die als belangrijkste indicatoren naar voren komen bij vele (honderden of duizenden) Decision Trees.

Om dit te relativeren, kan er gekeken worden naar de relatieve Gini-index.[42] De Gini-index wordt doorgaans gebruikt om de variabelen aan te duiden die het grootste verschil maken voor de voorspelling, zie bijvoorbeeld Figuur 15.

Een nadeel van Random Forest (of Gradient Boosting) is dat het uit de uitkomsten niet direct duidelijk wordt welke richting de variabele heeft, oftewel hoe de variabele precies van invloed is. Bijvoorbeeld, als leeftijd een belangrijke indicator is van inkomen, is het niet direct duidelijk of een hoge leeftijd of juist een lage leeftijd leidt tot een hoger inkomen. Domeinkennis kan hierbij helpen. Maar er kunnen ook additionele data science-technieken ingezet worden om de invloed te herleiden, we schetsen hieronder een mogelijke methode.

Om zicht te krijgen op de richting van de invloed van een bepaalde variabele, kan er een simpele techniek toegepast worden. Aanvullend op de Random Forest kan een Decision Tree gedraaid worden op de top-N indicatoren. Met behulp van de Decision Tree worden de richtingen die van invloed zijn op de variabelen nu wel duidelijk. Soms moeten er meerdere Decision Trees gedraaid worden (met aan andere random seed) om alle variabelen in beeld te kunnen krijgen. Tevens geven de Decision Trees ons meer inzicht in de belangrijkste combinaties van risicofactoren. Met dezelfde gedachte kan ook een Logistic Regression toegepast worden (in plaats van een Decision Tree). Logistic Regression geeft aan de hand van het teken (+ of -) van de coëfficiënten ook de richtingen die van invloed zijn op de variabelen zijn aan.

[42] Deze meet de ongelijkheid tussen waarden van een frequentieverdeling. Een Gini-index van nul geeft de perfecte gelijkheid weer, waarbij alle waarden hetzelfde zijn. Een Gini-index van 1 (of 100%) drukt de maximale ongelijkheid tussen waarden uit.



Figuur 15: Belangrijkste indicatoren van een Random Forest gesorteerd aan de hand van een Gini-index. Een simulatie van de resultaten van een voorbeeldonderzoek naar criminaliteit op bedrijventerreinen.

Het opnieuw draaien van een model met een subselectie van de topindicatoren maakt de uitslag van het model tevens robuuster, omdat de coëfficiënten dan ten opzichte van elkaar geoptimaliseerd worden. Het gebruik van een tweede Random Forest op de shortlist van indicatoren om de relatieve verhouding beter in beeld te brengen is aan te raden.

Naast het kijken naar de belangrijkste indicatoren, geeft het kijken naar de indicatoren die niet als belangrijk naar voren komen ook vaak extra inzichten.

Advies

Het runnen van een ensemble methode, zoals Random Forest, in combinatie met een transparantere methode, zoals een Decision Tree of een Logistic Regression, zorgt ervoor dat de output van een Random Forest interpreteerbaar is en beter visueel weergegeven kan worden. Ook worden de uitkomsten robuuster als een subset van topindicatoren opnieuw gemodelleerd, en zodoende de coëfficiënten ten opzichte van elkaar geoptimaliseerd worden. Naast het bekijken van de topindicatoren adviseren we ook om te kijken naar de indicatoren die juist niet als belangrijk naar voren komen, soms tegen de intuïtie in. Om uitleg daaraan te geven is domeinkennis nodig.

6.5 Meerwaarde van de adviezen

Vraag aan de gemeente Den Haag:

Wat is de meerwaarde van deze adviezen voor jullie businesscase?

Antwoord:

Voordeel is dat we voor onze voorspellingen een ander model in beeld hebben, dat minder transparant is maar krachtiger en scherper is in de berekening van de voorspelling. Uiteindelijk streef je niet alleen naar transparantie maar ook naar waarheidsgetrouwe voorspellingen. Deze instructie geeft tevens handreikingen om dit nieuwe model met behulp van een ander model transparanter te krijgen, zodat de uitkomst begrijpelijker en verklaarbaarder wordt. Door dit advies hebben we het beste van twee werelden: transparantie en krachtige voorspellingen.

Tevens geeft dit advies ons ook suggesties om tot scherpere klantsegmenten te komen. Door een ander model voor clustering toe te passen zijn we in staat om een variabele ook categorische waarden te geven, in plaats van alle variabelen een numerieke waarde toe te kennen zoals we voorheen deden. In onze dataset hebben wij veel meer categorische waarden dan numeriek. Categorische data levert meer op in de berekening van de afstanden tot clusters. Bijvoorbeeld als je voor de variabele geslacht de waarde 1 of 0 invoer dan berekent het model bijvoorbeeld 0.6 man, daarmee kom je niet verder dan overwegend man of vrouw. Door deze variabele categorisch in te vullen krijg je een veel scherpere uitkomst of geslacht überhaupt een patroon vormt in een betreffende groep of dat er andere kenmerken doorslaggevender zijn.

Daarbij moet je bij het K means model al van te voren aangeven hoeveel clustergroepen je wilt zien. Met de Elbow methode kun je kijken waar mogelijk een scherpe buiging zit bij het aantal clustergroepen. Als die veelal bij elkaar liggen en geen scherpe buiging laat zien, dan moet je alsnog zelf kiezen waar je de grens wil trekken. Bij de aanbevolen methode hoeft dit niet en laat het model zelf zien wat het optimaal aantal groepen zijn. Hierdoor laat je data echt het verhaal vertellen in plaats van je dat zelf moet gaan zoeken en experimenteren hoe de groepsindeling het scherpst, volledig en overzichtelijk is voor de business.