Machine Learning Pipeline Automation

Amar Kisoensingh

22 maart 2021

Voorwoord

Samenvatting

Summary

Afkortingen

PaaS Platform as a Service 18, 19

Begrippenlijst

cloud computing platform first 14, 19, 20
computer science is de studie van algoritmische processen, computationele machines en computatie zelf. 32
cross-validation is een betrouwbare manier om een schatting van de prestatie van een model te berekenen met behulp van training data. 36, 37

datamining 18

feature is een meetbaar attribuut waarmee een machine learning model traint. 32, 33
feature engineering is het maken van nieuwe features met bestaande features. 37
feature vector is 32, 33
fold is een subgroep van een dataset [**data-science-primer**]. 36

hyperparameter optimalisatie is het kiezen van een optimale set van hyperparameters [**hyperparameter-optimization-wikipedia**]. 37

machine learning is het ontwikkelen van algoritmes en technieken waarmee computers kunnen leren. 14, 18–20, 32
machine learning pipeline is een pijplijn waarin gedefiniëerd staat hoe data stroomt. first 14, 15, 18, 19

reinforcement learning 33

semi-supervised learning 33
supervised learning 32, 33

unsupervised learning 33

vendor lock-in is niet de mogelijkheid hebben om naar een andere dienst/provider te gaan die dezelfde service biedt. 18, 19

Lijst van figuren

Lijst van tabellen

Inhoudsopgave

1 Inleiding

1.1 Het bedrijf: NGTI

NGTI is een bedrijf gevestigd in Rotterdam dat apps en websites ontwerpt en ontwikkelt [**ngti-services**].

1.1.1 Klanten van NGTI

1.1.2 Tools die worden gebruikt

Om productief te zijn gebruikt NGTI een aantal tools en programma's om producten te maken en te communiceren met zowel collega's als klanten.

Slack

Interne communicatie gaat via Slack. Het programma faciliteren collega's om elkaar met een lage instap te benaderen en berichten die voor het hele bedrijf relevant zijn te versturen. Ook zijn er 'channels' beschikbaar over specifieke onderwerpen, zoals: *#dev*, *#ios* en *#test-automation*.

Google Workspace

Met Google Workspace kunnen bestanden en documenten gemaakt, opgeslagen en gedeeld worden. Omdat dit via een browser kan, hoeven werknemers geen software te installeren. NGTI gebruikt het ook om collaboratief en parallel te werken aan hetzelfde document.

Zoom

Voorheen werdt Zoom alleen gebruikt om te videobellen met collega's en interviewees. In de tijd van het pandemie is Zoom echter een belangrijke speler geworden om effectief samen te werken. Meetings zoals introducties van nieuwe collega's of demo's van producten worden online gehouden.

1.2 Opdracht

NGTI heeft voorzien dat ze haar applicaties 'slimmer' moet maken door machine learning in te zetten. Niet alleen zorgt dit voor een betere gebruikerservaring, maar geeft NGTI ook een voorsprong op haar concurrenten.

Om machine learning toe te passen is het raadzaam om een machine learning pipeline op te zetten. Een pipeline is een gestructureerde werkwijze om een model te trainen. Het opzetten van de pipeline en een competente model trainen is tijdrovend en vereist kennis in het domein. Vaak worden modellen getraind op een cloud computing platform zoals Azure, AWS of Google Cloud. Het probleem met platformen zoals deze is dat het ontzettend lastig is om te wisselen van platform en er is veel kennis vereist om een infrastructuur op te zetten.

De opdracht bestaat uit twee onderdelen:

1. onderzoek naar het automatiseren van het opzetten van een machine learning pipeline om het laagdrempelig en minder tijdrovend te maken.
2. onderzoek naar het maken van een platform agnostische oplossing

Hierbij zal een PoC gemaakt worden om aan te tonen of het haalbaar is. Een diepere duik in het probleem is te vinden in hoofdstuk 2.

1.3 Leeswijzer

2 Probleemanalyse

2.1 Probleemdefinitie

Zoals beschreven in paragraaf 1.2 is NGTI genoodzaakt om machine learning in te zetten om haar applicaties 'slimmer' te maken. Hier zijn een aantal redenen voor, namelijk:

1. gebruikerservaring verbeteren
2. voorsprong hebben op concurrenten

Het 'slimmer' maken van applicaties kan op verschillende manieren, maar met machine learning kan een platform gebouwd worden waar naar elke richting op gegaan kan worden. Om machine learning te implementeren in haar applicaties loopt NGTI tegen een aantal obstakels op, namelijk: expertise vereist in het machine learning pipeline domein, tijd om een pipeline op te zetten en vendor lock-in.

2.1.1 Expertise Machine Learning

Machine learning is geen triviaal onderwerp. Om een model te trainen is kennis nodig van verschillende domeinen: datamining, software engineering en statistieken. In een multidisciplinair team is het voor één teamlid niet nodig om alle domeinen te beheersen.

Doordat er voorkennis nodig is om een model te trainen, is het vaak te hoogdrempelig om te beginnen voor developers. De expertise is daar bovenop niet in een korte tijd te vergaren.

2.1.2 Opzetting pipeline

Er bestaan verschillende manieren om een model te trainen. Een machine learning pipeline opzetten is daar een van. In een pipeline wordt voor het trainen van het model de data voorbereid. Het opzetten van een pipeline kost tijd. Op zichzelf niet zo zeer veel tijd, maar als er veel wordt geëxperimenteerd met het trainen van modellen kan de tijd opstapelen. Ook zijn de stappen in een pipeline over het algemeen hetzelfde, ongeacht wat voor model je traint. Hierdoor worden taken vaak herhaalt tussen het opzetten van verschillende pipelines.

2.1.3 Vendor lock-in

Er bestaan een aantal diensten, zogenoemde Platform as a Service (PaaS), waarbij je een pipeline kan opzetten. Een van de problemen met een PaaS is vendor lock-in. Dit betekent dat, als er eenmaal een pipeline is opgezet, de overdraagbaarheid van de pipeline naar een andere PaaS vrijwel onmogelijk is. Ook zijn de opties en mogelijkheden om uit te breiden in de toekomst gelimiteerd.

2.2 Doelstelling

De doelstelling is om een systeem te ontwikkelen waarbij developers met weinig tot geen kennis een model kunnen trainen, onderdelen van de pipeline geautomatiseerd zijn en platform agnostisch is.

Het trainen van een model is een iteratief proces omdat de data waarmee het model getraind is verouderd waardoor het model niet meer optimaal presteert. Om een nieuw model te trainen en consistent te blijven met hoe een model getraind wordt kan een pipeline opgezet worden. Een pipeline is dus herbruikbaar.

2.3 Bestaand oplossingen

Er bestaan een aantal oplossingen voor vrijwel alle problemen dat NGTI ondervindt. Elk oplossing is een PaaS van een derde partij waarbij vendor lock-in inherent is. Dit maakt ze ongeschikt maar betekent echter niet dat ze nutteloos zijn. Er kan namelijk gekeken worden hoe een pipeline wordt opgezet en daar vervolgens (gedeeltelijk) het systeem op baseren.

De oplossingen kunnen gecategoriseerd worden in twee groepen:

1. Machine learning pipeline specifieke services
2. Cloud computing platformen

2.3.1 Machine learning pipeline specifieke service

Bedrijven zoals Algorithmia [1] en Valohai [2] bieden alleen diensten om pipelines op te zetten. Ze zorgen voor het databehoud dat door de gebruiker wordt geüpload en het trainen van het model. Verder kan er toezicht gehouden worden op de kosten, beschikbaarheid en prestatie van het model.

Valohai heeft documentatie een aantal blog posts die bij het ontwerpen van het systeem relevant zouden kunnen zijn.

2.3.2 Cloud computing platformen

De drie grote cloud computing platformen Amazon, Azure en Google hebben meer te bieden dan alleen een pipeline opzetten, zoals het hosten van een website, database of virtuele server. De cloud computing platformen hebben hetzelfde probleem als de machine learning pipeline specifieke services; vendor lock-in is onvermijdelijk. Wat wel een mogelijkheid zou kunnen zijn is dat de andere services van de cloud computing platformen gebruikt kunnen worden als onderdeel van het systeem.

2.4 Hoofd- en deelvragen

Uitgaand van de drie focuspunten in paragraaf 2.1 kan de hoofdvraag als volgt worden geformuleerd:

Hoe kan een machine learning pipeline worden geautomatiseerd onafhankelijk van de onderliggende cloud computing platform of lokale framework?

De hoofdvraag kan worden onderbouwd met vier deelvragen. Om te beginnen is het verstanding om te onderzoeken hoe een machine learning model wordt getraind:

Welke stappen moeten worden ondernomen om een machine learning-model te trainen?

Vervolgens kan er op de deelvraag voortborduurt worden om het opzetten van een pipeline in kaart te brengen:

Hoe wordt een machine learning-pipeline opgezet?

Verder zijn er verschillende cloud computing platformen en lokale frameworks waarmee machine learning modellen getraind kunnen worden. Doordat het systeem platform agnostisch moet zijn is het van belang om verschillende platformen en frameworks te onderzoeken:

Wat zijn de verschillende en overeenkomsten tussen cloud computing platforms en lokale frameworks waarmee machine learning-modellen kunnen worden getraind?

Ten slotte wordt een PoC gemaakt om te laten zien of het probleem oplosbaar is. Hiervoor is een doordachte voorbereiden onmisbaar:

Hoe ziet de architecturale blauwdruk van een applicatie, waarin een machine learning pipeline kan worden opgezet en die platform-onafhankelijk is, eruit?

3 Onderzoeksmethoden

D1: Welke stappen moeten worden ondernomen om een machine learning-model te trainen?	
Doel	Onderzoek naar wat machine learning is en hoe een model getraind kan worden. Een visualisatie van hoe een model getraind wordt, een PoC waarbij een model getraind wordt en een conclusie en advies met vooral waarop gelet moet worden bij het trainen.
Methode	Kwalitatief
Validatie	Er is uitleg gegeven wat machine learning is en hoe een model getraind kan worden op een simpele manier waarbij de visualisatie de uitleg compleet maakt. Met het conclusie en advies kan iemand die vrijwel geen kennis heeft aan de slag met het trainen van een model.

Tabel 3.1: Deelvraag 1

D2: Hoe wordt een machine learning pipeline opgezet?	
Doel	Onderzoek naar wat een machine learning pipeline is, waarom het gebruikt wordt en welke onderdelen van de pipeline geautomatiseerd kunnen worden. PoC waarbij een pipeline met het model van de vorige deelvraag wordt opgezet. Conclusie en advies.
Methode	Kwalitatief
Validatie	Het is duidelijk wat een pipeline is en waarom het handig/noodzakelijk is om te gebruiken. In de conclusie en advies is het onder andere duidelijk of onderdelen geautomatiseerd kunnen worden.

Tabel 3.2: Deelvraag 2

D3: Wat zijn de verschillen en overeenkomsten tussen cloud computing platforms en lokale frameworks waarmee machine learning-modellen kunnen worden getraind?	
Doel	De grote cloud computing platformen en lokale frameworks in kaart brengen. De features, voor- en nadelen worden met elkaar vergeleken. Waar van toepassing wordt een kosten berekening gedaan. Conclusie en advies bevat onder andere welke platform en/of framework het beste is om mee te beginnen.
Methode	Kwalitatief
Validatie	Een overzichtelijke vergelijking tussen de cloud computing platformen en lokale frameworks is gemaakt. Uit de conclusie en advies moet blijken welke platform en/of framework het beste is om mee te beginnen.

Tabel 3.3: Deelvraag 3

D4: Hoe ziet de architecturale blauwdruk van een applicatie, waarin een machine learning pipeline kan worden opgezet en die platform-onafhankelijk is, eruit?	
Doel	Er wordt research gedaan naar hoe een pipeline geautomatiseerd opgezet kan worden en naar verschillende tools/frameworks. Technische designs worden gemaakt ter validatie en als blauwdruk.
Methode	Kwalitatief
Validatie	Uit de research en designs is het duidelijk hoe een systeem gebouwd kan worden.

Tabel 3.4: Deelvraag 4

H: Hoe kan een machine learning pipeline worden geautomatiseerd onafhankelijk van de onderliggende cloud computing platform of lokale framework?	
Doel	Uit de requirements wordt de scope bepaald. Er worden designs gemaakt om de interface te testen. Een PoC wordt gebouwd.
Methode	Kwalitatief
Validatie	De PoC is werkzaam en voldoet aan de requirements.

Tabel 3.5: Hoofdvraag

4 Requirementsanalyse

4.1 Stakeholders

4.2 Requirements

5 Hoe werkt Machine Learning?

Machine learning (ML) is onderdeel van het domein computer science. Binnen machine learning worden algoritmische modellen gebouwd om statistische problemen op te lossen. Dit wordt gedaan met behulp van een dataset dat verzameld is uit de echt wereld of gefabriceerd door mensen. Het model kan met behulp van de dataset een redelijk nauwkeurige voorspelling maken [3, p. 7].

De termen deep learning (DL) en artificial intelligence (AI) komt vaak voor als het over ML gaat. DL is een uitbreiding van neural networks, wat weer bestaat uit een set algoritmes en is grotendeels gemodelleerd naar de hersenen [4]. Niet alle termen en concepten van ML zijn relevant binnen DL maar valt wel onder dezelfde paraplu.

ML is onderdeel van het AI domein. Bij AI wordt niet alleen ML toegepast, maar ook concepten zoals beredeneren, plannen en vooruitdenken en het onthouden en terug refereren. Een voorbeeld hiervan is dat een ML model kan voorspellen wat het volgende woord in een zin kan zijn, maar een AI kan beredeneren waarom de zin gebouwd is zoals het is en hoe het past binnen de context van de alinea past [5].

5.1 Hoe leert een machine learning model?

Een machine learning model leert met behulp van een dataset om een voorspelling te maken. Een dataset kan duizenden feature vectors (voorbeelden) bevatten, waarbij elke vector een of meerdere feature (attribuut) heeft. In een voorbeeld waarbij een model moet kunnen voorspellen of een persoon kanker heeft kan een feature vector één persoon zijn en de features het gewicht, leeftijd, lengte en of de persoon kanker heeft [6].

Het model gaat langs alle feature vectors en probeert correlaties te vinden tussen de features. Op basis hiervan kan het model vervolgens een voorspelling doet met een feature vector dat het model nooit heeft gezien.

Er zijn een aantal manieren om een algoritme te laten trainen, gegroepeerd in vier vormen.

5.1.1 Supervised learning

Met supervised learning wordt een model getraind door middel van een **gelabeld** dataset. Dit betekent dat een feature vector gepaard gaat met de gewenste label.

Feature	**Label**
woorden, afzender, tijd van versturen	spam/geen spam

Tabel 5.1: Voorbeeld gelabeld dataset

Het model kan leren dat woorden zoals 'gratis' in e-mails die tussen 01:00 en 03:00 worden verstuurd vaak als spam gelabeld zijn. Er kan dan geconcludeerd worden dat als een e-mail met soortgelijke features als input word gegeven, dat het een spam e-mail is [6].

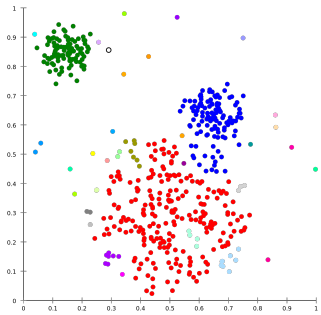
5.1.2 Unsupervised learning

Unsupervised learning lijkt grotendeels op supervised learning. Het verschil is dat de feature vectors in de dataset niet gepaard zijn met een label.

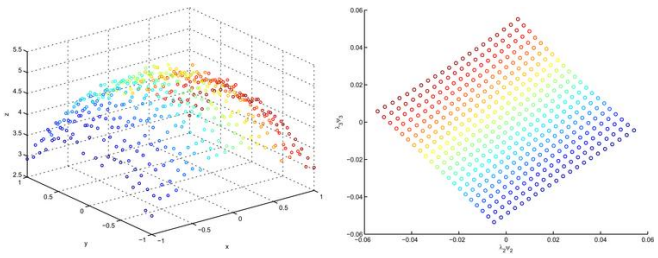
Feature	Label
woorden, afzender, tijd van versturen	?

Tabel 5.2: Voorbeeld niet gelabeld dataset

Het doel bij unsupervised learning is om de data op een bepaalde manier te transformeren om er iets waardevols uit te halen. Een voorbeeld hiervan is het *clusteren* van data. Hierbij wordt soortgelijke data gegroepeerd (Figuur 5.1). Een andere voorbeeld is *dimensionality reduction* (Figuur 5.2). Feature worden weggelaten om twee features tegen over elkaar te zetten om bijvoorbeeld uitschieters te vinden [3].



Figuur 5.1: Voorbeeld van *clustering*



Figuur 5.2: Voorbeeld van *dimensionality reduction*

5.1.3 Semi-supervised learning

Semi-supervised learning is een combinatie van supervised learning en unsupervised learning waarbij het doel dezelfde is als supervised learning. Het is gebruikelijk dat het merendeel van de dataset niet gelabelde data is [3].

5.1.4 Reinforcement learning

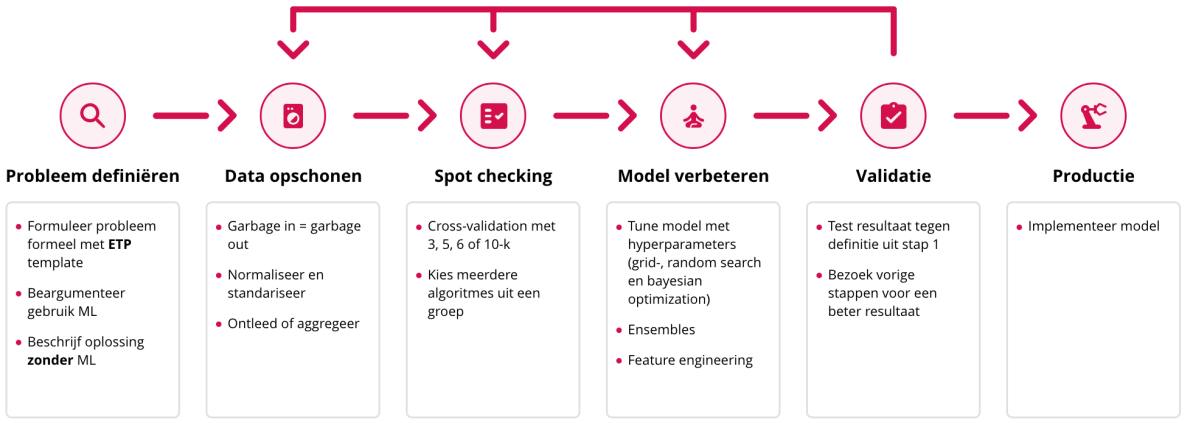
Reinforcement learning wordt ook wel 'survival learning' genoemd. Een analogie hiervoor is een organisme dat moet overleven in verschillende situaties door de juiste actie te verrichten. Acties kunnen een positief of een negatief effect hebben. Acties dat langdurig een positieve effect hebben moeten versterkt worden zodat het model zo veel mogelijk positieve acties verricht [7].

5.2 Stappenplan voor het trainen van een model

Er zijn een aantal stappen die altijd uitgevoerd moeten worden om een model te trainen, zoals: data opschonen, model trainen en een voorspelling maken met het model. Om deze

stappen kunnen extra stappen geplaatst worden om de accuraatheid van de voorspelling te maximaliseren. Brownlee [8] beschrijft een proces dat hij "Applied Machine Learning Process" noemt. Hierbij wordt vooraf het probleem in kaart gebracht, tussentijds gekeken welk algoritme de beste kans heeft op een hoge accuraatheid en na de selectie van het beste algoritme wordt het resultaat verbeterd als dit mogelijk is.

Het proces van Brownlee is misschien niet de beste en zeker niet de enige stappenplan. Zijn kwalificaties en toegankelijkheid van de bronnen heeft echter wel een rol gespeeld in de keuze van het leidende stappenplan en mijn eigen ervaring ondersteund de keuze.



Figuur 5.3: Stappenplan om een model te trainen

5.2.1 Het probleem definiëren

Om een ML model te trainen en het gebruik te verantwoorden, moet er gekeken worden naar **wat** het probleem is, **waarom** het een probleem is en de **hoe** het opgelost kan worden *zonder* het gebruik van ML. Als eerst kan het probleem informeel geformuleerd worden, bijvoorbeeld:

Ik wil het temperatuur voor de komende week voorspellen.

Informeel formuleren is wat makkelijker om mee te beginnen maar is optioneel. Nu kan het probleem formeel geformuleerd worden met behulp van Mitchells sjabloon [9]:

> A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.

Door de letters **E**, **T** en **P** in de zin in te vullen, kunnen we identificeren welke data er verzameld moet worden (**E**), wat het model moet kunnen doen/voorspellen (**T**) en hoe de accuraatheid gemeten kan worden (**P**).

Het informele probleem hierboven kan met behulp van Mitchells sjabloon formeel geformuleerd worden op de volgende manier:

- **T**: De temperatuur voor de komende 7 dagen voorspellen

- **E**: De gemeten temperatuur op dagen van voorgaande jaren
- **P**: Hoe dicht het voorspelde temperatuur bij de gemeten temperatuur lag

Vervolgens moet er gekeken worden waarom het probleem opgelost moet worden. De motivatie achter het gebruiken van ML kan het experimenteren zijn of omdat het van je werkgever moet. De voordelen van het gebruik van ML kan ook een reden zijn. ML toepassen kan bijvoorbeeld meer tijd vrijmaken voor andere taken.

Als laatste moet een plan geformuleerd worden waarin staat hoe het probleem opgelost kan worden. In het plan kan staan wat voor data er wordt verzameld en hoe technische tekeningen er uit zien. Hierbij kunnen prototypes en experimenten bij zitten. Het is belangrijk om dit *zonder* ML te doen om 'domain knowledge' , zoals Brownlee het noemt [10], naar boven te halen. Dit zijn praktische feiten zoals waar data is opgeslagen en wat voor kenmerken waardevol kunnen zijn.

5.2.2 Data prepareren

Bij het selecteren van de dataset is het niet verstandig alle beschikbare data te gebruiken. Meer is **niet** beter bij het trainen van een ML model. Een bekende gezegde binnen ML luid 'garbage in = garbage out' wat betekent dat de prestatie van het model afhangt van de kwaliteit van de dataset. Het is daarom goed om op papier te zetten wat voor data er beschikbaar is, wat er potentieel mist (en eventueel wat wel in de toekomst kan worden verzameld) en wat er niet gebruikt gaat worden.

Na de selectie moet er gekeken worden naar hoe de data gebruikt gaat worden. In deze stap wordt de data voorbewerkt. Hierbij wordt onder andere gekeken naar de formattering zoals typefouten of bestandstypes, het weghalen of repareren van missende data en het sampelen om te experimenten met een kleinere dataset zodat het snel gaat.

Als laatste stap kan de data getransformeerd worden. Het kan zijn dat deze stap weer bezocht wordt in een later stadium om de resultaten te verbeteren. Voor veel algoritmes is het handiger om met kleine waardes te werken, zoals tussen 0 en 1. Dit heet *data scaling*, of het schalen van data. Het normaliseren van data transformeert de waarde tussen 0 en 1. De formule hiervoor is:

$$y = \frac{(x - min)}{(max - min)}$$

De min is het kleinste getal in de dataset en max het grootste getal. De x is het getal wat genormaliseerd moet worden en y is het resultaat.

Een andere methode om te schalen is het standaardiseren. Hierbij wordt de verdeling van waardes herschaald zodat het gemiddelde 0 is en de standaardafwijking 1. De formule hiervoor is:

$$y = \frac{x - mean}{standard\_deviation}$$

Waarbij $mean$ en $standard\_deviation$ zijn:

$$mean = \frac{sum(x)}{count(x)} \qquad standard\_deviation = sqrt\left(\frac{sum((x - mean)^2)}{count(x)}\right)$$

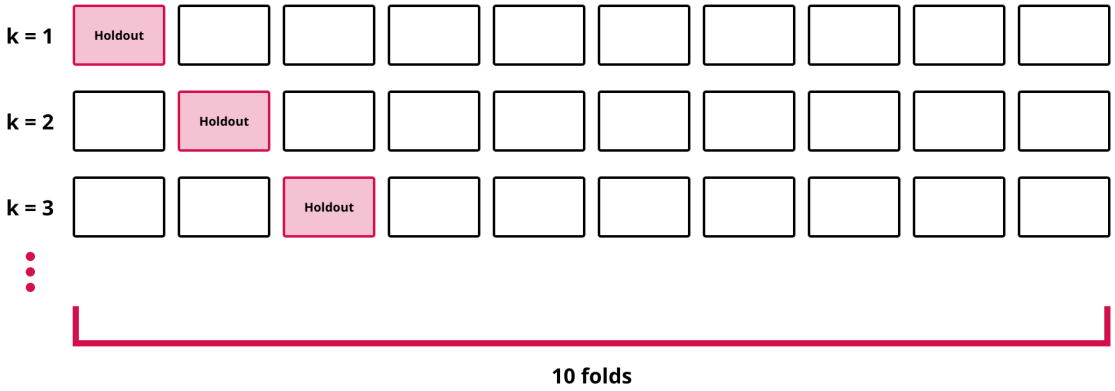
Het kan voorkomen dat features te complex zijn of los niet veel betekenen, maar samen meer betekenen. Deze features kunnen **ontleed** of **geaggregeerd** worden. Een voorbeeld van ontleden is dat de datum en tijd beschikbaar zijn, maar alleen de tijd relevant is. Een voorbeeld van aggregatie is vaak voorkomend data dat in één getal dezelfde waarde heeft, zoals de aantal logins van een gebruiker [11].

5.2.3 Spot check algoritmes

Er zijn ontzettend veel soorten algoritmes die elke effectief kunnen zijn voor een specifiek probleem. Een algoritme heeft daar bovenop nog een aantal parameters die aangepast kunnen worden. Het is daarom niet verstandig om één algoritme te kiezen en de prestatie daarvan te verbeteren. Een effectievere manier is om een 'test harness' op te zetten [12].

Het doel van de test harness is om snel en consistent algoritmes te testen tegen een representatie van het probleem. Met het resultaat uit de tests is het duidelijk met welke algoritme(s) het waard is om verder te gaan. Ook geeft het een idee hoe goed de dataset is en kan een algoritme accurater zijn dan eerst gedacht.

In de test harness wordt gebruik gemaakt van cross-validation. Bij cross-validation wordt de dataset verdeeld in delen, ook wel folds genoemd, waarvan 1 fold een *holdout* fold is. Het model wordt getraind met de folds en getest tegen de holdout fold. Dit proces wordt herhaald totdat elke fold een holdout fold is geweest. Aan het einde wordt de gemiddelde genomen van de prestatiewaarde van elke test. Het is gebruikelijk om een fold te nemen van 3, 5, 6 of 10. Een 10-k cross-validation methode is in Figuur 5.4 gevisualiseerd [12].



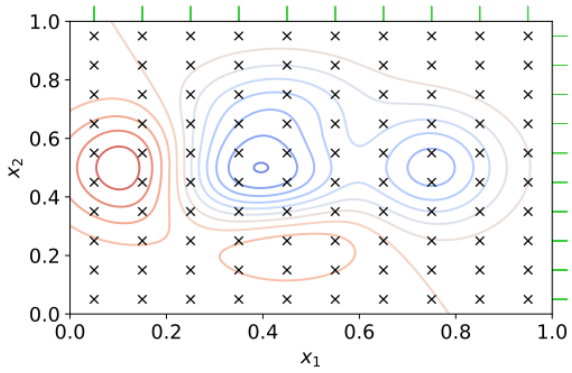
Figuur 5.4: Visualisatie van de 10-k cross-validation methode

Nu het duidelijk is wat de test harness gaat doen, kan de focus gelegd worden op welke algoritmes er wordt getest. Het is valide om één algoritme te testen, maar het is conventioneel om meerdere te testen. Algoritmes dat op dezelfde manier werken kunnen gegroepeerd worden, zoals regressie, classificatie bayesian, clustering, enz. Om te weten

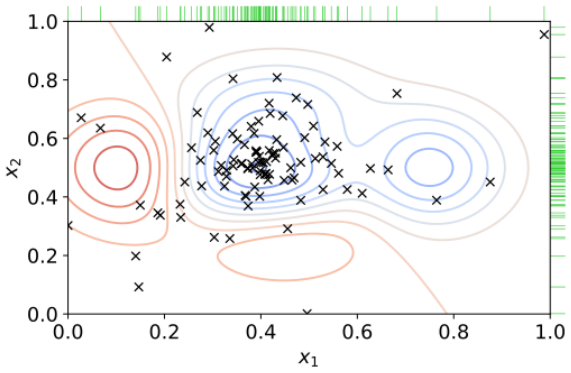
welke groep(en) relevant is, kan terug gerefereerd worden naar het probleem de eerste stap 1 (deelparagraaf 5.2.1).

5.2.4 Resultaten verbeteren

Als laatste stap moet het model geoptimaliseerd worden. Dit kan op drie manieren. Als eerste kan het model 'getuned' worden wat ook wel hyperparameter optimalisatie genoemd wordt. Een hyperparameter is een parameter dat wordt gebruikt om het leerproces van een model een bepaalde richting op te sturen. Er zijn een aantal manieren om de optimale waarde van een hyperparameter te achterhalen. Een aantal simpele methodes zijn grid search, random search en bayesian optimization. Met bijvoorbeeld de grid search methode wordt de minimale en maximale waarde van de parameter op een x- en y-as geplot. Deze grenzen moeten vaak zelf worden gespecificeerd. Vervolgens wordt er langs elke set gegaan en wordt doorgaans met cross-validation de prestatie berekend. In Figuur 5.5 is te zien dat de hyperparameters binnen de donkerblauwe lijnen het beste presteren. Het model presteert minder goed naarmate de lijnen rood worden.



Figuur 5.5: Hyperparameter optimalisatie met de grid search methode



Figuur 5.6: Hyperparameter optimalisatie met de bayesian optimization methode

Bij random search wordt hetzelfde proces doorlopen waarbij het enige verschil is dat de waardes willekeurig zijn gekozen. De bayesian optimization methode begint willekeurig, maar houd rekening met de vorige steekproeven om een keuze te maken om dichterbij of verder weg van de vorige waarde(s) te experimenteren (Figuur 5.6).

De tweede manier is met *ensembles*. Bij het gebruik van ensembles wordt het resultaat van verschillende methodes bijeengevoegd om tot een beter resultaat te komen. Er zijn een aantal strategieën om dit te doen. Een daarvan is **boosting**. Hierbij worden meerdere algoritmes van dezelfde groep getraind.

Als laatste manier kan feature engineering gebruikt worden op de prestatie te verbeteren. De data heeft een bepaalde structuur dat het model exploiteert om een besluit of voorspelling te maken. Met feature engineering worden bepaalde delen van de structuur 'benadrukt' zodat het model de focus hierop zal leggen. Er wordt als het ware tegen het model gezegd: "Deze features zijn belangrijk in de keuze/voorspelling dat wordt gedaan". Dit kan vervolgens resulteren in een accurater model.

5.2.5 Validatie en productie

Na het trainen en verbeteren van het model kan er validatie plaatst vinden. Gebruik hierbij de formulering uit stap 1 (deelparagraaf 5.2.1). Als het model niet of niet goed genoeg voldoet, kunnen de voorgaande stappen opnieuw bezocht worden. Als het model wel voldoet, kan het door naar productie.

5.3 Kennis vereist om een model te trainen

Waarom je niet alles kan voorkauwen. Wat is de middle ground?

5.4 Wachttijden voorspellen

5.5 Conclusie

5.6 Advies

6 Hoe werkt een Machine Learning pipeline

6.1 Literatuur

6.2 POC

6.3 Conclusie

7 Cloud computing platformen en lokale frameworks

7.1 Literatuur

7.2 Conclusie

8 Architecturale ontwerp van de oplossing

8.1 Literatuur

8.2 Design

8.3 Conclusie

9 Oplossing

9.1 Scope definiëren

9.2 Research techstack

9.3 Wireframe

9.4 Mockup

9.5 POC

9.6 Conclusie

10 Conclusie

11 Aanbeveling

12 Discussie

13 Reflectie

Bibliografie

[1] Algorithmia Inc. *Algorithmia*. 18 feb 2021. URL: https://algorithmia.com.

[2] Valohai. *Valohai*. 22 feb 2021. URL: https://valohai.com.

[3] A Burkov. *The Hundred-Page Machine Learning Book*. 1ste ed. 1 nov 2019. ISBN: 9781999579517.

[4] 16 mrt 2021. URL: https://wiki.pathmind.com/neural-network.

[5] 16 mrt 2021. URL: https://machinelearningmastery.com/think-machine-learning/.

[6] 7 mrt 2021. URL: https://developers.google.com/machine-learning/crash-course/framing/ml-terminology.

[7] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[8] 15 mrt 2021. URL: https://machinelearningmastery.com/process-for-working-through-machine-learning-problems/.

[9] T.M. Mitchell. *Machine learning*. 1ste ed. 1 nov 2019. ISBN: 9780071154673.

[10] 15 mrt 2021. URL: https://machinelearningmastery.com/how-to-define-your-machine-learning-problem/.

[11] 16 mrt 2021. URL: https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/.

[12] 16 mrt 2021. URL: https://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/.

Bijlagen