Machine Learning Pipeline Automation

Amar Kisoensingh

7 april 2021

Voorwoord

Samenvatting

Summary

Afkortingen

PaaS Platform as a Service 15

Lijst van figuren

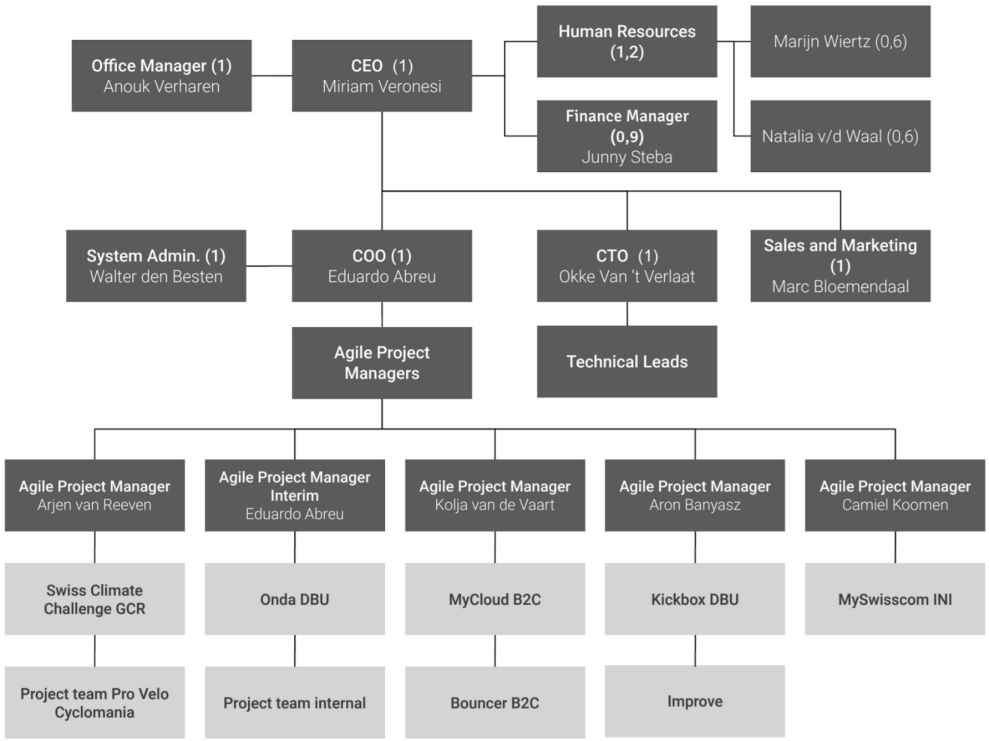
Lijst van tabellen

Inhoudsopgave

1 Inleiding

NGTI is een software ontwikkel bedrijf dat gevestigd is in Rotterdam. Met het starten van nieuwe projecten begint NGTI met de probleem stelling, mockups en wireframes en prototyping. Vervolgens wordt een applicatie voor mobiel en/of webgebruik ontwikkelt en wordt support geleverd voor bijvoorbeeld updates of het oplossen van bugs [1]. Naast het maken van een applicatie op maat biedt NGTI ook andere diensten, zoals een app framework of white label apps [2].

Kijkend naar hoe het bedrijf opereert is het een vrij 'platte' structuur. Formeel bestaat er wel een hiërarchie (Figuur 1.1), maar deze is in de praktijk niet zo gauw terug te vinden.



Figuur 1.1: Organogram van NGTI op 29-03-2021 [3].

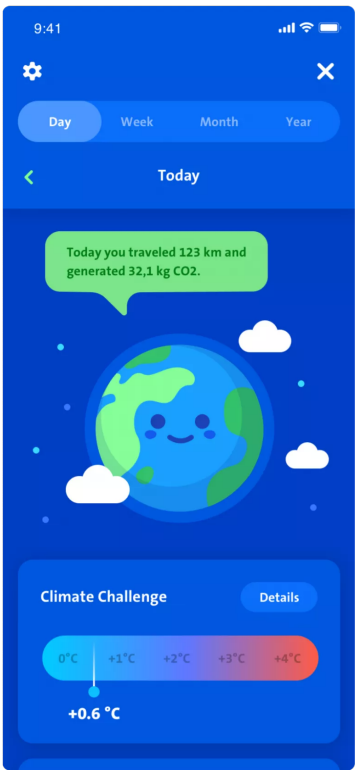
Swisscom heeft verschillende dochterbedrijven [4] waarvan NGTI er een van is. Sinds maart 2021 is het bekend gemaakt dat Swisscom van plan is om een afdeling, Swisscom DevOps Center, te fuseren met NGTI. Omdat de fusie onzekerheid met zich meebrengt voor de structuur en manier hoe NGTI werkt, zal de situatie vóór de fusie aangehouden worden gedurende het afstuderen.

1.1 Projecten

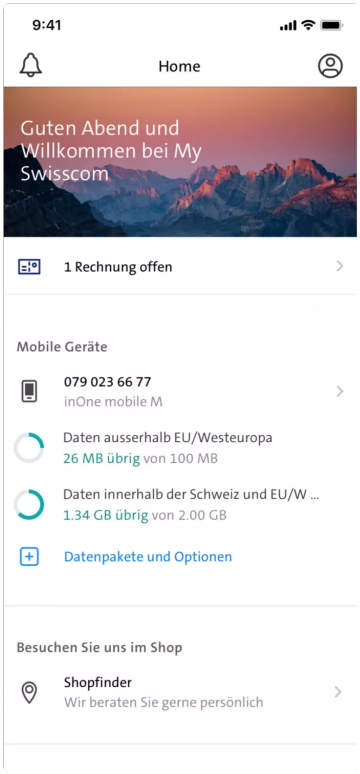
NGTI heeft een vrij breed portfolio met apps voor verschillende doeleinden. Een van deze apps is de Climate Challenge App [5]. Me deze app kunnen gebruikers hun C02-voetafdruk en impact in kaart brengen. Er wordt bijgehouden hoeveel kilometer de gebruiker reist en

met welke vervoersmiddel. De app is onderdeel van twee bestaande nieuws apps, Blick en Bluewin [6]. Het doel is om de gebruiker aan te sporen om groener te reizen. Een screenshot van de app is te zien in Figuur 1.2

Een andere oplossing is de My Swisscom App [7]. Dit is een native app voor Android en iOS waarbij Swisscom-klanten hun contract kunnen bestellen, wijzigen of beëindigen. In de app kunnen klanten ook de dataverbruik zien en instellingen voor abonnementen wijzigen. Een screenshot van de app is te zien in Figuur 1.3.



Figuur 1.2: Screenshot van de Swiss Climate Challenge app [5].



Figuur 1.3: Screenshot van de My Swisscom App [7]

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. De meest gebruikte en belangrijkste zijn Slack, Google Workspace en Zoom.

1.2.1 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*.

1.2.2 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.

1.2.3 Zoom

Voorheen werd Zoom alleen gebruikt om te videobellen met collega's en geïnterviewden. 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.

1.3 Aanleiding opdracht

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

De opdracht bestaat uit drie onderdelen:

1. Onderzoek naar hoe het opzetten van een pipeline en het maken van de acties in elke stap geautomatiseerd kunnen worden
2. Onderzoek naar hoe het trainen van een model versimpeld kan worden voor de developer
3. Onderzoek naar het maken van een platform-agnostische oplossing

Hierbij zal een proof-of-concept (PoC) gemaakt worden om aan te tonen of het haalbaar is. Een diepere duik in het probleem en het definiëren van de onderdelen is te vinden in hoofdstuk 2.

1.4 Leeswijzer

TODO: Schrijf leeswijzer voor elk hoofdstuk

2 Probleemanalyse

Zoals beschreven in paragraaf 1.3 is NGTI genoodzaakt om machine-learning in te zetten om haar applicaties 'slimmer' te maken. Hier zijn een aantal redenen voor, onder andere:

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 waarmee 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 domein, tijd om een pipeline op te zetten en vendor lock-in

2.1 Expertise Machine Learning

Machine learning is geen triviaal onderwerp. Om een model te trainen is kennis nodig van verschillende domeinen: data mining, 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 en een pipeline goed op te zetten, is het vaak te hoogdrempelig voor developers. De expertise is daar daarnaast niet in een korte tijd te vergaren.

2.2 Opzetten pipeline

Bovenop de complexiteit van machine learning zelf bestaan er verschillende manieren om een model te trainen. Een machine learning pipeline opzetten is daar een van. Een pipeline is een workflow dat bestaat uit een aantal stappen die doorgelopen kan worden om een model te trainen. In elke stap worden acties uitgevoerd, zoals het verwijderen van onbruikbare data of de prestatie van modellen vergelijken en een rapport met uitslagen genereren. Het opzetten van zo een pipeline én de actie(s) in de stappen definiëren kost tijd en vereist specifieke kennis. Daarnaast zijn de stappen en acties vaak hetzelfde voor verschillende pipelines. Het automatiseren en hergebruiken van stappen en acties tussen pipelines zou tot onder andere tijdwinst leiden.

2.3 Vendor lock-in

Er bestaan een aantal diensten, zogenoemde Platform as a Service (PaaS), waarbij je een pipeline kan opzetten en acties kan definiëren. 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.

2.4 Doelstelling

Om het probleem op te lossen is onderzoek en experimentatie nodig op verschillende vlakken. De gewenste oplossing is het ontwikkelen van een systeem waarbij developers met weinig tot geen kennis een model kunnen trainen. Het systeem moet de infrastructurele taken voor zich nemen, zoals het opzetten van een pipeline, de stappen en acties automatiseren. Daarnaast moet het systeem ook platform agnostisch zijn zodat het systeem niet aan één platform gebonden is.

2.5 Bestaande oplossingen om pipelines op te zetten

Er bestaan een aantal oplossingen die deels aan de eisen in de doelstelling voldoen. Elke 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, welke acties de stappen verricht en daar vervolgens (gedeeltelijk) het systeem op baseren. Daarnaast wordt bij alle oplossingen een expertise van ML op een bepaalde niveau verwacht.

De oplossingen kunnen gecategoriseerd worden in twee groepen:

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

2.5.1 Machine learning pipeline specifieke service

Bedrijven zoals Algorithmia [8] en Valohai [9] 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.

2.5.2 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.

Het systeem zou bijvoorbeeld een server kunnen aanmaken, een model trainen, het resultaat downloaden en vervolgens de server verwijderen. Om dit zonder de grafisch interface te doen kan er gebruik worden gemaakt van frameworks dat interacteert met het platform.

2.5.3 Frameworks om cloud computing platformen te beheren

Gedurende de vooronderzoek zijn frameworks dat cloud computing platformen beheert en frameworks waarmee een pipeline uitgerold kan worden naar boven gekomen. Deze frameworks kunnen deel uitmaken van de oplossing. Een framework dat cloud computing platformen kan beheren is Terraform [10]. Met Terraform is het mogelijk om een plan te schrijven waarin bijvoorbeeld staat welke type server nodig is. Bij het uitvoeren van het plan spreekt Terraform een cloud computing platform naar keuze aan om de server op te starten. Terraform kan vervolgens controleren of de server draait en de server afsluiten wanneer het niet meer nodig is.

Kubeflow [11] is een ander framework waarmee een pipeline kan worden opgezet. Het verschil met Terraform is dat Terraform flexibeler is met wat er aangemaakt kan worden op een cloud computing platform. Kubeflow kan alleen een pipeline uitrollen. Een andere framework zoals Kubeflow is Apache Beam [12].

2.6 Hoofd- en deelvragen

Uitgaand van de drie obstakels kan de hoofdvraag als volgt worden geformuleerd:

In welke mate kan een machine learning pipeline worden geautomatiseerd onafhankelijk van de onderliggende cloud computing platform?

De hoofdvraag kan worden onderbouwd met vier deelvragen. Om te beginnen is het verstandig om te weten welke stappen er in een machine learning pipeline zit:

Uit welke stappen bestaat een machine learning pipeline?

Vervolgens kunnen de verschillende cloud computing platformen in kaart worden gebracht:

Wat zijn de verschillen en overeenkomsten tussen cloud computing platformen waarmee een machine learning pipeline kan worden opgezet?

Verder kan het handig zijn om meerdere platformen aan te spreken. Dit zou kunnen met een bestaand framework. Hierbij is het, net als de vorige deelvraag, belangrijk om te weten welke frameworks er zijn:

Wat zijn de verschillen en overeenkomsten tussen frameworks waarmee cloud computing platformen beheerd kunnen worden?

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, waarmee een machine learning pipeline kan worden opgezet, die modulaire acties geautomatiseerd in stappen samenstelt, en die platform-onafhankelijk is, eruit?

3 Onderzoeksmethoden en scope

Om elke hoofd- en deelvraag te beantwoorden, wordt er bij elk gebruik gemaakt van een onderzoeksmethode. Volgens Scribbr [13] zijn er twee onderzoeksmethoden: kwantitatief en kwalitatief. Bij een kwantitatief onderzoeksmethode wordt data verzameld waarmee bijvoorbeeld grafieken of tabellen gemaakt kunnen worden. De focus bij een kwalitatief onderzoeksmethode ligt bij het verzamelen van verschillende interpretaties en opvattingen. Hierop kan optioneel een eigen interpretaties op gemaakt worden. [14].

Onder kwantitatief en kwalitatief vallen verschillende dataverzamelingsmethoden. Deze beschrijft simpelweg de manier hoe data wordt verzameld. Dit kan bijvoorbeeld met een enquête, literatuuronderzoek op websites en in boeken of een onderzoek over een lange periode [14].

Elke hoofd- en deelvraag is gekoppeld aan een onderzoeksmethoden. Vervolgens is beschreven welk(e) dataverzamelingsmethode(n) wordt gebruikt met een korte toelichting. Daarnaast wordt op een hoog niveau de scope bepaald.

D1: Uit welke stappen bestaat een machine learning pipeline?	
Methode(s)	Kwalitatief
Dataverzamelings methode(n)	Literatuuronderzoek, fundamenteel onderzoek, toegepast onderzoek
Scope	Binnen de scope: • In kaart brengen uit welke stappen een pipeline bestaat • Acties die in een stap worden uitgevoerd • Of het mogelijk is om stappen te versimpelen / abstraheren voor developers • Of het mogelijk is om stappen en acties te automatiseren Buiten de scope: • Automatisering van stappen en acties • Een versimpeling van machine learning
Toelichting	Er wordt gekeken naar welke stappen er in een pipeline zitten. De theorie wordt vervolgens toegepast in een experiment. De nadruk ligt vooral of de mogelijkheid er is om stappen en acties te automatiseren en of machine learning versimpeld kan worden, niet dat er een uitwerking is.

Tabel 3.1: Deelvraag 1

D2: Wat zijn de verschillen en overeenkomsten tussen cloud computing platformen waarmee een machine learning pipeline kan worden opgezet?	
Methode	Kwalitatief
Dataverzamelingsmethode(n)	Literatuuronderzoek, vergelijkend onderzoek
Scope	Binnen de scope: • Inventarisatie met de ”long list short list” methode • Welke functionaliteit bieden de platformen op een machine learning pipeline op te zetten • Ervaring opdoen doormiddel van een pipeline te maken binnen twee cloud computing platformen • Basale inventarisatie en vergelijking van alternatieve manieren om met een cloud computing platform te communiceren Buiten de scope: • Prijs, performance en snelheid
Toelichting	Cloud computing platformen worden in kaart gebracht. Vervolgens wordt met criteria bepaald in een later stadium de lijst verkort tot twee kandidaten. Alternatieve manieren om met een cloud computing platform te communiceren is afgebakend tot first-party tools en frameworks dat een of meerdere platformen tegelijk kan aanspreken.

Tabel 3.2: Deelvraag 2

D3: Wat zijn de verschillen en overeenkomsten tussen frameworks waarmee cloud computing platformen beheerd kunnen worden?	
Methode	Kwalitatief
Dataverzamelings methode(n)	Literatuuronderzoek, vergelijkend onderzoek
Scope	Binnen de scope: • Inventarisatie naar wat er aangemaakt, gewijzigd en verwijderd kan worden binnen een cloud computing platform • Hoe een machine learning pipeline op papier gemaakt zou worden met een framework • Experiment met het opzetten van een pipeline via het framework op een cloud computing platform
Toelichting	Er wordt gekeken naar welke frameworks er beschikbaar zijn en wat de verschillen/overeenkomsten zijn. Om de applicatie future proof te maken is het belangrijk om een framework te kiezen wat in de praktijk beproefd is en ondersteuning van een community heeft.

Tabel 3.3: Deelvraag 3

D4: Hoe ziet de architecturale blauwdruk van een applicatie, waarin een machine learning pipeline kan worden opgezet, die acties voorgeprogrammeerd zijn, en die platform-onafhankelijk is, eruit?	
Methode	Kwalitatief
Dataverzamelings methode(n)	Literatuuronderzoek
Scope	Binnen de scope: • Technische tekeningen Buiten de scope: -
Toelichting	De literatuuronderzoek slaat op of de technische tekeningen gemaakt zijn volgens een standaard zoals UML. Dit komt niet terug als theorie maar de bronnen worden wel vermeld.

Tabel 3.4: Deelvraag 4

H: In welke mate kan een machine learning pipeline worden geautomatiseerd onafhankelijk van de onderliggende cloud computing platform?	
Methode	Kwalitatief
Dataverzamelingsmethode(n)	Literatuuronderzoek
Scope	De scope wordt bepaald na de requirement analyse.
Toelichting	Onderzoek naar documentatie van gebruikte framework(s).

Tabel 3.5: Hoofdvraag

4 Stappen in een Machine Learning pipeline

Een machine learning (ML) pipeline is zoals beknopt beschreven in paragraaf 2.2 een collectie van stappen dat wordt doorlopen om een model te trainen. Elke stap bevat een aantal acties dat wordt uitgevoerd. Hapke en Nelson [15] benoemen een aantal voordelen bij het gebruik van een pipeline voor het trainen van modellen:

- Voorkomen van bugs

  De stap waarbij de data wordt voorbereid is gebonden aan het trainen van een model. Zonder een pipeline zou het kunnen voorkomen dat een model is getraind en achteraf het proces om de data voor te bereiden is aangepast. Volgens Hapke en Nelson kan dit zonder een geautomatiseerde workflows zorgen voor bugs [15, p. 2].

- Behulpzame broodkruimels

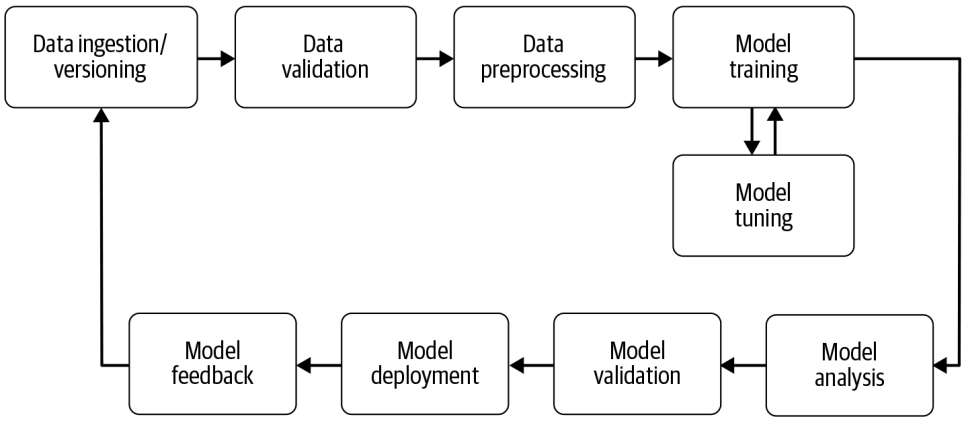
  Bij het doorlopen van de pipeline worden zaken bijgehouden zoals hyperparameters, gebruikte datasets en de modelstatistieken. Ook is te zien welk model momenteel is uitgerold. Mocht er wat fout gaan tijdens het trainen of uitrollen, is ook informatie terug te zien om het probleem te verhelpen.

- Standaardisatie binnen het team

  Binnen een team zal er één correcte manier zijn om een model te trainen; met de stappen in een pipeline. Dit zorgt ervoor dat er consistentie is tussen teamleden als een model wordt getraind en is het makkelijker voor nieuwe teamleden om te starten.

4.1 De stappen in een pipeline

Een machine learning pipeline begint met het opnemen van data en eindigt met het ontvangen van feedback om de prestatie van het model te verbeteren. De pipeline bevat een aantal stappen zoals data voorbereiden, het model trainen en het uitrollen van het model (Figuur 4.1). In totaal zijn er, zonder de feedback loop stap, acht stappen dat elke keer doorlopen moeten worden om een model te trainen. Om dit proces handmatig te herhalen is tijdrovend en kan gevoelig voor fouten zijn.



Figuur 4.1: Lifecycle van een model volgens Hapke en Nelson [15, p. 4].

Om een greep te krijgen van een machine learning pipeline zullen de stappen uit Figuur 4.1 als leidraad genomen worden. Bij elke stap in de komende subkoppen wordt code snippets gegeven. Het doel is om bij de laatste stap een werkend pipeline te maken. Een pipeline wordt doorgaans uitgevoerd in een cloud omgeving zoals Azure of Amazon Web Services (AWS). In dit hoofdstuk wordt de pipeline lokaal gemaakt en zal in hoofdstuk 5 een pipeline in een cloud omgeving opgezet worden.

De dataset waarmee het model wordt getraind bestaat uit 150 voorbeelden van drie iris soorten: Setosa, Virginica en Versicolor. Elk voorbeeld heeft de lengte en breedte van de kelk- en bloemblad met de bijbehorende soort iris. In Tabel 4.1 is een voorbeeld van de dataset te zien. Het model zal een van de drie soorten kunnen herkennen op basis van de lengte en breedte van een gegeven kelk- en bloemblad.

Kelkblad - lengte	**Kelkblad - breedte**	**Bloemblad - lengte**	**Bloemblad - breedte**	**Soort**
5.1	3.5	1.4	0.2	Iris-setosa

Tabel 4.1: Voorbeeld van de iris dataset

De gekozen dataset is een bekend ML voorbeeld welk meerdere malen is opgelost. Het doel is dus om de stappen te beproeven en niet om te experimenteren met een onopgelost probleem.

4.1.1 Data opname en versiebeheer

De eerste stap in de pipeline is het opnemen van data. Met deze data zal het model getraind, gevalideerd en getest worden. De dataset kan van een of meerdere bronnen komen, zoals lokaal, een storage bucket of van een database. Zodra de data is ingeladen, moet het verdeeld worden tussen een train, validatie en test dataset. Normaal gebeurt dit met een split ratio van 6:2:2. De train dataset is 60% en de validatie en test dataset zijn allebei 20% van de originele dataset [15, p. 27-37].

Een usecase van een pipeline is dat een nieuw model getraind kan worden door een geüpdatet dataset te gebruiken. Dit wordt gedaan door de voorgaande dataset te gebruiken waarbij nieuwe data is toegevoegd. Door het gebruik van verschillende datasets is het handig om versiebeheer toe te passen. Zo is goed te zien welke dataset welk model produceert. Een versie geven aan een dataset gebeurt voordat de dataset wordt ingeladen [15, p. 39-40]. Versiebeheer voor datasets kan bijvoorbeeld met DVC [16] of Pachyderm [17].

De iris dataset wordt in Figuur 4.2 geïmporteerd door middel van code. Normaal gesproken zou, voordat deze code uitgevoerd wordt, een versie gegeven worden aan de dataset. Omdat de schaal van dit voorbeeld klein is en om de voorbeeld reproduceerbaar te houden is dit niet gedaan. De dataset wordt aangemaakt met de namen voor de kolommen op regel 5 net zoals het voorbeeld in Tabel 4.1. Vervolgens is de dataset onder de variabel naam $dataset$ beschikbaar voor de komende stappen.

```
# URL to the dataset
url =
"https://gist.githubusercontent.com/UNRULYEON/7765e8bc37f928fc91aea3
d109c337f5/raw/890f6e5c11f740150d57c578cf06d5bf9f35000d/iris.csv"

# Names of the columns
names = ["sepal-length", "sepal-width", "petal-length", "petal-
width", "class", ]

# Create dataset
dataset = read_csv(url, names=names)
```

Figuur 4.2: Dataset importeren

4.1.2 Data validatie

Nu de dataset verdeeld is, een versie heeft en op een bereikbare plek is, kan de data gevalideerd worden. Deze stap is vooral belangrijk om te voorkomen dat een model wordt getraind dat niet nuttig is aangezien het trainen veel tijd in beslag kan nemen. Een bekende uitdrukking is ”garbage in = garbage out”. Dit betekent dat als de dataset niet goed is, het model ook niet goed zal zijn [15, p. 43]. Tijdens de validatie stap wordt gecontroleerd op het volgende:

- Afwijkingen in de dataset
- Wijzigingen in de structuur
- Algemene statistieken in vergelijkingen met voorgaand datasets [15, p. 44]

Er wordt eerst statistieken gegenereerd van het huidige dataset. Om voorbeelden te geven kan een fictief dataset van woningen in Rotterdam dat te koop staat genomen worden. Uit de statistieken van dit dataset kan blijken dat er meer woningen in Rotterdam Noord zijn dan Zuid. Dit kan een onrepresentatief voorspelling geven voor woningen in Zuid. Een afwijkingen en vergelijking zou kunnen zijn dat de prijs in voorgaande datasets cijfers waren, maar in het huidige dataset karakters zijn.

5 Cloud computing platformen

Een cloud computing platform stelt een dienst beschikbaar waarmee gebruikers onder andere rekenkracht of opslag kan huren. De gebruiker is niet verantwoordelijk voor de details zoals onderhoud of uptime. Daarnaast kan er gemakkelijk geschaald worden mocht een gebruiker behoefte hebben aan meer rekenkracht of opslag [18].

5.1 Inventarisatie van platformen

De PoC moet minimaal twee platformen ondersteunen om te voldoen aan de platform agnostische vereiste. Om de lijst van potentiële kandidaten te verkleinen wordt er gekeken naar een aantal criteria waaraan de platform moet voldoen. De criteria is opgelegd door NGTI en de volgorde van de lijst heeft geen belang. Een kort overzicht van de criteria is in de onderstaande lijst te vinden. Een uitgebreid overzicht met toelichting bij elk criteria is te vinden in de bijlage (Bijlage 12).

- ML Pipeline aanmaken
- Serverbeheer
- Database beheer
- Storage bucket beheer
- Uptime 99.9%
- Regionale beschikbaarheid
- Toegankelijkheid documentatie ML pipeline
- APIs
- Inhoud ML

Er bestaat geen lijst met waar alle platformen te vinden zijn. De platformen in Tabel 5.1 zijn gevonden op verschillende websites dat een handvol platformen noemt. De websites sorteert de platformen op basis van een bepaalde kenmerk (goedkoopst, veiligst, het beste voor beginners) en benoemt de lijst het beste van bijvoorbeeld 2019 of 2020. Omdat er al criteria (12) is gedefinieerd, is het niet nodig om verder te kijken dan de naam van het platform.

In Tabel 5.1 zijn de platformen met de criteria te vinden en of het platform wel of niet voldoet. Een uitgebreide analyse met links naar eventuele documentatie van een requirement is te vinden in de bijlage. In elke regel is te vinden welke specifieke bijlage het is.

Platform	ML pipeline aanmaken	Server be-heer	Database beheer	Storage bucket beheer	Uptime 99.9%	Regionale beschik-baarheid	Toegankelijkheid documentatie ML pipeline	APIs	Inhoud ML	Bijlage
AWS										Tabel 2
Azure										Tabel 3
Google Cloud										Tabel 4
DigitalOcean										Tabel 5
IBM Cloud										Tabel 6
Alibaba										Tabel 7
Oracle Cloud Infrastructure										Tabel 8
Kamatera Cloud										Tabel 9
Cloudways										Tabel 10
Vultr										Tabel 11
BigML Inc.										Tabel 12
H2O.ai Inc.										Tabel 13

Tabel 5.1: Overzicht van cloud computing platformen met criteria

6 Frameworks om platformen te beheren

7 Architecturale ontwerp van de oplossing

7.1 Literatuur

7.2 Design

7.3 Conclusie

8 Oplossing

8.1 Scope definiëren

8.2 Research techstack

8.3 Wireframe

8.4 Mockup

8.5 POC

8.6 Conclusie

9 Conclusie

10 Aanbeveling

11 Discussie

12 Reflectie

Bibliografie

[1] 22 mrt 2021. URL: https://www.ngti.nl/diensten/.

[2] 29 mrt 2021. URL: https://www.ngti.nl/oplossingen/.

[3] NGTI B.V. „Organogram".

[4] 29 mrt 2021. URL: https://www.swisscom.ch/en/about/beteiligungen-swisscom-uebersicht.html.

[5] 29 mrt 2021. URL: https://www.ngti.nl/cases/swiss-climate-challenge/.

[6] 29 mrt 2021. URL: https://www.swissclimatechallenge.ch.

[7] 29 mrt 2021. URL: https://www.ngti.nl/cases/my-swisscom-app/.

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

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

[10] 5 apr 2021. URL: https://www.terraform.io.

[11] 5 apr 2021. URL: https://www.kubeflow.org.

[12] 5 apr 2021. URL: https://beam.apache.org.

[13] 25 mrt 2021. URL: https://www.scribbr.nl/scriptie-structuur/methodologie-in-je-scriptie/.

[14] 25 mrt 2021. URL: https://www.scribbr.nl/onderzoeksmethoden/kwalitatief-vs-kwantitatief-onderzoek/.

[15] H. Hapke en C. Nelson. *Building Machine Learning Pipelines*. 1ste ed. 2020. ISBN: 9781492053194.

[16] 6 apr 2021. URL: https://dvc.org.

[17] 6 apr 2021. URL: https://www.pachyderm.com.

[18] 5 apr 2021. URL: https://nl.wikipedia.org/wiki/Cloud_computing.

Bijlagen

Criteria met toelichting voor cloud computing platformen

Criteria	Toelichting
ML Pipeline aanmaken	Het platform moet het aanmaken van een machine learning pipeline kunnen ondersteunen.
Serverbeheer	Aanmaken, wijzigen en verwijderen van een server. Een server kunnen aanmaken bij het platform is praktisch omdat een server voor meerdere doeleinde gebruikt kan worden.
Database beheer	Aanmaken, wijzigen en verwijderen van een database. De PoC moet data ergens kunnen opslaan en vandaan halen; dit is mogelijk in een database.
Storage bucket beheer	Aanmaken, wijzigen en verwijderen van een storage bucket. In een storage bucket kan grote hoeveelheden data opgeslagen worden zoals bestanden en afbeeldingen. Dit kan handig zijn om bijvoorbeeld train en test data op te slaan.
Uptime 99.9%	Het platform moet een vorm van uptime kunnen garanderen waarbij het percentage zo dicht mogelijk bij 99.9 ligt. Hierdoor is de kans dat NGTI door een storing bij een platform niet productief kan zijn zo klein mogelijk.
Regionale beschikbaarheid	Waar data wordt opgeslagen en de servers draaien is vrij belangrijk. Het platform moet ten minste West-Europa ondersteunen in verband met de gegevensbescherming in de EU (GDPR). In een ideale situatie zou het platform ook specifiek Zwitserland ondersteunen aangezien de meeste klanten van NGTI vandaar komen.
Toegankelijkheid documentatie ML pipeline	In het geval dat de PoC wordt uitgebreid tot een applicatie is het belangrijk om onderhoud uit te voeren en eventueel nieuwe functies toe te voegen. Het is daarom van belang dat het platform documentatie heeft over ML pipelines.
APIs	Om het platform programmatisch aan te sturen zijn APIs dat het platform beschikbaar stelt onmisbaar. Op deze manier kunnen frameworks het platform beheren en kan eventueel een op maat gemaakte oplossing gebruikt worden.
Inhoud ML	Voor NGTI is het belangrijk dat het platform niet alleen ondersteuning heeft om modellen te trainen, maar meer mogelijkheden biedt zoals het schrijven van een eigen algoritme of out-of-the-box oplossingen.

Tabel 1: Criteria voor cloud computing platformen

Gedetailleerd overzicht cloud computing platformen

AWS

Criteria	Toelichting
ML Pipeline aanmaken	Met Amazon SageMaker kunnen pipelines gemaakt worden. Daarnaast heeft Amazon specifieke services om bijvoorbeeld vooroordelen te detecteren, statistieken te meten en data te verzamelen (overzicht).
Serverbeheer	Met Amazon EC2 kunnen servers worden opgestart, gewijzigd of verwijderd worden. Een overzicht wat Amazon EC2 allemaal kan
Database beheer	Amazon biedt een database service aan waarbij een database aangemaakt, gewijzigd of verwijderd kan worden. De databases zijn gesorteerd op type in dit overzicht.
Storage bucket beheer	Wat betreft opslag biedt Amazon twee soorten aan: Amazon S3 en Amazon Elastic Block Store. Het verschil is dat S3 data opslaat als een object ten opzicht van Elastic Block Storage, dat data op de conventionele manier opslaat.
Uptime 99.9%	De uptime staat beschreven in de Service Level Agreement (SLA) en is voor elke service anders. Over het algemeen biedt Amazon voor elke service een uptime van 99.9%: Amazon SageMaker, Amazon EC2 en Elastic Block Store, Amazon Relational Databases en Amazon S3.
Regionale beschikbaarheid	Amazon heeft in totaal zes fysieke datacenters in Europa (overzicht). Ook is het mogelijk om een specifiek regio te kiezen zoals het hier beschreven staat. Helaas is Zwitserland geen beschikbare regio.
Toegankelijkheid documentatie ML pipeline	Amazon heeft een uitgebreide services voor ML binnen AWS. Alle services zijn te vinden in dit (overzicht).
APIs	Voor al haar services stelt Amazon een API beschikbaar. Een overzicht voor de documentatie is hier te vinden. API documentatie voor ML en SageMaker is ook beschikbaar.
Inhoud ML	Amazon heeft een groot aantal ML specifieke services waar gebruik van gemaakt kan worden. Een paar voorbeelden daarvan is Amazon Translate, Amazon Healthlake en Amazon Forecast.

Tabel 2: AWS tegenover criteria op 06-04-2021

Azure

Criteria	Toelichting
ML Pipeline aanmaken	
Serverbeheer	
Database beheer	
Storage bucket beheer	
Uptime 99.9%	
Regionale beschikbaarheid	
Toegankelijkheid documentatie ML pipeline	
APIs	
Inhoud ML	

Tabel 3: Azure tegenover criteria op 06-04-2021

Google Cloud

Criteria	Toelichting
ML Pipeline aanmaken	
Serverbeheer	
Database beheer	
Storage bucket beheer	
Uptime 99.9%	
Regionale beschikbaarheid	
Toegankelijkheid documentatie ML pipeline	
APIs	
Inhoud ML	

Tabel 4: Google Cloud tegenover criteria op 06-04-2021

DigitalOcean

Criteria	Toelichting
ML Pipeline aanmaken	
Serverbeheer	
Database beheer	
Storage bucket beheer	
Uptime 99.9%	
Regionale beschikbaarheid	
Toegankelijkheid documentatie ML pipeline	
APIs	
Inhoud ML	

Tabel 5: DigitalOcean tegenover criteria op 06-04-2021

IBM Cloud

Criteria	Toelichting
ML Pipeline aanmaken	
Serverbeheer	
Database beheer	
Storage bucket beheer	
Uptime 99.9%	
Regionale beschikbaarheid	
Toegankelijkheid documentatie ML pipeline	
APIs	
Inhoud ML	

Tabel 6: IBM Cloud tegenover criteria op 06-04-2021

Alibaba

Criteria	Toelichting
ML Pipeline aanmaken	
Serverbeheer	
Database beheer	
Storage bucket beheer	
Uptime 99.9%	
Regionale beschikbaarheid	
Toegankelijkheid documentatie ML pipeline	
APIs	
Inhoud ML	

Tabel 7: Alibaba tegenover criteria op 06-04-2021

Oracle Cloud Infrastructure

Criteria	Toelichting
ML Pipeline aanmaken	
Serverbeheer	
Database beheer	
Storage bucket beheer	
Uptime 99.9%	
Regionale beschikbaarheid	
Toegankelijkheid documentatie ML pipeline	
APIs	
Inhoud ML	

Tabel 8: Oracle Cloud Infrastructure tegenover criteria op 06-04-2021

Kamatera Cloud

Criteria	Toelichting
ML Pipeline aanmaken	
Serverbeheer	
Database beheer	
Storage bucket beheer	
Uptime 99.9%	
Regionale beschikbaarheid	
Toegankelijkheid documentatie ML pipeline	
APIs	
Inhoud ML	

Tabel 9: Kamatera Cloud tegenover criteria op 06-04-2021

Cloudways

Criteria	Toelichting
ML Pipeline aanmaken	
Serverbeheer	
Database beheer	
Storage bucket beheer	
Uptime 99.9%	
Regionale beschikbaarheid	
Toegankelijkheid documentatie ML pipeline	
APIs	
Inhoud ML	

Tabel 10: Cloudways tegenover criteria op 06-04-2021

Vultr

Criteria	Toelichting
ML Pipeline aanmaken	
Serverbeheer	
Database beheer	
Storage bucket beheer	
Uptime 99.9%	
Regionale beschikbaarheid	
Toegankelijkheid documentatie ML pipeline	
APIs	
Inhoud ML	

Tabel 11: Vultr tegenover criteria op 06-04-2021

BigML Inc.

Criteria	Toelichting
ML Pipeline aanmaken	
Serverbeheer	
Database beheer	
Storage bucket beheer	
Uptime 99.9%	
Regionale beschikbaarheid	
Toegankelijkheid documentatie ML pipeline	
APIs	
Inhoud ML	

Tabel 12: BigML Inc. tegenover criteria op 06-04-2021

H2O.ai Inc.

Criteria	Toelichting
ML Pipeline aanmaken	
Serverbeheer	
Database beheer	
Storage bucket beheer	
Uptime 99.9%	
Regionale beschikbaarheid	
Toegankelijkheid documentatie ML pipeline	
APIs	
Inhoud ML	

Tabel 13: H2O.ai Inc. tegenover criteria op 06-04-2021